PROCEEDINGS OF THE
MULTIDISCIPLINARY DOCTORAL SCHOOL
2011-2012 ACADEMIC YEAR
FACULTY OF INFORMATION TECHNOLOGY
PÁZMÁNY PÉTER CATHOLIC UNIVERSITY
BUDAPEST
2012

Faculty of Information Technology
Pázmány Péter Catholic University

Ph.D. PROCEEDINGS

PROCEEDINGS OF THE MULTIDISCIPLINARY DOCTORAL SCHOOL 2011-2012 ACADEMIC YEAR FACULTY OF INFORMATION TECHNOLOGY PÁZMÁNY PÉTER CATHOLIC UNIVERSITY BUDAPEST

July, 2012

Pázmány University ePress
Budapest, 2012

Kiadja a Pázmány Egyetem eKiadó
2012
Budapest

Felelős kiadó
Ft. Dr. Szuromi Szabolcs Anzelm O. Praem.
a Pázmány Péter Katolikus Egyetem rektora

Készült a
TÁMOP-4.2.1.B-11/2/KMR-2011-0002 és a TÁMOP-4.2.2/B-10/1-2010-0014
projekt keretében, és az Új Széchenyi Terv támogatásával

Cover image by Csaba Nemes, Partition and floorplan of a layered graph generated from a mathematical expression related to computational fluid dynamics (CFD).

A borítón Nemes Csaba ábrája látható: Egy áramlástani példa megoldása során használt adatfolyam gráf particionálása és a gráf csúcsainak elhelyezése a síkban.

HU ISSN 1788-9197

Contents

Introduction

It is our pleasure to publish this Annual Proceedings again to demonstrate the genuine multidisciplinary research done at the Jedlik Laboratories by young talents working in the Interdisciplinary Doctoral School of the Faculty of Information Technology at Pázmány Péter Catholic University. The scientific results of our PhD students show the main recent research directions in which our faculty is engaged. Thanks are also due to the supervisors and consultants, as well as to the five collaborating National Research Laboratories of the Hungarian Academy of Sciences and the Semmelweis Medical School. The collaborative work with the partner universities, especially, Katolieke Universiteit Leuven, Politecnico di Torino, Technische Universität München, University of California at Berkeley, University of Notre Dame, Universidad de Sevilla, Universita di Catania is gratefully acknowledged.

As an important development of this special collaboration, we were able to jointly accredit a new undergraduate curriculum on Molecular Bionics with the Semmelweis Medical School, the first of this kind in Europe.

We acknowledge the many sponsors of the research reported here. Namely,

- the Hungarian National Research Fund (OTKA),
- the Hungarian Academy of Sciences (MTA),
- the National Development Agency (NFÜ),
- the Gedeon Richter Co.,
- the Office of Naval Research (ONR) of the US,
- NVIDIA Ltd.,
- Eutecus Inc., Berkeley, CA,
- MorphoLogic Ltd., Budapest,
- Analogic Computers Ltd., Budapest,
- AnaFocus Ltd., Seville,

and some other companies and individuals.

Needless to say, the resources and support of the Pázmány Péter Catholic University is gratefully acknowledged.

Budapest, July 2012.

Tamás Roska	Gábor Prószéky	Péter Szolgay
Head of the Jedlik Laboratory	Chairman of the Board of the Doctoral School	Head of the Doctoral School

Determining Sparse Realizations of Biochemical Reaction Networks Using Parallel Optimization

Zoltán A. Tuza
(Supervisors: Gábor Szederkényi, Kristóf Karacs)
tuza.zoltan@itk.ppke.hu

***Abstract*—In this paper, all sparse realization of a given kinetic system will be computed using an optimization-based algorithm. For demonstration purpose we select the 3-dimensional Lorenz system and transform it into kinetic form. The transformation is carried out using state-dependent time-scaling. The resulting sparse realizations are briefly evaluated based on the criteria described in the paper.**

***Keywords*-reaction networks; chaotic systems; optimization**

I. Introduction

Kinetic systems (also called chemical reaction networks, or simply CRNs) are known to possess advantageous dynamical descriptive properties being able to produce all the important qualitative features like stable and unstable equilibria, multiple equilibria, bifurcation phenomena, oscillatory and even chaotic behaviour [2], [7]. Thus the CRN model can describe the nonlinear nature of the multireaction networks [7]. Since it is a lumped model, it can be described by a system of ordinary differential equations. The structure of a (bio)chemical reaction network represents the interactions between chemical compounds. It shows how products are formed from reactants. The parameters of this network are the reaction rates between reactants and products.

Necessary and sufficient conditions for a polynomial system to be kinetic were first given in [6], where a constructive proof was given to build the so-called "canonic mechanism" for kinetic polynomial models. It has been known since at least the 1970's that multiple different CRN structures/parametrizations can generate exactly the same dynamics of the concentrations [5], [7]. This phenomenon is called *macro-equivalence* or *dynamical equivalence*. The geometric conditions of macro-equivalence were first studied in [1]. The first optimization-based numerical procedures for generating dynamically equivalent structures with prescribed properties were published in [9], [11], [10], [12].

This dynamical equivalence raises the issue of ambiguity in the identification procedure, since reaction networks with the same complex set, but with different structure and flux rates exhibit the same dynamic behavior in the concentration space. In this paper we propose an optimization-based method to determine all sparse realization of a given kinetic system.

II. Background

A. Mathematical models of deterministic chemical reaction networks obeying the mass action law

In this paper, only deterministic mass-action type models are meant by CRNs.

The mathematical description of the Chemical Reaction Networks originates from [3], using that formulation a CRN is characterized by three sets:

1) $\mathcal{S} = \{X_1, \ldots, X_n\}$ is the set of *species* or chemical substances.
2) $\mathcal{C} = \{C_1, \ldots, C_m\}$ is the set of *complexes*. Formally, the complexes are represented as linear combinations of the species, i.e.
$$C_i = \sum_{j=1}^{n} \alpha_{ij} X_j, \qquad i = 1, \ldots, m, \tag{1}$$
where α_{ij} are nonnegative integers and are called the *stoichiometric coefficients*.
3) $\mathcal{R} = \{(C_i, C_j) \mid C_i, C_j \in \mathcal{C}$, and C_i is transformed to C_j in the CRN$\}$ is the set of *reactions*. The relation $(C_i, C_j) \in \mathcal{R}$ will be denoted as $C_i \to C_j$. Moreover, a nonnegative weight, the *reaction rate coefficient* denoted by k_{ij} is assigned to each reaction $C_i \to C_j$. Naturally, if the reaction $C_i \to C_j$ is not present in the CRN then $k_{ij} = 0$.

Given these sets, a weighted directed graph can be built $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the vertexes which are the complexes of the reaction network, $\mathcal{V} = \{C_1, C_2, \ldots, m\}$. The $\mathcal{E}$ set contains the directed edges representing the reactions between the complexes, i.e. $(C_i, C_j) \in \mathcal{E}$ if reaction $C_i \to C_j$ occurs. For each edges of the CRN a positive weight k_{ij} is assigned as reaction rate for the reaction $C_i \to C_j$. Edges with zero rates are omitted from the reaction graph.

To describe the time-evolution of species concentration, several options are available. With the mass action law assumption, we will use the following nonlinear vector-valued differential equation, which was proposed in [3]

$$\dot{x} = Y \cdot A_k \cdot \Psi(x) \tag{2}$$

where $x \in \mathbb{R}^n$ is the concentration vector of the species from $\mathcal{S}$. The $Y \in \mathbb{R}^{n \times m}$ called complex composition matrix where the jth column encodes the composition of the C_j complex, namely $Y_{i,j} = \alpha_{j,i}$. The $A_k \in \mathbb{R}^{n \times n}$ matrix stores

the structure and parameters of the reaction graph, this is a column conservation matrix (also called the *Kirchhoff matrix* of the CRN) defined as follows:

$$[A_k]_{i,j} = \begin{cases} -\sum_{l=1,l\neq i}^{m} k_{il}, & \text{if} \quad i = j \\ k_{ji}, & \text{if} \quad i \neq j. \end{cases} \tag{3}$$

The last term in the equation is a monomial-type vector mapping defined by

$$\psi_j(x) = \prod_{i=1}^{n} x_i^{Y_{i,j}}, \quad j = 1, \ldots, m. \tag{4}$$

1) Dynamical equivalence of mass-action networks: Let us denote two CRNs with the following matrix pairs $(Y^{(1)}, A_k^{(1)}$ and $(Y^{(2)}, A_k^{(2)})$ and we call them dynamically equivalent, if

$$Y^{(1)} A_k^{(1)} \psi^{(1)}(x) = Y^{(2)} A_k^{(2)} \psi^{(2)}(x) = f(x), \quad \forall x \in \bar{\mathbb{R}}_+^n, \tag{5}$$

where $f(x)$ denotes the right hand side of the differential equation and for $i = 1, 2$, $Y^{(i)} \in \mathbb{R}^{n \times m_i}$ have nonnegative integer entries, $A_k^{(i)}$ are valid Kirchhoff matrices. At this point, we can define the realizations of a kinetic vector f such as $(Y^{(i)}, A_k^{(i)})$ for $i = 1, 2$. Hereinafter, it is assumed that the set of complexes are fixed and known before the computations, therefore the above equivalence definition can be rewritten as

$$Y \cdot A_k^{(1)} = Y \cdot A_k^{(2)} =: M \tag{6}$$

M is the invariant matrix containing the coefficients of the monomials.

Clearly, if $A_k^{(1)}$ and $A_k^{(2)}$ are valid realization with fixed Y, then $A_k^{(3)} = \frac{A_k^{(1)} + A_k^{(2)}}{2}$ is also a valid realization, thus a CRN can have infinitely many realizations. Therefore, we need specific computational methods to calculate those realizations which fulfills predefined properties, such as dense or sparse realization, weakly reversibly realization, etc. The computational methods to calculated these dynamically equivalent realization with desired properties was introduced in [9].

B. Optimization methods to find sparse and dense realizations

For fixed stoichiometric matrix Y, the sparsest realization contains minimum number of element in the A_k matrix, on the other hand the dense realization contains the maximal number of reactions.

These calculations can be formulated as mixed integer linear programing taks, where we assume that we have a canonical CRN and its parameters are known. Solving the MILP optimization, we want to find another valid A_k matrices that fulfill the given requirements. One of the possible requirements is the mass-action dynamics and it can be expressed as equality and inequality constraints

$$Y \cdot A_k = M \tag{7}$$

$$\sum_{i=1}^{m} [A_k]_{ij} = 0, \quad j = 1, \ldots, m \tag{8}$$

$$[A_k]_{ij} \geq 0 \quad i, j = 1, \ldots, m \quad i \neq j \tag{9}$$

$$[A_k]_{ii} \leq 0, \quad i = 1, \ldots, m \tag{10}$$

where the elements of A_k are the decision variables. We also put lower and upper bound constraint on the decision variables to make the optimization problem computationally tractable.

$$0 \leq [A_k]_{ij} \leq l_{ij}, \quad i, j = 1, \ldots, m, i \neq j \tag{11}$$

$$l_{ii} \leq [A_k]_{ii} \leq 0, \quad i = 1, \ldots, m \tag{12}$$

On the top of these constraints we want to find such A_k matrices where the number of nonzero off-diagonal elements are minimal or maximal with respect to the definition of dynamical equivalence. To achieve this, we introduce logical variables denoted by δ and construct the following compound statements.

$$\delta_{ij} = 1 \leftrightarrow [A_k]_{ij} > \epsilon, \quad i, j = 1, \ldots, m, i \neq j \tag{13}$$

where "$\leftrightarrow$" encodes the *if and only if* logical statement, ϵ is an arbitrary small threshold variable. The introduction of the δ logical variable yields the objective function

$$C_1(\delta) = \sum_{i,j=1 i \neq j}^{m} \delta_{ij} \tag{14}$$

given this objective function and the above set of constraints, we are able to compute realizations with maximal or minimal number of reactions, thus the sparse or dense realizations of the canonical CRN.

It is known that the sparse realizations are generally non-unique, meanwhile the structure of the dense realization given complex set is unique and contains every possible dynamically equivalent structure as a proper subgraph [11]. The result section will elaborate on these sparse realizations and investigates their properties.

1) Core reactions: Core reactions are such reactions that must be present in any dynamically equivalent realizations of a CRN. Therefore, if the following constraint included in the CRN realization computation

$$[A_k]_{ji} = 0 \tag{15}$$

and it yields infeasible LP, then the reaction $C_i \to C_j$ is a core reaction.

III. Transformation of polynomial models into kinetic form

In this section a general polynomial ODE will be converted into kinetic system in three steps. First, we have to ensure that the trajectory of the system remains in the positive orthant. Then, we apply the state-dependent time-scaling to make the system kinetic. Finally, a canonical realization is made the algorithm proposed in [6]

The general polynomial ODE system can written in the following form

$$\dot{x} = M\Psi(x) \tag{16}$$

$$\Psi_j(x) = \prod_{i=1}^{n} x_i^{[B]_{ji}}, \quad j = 1, \ldots, m. \tag{17}$$

where n and m denote the number of state variables and monomials, respectively. The $M \in \mathbb{R}^{n \times m}$ contains the

coefficients of the monomials in the polynomial ODE and $B \in \mathbb{R}^{m\times n}$ matrix encodes the monomials of the system. Hence, a polynomial ODE can be completely described with (M, B) pair of matrices.

A. Shifting the state variable into the positive orthant

The first step toward a kinetic form is the shifting the state variables into the positive orthant.

$$\tilde{x} = x + C \tag{18}$$

where each component of C are

$$c_j > |min(x_j)| \qquad j = 1, \ldots, n \tag{19}$$

This yields the shifted ODE

$$\dot{\tilde{x}} = M\Psi(\tilde{x} - C) \tag{20}$$

B. State-dependent Time Scaling

With state-dependent time scaling we can transform our system into a new time domain where the connection between the two domains are described by a scalar differential equation (21). That means that the nonlinear mapping function between the two time domains depends on the state variables. This transform is widely used for feedback linearization [4] and navigation planning [8]. However we will apply this transform for our shifted polynomial system to make it a kinetic system.

If the initial conditions and the all the state variables remain in strictly positive throughout the solution ($x_0 \in R^n_+$ and $x(t) \in R^n_+$), then one can apply the following state-dependent time scaling for the system equations

$$dt = \prod_{i=1}^{n} \chi_i^{x_i} dt \tag{21}$$

where $\chi_i \in \{0, 1\}$ for $i = 1, \ldots, n$, there always exist such a binary combination of the state variables which yields a kinetic system.

C. Building reaction kinetic realization

At this point we have our polynomial system in a kinetic form and want to assign a chemical reaction network (CRN) where the nodes represent the complexes and the edges are the reactions transforming C_j complex into C_i form as it was described in section II-A. To achieve this directed graph the algorithm from [6] was used. The output of the algorithm creates the so called canonical CRN realization of the original polynomial system, such system is shown on Figure 1. This directed graph will be our base structure in the further investigation.

IV. Results

For our investigation we choose a well-known chaotic system, namely the 3-dimensional Lorenz system.

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= xy - \beta z \end{aligned} \tag{22}$$

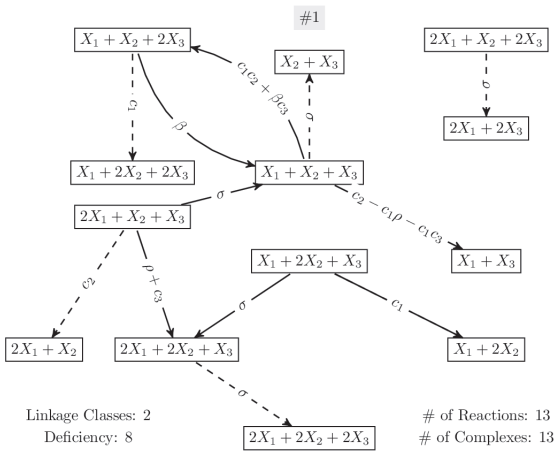


Figure 1. Canonical realization of Lorenz system with state-dependent time scaling. The dotted edges represent the core reactions and the parameters on the edges indicate the reaction rates for each reactions. Again, $\{c_1, c_2, c_3\}$ are the state space shifting constants, and $\{\sigma, \rho, \beta\}$ are parameters of the original systems.

where σ, ρ and β are real valued constants, we set the values of the constants as $\sigma = 10, \rho = 28, \beta = 8/3$, with these values the chaotic behavior of Lorenz system is reported in several papers. In each simulation, we started the system from $x_0 = \begin{bmatrix} 30, & 30, & 30 \end{bmatrix}$. In the sequel we transform the shifted system into the kinetic form with the above discussed method.

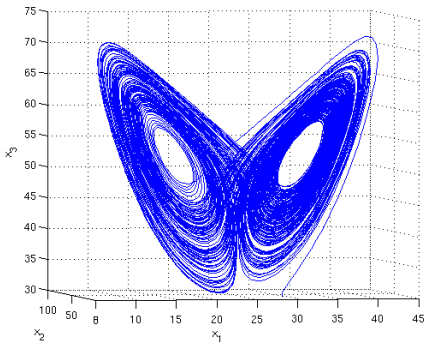


Figure 2. The State-space behavior of the chemical realization of the Lorenz system. The Figure shows that the chemical Lorenz system exhibits the qualitatively the same state-space behavior as the original system, but the state trajectory remains in the positive orthant. A corresponding parameters are: $\sigma = 10$, $\gamma = 28$, $\beta = 8/3$, $c_1 = 24$, $c_2 = 25$, $c_3 = 26$, $x_0 = [30, 30, 30]$

First, we shifted the state-variables into the positive orthant. To agree with (19) and to avoid cancellation in the last monome in the first equation of the Lorenz system we choose the shifting constant as $C = \begin{bmatrix} 24, & 25, & 26 \end{bmatrix}$

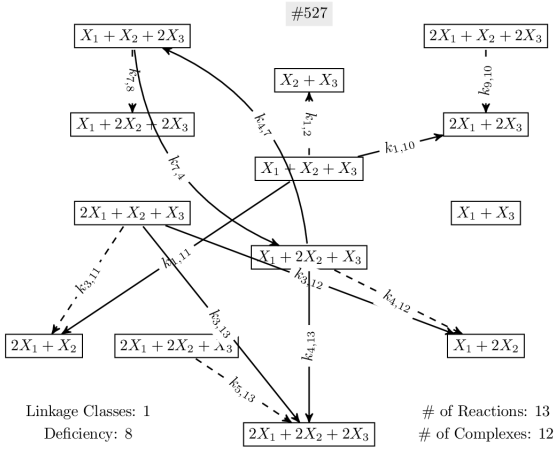


Figure 3. This realization shows only 11 complexes are enough to generate the dynamics of a Lorenz system. The dotted edges are the core reactions, the parameters on the edges are the corresponding elements from the A_k matrix.

A. State-dependent Time scaling

We applied the time scaling with a specific monomial, namely $\chi_{1,2,3} = xyz$. This yields the following equations:

$$\begin{aligned} x &= \sigma xy^2z - \sigma x^2yz + \sigma(c_1 - c_2)xyz \\ y &= \rho x^2yz + (c_2 - c_1\rho - c_1c_3)xyz \\ &\quad -xy^2z - x^2yz^2 + c_3x^2yz + c_1xyz^2 \\ z &= x^2y^2z - c_2x^2yz - c_1xy^2z \\ &\quad +(c_1c_2 + \beta c_3)xyz - \beta xyz^2 \end{aligned} \tag{23}$$

With algorithm in [6] we can transform this system into CRN, the result is shown on Figure 1. From now on, the three species $\{X, Y, Z\}$ will be noted as $\{X_1, X_2, X_3\}$ respectively. The corresponding reaction network are given on Figure 1, it also shows the reaction rate constants on the edges. The state space behavior of the CRN are shown on Figure 2, which shows the solution remains in the positive orthant, this property reflects the fact that the concentration of chemical species cannot be negative.

At this points we have reached the canonical reaction kinetic of the 3-dimensional Lorenz system. To compute the all possible sparse structures of this CRN, we need further information such a sparse realization, the dense realization and the core reaction set. Using the computation method introduced in section II-B, we calculated the corresponding data. According to that sparse realization contains $R_s = 13$ reactions, whereas the dense has $R_d = 51$. The number of core reactions are $R_c = 6$, these reactions can be seen on Figure 1 and 3 with dotted edges.

B. Computation of all Possible Sparse Structures

To compute all the possible dynamically equivalent sparse realization of the reaction kinetic Lorenz system the following should be considered. The dense realization of a CRN is a unique realization and contains all the possible equivalent realization as sub-graphs. Our goal here is to select minimum number of reactions (edges) from the dense graph in a way that the resulting CRN still fulfills the dynamical equivalence properties. But we have to do this in that way the core reactions are remain in the graph as a conserved structure. To solve this problem we delete as many edges from dense graph as difference between reaction number in the sparse and dense realization: $R_d - R_s$ and we have to do this in the all possible combination,

$$\begin{pmatrix} R_d - R_c \\ R_d - R_s \end{pmatrix} = \frac{(R_d - R_c)!}{(R_d - R_s)!(R_s - R_c)!} \tag{24}$$

After substituting the corresponding numbers into (24), it yields $45.379.620$ possible graphs, this means that we have to solve the linear program stated in section II-B that many times to find feasible realizations. Clearly, that means a lot of time even on a high-end computer. To workaround the problem, we identified some reaction pairs, which are non-core reactions, but if none of them are present in the reaction graph, then the corresponding dynamically equivalent realization cannot be sparse. This is due to the fact that the exclusion such a reaction pair will brings more reactions into the graph than what we have excluded. Hence, it is clearly violates the sparsity constraints. The reaction pairs with the above properties are $\{\{C_3 \to C_5, C_3 \to C_{13}\}, \{C_4 \to C_5, C_4 \to C_{13}\}, \{C_1 \to C_6, C_1 \to C_{10}\}, \{C_1 \to C_6, C_1 \to C_{11}\}\}$ With that information we can omit those possibilities that exclude the above reaction pairs and this procedure significantly reduce the search space, to be exact only 442.454 graphs must be checked in that case. After evaluating the graphs with the linear program from section II-B, 5376 realizations are found feasible. This means that with fixed the complex set and the maximum number of reactions for a given dynamics, we can tell how many sparse CRN can produce exactly the same dynamics.

C. Comment on the results

Now, we have all the possible sparse realization of the kinetic Lorenz system we can investigate the properties of the different CRN realizations. From the previous analysis we know that the kinetic Lorenz dense and sparse realizations have $R_d = 51$ and $R_s = 13$ reactions, respectively. There is also $R_c = 6$ core reactions in the CRN. Among the possible sparse realizations the minimum and maximum linkage classes are 1 and 3, respectively. More importantly, none of the realizations is weakly reversible, the implication of that fact can be found in [12].

V. Conclusion

We have transformed a polynomial ODE system into a chemical kinetic system, the corresponding reaction graph has been given. We also have shown that the chemical kinetic version of the original polynomial system exhibits the same state-space behavior, but its trajectory confined into the positive orthant. Furthermore, with an optimization-based algorithm we have calculated the dynamically equivalent realizations of the canonical CRN with prescribed properties, e.g. minimal or

maximal number of reactions in the reaction network. Based on the fact that sparse realization of the CRN is not unique, but the dense is, we have calculated all the sparse realization of the canonical system and investigated their properties.

REFERENCES

[1] G. Craciun and C. Pantea. Identifiability of chemical reaction networks. *Journal of Mathematical Chemistry*, 44:244–259, 2008.

[2] I. R. Epstein and J. A. Pojman. *An Introduction to Nonlinear Chemical Dynamics: Oscillations, Waves,Patterns and Chaos (Topics in Physical Chemistry)*. Oxford University Press, 1998.

[3] M. Feinberg. *Lectures on chemical reaction networks*. Notes of lectures given at the Mathematics Research Center, University of Wisconsin, 1979.

[4] K. Furuta. On time scaling for nonlinear systems: Application to linearization. *Automatic Control, IEEE Transactions on*, 1986.

[5] F. Horn and R. Jackson. General mass action kinetics. *Archive for Rational Mechanics and Analysis*, 47:81–116, 1972.

[6] V. Hrs and J. Tth. On the inverse problem of reaction kinetics. In M. Farkas and L. Hatvani, editors, *Qualitative Theory of Differential Equations*, volume 30 of *Coll. Math. Soc. J. Bolyai*, pages 363–379. North-Holland, Amsterdam, 1981.

[7] P. rdi and J. Tth. *Mathematical Models of Chemical Reactions. Theory and Applications of Deterministic and Stochastic Models*. Manchester University Press, Princeton University Press, Manchester, Princeton, 1989.

[8] E. Szdeczky-Kardoss and B. Kiss. On-line trajectory time-scaling to reduce tracking error. In *Intelligent Engineering Systems and Computational Cybernetics*, pages 3–14. Springer, 2009.

[9] G. Szederknyi. Computing sparse and dense realizations of reaction kinetic systems. *Journal of Mathematical Chemistry*, 47:551–568, 2010.

[10] G. Szederknyi and K. M. Hangos. Finding complex balanced and detailed balanced realizations of chemical reaction networks. *Journal of Mathematical Chemistry*, 49:1163–1179, 2011.

[11] G. Szederknyi, K. M. Hangos, and T. Pni. Maximal and minimal realizations of reaction kinetic systems: computation and properties. *MATCH Commun. Math. Comput. Chem.*, 65:309–332, 2011.

[12] G. Szederknyi, K. M. Hangos, and Zs. Tuza. Finding weakly reversible realizations of chemical reaction networks using optimization. *MATCH Commun. Math. Comput. Chem.*, 67:193–212, 2012.

Distributed solution of large MILP problems applied to the analysis and control of complex dynamical systems

János Rudan
(Supervisors: Ton van den Boom, Gábor Szederkényi and Katalin Hangos)
rudan.janos@itk.ppke.hu

***Abstract*—In this paper a set of techniques are presented to solve Mixed Integer Linear Programming (MILP) problems for the model predictive control of complex dynamical systems. Our aim is to reduce the amount of dealy in railway networks using dynamic traffic management by the rescheduling of trains. Due to the size of the emerging MILP problem and the given constraints on solution time, a thorough analysis of different MILP solution techniques was necessary. It has been proven that a significant speedup of the solution time can be achieved by the proper restructuring of the matrices of the MILP problem. The simulation results also confirm the effectiveness of the proposed control technique.**

***Keywords*-Mixed Integer Linear Programming; railway scheduling; optimization based control**

I. Introduction

Controlling complex dynamical systems with the help of optimization techniques is a well-known technique. Especially the developments in the field of Model Predictive Control (MPC) gives us the opportunity to exploit the advantages of these approaches. Scheduling problems are emerging when the synchronization of the work of several agents is necessary because of the limited capacity of some resources in the system.

In this paper, a model predictive framework is used which gives us the ability to reformulate the dynamic traffic management of railway networks as a Mixed Integer Linear Programming (MILP) problem [1], [2]. The model is formulated in a way that the the order of the trains using the critical resource (namely the tracks) is controlled by binary variables. During the optimization this set of binary variables is determined while minimizing the given cost function.

The proposed technique is capable of predicting a given network's future behavior in case of a cyclic timetable and find an optimal rescheduling of the trains to minimize the summarized delay in the network along the prediction horizon.

II. Formulating the model

Consider a railway network having a periodic timetable with cycle time T. In case of nominal operation the trains follow a pre-scheduled route repeated in every T minutes where the order of the trains on the tracks is naturally defined by the schedule. If a delay is introduced into the network forcing it to deviate from the nominal operation, we call the resulting schedule as perturbed mode.

In this section only a short review of the model will be discussed. More detailed explanation of the model (constraints, definition of the model predictive control task, MILP-formulation) can be found in [3], [4].

A. Modelling the trains and the tracks

Let's say train i is moving on (virtual) track i, which begins at (virtual) station i. Let us note that some of the virtual tracks or stations could refer to the same physical track or station meaning that the number of physical tracks in the network usually smaller than the number of virtual tracks.

Let $d_i(k)$ be the departure time of train i from its departure station in the kth cycle, and let $a_i(k)$ be the arrival time of the same train to the other end of the given track. Let $r_i^d(k)$ and $r_i^a(k)$ be the scheduled (nominal) departure and arrival time of the same train, respectively.

B. Building up the constraint set

The departure and arrival times of the trains have to meet with the following constraints:

- *Time schedule constraint:* a departure may not occur before its scheduled departure time, so we have to satisfy the timetable constraint

$$d_i(k) \geq r_i^d(k) \tag{1}$$

  where $r_i^d(k)$ is the scheduled departure time for the ith train in the kth cycle. For the arrival time a similar constraint can be expressed:

$$a_i(k) \geq r_i^a(k) \tag{2}$$

- *Running time constraint:* let $t_i(k)$ be the running time of the ith train. The running time constraint then becomes

$$a_i(k) \geq d_i(k - \delta_{ii}) + t_i(k) \tag{3}$$

  where $\delta_{ii} = 0$ if the departure time and the arrival time are in the same cycle and $\delta_{ii} = \mu$ if there is μ cycle difference between departure and arrival.
- *Dwell time constraint:* let p_i be the preceding train of train i, which means that train i and p_i are physically the same train. Let $s_{p_i}(k)$ be the minimum dwell time between arrival of train p_i and the departure of train i, then we have to satisfy the dwell time constraint

$$d_i(k) \geq a_{p_i}(k - \delta_{ip_i}) + s_{p_i}(k) \tag{4}$$

where $\delta_{ip_i} = \mu$ if the $(k-\mu)$th train p_i arriving at the physical station corresponding to virtual station i continues as the kth train i.

- *Headway constraints:* let $\mathcal{F}_i(k)$ be the set of trains that move over the same track and in the same direction as train i, and are scheduled before train i. Let $j \in \mathcal{F}_i(k)$ and let h_{ij} denote the minimum headway time between train j and train i. For each train $j \in \mathcal{F}_i(k)$ we have headway constraints for both departure and arrival

$$\begin{aligned} d_i(k) &\geq d_j(k-\delta_{ij}) + h_{ij} + u_{ij}(k)\beta,\ \forall j \in \mathcal{F}_i(k) \\ d_j(k-\delta_{ij}) &\geq d_i(k) + h_{ij} + (1-u_{ij}(k))\beta,\ \forall j \in \mathcal{F}_i(k) \end{aligned} \tag{5}$$

$$\begin{aligned} a_i(k) &\geq a_j(k-\delta_{ij}) + h_{ij} + u_{ij}(k)\beta,\ \forall j \in \mathcal{F}_i(k) \\ a_j(k-\delta_{ij}) &\geq a_i(k) + h_{ij} + (1-u_{ij}(k))\beta,\ \forall j \in \mathcal{F}_i(k) \end{aligned} \tag{6}$$

where β is a large negative number (so $\beta \ll 0$) and $u_{ij}(k)$ is a binary variable. Note that for $u_{ij}(k) = 0$, the constraints (5.a) and (6.a) will be active, and for $u_{ij}(k) = 1$, the constraints (5.b) and (6.b) will be active.

- *Meeting constraints:* let $\mathcal{W}_i(k)$ be the set of trains that move over the same track and in the opposite direction as train i, and are scheduled before train i. Let $j \in \mathcal{W}_i(k)$ and let w_{ij} denote the minimum separation time between arrival of train j and departure of train i. For each train $j \in \mathcal{W}_i(k)$ we have separation constraint

$$\begin{aligned} d_i(k) &\geq a_j(k-\delta_{ij}) + w_{ij} + u_{ij}(k)\beta,\ \forall j \in \mathcal{W}_i(k) \\ d_j(k-\delta_{ij}) &\geq a_i(k) + w_{ij} + (1-u_{ij}(k))\beta,\ \forall j \in \mathcal{W}_i(k) \end{aligned} \tag{7}$$

where the role of β and $u_{ij}(k)$ is similar to the previous case.

It should be noted that during nominal operation all δ_{ij} values are equal to zero or one, but in perturped operation other values are also possible. For sake of simplicity, we will consider only the case when $\delta_{ij} = \{0, 1\}$ which means that $\mu_{max} = 1$.

C. Model predictive control and MILP formulation

As it is defined previously, let us consider a network having n train runs (or shortly trains), and define the following vectors:

$$z(k) = \begin{bmatrix} d_i(k) \\ \vdots \\ d_n(k) \\ a_1(k) \\ \vdots \\ a_n(k) \end{bmatrix} \in \mathbb{R}^{2n}; \quad r(k) = \begin{bmatrix} r_1^d(k) \\ \vdots \\ r_n^d(k) \\ r_1^a(k) \\ \vdots \\ r_n^a(k) \end{bmatrix} \in \mathbb{R}^{2n} \tag{8}$$

and the elements $u_{i,j}(k)$ with $j \in \mathcal{F}_i(k)$ or $j \in \mathcal{W}_i(k)$ can be stacked in one vector $u(k) \in \mathbb{R}^{n_u}$.

The goal of the model predictive controller is to minimize the sum of all delays, and so we come to the object function (or performance index):

$$J(k) = \sum_{j=0}^{N_p} \left(\sum_{i=1}^{n} \sigma_i \left(z_i(k+j) - r_i(k+j) \right) + \sum_{l=1}^{n_u} \rho_l\, u_l(k+j) \right) \tag{9}$$

Here N_p is the prediction horizon, and σ_i, ρ_l are weighting scalars. The first term of (9) is related to the sum of all predicted departure and arrival delays, and the second term denotes the penalty for all train orders changes and broken connections during cycle $k+j$.

Now define the extended vectors

$$\tilde{z}(k) = \begin{bmatrix} z(k) \\ z(k+1) \\ \vdots \\ z(k+N_p) \end{bmatrix} \in \mathbb{R}^{2nN_p}; \tilde{u}(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p) \end{bmatrix} \in \mathbb{R}^{n_u N_p} \tag{10}$$

The object function can be rewritten as:

$$J(k) = c^T \begin{bmatrix} \tilde{z} \\ \tilde{u} \end{bmatrix} \tag{11}$$

where c contains all the values σ and ρ. Also the network constraints (1)-(7) can be written using the vectors $\tilde{z}$ and $\tilde{u}$ and we obtain:

$$\begin{bmatrix} A_z & A_u \end{bmatrix} \begin{bmatrix} \tilde{z} \\ \tilde{u} \end{bmatrix} \leq b(k) \tag{12}$$

where matrix A_z only consists of entries $[A_z]_{ij} \in \{-1, 0, 1\}$, matrix A_u only consists of entries $[A_u]_{ij} \in \{-\beta, 0, \beta\}$ and $b(k)$ contains the schedule times $r_d(k)$, $r_a(k)$, the running time $t(k)$, the dwell times $s(k)$, the heading times $h(k)$, the separation times $w(k)$ and the past values $z(k-j)$, $j = 1, \ldots, \mu_{\max}$, where $\mu_{\max}$ is the largest cycle shift μ in the system.

We have now recasted the model predictive control problem into a MILP structure where the objective function is defined in (11) and the constraints in (12).

III. Simulation results

Simulations were done using the model of the Dutch railway system, with a cycle time $T = 60min$. The network model consists of 41 stations and 118 tracks. In the timetable only the international and interregional trains are included, the local trains are excluded. For the simulations $N_p = 2$ is selected, so the prediction is done for 2 hours in the future.

The generated problem has 26712 constraints and 5316 variables from which 1728 are continous variable (departure and arrival times) and the remaining 3588 are binary.

A. Scenario generation

In order to simulate the pertubed operation of the system, a new parameter vector Θ_p is generated containing all the deviations from values in the original schedule described by Θ. It is shown in [5] that the distribution of delays appearing in a train network show a Weibull distribution. In our case delays were generated according to a Weibull-distribution having

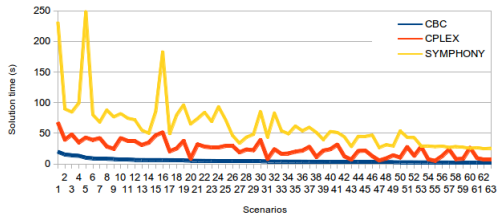


Fig. 1: Comparsion of solution times in case of different MILP problems. CBC outperforms all the other solvers. In the scenarios 8% of trains were selected to be delayed.

shape parameter 0.8 and scale parameter 5. Maximal delay was set to $40min$ and the average introduced delay to $10min$.

A given percentage of the trains were selected from all the trains and the delay values were added to their departure and arrival data, according to two different scenarios:

- *case 1*: 8% of the trains were selected to be delayed.
- *case 2*: 20 % of the trains were selected to be delayed.

We used *case 1* in simulations during the selection of the solver (see Section III-B) and we used *case 2* in other tests.

It should be noted that according to available data on average 7.5% of trains are delayed in real life in the Dutch railway network which means that *case 2* is related to a much more complex scenario than normal cases.

B. Selecting the proper MILP solver

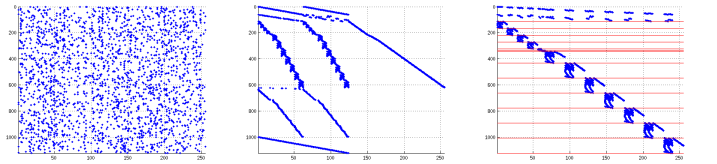
There are several available MILP solvers both originating from the free software community and commercial ones [6]. Comparsion of the solvers considering the used techniques and algorithms can be found in [7]. Another interesting and continously maintaned review of the solver's performance can be found at [8].

State-of-art solvers implements stronly heuristic-driven branch-and-cut algorithms to solve MILP problems [9]. Because of their crucial role in the solution speed, heuristics applied in MILP solvers are the subject of extensive research in the recent years [10], [11].

During the present work the following solvers were investigated: from the COIN-OR [12] community: DIP (version 0.83.2), SYMPHONY (version 5.4.4), CBC (version 2.7.6). Also GLPK (version 4.32) and CPLEX (version 12.1) were involved in the tests. The selection process was based on two criteria: on one hand the ability of parallel solution and on the other hand the average solution time.

The solvers were compared based on their average solution time processing a given problem set. The method of the problem generation is detailed in Section III. DIP and GLPK were unable to solve any of the problems in the given time limit. The results of the three remaining solvers can be seen in Figure 1.

Based on the results CBC was selected as the fastest solver. According to this we used CBC to complete all the other tests and simulations.



(a) Randomly ordered (b) Partially ordered (c) Track-based ordering. Borders of the blocks are marked.

Fig. 2: Structure of the constraint matrix where the variables are on the x-axis and the equations are on the y-axis. As it can be seen in Fig. 2c. the result of the track-based ordering is a clear block-angular structure.

C. Track-based reordering of the constraint matrix

In order to speed up the solution, the constraint matrix has been reordered on a per-track basis.

Reordering on track-basis means that we collect into one block all the constraints and control variables corresponding to a given track. It is clear that the order of the trains on one track is independent of the order on another track. By reordering the constraint set on track-basis we can see similar structure in the MILP formulation. The different constraint blocks corresponding to a given track are independent from each other, and the blocks themselves are connected via the continuity contstraints.

In Fig. 2. the reader can see the result of the track-based ordering approach. The formulated blocks can be seen clearly. Of course any other algorithm generating the constraint set could give some structure to the constraint matrix (as it can be seen on Fig. 2b.) but, among the examined cases the track-based approach gives the best result. The increase in solution speed gained by the proposed reordering is detailed in Secion III-D.

D. Effect of reordering on the solution time

As it was mentioned before, the effect of proper restructuring of the given MILP problem can have huge impact on the solution time. In the present work we selected a track-based ordering which leads to a clear block-angular structure. The following results are verifying that even in the presence of the solver's preprocessor serious gain can be achieved with proper problem formulation.

In the tests we considered two case. First the columns of the formulated matrices were mixed up randomly. In the other case we built up the same problem's matrix based but with track-based ordering (see Sec. III-C).

The solution times in both cases for several different scenario can be seen in Fig. 3 for several different scenarios. In the presented case the test set consists of 97 different scenarios. The average solution time was $23,97sec$ (std. dev. $14,51sec$) without and $17,11sec$ (std. dev. $10,13sec$) with reordering so the time gain obtained by the reordering was $6.85sec$ on average (std. dev. $7.04sec$) which means a speedup ratio $1,407$.

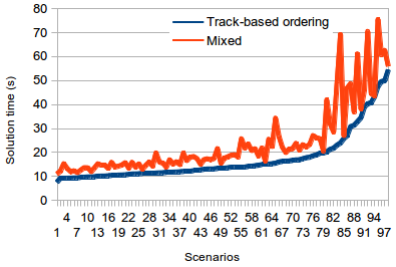


Fig. 3: Solution times in case of a random constraint matrix structure or in case of the track-based reordering. The simulation results show that even in the presence of the solver's preprocessor proper reordering can yield to significant speedup.

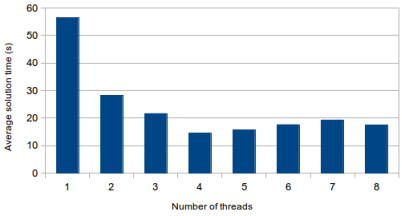


Fig. 4: Average elapsed time of solving a MILP with multiple threads on a 4-core (8 cores with Hyper-Threading) computer. Using 4 threads gives the best result.

E. Results in case of distributed solution

As it is mentioned in Section III-B. several available solvers offers the capabiliy of parallel solution of MILP problems [13].

The simulation result using CBC shows the advantage of the distributed solution. In Fig. 4. the average solution time of a problem set having 100 elements can be seen in case of single-thread and multi-thread setups.

During the simulations we used a 4-core computer. The maximal speedup can be achieved if the number of threads is equal to the number of physical cores in the processor.

F. Effectiveness of the proposed control technique

To measure the performance of the control technique, the following test were completed. Having a given scenario with initial (primary) delays we have simulated the uncontrolled behaviour of the network. During this open-loop simulation all control inputs were set to zero. At the end of the prediction horizon the difference between the result and the reference schedule is calculated as the sum of the delays. For the same scenario we have done the simulation and the calculation of the sum of the delays in controlled mode, too.

The comparsion of the sum of the delays shows the performance of the control techniques as it can be seen in Fig. 5. We found that the proposed technique can decrease the sum of the delays with 30% compared to the uncontrolled case.

IV. Summary and Future Work

In the present paper the details of a previously proposed control technique [4] were investigated. In this framework a general scheduling problem is considered as a MILP problem and solved in an MPC architecture. Due to the computational complexity of the emerging MILP problem and the time constraints on the solution it was necessary to speed up the solution with proper handling of the original problem.

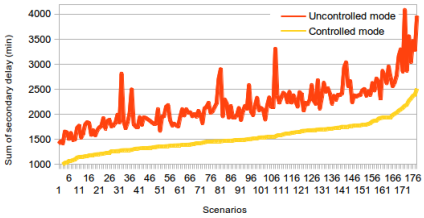


Fig. 5: Effect of optimal reordering of the trains in case of delays. As it can be seen the sum of the delays in the network can be significantly reduced over the control horizon with the help of the proposed control technique.

Beyond the selection of a proper solver to solve the MILP problems, the importance of the track-based reordering and the effectiveness of the proposed control technique is shown.

The general approach - namely the formulation of a scheduling problem as a MILP problem in an MPC framework - could be utilized in the optimal control of complex nonlinear systems by using the PWA approximation of the system and the modelling of reaction kinetic networks.

References

[1] S. P. Bradley, A. C. Hax, and T. L. Magnanti, *Applied Mathematical Programming*. Addison-Wesley, 1977.

[2] J. W. Chinneck, *Practical Optimization: A Gentle Introduction*. Carleton University, 2009.

[3] T. van den Boom and B. D. Schutter, "On a model predictive control algorithm for dynamic railway network management," in *2nd International Seminar on Railway Operations Modelling and Analysis (RailHannover2007)*, 2007.

[4] T. J. van den Boom, N. Weiss, W. Leune, R. M. Goverde, and B. D. Schutter, "A permutation-based algorithm to optimally reschedule trains in a railway traffic network," in *IFAC World Congress*, 2011.

[5] J. Yuan, "Capturing stochastic variations of train event times and process times with goodness-of-fit tests," Delft University of Technology, Tech. Rep., 2007.

[6] A. Lodi and J. T. Linderoth, *Encyclopedia for Operations Research and Management Science*. Wiley, 2011, ch. MILP Software.

[7] J. T. Linderoth and T. Ralphs, "Noncommercial software for mixed-integer linear programming," in *Integer Programming: Theory and Practice*, J. Karlof, Ed. CRC Press, 2005.

[8] *Benchmarks for Optimization Software*. [Online]. Available: http://plato.asu.edu/bench.html

[9] M. Galati, "Decomposition methods for integer linear programming," Ph.D. dissertation, Lehigh University, 2010.

[10] B. Gendron and T. G. Crainic, "Parallel branch and bound algorithms: Survey and synthesis," *Operations Research*, vol. 42, pp. 1042–1066, 1994.

[11] L. Bertacco, M. Fischetti, and A. Lodi, "A feasibility pump heuristic for general mixed-integer problems," University of Padova, Italy, Tech. Rep. Technical Report OR-05-5, 2005.

[12] *COmputational INfrastructure for Operations Research*. [Online]. Available: http://coin-or.org

[13] T. K. Ralphs, *Parallel Combinatorial Optimization*. Wiley, 2006, ch. Parallel Branch and Cut, pp. 53–101.

Efficient mapping of the Sphere Detector Algorithm on GPU Based on Complexity Analysis

Csaba M. Józsa
(Supervisor: Géza Kolumbán)
jozsa.csaba@itk.ppke.hu

***Abstract*—Data rate cannot be improved significantly in single-input single-output (SISO) systems used currently because of the strict bandwidth requirements. A promising solution to significant increase of bandwidth efficiency, transmission capacity and system robustness is the exploitation of the spatial dimension. Multiple–input multiple-output (MIMO) technology has attracted attention in wireless communications, because it offers significant increases in data throughput and link range without additional bandwidth or increased transmit power. The implementation of wideband MIMO system posts a major challenge to hardware designers due to the huge computing power required for MIMO detection. With the help of the General Purpose Graphical Processing Unit (GP-GPU) the computing power is not a limiting factor anymore. In this paper after introducing the basic model of the MIMO system, the sphere detector algorithm is presented, finally the efficient mapping of the sphere detector algorithm to GP-GPU architecture is given.**

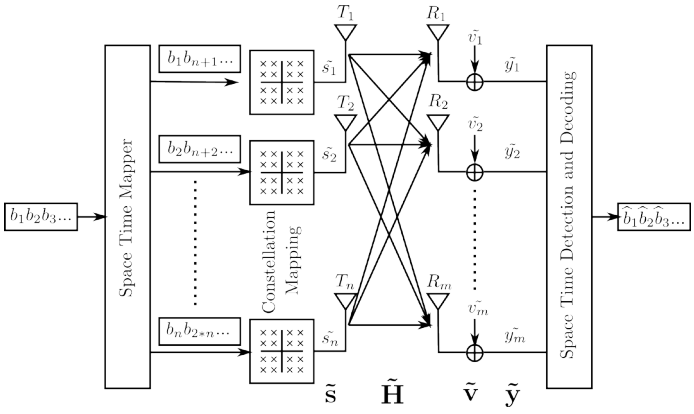


Figure 1. MIMO system model

I. Introduction

The challenges in wireless communications systems are: increasing the link throughput (i.e., bit rate) and the network capacity. The limiting factors of such systems are usually, the equipment cost, the radio propagation and the frequency spectrum. However to fulfill the above goals, future systems should be characterized by improved spectral efficiency. Research in the information theory, has revealed that important improvements in information rate can be achieved when multiple antennas are applied at both the receiver and transmitter side. The key feature of multiple-transmit multiple-receive antenna, i.e. Multiple-Input Multiple-Output (MIMO), systems is the ability to turn multipath propagation, traditionally a pitfall of wireless transmission, into a benefit for the user. MIMO effectively takes advantage of random fading and when available, multipath delay spread, for multiplying transfer rates. The success of MIMO lies in the fact that the performance of wireless systems are many orders of magnitude improved at no cost of extra spectrum only complexity is added to the different algorithms and hardware. The MIMO techniques can increases the robustness of the wireless communication system by transmitting different representations of the same data stream (by means of coding) on the different transmit branches, or they can achieve a higher throughput by transmitting independent data streams on the different transmit branches simultaneously and at the same carrier frequency. However the complexity of the detector algorithms used in the receiver structures is defined by many factors (coding, channel, antenna mapping) there are some cases, when the detection cannot be done by simple hardware components, instead it's worth using many-core architectures such GP-GPUs, or field programmable gate arrays (FPGAs).

II. System Model

The block diagram of an $n \times m$ MIMO system is given in Figure 1, where n denotes the number of transmit antennas, and m denotes the number of receive antennas. The transmit antennas are sending a complex signal vector of size n. The complex signal vector $\tilde{s}$ is transmitted during one symbol period. Assuming rich-scattering and flat-fading channel over one symbol period the system model is given as follows:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{s}} + \tilde{\mathbf{v}} \tag{1}$$

where $\tilde{\mathbf{s}} = [\tilde{s}_1, \tilde{s}_2, .., \tilde{s}_n]^T$ is the transmitted symbol vector, having each component drawn from a complex constellation, $\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_{N_r}]$ is the received complex symbol vector, and $\tilde{\mathbf{v}} = [\tilde{v}_1, \tilde{v}_2, ..., \tilde{v}_{N_r}]$ is an independent and identically distributed(i.i.d) $CN(0, K)$ circular symmetric complex multivariate random variable, where the covariance matrix $K = \sigma_n^2 I_n$, and the entries $\tilde{h}_{ij}$ of the channel matrix $\tilde{H}$ are assumed to be i.i.d zero-mean complex Gaussian variables with unit variance. Without any loss of generality we assume that $n = m$ and the channel matrix has full rank. Furthermore we assume perfect channel state information(CSI), i.e. the channel matrix $\tilde{\mathbf{H}}$ is known, at the receiver side.

The optimal solution for the system model (1) is:

$$\tilde{s}_{ML} = \underset{\tilde{s}\epsilon\Omega^n}{\operatorname{argmin}} \|\tilde{y} - \tilde{H}\tilde{s}\|^2 \tag{2}$$

where Ω is the set of the complex symbols.

Because the sphere detector algorithm is more efficient using real values, we transform the original complex representation of the system model (1) into a real valued system model, at the cost of increasing dimension:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{v} \tag{3}$$

where $\mathbf{y}_{2\times n,1} = \begin{bmatrix}\Re(\tilde{y})\\ \Im(\tilde{y})\end{bmatrix}$, $\mathbf{s}_{2\times n,1} = \begin{bmatrix}\Re(\tilde{s})\\ \Im(\tilde{s})\end{bmatrix}$, $\mathbf{v}_{2\times m,1} = \begin{bmatrix}\Re(\tilde{v})\\ \Im(\tilde{v})\end{bmatrix}$, and $\mathbf{H}_{2\times m,2\times n} = \begin{bmatrix}\Re(\tilde{H}) & -\Im(\tilde{H})\\ \Im(\tilde{H}) & \Re(\tilde{H})\end{bmatrix}$.

For the real valued system the ML metric is:

$$s_{ML} = \underset{s\epsilon\Omega^{2\times n}}{\operatorname{argmin}} \|y - Hs\|^2 \tag{4}$$

where $y, H, s, \hat{s}$ are real valued. After the transformation the symbol vector s will be real valued which implies that the symbols in Ω are also real valued. From equation (4) it can be seen that the maximum likelihood estimate of the symbol vector is given by solving an integer least-squares(ILS) problem, which is analogous [1] of finding the closest lattice point of lattice $\Lambda = \{Hs : s\epsilon\mathbb{Z}^n\}$ to a given point y [2]. In lattice theory this problem is often called the closest lattice point search (CLPS) [3], [4]. The exhaustive search implementation of ML decoding, or the enumeration of all lattice points has a complexity that grows exponentially with the size of Ω or with the number of the antennas, thus the required computational performance will be unattainable. For general lattices the problem was shown to be NP-hard([5]). However significant complexity reduction can be achieved by exploiting the structure of the lattice (e.g., [6], [7]). In the case of many wireless communication applications, i.e. detection, the integer symbol vector is uniformly distributed over a discrete and finite set $C \subset \mathbb{Z}^n$, which represents the transmitter codebook.

III. Sphere decoding

A. Problem Statement

The fundamental aim of the sphere detector algorithm is to reduce the search over only lattice points $s\epsilon\Omega^{2n}$ that lie in a certain sphere of radius d around a given vector. Reducing the search space and the required computations will not affect the performance of the detection, because the closest lattice point inside the sphere will also be the closest lattice point for the hole lattice. However, the reduction of the search space is necessary in order to deal with the high computational complexity required by the ML detection, there are several questions that have to be taken into consideration.

Taking the unconstrained least-squares solution $\hat{s} = (H^T H)^{-1} H * y$ of the real system shown in (3) and applying the QR factorization to the real channel matrix $H = QR$, the ML solution can be defined as follows:

$$\begin{aligned} s_{ML} &= \underset{s}{\operatorname{argmin}} \|y - Hs\|^2 \\ &= \underset{s}{\operatorname{argmin}} \|R(s - \hat{s})\|^2 \end{aligned} \tag{5}$$

where matrix Q is orthogonal and matrix R is upper triangular. The constraint condition $Hs \in S(y, d)$, where $S(y, d)$ denotes a hypersphere with center point y and radius d, can be formulated as follows:

$$\begin{aligned} &\|R(s - \hat{s})\|^2 \leqslant d^2 \\ &|r_{mm}(s_m - \hat{s}_m)|^2 + \\ &|r_{m-1m-1}(s_{m-1} - \hat{s}_{m-1}) + r_{m-1m}(s_m - \hat{s}_m)|^2 + \\ &\vdots \\ &|r_{11}(s_1 - \hat{s}_1) + r_{12}(s_2 - \hat{s}_2) + \cdots + r_{1m}(s_m - \hat{s}_m)| \leqslant d^2 \end{aligned} \tag{6}$$

Instead of enumerating all the possible symbol combinations, a recursion can be given based on the dependency hierarchy of the terms. In every iteration one symbol s_i is being selected based on the previous selections $s_{i+1}, \cdots, s_m$. Let $s_i^m \triangleq (s_i, s_{i+1}, \cdots, s_m)^T$ denote the last $m - i + 1$ components of the vector s. Considering vector s_{i+1}^m fixed, an interval $I_i(s_{i+1}^m) = \left[A_i(s_{i+1}^m), B_i(s_{i+1}^m)\right]$ can be defined, where

$$A_i(s_{i+1}^m) = \hat{s}_i - \frac{1}{r_{ii}} \left(\sqrt{d^2 - \sum_{j=i+1}^{m} \left| \sum_{k=j}^{m} r_{jk}(s_k - \hat{s}_k) \right|} - \sum_{l=i+1}^{m} r_{il}\left(s_l - \hat{s}_l\right) \right)$$

$$B_i(s_{i+1}^m) = \hat{s}_i + \frac{1}{r_{ii}} \left(\sqrt{d^2 - \sum_{j=i+1}^{m} \left| \sum_{k=j}^{m} r_{jk}(s_k - \hat{s}_k) \right|} - \sum_{l=i+1}^{m} r_{il}\left(s_l - \hat{s}_l\right) \right). \tag{7}$$

If $A_i(s_{i+1}^m) > B_i(s_{i+1}^m)$ or $I_i(s_{i+1}^m) \cap \Omega = \{\}$, means that there is no s_i that satisfies the inequality (6), thus the chosen s_{i+1}^m is outside the sphere $S(y, d)$.

The continuous change of vector s_i^m is analogous of a depth-first tree traversal process. Vector s_i^m can be regarded as a path at depth $m - i + 1$ and by defining path metric as follows:

$$M(s_i^m) = \sum_{j=i}^{m} \left| \sum_{k=j}^{m} r_{jk}(s_k - \hat{s}_k) \right|^2 \tag{8}$$

then each node in the tree can have a weight, thus the sphere decoding becomes a bounded tree search.

IV. Mapping the SD algorithm on GP-GPU architecture

A. Algorithm design principles

In Section III it has been shown, that sphere decoding can be traced back to a branch and bound tree search. Due to the decoding algorithm structure, the system specification and the parallel architecture it is plausible to define several levels of parallelization of the system. It is convenient to define the system level paralellization as decoding multiple symbols at the same time, namely each symbol will be assigned to a thread block in the grid. In ([8]) Khairy et al. showed that significant speed-ups can be achieved by executing multiple sphere decoders at once with the conventional sequential algorithm. However to achieve peak performance on the GP-GPU it is mandatory to redesign the sequential algorithm, taking into considerations the limitations imposed by the architecture exploiting it by using several effective paralell design patterns. In ([9]) presented a mapping of a soft MIMO detector on GPU, but they used several important simplifications which comes at the expense of the quality of the ML solution.

The complexity analysis of the SD algorithm has been seriously investigated by the researchers ([10], [11], [12], [13], etc.) because the exact solution of the decoding process outperforms even the best heuristics. It has been shown that the complexity of the sphere detector algorithm is directly proportional to the number of explored points. The search space is highly influenced by the chosen sphere radius. Choosing an inappropriate sphere radius will lead to an increased complexity, however finding the covering radius is NP hard. Defining the complexity of the sphere decoding algorithm with radius update analytically seems to be an open problem, however updating the size of the sphere radius to $d^2 = \|y - Hs_{in}\|$, where s_{in} is a lattice point inside the sphere, can lead to significant complexity reduction.

The two fundamental tree search algorithms are the breadth-first (BF) search and the depth-first (DF) search techniques. The sphere decoding algorithm is based on a DFS strategy. It is known that finding the least cost leaf in the case of a large tree could take excessively long time, but at the same time the DF search is the one trying to find a leaf node as fast as possible. In the case of the BF search the result will be the least cost node, but the memory requirements may be to excessive and the criteria of finding a lattice point inside the sphere as fast as possible are are not met. Lai et al. in [14] have examined different hybrid tree search algorithms and they could achieve significantly lower complexity with a moderate increase in the memory requirement. The speed-up was achieved because the decoding process started with a BF search, and it was continued with a DF search based on the extracted nodes branch metric, thus a leaf node can be found more efficiently.

B. Algorithm building blocks

As discussed in the previous section one of the most important motivating factors is to find a lattice point inside the sphere as fast as possible because then it is possible to adjust the sphere radius, thus the search space will become much smaller. A good searching strategy seems to be the combination of the BF search with the DF search in such a way that, after the BF search the further path exploration is based on the actual path metric. The challenge is to map these strategies on the GP-GPU taking in consideration the limits imposed by the hardware such as memory, throughput, latency.

The system level parallelism is equal of mapping each received symbol vector y to a thread block in the grid. Based on the chosen communication standard the size of the grid can be determined. Having a high number of thread blocks is important, because it keeps the GP-GPU utilized, in addition it can hide the latency appearing during the calculations.

The algorithm level parallelism is more difficult, but significant speed-up can be achieved by redesigning the sequential algorithm. In order to describe the parallel implementation of the sphere decoding (PSD) algorithm we introduce the following parameter notation:

1) $\boldsymbol{tt}$ - total number of threads, used in a thread block, decoding one received symbol vector (system parameter)
2) $\mathbf{t_{id}^{k}}$ - thread with identifier k in a given thread block
3) $\mathbf{wt_{lvl_x}}$ - the number of concurrently working threads on level lvl_x
4) $\mathbf{lvl_{nr}}$ - PSD algorithm parameter specifying the number of levels used for BF search strategy (i.e. 1,2,3,...)
5) $\mathbf{lvl_x}$ - levels assigned for BF search
6) $\mathbf{max_{lvl_x}}$ - the maximum number of paths on level lvl_x
7) $\mathbf{paths_{lvl_x}}$ - the number of symbol vectors selected on level lvl_x for simultaneous process with DF search strategy
8) $\mathbf{exp_{lvl_x}}$ - the number of simultaneously explored paths on level lvl_x
9) $\mathbf{mem_{lvl_x}}$ - memory required for level lvl_x
10) $\mathbf{s_{lvl_x}^{N<j>}}$ - symbol vector on level lvl_x, where $< j >$ is an optional parameter showing that the symbol vector is placed in buffer buf_{lvl_x} at index j
11) $\mathbf{M(s_{lvl_x}^{N<j>})} \equiv \mathbf{M_{lvl_x}^{<j>}}$ - the path metric of symbol vector $s_{lvl_x}^{N<j>}$, where $< j >$ is an optional parameter showing that the symbol vector's path metric is placed in buffer $bufPM_{lvl_x}$ at index j
12) $\mathbf{buf_{lvl_x}}$ - shared memory buffer containing the symbol vectors $s_{lvl_x}^{N<j>}$ for level lvl_x, where $0 \leq j < exp_{lvl_x}$
13) $\mathbf{bufPM_{lvl_x}}$ - shared memory buffer containing the path metric of the symbol vectors $M(s_{lvl_x}^{N<j>})$ for level lvl_x, where $0 \leq j < exp_{lvl_x}$
14) $\mathbf{vt_{lvl_x}^{i<j>}}$ - virtual thread identifier i on processing level lvl_x, where $< j >$ is an optional parameter denoting that the virtual thread identifier will form the symbol vector $s_{lvl_x}^{N<j>}$ at index j in buf_{lvl_x}
15) $\mathbf{vb_{lvl_x}^{i<j>}}$ - virtual block identifier i on processing level lvl_x, where $< j >$ is an optional parameter denoting that the virtual block identifier will form the symbol vector $s_{lvl_x}^{N<j>}$ at index j in buf_{lvl_x}

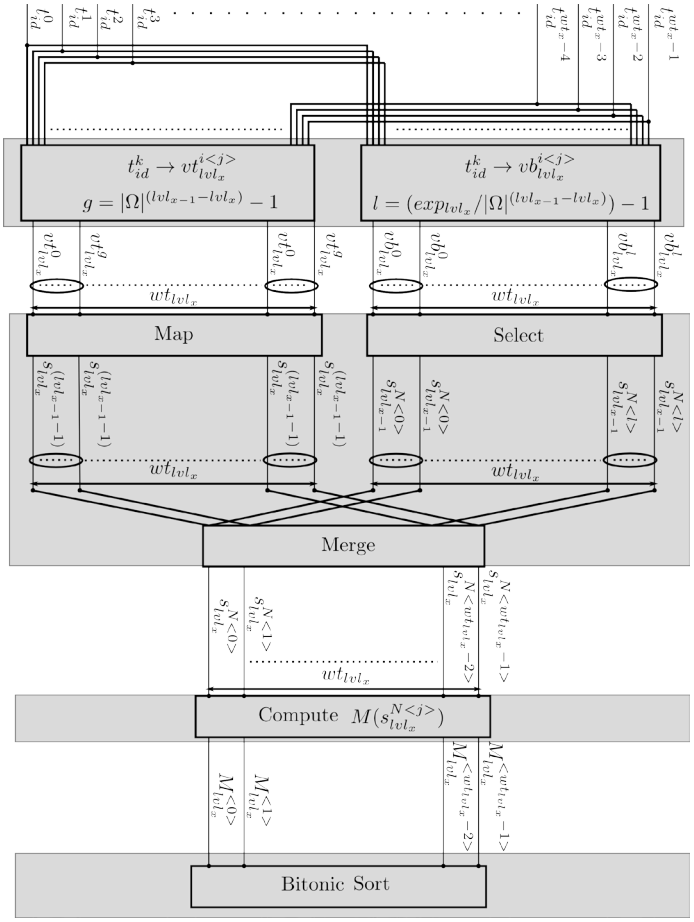


Figure 2. Parallel sphere detector *explore and evaluate* pipeline

Due to the high number of the working threads it is possible to simultaneously explore, fully or partially, a given level of the tree. During the decoding process only distinguished levels will be expanded in such a way that $N = lvl_0 > lvl_1 > lvl_2 > \cdots > lvl_{nr}$, where N denotes the depth of the tree. The number of total paths on lvl_x is $max_{lvl_x} = |\Omega|^{N-(lvl_x-1)}$, however the number of simultaneously explored paths exp_{lvl_x} is $exp_{lvl_x} = paths_{lvl_{x-1}} * |\Omega|^{(lvl_{x-1}-lvl_x)}$ determined by the number of symbol vectors $paths_{lvl_{x-1}}$ selected on level lvl_{x-1}. In the case if $paths_{lvl_{x-1}} = max_{lvl_{x-1}}$ holds, then $exp_{lvl_x} = max_{lvl_x}$ which means that a full BF search will take place on lvl_x.

The result of the path exploration has to be stored in memory, thus the memory required for a specific level is directly proportional with the number of explored paths $mem_{lvl_x} \sim K * exp_{lvl_x}$, where K is the size of the data structure representing one symbol vector and the associated path metric. This is really important because the memory requirements for this algorithm are influenced by the number of chosen paths at each level.

After considering the general conditions for the PSD algorithm, it is possible to define the main building blocks of the detection, which forms the *explore and evaluate* pipeline of the algorithm. One iteration of the detection process is presented in Figure 2., showing the main processing stages. A more detailed explanation is given in the following:

1) By definition for every level lvl_x the number of working threads is given by wt_{lvl_x}. In order to form a symbol vector $s_{lvl_x}^{N<j>}$ on level lvl_x, i.e. one branch of the tree starting from level lvl_0 to level lvl_x, parameters like $vt_{lvl_x}^{i<j>}$ and $vb_{lvl_x}^{i<j>}$ have to be defined. The amount of work given to a thread depends whether the conditions of the following inequality holds:

$$exp_{lvl_x} \leq wt_{lvl_x} \leq tt. \tag{9}$$

In the case when the number of working threads wt_{lvl_x} associated to level lvl_x are higher or equal than the number paths needed to be processed, then one thread is responsible for exploring and evaluating a single symbol vector, contrary one thread has to process at most $\lceil exp_{lvl_x}/wt_{lvl_x} \rceil$ symbol vectors. Assuming that exp_{lvl_x} is divisible by wt_{lvl_x}, two sets are defined for each thread with identifier t_{id}^k working on level lvl_x: $VT_{lvl_x}^k$ containing the virtual thread identifiers and $VB_{lvl_x}^k$ containing the set of the virtual block identifiers

$$\begin{aligned} VT_{lvl_x}^k = \{vt_{lvl_x}^{i<j>} | i = (t_{id}^k + n * wt_{lvl_x}) \\ \bmod |\Omega|^{(lvl_x - lvl_{x-1})}, \\ j = (t_{id}^k + n * wt_{lvl_x}), \\ n = 0 : \lceil exp_{lvl_x}/wt_{lvl_x} \rceil - 1\}, \end{aligned} \tag{10}$$

$$\begin{aligned} VB_{lvl_x}^k = \{vb_{lvl_x}^{i<j>} | i = \lfloor (t_{id}^k + n * wt_{lvl_x})/|\Omega|^{(lvl_{x-1} - lvl_x)} \rfloor, \\ j = (t_{id}^k + n * wt_{lvl_x}), \\ n = 0 : \lceil exp_{lvl_x}/wt_{lvl_x} \rceil - 1\}, \end{aligned} \tag{11}$$

Each thread has to compute its own set of identifiers on every level storing them in the thread's registers. This first stage, called the **preparatory phase** of the explore and evaluate pipeline, is completed when each thread computed the virtual identifiers.

2) In the second stage the task is to determine the symbol vectors $s_{lvl_x}^{N<j>}$ for the buffer buf_{lvl_x}, where $0 \leq j < exp_{lvl_x}$. A **selecting mapping and merging** is required in order to generate the symbol vector $s_{lvl_x}^{N<j>}$. After selecting a precomputed symbol vector $s_{lvl_{x-1}}^{N<l>}$, the virtual thread identifiers computed in the previous stage have to be mapped to partial symbol vectors $s_{lvl_x}^{lvl_{x-1}-1<j>}$ and finally the selected vector and the partial symbol vector have to be merged.
In the selecting phase $paths_{lvl_{x-1}}$ number of previously explored and evaluated symbol vectors $s_{lvl_{x-1}}^{N}$ are selected from lvl_{x-1}. The selection is done by each thread t_{id}^k based on the virtual block identifiers. Iterating over the elements $vb_{lvl_x}^{i<j>}$ of the set $VB_{lvl_x}^k$ a new set $S_{lvl_x}^k$ is formed for each thread. The elements of this set will be determined by the virtual block identifier $vb_{lvl_x}^{i<j>}$

in such a way that the symbol vector $s_{lvl_{x-1}}^{N<i>}$ will be selected from the buffer $buf_{lvl_{x-1}}$ at index i and it will be used to form the symbol vector $s_{lvl_x}^{N<j>}$ which will be placed in buffer buf_{lvl_x} at index j. In Figure 2. a special scenario is presented, namely when every thread has a one element set of virtual thread and block identifiers.
In the mapping stage we transform each element $vt_{lvl_x}^{i<j>} \in VT_{lvl_x}^k$ to a binary vector of size

$$s = \log_2 |\Omega| * (lvl_{x-1} - lvl_x),$$

using the conversion function:

$$B : (\mathbb{N}, s \in \{0, \cdots, N\}) \rightarrow \mathbb{B}^{s*\log_2 |\Omega|} = \{0,1\}^{s*\log_2 |\Omega|}.$$

Having defined the binary vector $b^s = B(vt_{lvl_x}^i \in VT, lvl_{x-1} - lvl_x)$ we can construct one element of the symbol vector $s_{lvl_x}^{lvl_{x-1}+1}$ by grouping the binary elements of vector b^s in groups of $\log_2 |\Omega|$ and defining a one to one mapping $M : \mathbb{B}^{\log_2 |\Omega|} \rightarrow \Omega$ function between the binary groups and the the symbol set Ω. The formula below shows the grouping and the one to one mapping:

$$s_i = M(b_{(1+i*\log_2 |\Omega|):((1+i)*\log_2 |\Omega|)}),$$

where $0 < i \leq (lvl_{x-1} - lvl_x)$.
In the merging process the result of the selection and mapping is merged, namely each selected vector $s_{lvl_{x-1}}^N$ and mapped symbol vector $s_{lvl_x}^{lvl_{x-1}-1}$ based on $vt_{lvl_x}^i$, is merged, thus forming the symbol vector:

$$s_{lvl_x}^N = (s_{lvl_x}, \cdots s_{lvl_{x-1}-1}, s_{lvl_{x-1}}, \cdots, s_N).$$

3) The advantage of the multi-threaded environment shows up in the **path metric evaluation** stage. The metric computation is one of the most time-consuming stages, but the fact that each thread computes the metric of a different path a significant speed-up can be achieved. Having buffered the path metric $M(s_{lvl_{x-1}}^N)$ of $s_{lvl_{x-1}}^N$ in $bufPM_{lvl_{x-1}}$ only the metric $M(s_{lvl_x}^{lvl_{x-1}-1})$ of the mapped symbol vector has to be computed, thus the path metric of the merged symbol vector can be computed as follows:

$$M(s_{lvl_x}^N) = M(s_{lvl_{x-1}}^N) + M(s_{lvl_x}^{lvl_{x-1}+1}).$$

4) The **sorting stage** is one of the most important stages during the detection. As discussed in Subsection IV-A the complexity of the algorithm can be significantly reduced after finding a leaf of the tree and adjusting the radius of the sphere. In the PSD algorithm this is achieved with the help of the sorting networks [15], [16]. Due to their data-independent structure, their operation sequence is completely rigid, what makes this algorithm parallelizable for the GP-GPU architecture. After the path metric evaluation stage the candidate paths $s_{lvl_x}^{N<k>}$ will be sorted based on their path metric and the next iteration continues with the selection of the $paths_{lvl_x}$ best number of them. This kind of greedy behavior and the hybrid searching strategy makes it possible to find a leaf node after several iterations. In the PSD algorithm one thread block is responsible for decoding one symbol, thus the data sets that have to be sorted are small enough to fit in the shared memory, so we are not limited by the global memory bandwidth.

The PSD algorithm kernel code is presented in IV-B.

Algorithm 1 The PSD algorithm GP-GPU kernel pseudo-code

Require: $\hat{s}, R, d^2, \Omega, N, lvl_{nr}$
1: shared $M(s_{ML}) \leftarrow \infty$, shared $s_{ML} \leftarrow \{\}$
2: **procedure** INIT
3: **for** $x = 0 \rightarrow lvl_{nr}$ **do**
4: Fetch from global memory to shared memory: lvl_x, $paths_x$, wt_x
5: shared $max_{lvl_x} \leftarrow |\Omega|^{N-(lvl_x-1)}$
6: shared $off_{lvl_x} \leftarrow 0$ ▷ Offset for lvl_x
7: **if** $x > 0$ **then**
8: shared $exp_{lvl_x} \leftarrow paths_{lvl_{x-1}} * |\Omega|^{(lvl_{x-1}-lvl_x)}$
9: shared $buf_{lvl_x}[exp_{lvl_x}]$
10: shared $bufPM_{lvl_x}[exp_{lvl_x}]$
11: **else**
12: shared $buf_{lvl_0} \leftarrow \Omega$
13: shared $bufPM_{lvl_0} \leftarrow M(\Omega)$
14: **end if**
15: **end for**
16: PROCESS(lvl_1,$off_{lvl_0} \leftarrow 0$)
17: **end procedure**
18: **procedure** PROCESS(lvl_x,$off_{lvl_{x-1}}$)
19: **if** $t_{id}^k \geq wt_{lvl_x}$ **then** wait for next level
20: **if** $off_{lvl_{x-1}} < exp_{lvl_{x-1}}$ **then**
21: EXPLORE AND EVALUATE
22: $off_{lvl_{x-1}} \leftarrow off_{lvl_{x-1}} + paths_{lvl_{x-1}}$
23: **return** PROCESS($lvl_{x+1}, off_{lvl_x} \leftarrow 0$)
24: **else**
25: **if** $x > 1$ **then**
26: $off_{lvl_{x-1}} \leftarrow 0$
27: **return** PROCESS($lvl_{x-1}, off_{lvl_{x-2}}$)
28: **else**
29: **if** s_{ML} is empty **then**
30: Increase radius $d^2 \leftarrow 2 * d^2$
31: PROCESS(lvl_1,$off_{lvl_0} \leftarrow 0$)
32: **else**
33: Transfer s_{ML} to global memory
34: **return**
35: **end if**
36: **end if**
37: **end if**
38: **end procedure**

V. Experiments and Analysis

The PSD algorithm has a lot of parameters needed to be set. Until now we have not discussed how to set these parameters,

procedure EXPLORE AND EVALUATE
▷ The procedure was detailed in Subsection IV-B
Determine $vt_{lvl_x}^{i<j>}$ for thread t_{id}^k as stated in eq. 10
Determine $vb_{lvl_x}^{i<j>}$ for thread t_{id}^k as stated in eq. 11
Select path $s_{lvl_{x-1}}^{N<l>}$ from buffer $buf_{lvl_{x-1}}$ with the help of $vb_{lvl_x}^{(i+off_{lvl_{x-1}})<j>}$
Map $vt_{lvl_x}^{i<j>}$ to a partial symbol vector $s_{lvl_x}^{lvl_{x-1}-1<j>}$
Merge selected path $s_{lvl_{x-1}}^{N<l>}$ with partial symbol vector $s_{lvl_x}^{lvl_{x-1}-1<j>}$ resulting in $s_{lvl_x}^{N<j>}$
Store $s_{lvl_x}^{N<j>}$ in buf_{lvl_x} at index j
Compute path metric for the merged symbol vector $M(s_{lvl_x}^{N<j>})$
Store $M(s_{lvl_x}^{N<j>})$ in $bufPM_{lvl_x}$ at index j
syncthreads
if $lvl_x \neq 1$ **then**
Bitonic sort buf_{lvl_x} based on the corresponding path metric buffer $bufPM_{lvl_x}$
else
Find the best path metric M' in $bufPM_{lvl_x}$ and the associated symbol vector $s_1^{N\prime}$ with a parallel max scan
if $M' < M(s_{ML})$ **then**
$s_{ML} = s_1^{N\prime}$
Update radius $d^2 \leftarrow M(s_{ML})$
end if
end if
syncthreads
end procedure

but they value seriously influence the performance of the PSD algorithm. In this article we are not focusing on how to set the optimal values, however in order to present the top performance of the PSD algorithm an exhaustive search of the parameters was done. The simulated model was a $4x4$ MIMO system using $16-QAM$ modulation. The optimal parameters are: $tt = 64$, $lvl_{nr} = 3$, $lvl_0 = N = 8$, $lvl_1 = 6$, $lvl_2 = 4$, $lvl_3 = 1$, $paths_0 = 4$, $paths_1 = 2$, $paths_2 = 2$. Table I shows the achieved throughput comparing to other solutions:

Table I
THROUGHPUT COMPARISON OF THE SD ALGORITHM

	QPSK	16-QAM	64-QAM	64-QAM
ASIC[17]	19 Mbps	38 Mbps	NA	NA
ASIC[18]	NA	53 Mbps	NA	NA
FPGA[19]	NA	NA	8.6 Mbps	NA
GPU[9]	46 Mbps	74 Mbps	15 Mbps	1.3 Mbps
PSD	NA	110 Mbps	NA	NA

The PSD algorithm has the highest throughput and it will find the ML solution because the entire search space will be analysed. The presented algorithms in Table I are truncating the search space and they still can not reach the speed of the PSD algorithm.

VI. CONCLUSION

In this paper we showed the importance of MIMO communication systems, presented the basic models, with the most important detection algorithms, and proved that GPU's can significantly improve the computational time of digital signal processing algorithms. Reaching high data rates while keeping the bit error rate as low as possible is now possible with this powerful devices.

REFERENCES

[1] M. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *Information Theory, IEEE Transactions on*, vol. 49, no. 10, pp. 2389–2402, 2003.
[2] J. H. Conway, N. J. A. Sloane, and E. Bannai, *Sphere-packings, lattices, and groups*. New York, NY, USA: Springer-Verlag New York, Inc., 1987.
[3] A. Murugan, H. El Gamal, M. Damen, and G. Caire, "A unified framework for tree search decoding: rediscovering the sequential decoder," *Information Theory, IEEE Transactions on*, vol. 52, no. 3, pp. 933–953, 2006.
[4] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *Information Theory, IEEE Transactions on*, vol. 48, no. 8, pp. 2201–2214, 2002.
[5] D. Micciancio and S. Goldwasser, *Complexity of lattice problems: a cryptographic perspective*. The Kluwer international series in engineering and computer science, Kluwer Academic, 2002.
[6] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, no. 170, pp. pp. 463–471, 1985.
[7] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical Programming*, vol. 66, pp. 181–199, 1994. 10.1007/BF01581144.
[8] M. Khairy, C. Mehlfuhrer, and M. Rupp, "Boosting sphere decoding speed through graphic processing units," in *Wireless Conference (EW), 2010 European*, pp. 99–104, IEEE, 2010.
[9] M. Wu, Y. Sun, S. Gupta, and J. Cavallaro, "Implementation of a high throughput soft mimo detector on gpu," *Journal of Signal Processing Systems*, pp. 1–14, 2010.
[10] H. Vikalo and B. Hassibi, "On the sphere-decoding algorithm ii. generalizations, second-order statistics, and applications to communications," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2819–2834, 2005.
[11] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm i. expected complexity," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2806–2818, 2005.
[12] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *Signal Processing, IEEE Transactions on*, vol. 53, pp. 1474 – 1484, april 2005.
[13] J. Fink, S. Roger, A. Gonzalez, V. Almenar, and V. Garciay, "Complexity assessment of sphere decoding methods for mimo detection," in *Signal Processing and Information Technology (ISSPIT), 2009 IEEE International Symposium on*, pp. 9 –14, dec. 2009.
[14] K. Lai, J. Jia, and L. Lin, "Hybrid tree search algorithms for detection in spatial multiplexing systems," *Vehicular Technology, IEEE Transactions on*, no. 99, pp. 1–1, 2011.
[15] P. Kipfer and R. Westermann, "Gpu gems 2, chapter 46," 2005.
[16] K. E. Batcher, "Sorting networks and their applications," in *AFIPS Spring Joint Computing Conference*, pp. 307–314, 1968.
[17] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G. Knagge, "Silicon complexity for maximum likelihood mimo detection using spherical decoding," *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 9, pp. 1544–1552, 2004.
[18] Z. Guo, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *Areas in Communications, IEEE Journal on*, vol. 24, no. 3, pp. 491–503, 2006.
[19] S. Chen, T. Zhang, and Y. Xin, "Relaxed K -Best MIMO Signal Detector Design and VLSI Implementation," vol. 15, no. 3, pp. 328–337, 2007.

An Experimental Study on Metastable Periodic Rotating Waves

Miklós Koller
(Supervisors: Luca Pancioni, Mauro Forti, Barnabás Garay, György Cserey)
kolmi@digitus.itk.ppke.hu

***Abstract*—In this paper, a one-dimensional ring of bidirectionally coupled, spatially-invariant Cellular Neural Network (CNN) is considered. The cells have the standard piecewise linear output function. Numerical simulations show long transient oscillations in a wide range of parameters and initial conditions, before the system converges toward an equilibrium point. The main goal of this paper is the experimental verification of such long oscillations in a circuit made of discrete-components.**

***Keywords*-CNN (Cellular Neural Network); long transient oscillations**

I. Introduction

The paper considers the dynamical behaviour of a one-dimensional CNN-ring, with bidirectional coupling and piecewise linear (PL) activation function.

Our analysis dealt with the positive part of the coupling-parameter space. In this case a CNN ring generates a monotone semiflow, but due to the squashing effect of the PL activations this semiflow is not eventually strongly monotone (ESM) [1]. Regardless of this fact, it has a Limit Set Dichotomy and convergence properties analogous to those of ESM semiflows [2], [3], [4], [5], [6], which results in convergence toward an asymptotically stable equilibrium point.

Numerical simulations (both with MATLAB and C++ program) show this convergence. However, if the number of neurons greater than or equal to 6, with a wide range of coupling parameters and with a wide set of initial conditions we can observe long transient oscillatic behaviour, before the system goes to the stable point.

The main result of this paper is to experimentally validate this phenomenon by an electrical circuit realization of the one-dimensional, bidirectionally coupled CNN-ring. The experiments show, that this long transient oscillatic behaviour is not a numerical artifact of the simulation, moreover, it is physically robust (taking into consideration the imprecision of the applied discrete components).

II. The model

Let us consider an autonomous one-dimensional standard CNN array satisfying the following system of differential equations

$$\tau \dot{x}_i = -x_i + \alpha g(x_{i-1}) + \beta g(x_{i+1}) \qquad (1)$$

where $\tau > 0$ is the neuron time constant, x_i, $i = 1, 2 \ldots, n$, are the neuron state variables and g is the standard PL output function

$$g(\rho) = \frac{1}{2}(|\rho + 1| - |\rho - 1|).$$

Each neuron has two-sided interactions α, β with its nearest neighboring neurons. We suppose periodic boundary conditions.

Let us take into consideration the parameter space $\alpha, \beta > 0$. Also suppose for simplicity that $\alpha + \beta > 2$. Consider the curve

$$C = \left\{(\alpha, \beta) : (\alpha + \alpha\beta - \beta^2 - 1)(\beta + \alpha\beta - \alpha^2 - 1) = 0\right\},$$

the region within C given by

$$R_\phi = \left\{(\alpha, \beta) : \alpha > \frac{\beta^2 + 1}{\beta + 1}; \beta > \frac{\alpha^2 + 1}{\alpha + 1}\right\}$$

and the region outside C given by

$$R_\sigma = \{(\alpha, \beta) : \alpha + \beta > 2\} \setminus R_\phi$$

.

Although the system (1) has only three equilibrium points within R_σ, there are several additional equilibrium points within R_ϕ. Within the region R_ϕ, the general solution of (1) immediately converges to one of the equilibrium points, however, within the region R_σ one can observe (with a wide set of initial conditions) unexpectedly long transient oscillatic behaviour before converging to one of the stable points. Numerical simulations both with MATLAB and with C++ show the same behaviour.

The main results of this paper are to validate the existence of this long transient oscillatic behaviour by an electrical circuit as well as to inspect some of its salient features.
From theoretical considerations [7], this long transient oscillatic behaviour is due to the presence of metastable rotating waves, whose instability exponentially decreases with the increasing dimension of the CNN ring.

III. The experimental setup

The implemented circuit is built up of discrete components. The base of this implementation was originally proposed by Chua and Yang in [8]. We can divide the architecture of one cell to four stages, which take place in the followings.

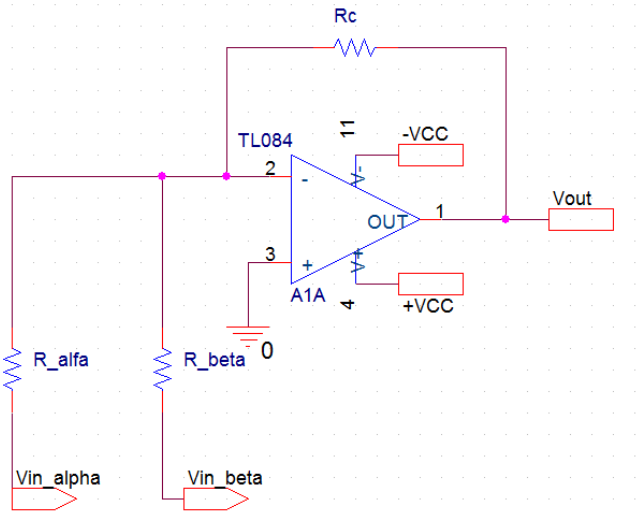


Fig. 1. First stage of the implemented circuit

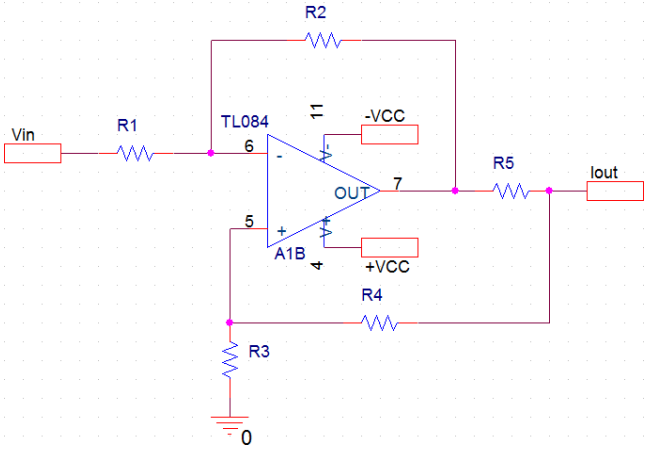


Fig. 2. Second stage of the implemented circuit

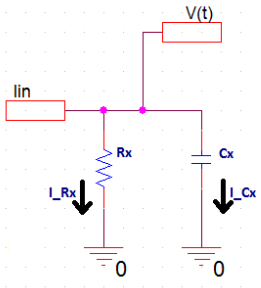


Fig. 3. Third stage of the implemented circuit

A. *First stage*

In the first step (Fig. 1), we have an inverting adder implementing the weighted sum of the inputs to the i-th neuron. These are the output voltages of the neighbouring cells $Vin\_alpha = g(x_{i-1})$ and $Vin\_beta = g(x_{i+1})$. Namely, we have

$$Vout = -\frac{Rc}{R\_alfa}Vin\_alpha - \frac{Rc}{R\_beta}Vin\_beta = \\ = -\alpha g(x_{i-1}) - \beta g(x_{i+1})$$

i.e., the dimensionless positive interaction parameters α, β are obtained as

$$\alpha = \frac{Rc}{R\_alfa}, \quad \beta = \frac{Rc}{R\_beta}.$$

In the actual circuit we have chosen $Rc = 560\ \Omega$, so that the design equations for R_alfa, R_beta are given as

$$R\_alfa = \frac{Rc}{\alpha} = \frac{560}{\alpha}\ \Omega, \quad R\_beta = \frac{Rc}{\beta} = \frac{560}{\beta}\ \Omega.$$

B. *Second stage*

The second step (Fig. 2) implements a voltage controlled current source (see Appendix of [8]), under the following constraint

$$\frac{R2}{R1} = \frac{R4 + R5}{R3} \tag{2}$$

it can be shown that we have

$$Iout = -\frac{R2}{R1R5}Vin.$$

In the actual circuit we have chosen, in accordance with (2), $R1 = 1.8$ kΩ, $R2 = 2.7$ kΩ, $R3 = 1.8$ kΩ, $R4 = 1.2$ kΩ and $R5$=1.5 kΩ, resulting

$$Iout = -\frac{1}{1000}Vin = \frac{\alpha}{1000}g(x_{i-1}) + \frac{\beta}{1000}g(x_{i+1}).$$

C. *Third stage*

The third step (Fig. 3) realizes the state x_i of the i-th neuron, which can be measured at the capacitor Cx. The time constant of the neuron is $\tau = RxCx$. By the Kirchhoff current law we obtain

$$Iin = I\_Cx + I\_Rx = Cx\dot{V}(t) + \frac{V(t)}{Rx} = Cx\dot{x}_i + \frac{x_i}{Rx}.$$

We have chosen $Rx = 1$ kΩ, $Cx = 680$ nF, hence we have

$$680 \times 10^{-9}\,\dot{x}_i = -\frac{x_i}{1000} + \frac{\alpha}{1000}g(x_{i-1}) + \frac{\beta}{1000}g(x_{i+1})$$

which coincides with the equation (1) describing the dynamics of the i-th neuron with a time constant $\tau = 6.8 \cdot 10^{-4}$ sec.

D. *Fourth stage*

The fourth step realizes the PL output-function $g(x_i)$ of the i-th neuron. This stage contains an amplification-division pair. Due to this overriding of the operational amplifier we are able to get the desired saturation-regions of the PL output-function. More precisely

$$\frac{R6 + R7}{R6}.$$

and

$$\frac{1}{\frac{R_9}{R_8 + R_9}}$$

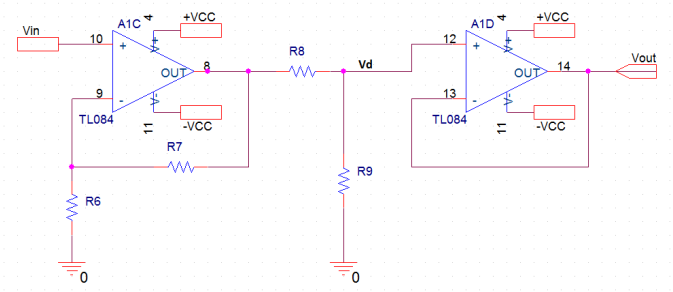


Fig. 4. Fourth stage of the implemented circuit

should be equal. In this way $Vd = g(x_i)$.
In the circuit, the parameter values are chosen as $R6 = 1$ kΩ, $R7 = 18$ kΩ, $R8 = 18$ kΩ and $R9 = 1$ kΩ. Finally, the fourth op amp implementing a voltage-follower with $Vout = Vd = g(x_i)$ is used for decoupling the voltage divider from the input stage of each connected neuron.

The initial condition is set by a switch at every cell, which is connected to the third stage's $V(t)$ point. When the switch is set in 'load' position, it disconnects the third stage from the second one, moreover, it connects the third stage to the voltage generator providing initial condition. In 'run' position it sets the normal setup, what is depicted on Fig. 3. The implemented

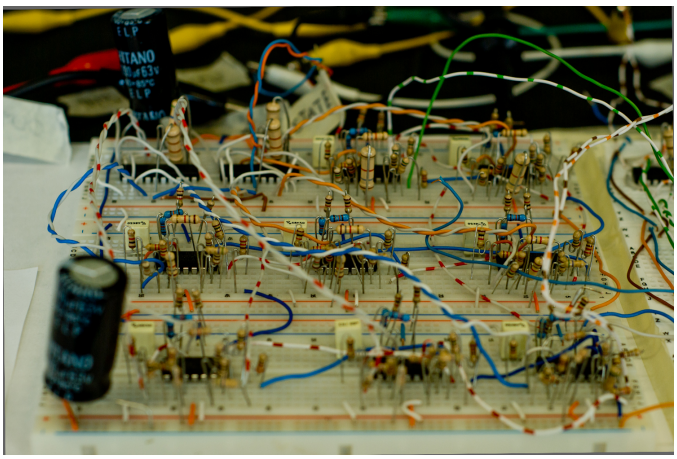


Fig. 5. Half of the built circuit's 16 cells.

circuit took place on a breadboard, the half of it can be seen on Fig. 5.

IV. Experimental results

We have built an electrical circuit of the CNN ring (1), with size $N = 16$. The interconnecting lines are configurable, in this way we can modify the weights of the connections as well as the size of the ring itself. The vast majority of our experiments were made with the following sizes $N = 4, 8, 16$. The used resistors have 5% tolerances, the capacitors have 10% tolerances; the type of the operational amplifiers is TL084. The supply voltage for the operational amplifier was set to ± 20 V. We used the MAX333 IC as switch, which has 130Ω series internal resisistance.

In the case of $N = 4$, we were not able to observe oscillation, however, with size $N = 6$ and above it, the appearing oscillation got more and more longer in time (as we saw earlier by numerical simulations, too). We tried more points of the (α, β)-plane, especially with

$$(\alpha, \beta) \in R_\sigma.$$

We tested also more initial conditions, in the following some of these "parameter – initial condition" pairs will be presented. In all of the cases, 16 cells were applied. By every figure, we made a comparison among three waveforms:

- the waveform of the differential-equation system's (1) pure solution;
- the waveform acquired from the PSpice circuit-simulator;
- the waveform measured by the oscilloscope on the experimental circuit.

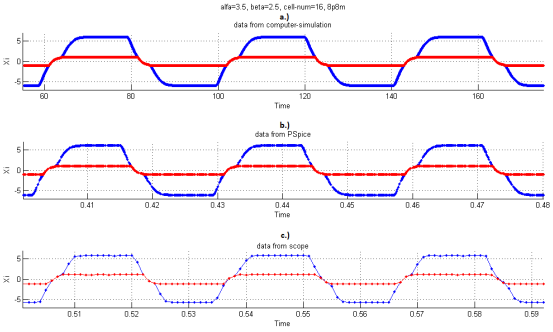


Fig. 6. Simulation result with the following parameters: $\alpha = 3.5$, $\beta = 2.5$, initial-condition:x'_0. a.) depicts the state- and output-waveform of a MATLAB simulation; b.) shows the signal levels of the state- and output-terminal, in a circuit-simulator; c.) illustrates the signal levels on the state- and output-terminal of the experimental circuit.

The first parameter pair is $\alpha = 3.5$, $\beta = 2.5$, and the initial condition is

$$x'_0 = 2.9 \cdot (1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1)'.$$

We can see the obtained waveforms on Fig. 6. In this case, the waveforms are very similar, however, we have to mention, the real oscillation held much shorter time, than the simulation of the ideal-model. The main reason is the imprecise value of the discrete components in the experimental circuit.

The second parameter pair is $\alpha = 3.5$, $\beta = 2.5$, and the initial condition is

$$x''_0 = 2.9 \cdot (1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1)'.$$

We can see the obtained waveforms on Fig. 7. In this case, due to the combinatorics of the initial condition, the waves have shorter period. Both the waveform of the circuit simulator and the real, oscilloscope-measurement have stronger instability. Moreover, we can recognize the two different forms of the death/dissolve of a metastable periodic waves over time: the result of the PSpice simulation shows direct convergence to the stable point; the waveform of the oscilloscope illustrates

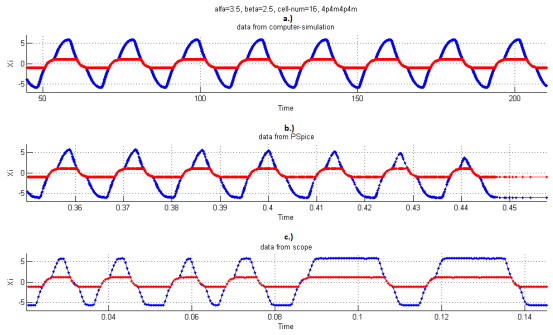


Fig. 7. Simulation result with the following parameters: $\alpha = 3.5$, $\beta = 2.5$, initial-condition:x_0''. a.) depicts the state- and output-waveform of a MATLAB simulation; b.) shows the signal levels of the state- and output-terminal, in a circuit-simulator; c.) illustrates the signal levels on the state- and output-terminal of the experimental circuit.

the transformation/metamorphosis from one periodic orbit to the other one.

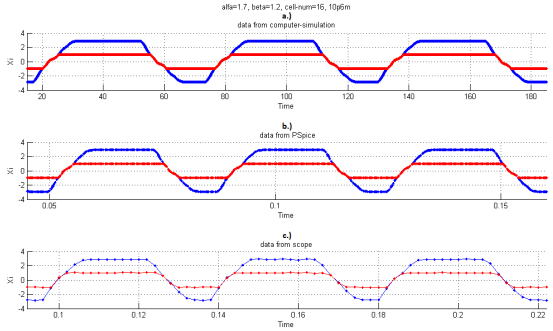


Fig. 8. Simulation result with the following parameters: $\alpha = 1.7$, $\beta = 1.2$, initial-condition:x_0'''. a.) depicts the state- and output-waveform of a MATLAB simulation; b.) shows the signal levels of the state- and output-terminal, in a circuit-simulator; c.) illustrates the signal levels on the state- and output-terminal of the experimental circuit.

The third parameter pair is $\alpha = 1.7$, $\beta = 1.2$, and the initial condition is

$$x_0''' = 2.9 \cdot (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1)'.$$

This is a highly asymmetric initial state, as it can be seen on Fig. 8. This measurement had a shorter oscillatic-regime, than the system with x_0'. The asymmetric initial state leads easier to the death of the metastable periodic wave.

V. Conclusion

This paper considers a one-dimensional, autonomous Cellular Neural Network with positive interconnections. With a wide range of interconnection parameters and with a wide set of initial conditions we can observe long-lasting oscillations. Due to theoretical considerations and analysis in [7] we suspect, that the main cause of these oscillations is the presence of metastable periodic solutions.

The numerically observed oscillations occured through the presented experimental results, moreover, we experienced them quite robust taking into consideration the deviation of the applied discrete components.

Acknowledgment

The author is grateful to Professor Tamás Roska for the discussions and his suggestions.
The support of the grants TÁMOP-4.2.1.B-11/2/KMR-2011-0002 and TÁMOP-4.2.2/B-10/1-2010-0014 is gratefully acknowledged.

References

[1] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, "The dichotomy of omega-limit sets fails for cooperative standard cnns," in *Cellular Nanoscale Networks and Their Applications (CNNA), 2010 12th International Workshop on*, pp. 1–5, IEEE, 2010.

[2] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, "Limit set dichotomy and convergence of semiflows defined by cooperative standard cnns," *International Journal of Bifurcation and Chaos*, vol. 20, no. 11, p. 3549, 2010.

[3] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, "Limit set dichotomy and convergence of cooperative piecewise linear neural networks," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, no. 99, pp. 1–1, 2011.

[4] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, "Convergence of a class of cooperative standard cellular neural network arrays," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, no. 99, pp. 1–1, 2012.

[5] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, "Further results on convergence of cooperative standard cellular neural networks," in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pp. 2161–2164, IEEE, 2011.

[6] M. Hirsch and H. Smith, "Monotone dynamical systems," *Handbook of differential equations: ordinary differential equations*, vol. 2, pp. 239–357, 2006.

[7] M. Forti, B. Garay, M. Koller, and L. Pancioni, "Floquet multipliers of a metastable rotating wave," *Linear Algebra Applications*. in preparation.

[8] L. Chua and L. Yang, "Cellular neural networks: Theory," *Circuits and Systems, IEEE Transactions on*, vol. 35, no. 10, pp. 1257–1272, 1988.

Accelerating Computational Quantum Chemistry with Automated Compilation of Exchange Integrals on GPU

Ádám Rák
(Supervisor: Dr. György Cserey)
rakad@digitus.itk.ppke.hu

***Abstract*—We demonstrate the use of graphical processing units (GPUs) to carry out quantum chemical exchange integral calculations for molecules, with utilizing a novel compiler architecture, and specialized mathematical formalism. This compiler includes our specifically modified generalized braket, integral solution rules, and also a special register allocation algorithm which enables better GPU utilization. Speedups ranging from 4x to 150x are achieved as compared to a third-party quantum chemistry integrator library (Libint) running on a traditional CPU.**

***Index Terms*—GPU, quantum chemistry, exchange integral, register allocation, compiler**

I. Introduction

Simulation techniques of quantum chemistry have been developed since the rise of computer technology [1][2]. In the course of the simulation of atoms and molecules, the molecular orbitals are described by series of atomic functions. All atoms has basis functions, their quantity and types are relevant because of accuracy. Simple basis functions can be characterized by the angular momentum $(s, p, d, f, ...)$ and the exponent, while complex (called contracted) basis functions are featured by linear coefficients (called contraction coefficients) as well. One of the speed limitation of quantum chemistry calculations is the calculation of integrals, especially the calculation of exchange integrals, because in case of n basis function n^4 exchange integrals have to be computed. This number can be over millions even in case of a simple molecule. Integrals can be calculated independently, therefore these calculations can be parallelizable by their nature. Because of the huge computational demands, a more efficient computational architecture is required with fully optimized implementation. Currently, regarding computational performance, GPU has one of the fastest evolving parallel architecture, therefore taking into account technological advances, a GPU architecture optimized integral calculation method should be developed.

The accuracy of quantum chemical calculations is dependent on the number and the quality (angular momentum) of the basis functions. For the general calculation of even simple organic molecules at least f-type functions are needed, for more accurate calculations g and h-type functions are required. The problem is that increasing the angular momentum of basis functions, integrals are more and more complicated and their calculation is computationally expensive. The previous GPU solutions hardly [3] or could not calculate the basis functions over d. The reason of this that the calculation of integrals usually needs recursion which has excessive memory requirements, the lot of GPU memory access is not optimal. Other possibility, which is not used normally, is the unrolling of numerical integration formula. In this case calculation of only one integral indicates calculations of millions of multiplications and additions meanwhile the register allocation is an NP difficult task. In this case another problem is that the source code is very large (often over millions of lines) and because of practical reasons such a huge source code can not be compiled to GPU by traditional compilers. Despite of these difficulties, we chose the way of unrolling of the numerical formula and our solution takes advantage of the memory hierarchy of GPUs for the high efficiency.

II. Exchange integral formula

The two electron four center exchange integral can be seen on equation 1. Where $\mathbf{r_{12}}$ is a distance function, usually $|\mathbf{r_1} - \mathbf{r_2}|$. The symbolic depiction of this is $(ab|cd)$ which is a quantum chemical braket, where the $abcd$ letters identify different basis functions, each with it's own center, and angular moment. For centers we will use $\mathbf{ABCD}$.

$$(ab|cd) = \iiint_{\mathbf{r_1}} \iiint_{\mathbf{r_2}} \Phi_a(\mathbf{r_1})\,\Phi_b(\mathbf{r_1})\,\frac{1}{\mathbf{r_{12}}}\,\Phi_c(\mathbf{r_2})\,\Phi_d(\mathbf{r_2})\,\mathrm{d}\mathbf{r_1}\mathrm{d}\mathbf{r_2} \quad (1)$$

Each basis function has two distinct parts, one is the angular polynom which is equation 3, and the gaussian functions on equation 2. The basis function is computed from a linear combination of gaussians multiplied by the angular polynom, as showed on equation 4. The coefficients and the exponents for gaussians are specific to basis functions. In the equations, the angular moment 3D vector is depicted by bold $\mathbf{a}$, later we will use $\mathbf{abcd}$ to differentiate between the four functions.

$$g(\mathbf{r}, \mathbf{R}, \alpha) = \mathbf{e}^{-\alpha|\mathbf{r}-\mathbf{R}|^2} \quad (2)$$

$$\mathcal{A}_{poly}(\mathbf{r}, \mathbf{R}, \mathbf{a}) = (r_x - R_x)^{a_x}(r_y - R_y)^{a_y}(r_z - R_z)^{a_z} \quad (3)$$

$$\Phi(\mathbf{r}) = \mathcal{A}_{poly}(\mathbf{r}, \mathbf{R}, \mathbf{a}) \sum_{\mathbf{i}} \mathbf{c_i g}(\mathbf{r}, \mathbf{R}, \alpha_\mathbf{i}) \quad (4)$$

III. Contraction, General contraction

Because each basis function is made up from a linear combination of gaussians which is called contraction, we can move the linear combination out from the integral. This is computationally still very expensive, that's why basis functions usually have "general contraction", where more basis functions share all their parameters, except c_i coefficients, so we only need to compute the integrals once for many basis functions.

IV. Mapping integrals to GPUs

GPUs are practical many core architectures, they can have 2048 cores or even more. Each core can do basic floating point and integer arithmetic operations. It is established that computing exchange integrals is computationally very expensive, and it is mostly accomplished by floating point multiplications and additions, so using GPUs for the task seems practical. On the other hand it is very memory intensive too, because when computing we need to traverse a deep arithmetic tree. At each level we have to store temporaries, which translates to a huge memory bandwidth need in the case when we have thousands of cores running integral calculations. While the memory bus bandwidth of the GPUs is very huge (200Gbyte/s), if we divide it by the number of cores, we rarely got much bigger than 100Mbyte/s per core. On the other hand, the arithmetic throughput of a single core is around 1-3Gflop/s. Which means that for every transferred byte we should compute 10-30 flop, for single precision arithmetic (32bit float), it means 40-120 flop / value transferred. This is much more than what what we can compute according to the conventional algorithms. Depending on the integral solving rule set, we only do a few flops in a single step, before we need to do memory transfers again.

Our approach uses the well researched exchange integral solving algorithms, in a more fitting way for the GPU. We execute the integral solution formula symbolically to obtain the entire arithmetic tree. We feed this arithmetic to our specially designed compiler, to transform and optimize it. This way we can do significant offline optimizations on the code, one of such, is using GPU registers for storing temporary values, and dynamically (per integral) balancing the computational tree between the memory transfer and arithmetic trade-off. This is quite unorthodox, because it generates very huge amounts of code, and the compilation can last for days for higher integrals. Unlike older CPUs, huge (500Mbyte) code sizes poses no challenges, because optimally every core on the GPU executes the same code instructions, but on different data, so we need to fetch the code only once. The especially big compile times pose no problems either because we only need to compile one type of integral only once, we can refit it to use other basis functions later.

A. Optimized contractions

Due to the unrolling of the whole algorithm, we can do a significant optimization on contracted and often used integrals. While the atomic centers in the basis functions is different in every molecule, the gaussian exponents only depend on the type of atom, and the basis set library. It means that we can compile specific integral solvers for specific atom types, which enables us to compute the gaussian exponent part of the solution offline. Contraction is usually used on lower orbitals (s, p), where the integral computation is much simpler, which means that doing this offline optimization is computationally cheap. The only drawback is that we generate much more integral solvers. We can choose to optimize only the very often used configurations, so we can keep the amount of compilation work under control.

V. Compiler architecture

A compiler generally has three main parts:

- The front-end which is input language specific and handles translating the input language into the internal representation of the compiler
- The middle-end where most of the code optimizations passes are. The middle-end is mostly hardware and input language independent
- The back-end where very machine specific optimizations are. And it translates the internal representation into machine code

We designed a compiler for converting the symbolic form of exchange integrals to GPU optimized machine code. The conventional solutions on CPU generally include similar but less through approaches. For example Libint generates C code, a different source file for every general step of the integral computations, without recursions. This is very efficient on CPUs because for most integrals the temporaries fit into the L1 cache, and many compiler optimizations are possible, due to the unrolled recursions.

In our case we go further, we execute it symbolically and completely unroll the whole algorithm into a single arithmetic tree, that serves as the front-end of our compiler. We can also generate C code at this stage, for testing our system. But it is impractical because no general purpose compilers exits to out knowledge, which can compile the code of even mid sized integrals in reasonable time (1 week), and also produce reasonably optimal code.

The middle-end of our compiler contains various optimization passes, most of the them are widely used in state of the art compilers. These optimizations:

- Conversion to Load-Store format
- Conversion to SSA format
- Eliminating dead arithmetic
- Constant propagation
- Transforming simple cases of $a \cdot c + b \cdot c \Rightarrow (a \cdot b) \cdot c$
- Arithmetic tree reordering by heuristics
- Hash based elimination of re-computation of values

The role of the back-end is to do GPU specific optimizations, and map the virtual registers to actual hardware registers of the hardware.

- Recognize FMA (Fused Multiply Add) instructions from basic arithmetic
- Constant value handling optimizations

- Special handling of negative signs
- Generating memory indexing
- Ordering memory stores
- **Mapping virtual registers**
- Generating machine code

VI. Compiler internal representation

Our compiler uses an optimized approach for storing and transforming program code internally, because we usually have to deal with huge amount of code. The internal format is double sided, on one hand the instructions are stored sequentially, on the other hand when we ensure it, the code can be efficiently traversed like a data-flow tree along the temporary values.

We use Static Single Assignment (SSA) representation, like modern compilers (GCC 4.0 [4], HotSpot JVM [5] and LLVM [6]). In this representation, one value can be given a value only once, we call that value assignment, the definition of the value. Later it can used any number of times. To achieve this, we have to introduce new temporary values into the code, the maximum amount is the number of the instructions, where every instruction get assigned to a temporary value.

VII. Mapping virtual registers

Register allocation happens in the back-end of our compiler, it is the mapping of virtual registers to hardware registers. This is a very important step, because it enables straightforward generation of machine code. When the processed program code does not use more registers than the number of registers in the hardware, this algorithm is trivial. Usually we have at 10x to 100x times more virtual registers, which make its necessary to use other kinds of memories to store temporary values.

It has been proved that SSA based register allocation can be done in polynomial time [7][8][9], and even non-SSA allocators in modern compilers have polynomial or near linear [10][11] complexity. We have tested our system with C language output, where the register allocation was done with state of the art compilers. In our case, due to the extreme amount of code, polynomial complexity proved to be prohibitively expensive.

We implemented a our own greedy register allocator which works with a linear scan, and takes into account the local memory present in GPUs. The register allocator computes local heuristics about where it should store the values.

VIII. Generalized braket formula

The generalized braket formula is mathematical powerful tool for describing exchange integral solving algorithms, it was used to define the PRISM algorithm [12]. The basic idea is to collect the exponents of various weights generated by the integral solving recursions, into a simple form.

The exchange integral problem contains four gaussians, each has a different center A, B, C, D, and an angular polynomial. The angular polynomials can be described by 3D angular moment vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$. Due to gaussian identities we can transform our integral according to equations 5 6.

$$g(\mathbf{r}_1, \mathbf{A}, \alpha_i) g(\mathbf{r}_1, \mathbf{B}, \beta_j) \Rightarrow g(\mathbf{r}_1, \mathbf{P}, \zeta_{ij}) \tag{5}$$

$$g(\mathbf{r}_2, \mathbf{C}, \gamma_k) g(\mathbf{r}_2, \mathbf{D}, \delta_l) \Rightarrow g(\mathbf{r}_2, \mathbf{Q}, \zeta_{kl}) \tag{6}$$

For these P, Q virtual centers we can assign $\mathbf{p}, \mathbf{q}$ angular moments and angular polynomials. The conversion in this case is not trivial, but this is the subject of exchange integral transforming formulas. We can continue on this way, and define an R virtual center and angular moment $\mathbf{r}$, to substitute the previous two gaussians.

While in the literature it was not favored, using the R virtual center, and not just $\mathbf{r}$ in the general brakets, greatly simplifies our method. With these mathematical aids the literature defines transformation rules to move angular moment from $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ to $\mathbf{p}, \mathbf{q}$ to $\mathbf{r}$, and from $\mathbf{r}$ we can actually eliminate angular moments, thus reducing the integral problem of integrating a single gaussian. The price of these transformations is that a braket is reduced to a weighted sum of brakets. The weights can be composed of the exponents of the gaussians, centers, and the current integer value of the angular moment which we are reducing. We can represent the powers of the exponents and centers in a form, along with all virtual or actual angular moments, which is called the generalized braket.

The simplest generalized braket can be seen on equation 7, where all powers and angular moments are zero, and (m) means derivation. In this form the braket is simple equal to a four center simple gaussian exchange integral, which can be very well approximated by special algorithms based on power series. The meaning behind the non-angular moment parts of the generalized braket is explained by equations 8 9.

We have also designed a novel way of computing $\mathcal{F}$ in equation 7. Where we specially took in account the preferences of GPUs. We have eliminated the control overhead, and used only a forward series expansion. We used a Chebyshev approximation to numerically stabilize the otherwise unstable forward algorithm.

The algorithm of solving an exchange integral is composed of eliminating angular moments, and solving the brakets we get after the elimination. Our compiler front-end uses this generalized braket representation to solve the exchange integral symbolically and to convert these brakets into program code, which can optimized later.

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{(m)} = \iiint_{\mathbf{r_1}} \iiint_{\mathbf{r_2}} \left(g(\mathbf{r}, \mathbf{A}, \alpha) g(\mathbf{r}, \mathbf{B}, \beta) \frac{1}{\mathbf{r_{12}}} g(\mathbf{r}, \mathbf{C}, \gamma) g(\mathbf{r}, \mathbf{D}, \delta) \right)^{(m)} \mathrm{d}\mathbf{r_1} \mathrm{d}\mathbf{r_2} \approx \mathcal{F}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \alpha, \beta, \gamma, \delta, m) \tag{7}$$

Shell quartet	Speedup on the GPU
(ssss)	6
(sspp)	67
contracted (sspp)	107
contracted (ssps)	150
contracted (pspp)	125
(ssdd)	52
(dsds)	69
(pppp)	43
(ppdd)	30
(dddd)	49
(ppff)	13
(fsfd)	23
(fdfd)	17

Fig. 1. Few examples of the measured speedup for quartets. The average speedup in this benchmark was 60x compared to the CPU implementation

$$\left[\begin{array}{cccc|cccc} \mathbf{a} & \mathbf{b} & \mathbf{p} & & \mathbf{c} & \mathbf{d} & \mathbf{q} & \mathbf{r} \\ a' & b' & p' & & c' & d' & q' & \\ \mathfrak{a}^* & \mathfrak{b}^* & \mathfrak{c}^* & & \mathfrak{e}^* & \mathfrak{f}^* & \mathfrak{g}^* & \mathfrak{r}^* \end{array}\right]^{(m)} = \mathcal{P}_{prefactor}(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D},\alpha,\beta,\gamma,\delta)\cdot \left[\begin{array}{cccc|cccc} \mathbf{a} & \mathbf{b} & \mathbf{p} & & \mathbf{c} & \mathbf{d} & \mathbf{q} & \mathbf{r} \\ 0 & 0 & 0 & & 0 & 0 & 0 & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right]^{(m)} \tag{8}$$

$$\begin{aligned} \mathcal{P}_{prefactor}(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D},\alpha,\beta,\gamma,\delta) = \\ (\mathbf{B}-\mathbf{A})^{\mathfrak{a}^*}\cdot(\mathbf{C}-\mathbf{D})^{\mathfrak{b}^*}\cdot(\mathbf{D}-\mathbf{B})^{\mathfrak{c}^*}\cdot \\ (\mathbf{C}-\mathbf{A})^{\mathfrak{e}^*}\cdot(\mathbf{D}-\mathbf{A})^{\mathfrak{f}^*}\cdot(\mathbf{C}-\mathbf{B})^{\mathfrak{g}^*}\cdot \\ (\mathbf{R})^{\mathfrak{r}^*}\cdot \\ \frac{(2\alpha)^{a'}(2\beta)^{b'}}{(2\zeta)^{p'}}\cdot\frac{(2\gamma)^{c'}(2\delta)^{d'}}{(2\eta)^{q'}}\cdot \end{aligned} \tag{9}$$

IX. Results

We measured our GPU implementation speedup against a state of the art quantum chemical integral computer library Libint [13], which runs on CPU. In the benchmark we used an Nvidia GTX580 graphics card for the GPU, and an AMD Athlon(tm) II X4 635 processor. We added sufficient amount of system memory, and run all tests several times to obtain the average performance. All timing results had the statistical deviation below 10% or their average.

We have measured shell quartets as can be seen on figure 1, and not single integrals, because it is more efficient to compute all integrals together which belong to the same shell quartet, with the same molecular centers. The Libint software library does the same optimization.

X. Conclusion

We have designed and implemented a novel compiler architecture for compiling quantum exchange integrals to the GPU. This compiler includes our specifically modified generalized braket, integral solution rules, and also a special register allocation algorithm which enables better GPU utilization. We have measured the performance of our system up to f shells, and it have performed better than state of the art GPU implementations.

With the GPUs being cheap commodity, using our compiler as part of a quantum chemistry software is feasible, and should result in significant performance gains compared to the CPU based solutions. Especially because, the cost-performance ratio in this case is far better than for traditional CPU systems.

Acknowledgment

The support of NVIDIA Professor Partnership Program and the Bolyai János Research Scholarship is gratefully acknowledged. The authors are also grateful to Tibor Höltzl, Gergely Feldhoffer, Gergely Balázs Soós and Balázs Oroszi.

References

[1] W. Hehre, "Ab initio molecular orbital theory," *Accounts of Chemical Research*, vol. 9, no. 11, pp. 399–406, 1976.

[2] J. Pople and D. Beveridge, *Approximate molecular orbital theory*. McGraw-Hill, 1970.

[3] I. Ufimtsev and T. Martinez, "Quantum chemistry on graphical processing units. 2. direct self-consistent-field implementation," *Journal of Chemical Theory and Computation*, vol. 5, no. 4, pp. 1004–1015, 2009.

[4] B. Gough and R. Stallman, "An introduction to gcc," *Network Theory, Ltd*, 2004.

[5] J. Team, "The java hotspot virtual machine," Technical Report Technical White Paper, Sun Microsystems, Tech. Rep., 2006.

[6] C. Lattner and V. Adve, "Llvm: A compilation framework for lifelong program analysis & transformation," in *Code Generation and Optimization, 2004. CGO 2004. International Symposium on*. IEEE, 2004, pp. 75–86.

[7] F. Bouchez, "Allocation de registres et vidage en mémoire," *Master's thesis, ENS Lyon*, 2005.

[8] P. Brisk, F. Dabiri, J. Macbeth, and M. Sarrafzadeh, "Polynomial time graph coloring register allocation," in *14th International Workshop on Logic and Synthesis*, vol. 1, no. 1, 2005.

[9] S. Hack, D. Grund, and G. Goos, "Register allocation for programs in ssa-form," in *Compiler Construction*. Springer, 2006, pp. 247–262.

[10] C. Wimmer and H. Mössenböck, "Optimized interval splitting in a linear scan register allocator," in *Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*. ACM, 2005, pp. 132–141.

[11] A. Evlogimenos, "Improvements to linear scan register allocation," *University of Illinois, Urbana-Champaign*, 2004.

[12] P. Gill and J. Pople, "The prism algorithm for two-electron integrals," *International journal of quantum chemistry*, vol. 40, no. 6, pp. 753–772, 1991.

[13] J. Fermann and E. Valeev, "Libint: Machine-generated library for efficient evaluation of molecular integrals over gaussians," *Freely available at http://libint.valeyev.net/ or one of the authors*, 2003.

The design of a biomimetic joint

Norbert Sárkány
(Supervisors: Péter Szolgay, György Cserey)
sarno@digitus.itk.ppke.hu

***Abstract*—This paper presents a design of an anthropomorphic bio-mimetic joint, focusing on the design of a fingers and its bio-inspired flexor-extensor like control. The kinematic description, the detailed explanation and presentation of the 3D CAD design are included. The description of the intend to use 3D tactile sensors is also detailed in the article. Matlab simulation results and also the first functional test of the hardware prototype gave promising results and show that the approach can be an effective solution for the need of a hand-like actuator in robotics or in prosthesis.**

***Index Terms*—artifacial hand, bio-mimetic, robotics, bionics**

I. Introduction

In the last twenty years there was an extensive research about robotic hands, which goal was to design and develop an anthropomorphic dexterous hand [1], [2], [3], [4], [5], [6]. There are two main concept of designs, one with a local control where the actuators are in the hand [1], [3], [4], its reduces the amount of space which it requires, and the weight. The small weight is always an important aspect but in many cases this reduces the DOF. The second design is where the actuated structure and the actuator mechanism are separated and connected with artificial tendons, such a hand is capable to do manipulation tasks like a human hand can do, here every joint has an independent control, and there are no passively controlled joints.

The commercially available prosthetics are similar to the first type but are limited in their movement capability and they have a lack of sensory information and a not so sophisticated control.

In this paper a design of a bio-mimetic joint will be presented. Primarily the aim of the research is to make a full functional finger with a extensor-flexor control system.

The final goal is to have an artificial hand which can be used in robotic applications and which could give a basis of new prosthetics design too.

In Section II. a general descrption will be presented of the human joints, Section III. the basic design concepts of a bio-mimetic joint is presented, Section IV. shows a detailed mechanical structure of a finger. Section V. discuss the type of sensors in the human hand and our selections which correspond to them. Finally, conclusions and future work are discussed in Section VI. .

II. General description of human joints

The bones in our sceleton are connected fixd and either partially motile. Ther are so called interrupted bone connections in our fingers. The connection between the opposite bone parts are made via membranous arthritis case and fibers.

The components of the joint:

- **Joint head**(1. Figure (10)):
  the connecton of the oposit bone faces, one of tham is called convex joint head and the other is concave joint head
- **Joint case**(1. Figure (7)):
  it separatesthe articular from the enviremant, it is a bag like connective tissue
- **Articular fiber**(1. Figure (5)):
  dense collagen fibrous connective tissue bundle,it serve to strenth the Joint case

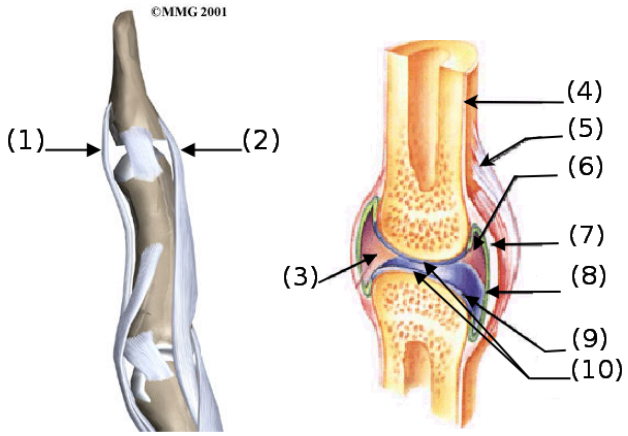


Fig. 1: The tensor, felexor tendons of the fingers and its tendon case.

The layers wich move on each other – wich can be two muscles, tendon muscle, tendon bone, skin and muscle bandage – the lubrication is achived by lymph, what is produced by articular sheath. The forward move of a finger is called flexio(Figure. 1. (1)), the opposite of it is backword move waht is calld extensio(Figure. 1 (2)).

III. Bio-mimetic joint design [12]

In the last 30 years, robotics and prosthesis from the moment of their presence use a basic concept which is nothing

else that the joint which connect the links in the finger or else where in the whole structure have fix and rigid rotatin axis. This kind of mechanical structure is very vulnerable for unplanned forces. When a Mechanical Engineer has a task to build an articifial hand or chain of links he use the basic concept. But is ther no other solution? Nature designed a robust and flexible structure , we have the result of this design and the only thing we need to do is to try to imitate it.

Figure 2. depict the kinematik structure of a joint. It can be given two description of the joint, one before the transiton to the q_i' from q_i which is described in the Table I. and one after the transiton what the Tabel II shows.

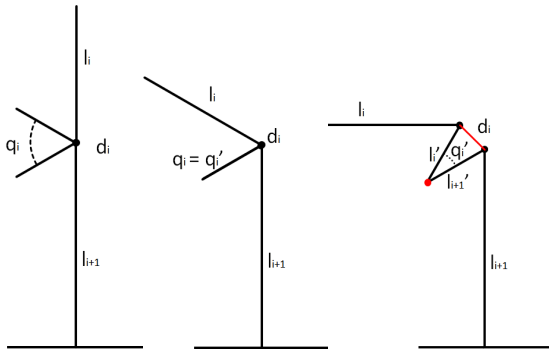


Fig. 2: The kinematic plane design of the bio-mimetic joint, l_i is the i. th link with fix size, d_i is the translation lenght between the end of l_i and the begining of l_{i+1},l_i', l_{i+1}' are the distance of the two rotation joints, q_i is the angle between the l_i' and l_{i+1}' link, is the angle between the l_i and l_{i+1} link after the axis of rotation are moved

In Table I and II the parameter q_{fix} is the angle between l_i – l_i' and l_{i+1} – l_{i+1}', and the parameter q_d is the angle between d_i and l_i'.

Link	a_i	α_i	d_i	Θ_i
1	l_{i+1}	0	0	90
2	l_{i+1}'	0	0	q_{fix}
3	l_i'	0	0	q_i
4	l_i	0	0	q_{fix}

TABLE I: The Denavit - Hartenberg parameters of the bio - mimetic joint

Link	a_i	α_i	d_i	Θ_i
1	l_{i+1}	0	0	90
2	l_{i+1}'	0	0	q_{fix}
3	l_i'	0	0	q_i'
4	l_i	0	0	q_{fix}
5	0	0	d_i	q_d

TABLE II: The Denavit - Hartenberg parameters of the bio - mimetic joint

In Figure. 3 we can see the first realized prototipe of the bio-mimetic joint and its CAD design. We can see the structural elements of a those main part which were listed in Section II. They are indicated on the CAD design, the part with label (a) is the joint case, (b) is the the joint hesd and (c) is the articular fiber.

The joint case was made from liquid latex after it becomes dry it forms a coherent elastic surface. In the bio - mimetic joint the components of the joint head are 3D printed concave sockets and a glass ball, to reduce the friction between the sufaces I greased them with sintetic fet. For the articular fiber I used kevlar fibers, to mount the kevlar fibers I designed boros on the end of the links.

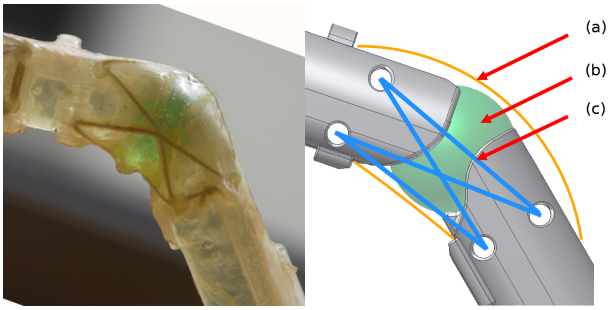


Fig. 3: The first fabricated joint and its CAD design

IV. Finger prototype design

The finger prototype with the bio-mimetic joint is designed by reproducing, as close as possibleto the size and kinematics of the human finger.

Each finger has two interphalangeal joints (IP's), distal (DIP) and proximal (PIP). Between the proximal phalanges and the metacarpals are the knuckles or metacarpophalangeal (MCP) joints. The IP and MCP joints are capable of flexion (bending) and extension (straightening). In addition, the MCP joints are capable of abduction (spreading of the fingers) and adduction (bringing the fingers together) [9] [10].

For the kinematic description of the finger I used a three segment anthropomorphic model, showed in Fig. 4 (C).

Where q_i is the angel between l_{i-1} and l_i, l_i is the length of the given phalanges, and d_i is the translation of the joint, this translation represents the displace ment of thejoint centrs what is caused by the straching of the elastic tissue. This is a not controled DOF.

V. Sensors

The human hand is not just a mechanical but a sensing tool as well. The hand can distinguish between different grasped objects based on tactile information (size, shape, material, and surface properties). The five main *tactile* sensors in the hands are: Meisner, Merkel, Ruffini, Vater-Pacini for actuali touch sensing and free nerv endings for pain.

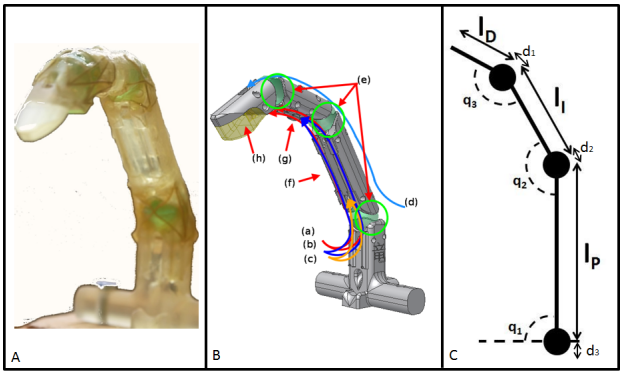


Fig. 4: (A) The full finger prototype. (B) The guiding of the flexor tendons(a)(b)(c), extensor (d), bio-mimetic joint(e), knucklebones (h)(g)(f). (C) The kinematic description of the finger, l_i is the length of the given phalanges, q_i is the angle between the i and $i+1$ phalanges and d_i is the translation in the joint

Besides the sense of touch the other important sens is *proprioception*. Proprioception is the sense of the relative position of neighboring parts of the body. Proprioception provides slow feedback on the status of the body internally. It is the sense that indicates whether the body is moving with the required effort, as well as where the various parts of the body are located in relation to each other.

In order to achive such a big modality of sensing in the bio-mimetic joint it would be needed a lot of different types of sensors which would required more space, computation power and draining. We selected the following sensors for the different senses.

A. Touch sensing

1) 3D Force and Torsion sensor [13]: The sensor (Optoforce, HUN) is a new and low-cost 3D optical comliant tactile sonsor that is capable of measuring a three-axial directional force component as well as an incipient slip. The sensor is easily scalable(finger tip or palm sized) and it's compliant surface makes it prone to contact and will increase the contact stability. It has a robust structure with a respectable overload capability, high sensitivity, high dinamic range boath in time and force domain at low noise operation. A picture of the sensor can be seen on Figure 5.

2) Pressure sensor: A $44x44$ pressure array (Model: 5051 , TekScan, USA), consists of two thin, flexible polyester sheets. The electrically conductive electrodes, wThis sensor is designed in to the palm.

VI. Future Works & Conclusions

The functional test showed promising results, but there is still room for improvement. First of all the investigation of the movement of the human hand to achieve "human-like" behavior, the muscle structure of the forearm to improve a better optimal strategy for the actuation system to reduce the amount of them. A low-level control which works as reflexes. A second important objective that we are pursuing is to implement a global–control, based on the functional, biological structure of the cerebellum. This very challenging goal could ultimately lead to the development of a novel biomechatronic hand.

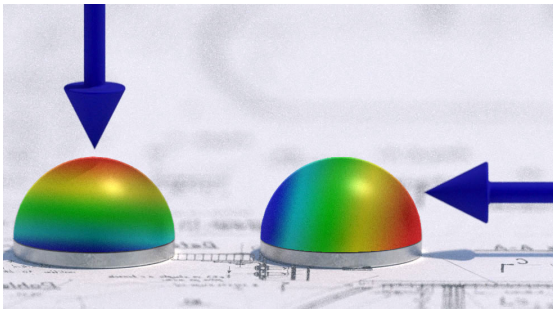


Fig. 5

Acknowledgment

I would like to thank the multidisciplinary doctoral school at the Faculty of Information Technology of the Pázmány Péter Catholic University,the Bolyai János Research Scholarship of the Hungarian Academy of Sciences for they support. The authors are also grateful to Professor Tamás Roska, and the members of the Robotics lab for the discussions and their suggestions. And special thanks go to Varinex Zrt. for the prototyping.

References

[1] H. Kawasaki, T. Komatsu, and K. Uchiyama, "Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II," *Mechatronics, IEEE/ASME Transactions on*, vol. 7, no. 3, pp. 296–303, 2002.

[2] V. Weghe, M. Rogers, M. Weissert, and Y. Matsuoka, "The ACT hand: design of the skeletal structure," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4, pp. 3375–3379, IEEE, 2004.

[3] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR-Hand II: Next generation of a dextrous robot hand," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1, pp. 109–114, IEEE, 2006.

[4] M. Carrozza, B. Massa, S. Micera, R. Lazzarini, M. Zecca, and P. Dario, "The development of a novel prosthetic hand-ongoing research and preliminary results," *Mechatronics, IEEE/ASME Transactions on*, vol. 7, no. 2, pp. 108–114, 2002.

[5] S. Jacobsen, E. Iversen, D. Knutti, R. Johnson, and K. Biggers, "Design of the Utah/MIT dextrous hand," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, pp. 1520–1532, IEEE, 2002.

[6] C. Lovchik and M. Diftler, "The robonaut hand: A dexterous robot hand for space," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2, pp. 907–912, IEEE, 2002.

[7] R. Tubiana, "Architecture and functions of the hand," *The hand*, vol. 1, 1981.

[8] G. Monkman, S. Hesse, and R. Steinmann, *Robot grippers*. John Wiley and Sons, 2007.

[9] R. Drake, W. Vogl, and A. Mitchell, *Gray's anatomy for students*. Elsevier/Churchill Livingstone Philadelphia, 2005.
[10] J. Doyle and M. Botte, *Surgical anatomy of the hand and upper extremity*. Lippincott Williams & Wilkins, 2003.
[11] G. Vasarhelyi, M. Adam, E. Vazsonyi, Z. Vizvary, A. Kis, I. Barsony, and C. Ducso, "Characterization of an integrable single-crystalline 3-d tactile sensor," *Sensors Journal, IEEE*, vol. 6, no. 4, pp. 928–934, 2006.
[12] J. Criag, "Introduction to robotics: mechanics and control," 2005.
[13] A. Tar, and G. Cserey, "Development of a low cost 3D optical compliant tactile force sensor, "2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), IEEE,pp. 236–240, 2011.

Medical imaging algorithms on kiloprocessor architectures

Gábor János Tornai
(Supervisors: Tamás Roska, György Cserey)
tornai.gabor@itk.ppke.hu

***Abstract*—This paper presents two fields that are motivated by real medical imaging problems. The first is the generation of Digitally Reconstructed Radiographs (DRRs) that is the most time consuming step in intensity based X-ray to CT registration. The second is the fast calculation of Level set dinamics that is widely used in detection and segmentation. The first part of this work presents optimized DRR rendering on graphical processor units (GPUs) and compares performance achievable on four devices. A ray-cast based DRR rendering was realized for a 512×512×72 CT volume. The block size parameter was analized for four different GPUs. Performance was statistically evaluated and compared for the four GPUs. The method and the block size dependence were validated on the latest GPU with a public gold standard dataset (512×512×825 CT).The rendering requires 0.3-5.2 ms. The presented results out-perform other results from the literature, thus automatic 2D to 3D registration can be performed quasi on-line, in less than a second. The second part of this work presents modified initial conditions for the level set framework. Their evolution can be calculated on manycore architectures effectively. Results showed that on manycore architecture the iteration number can be smaller than 4. Furthermore, new problems can be solved effectively. This is demonstrated with a gray matter segmentation on a T1 weighted MR image.**

***Keywords*-registration; DRR; GPU; block size; level set; optimal initial condition**

I. Introduction

Digitally Reconstructed Radiographs (DRRs) are simulated X-ray images generated by projecting 3D computed tomography (CT) images or 3D reconstructed rotational X-ray images. There are numerous papers presenting a wide spectrum of results connected to acceleration of DRR generation or reducing the required number of renderings. Table I presents a condensed summary of the reported results in acceleration of DRR rendering. In addition, it presents specific results in 2D to 3D registration that are straightforward in the way of DRR rendering, as well.

In many applications closed curves or (hyper)surfaces around a region need to be foud or traced. This task is handled with the level set framework [1]. In this paper a subset of these tasks are considered, where the exact time evolution of the model is out of our interest but the steady state solution of the given PDE should be approximated. This subset has especially great importance in medical imaging.

The paper is organized as follows. First the DRR rendering material is presented in sections II,III. Than the level set connected topics follow in sections IV,V. Section VI concludes the paper.

II. Methods – DRR

A. GPU overview

Recent GPU models are capable of non-graphics operations and are programmable through general purpose application programming interfaces (APIs) like C for CUDA [8] or OpenCL [9]. In this paper, C for CUDA nomenclature is used. The description below is brief overview of GPUs, and only those notations are summarized that have an impact on the performance of the DRR rendering.

A function that can be executed on the GPU is called a kernel. Any call to a kernel must specify an execution configuration for that call. This defines not just the number of threads to be launched but the arrangement of groups of threads to blocks and blocks to a grid. The dimensionality of a block can be one, two or three while the grid is either one or two dimensional. The number of threads in a block is referred to as block size. For a given number of threads the block size has a great impact on performance and there are no explicit rules to find its optimal value so far. Threads of a block are organized into warps. Threads in a warp are scheduled physically together on the device. The warp size is 32 on all GPUs that are employed in this work. As a consequence, the vendor advises block sizes that are multiples of 32. The parallel thread execution (PTX) [10] is an intermediate, device independent GPU language above Assembly. During the compilation, the kernel is translated first to PTX and then compiled to device dependent code. The used devices: 8800 GT, 280 GTX, Tesla C2050 and 580 GTX.

B. Data and measurement

The ROI is sampled randomly: the locations of the pixels are choosen by a 2D uniform distribution. Seven sampling ratios and full sampling are investigated: 1024, 1536, 2048, 3072, 4096, 6144, 8192 and 90000 pixels (full sampling, 400×225). This last case is referred to as full ROI DRR. Each pixel intensity is calculated by one thread on the GPU. So the number of pixels are equal to the number of threads launched on the device. The block size parameter was optimized for each number of pixels and GPU pair. This amounts to 32 cases (4 GPUs, 8 thread values). For each block size value, more than 1000 kernel executions were measured. In this work the dimensionality of the block size parameter is one so is the grid size.

The pixel locations were resampled for each kernel execution. Similarly, for each kernel execution the initial reference

Table I
SUMMARY OF PUBLISHED RESULTS CONNECTED TO FAST DRR RENDERING AND 2D TO 3D REGISTRATION.

Hardware	Year	ROI size	Perf. DRR	Perf. Reg.	Comments	Group
Intel Xeon (2.2 GHz)	2005	256×256	50 ms	100 s	SW cache, 0.4-6 h preproc	Russakoff [2]
Intel Pentium 4 (2.4 GHz)	2005	200×200	-	160-1100 s	SW cache, no preproc	[3]
Intel Xeon	2009	256×256	-	30-120 s	ROI randomly sampled (5%)	[4]
Intel Core (1.8 GHz octal)	2009	256×172	50 ms	-	multithreaded ray cast	Dorgham [5]
GeForce 7600 GS (12 cores)	2007	512×512	54-74 ms	-	voxel based, ($6 \cdot 10^6$ voxel)	[6]
GeForce 7800 GS (16 cores)	2008	256×256	73 ms	-	X-ray postprocessing imitaiton	[7]

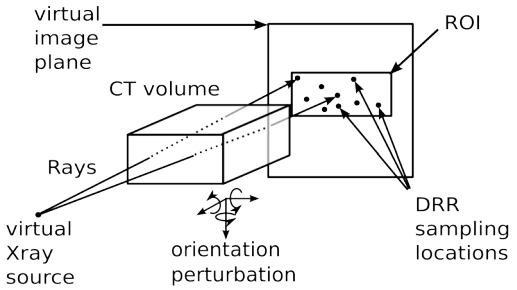


Figure 1. DRR rendering geometry. Rays are determined by virtual X-ray source and sampling locations on the virtual Image plane, inside the ROI. Pixel intensities are approximated line integrals along the dashed line segments (volume interior). ROI is resampled for each DRR rendering. Similarly, the CT position and orientation are varied by uniform distribution.

pose of the CT volume was varied (perturbed) in the range of ± 20 mm and ± 15 deg by uniform distribution. The perturbation of the volume pose and the resampling of the pixel locations mimics the repetitive DRR rendering need of a 2D to 3D registration process. It shall be noted, other results [4], [11], [12] showed that 2D to 3D image registration algorithms can robustly converge with good accuracy even if only a few percent of the pixels are sampled randomly.

For the execution time measurements a CT scan (manufactured by GE Healthcare, CT model Light Speed 16) of a radiological torso phantom (madufactured by Radiology Support Devices, Newport Beach, CA, model RS-330) was used. The resolution of the reconstructed image was $512 \times 512 \times 72$ with data spacing (0.521 mm, 0.521 mm, 1.25 mm). For validation purposes from the gold standard data set [13] a $512 \times 512 \times 825$ CT with data spacing (0.566 mmm, 0.566 mm, 0.4 mm) was used. Measurements were done on four Nvidia (Santa Clara, CA) GPUs which were state of the art devices. GPU driver and CUDA toolkit (compilation tools and libraries) were developed by Nvidia, their version were 260.16.21 and 3.2 respectively. The hosting PC contained an Intel Core2 Quad CPU (2.66GHz) and 4GB of system memory, running Linux kernel 2.6.32-5-amd64.

III. RESULTS – DRR

Table II presents mean and standard deviation of optimized execution times together with the optimal block size value. For comparision, the mean of execution times with naive block size value (256) are also presented. As an example, Fig 2 shows the block size dependences of rendering 1024, 1536 and 2048 pixels on a 580 GTX GPU.

Computation time dependence on block size is in the range of $10 - 60\%$ on 8800 GT and 280 GTX GPUs in the case of sampled DRRs if optimal execution time is considered as 100%. This variation is in the range of $23 - 145\%$ on Tesla C2050 and 580 GTX GPUs. If full ROI DRRs are considered, the variation is in the range of $10 - 34\%$ on all devices. The mean of texture cache hit ratio is 15% in the case of sampled DRRs on all devices. However, in the case of full ROI DRRs the hit rate is nearly perfect, so the proportional time to compute a pixel intensity is reduced by 90% in average.

Rendering times of full ROI DRRs are comparable with other results from the literature (Table I). In this work approximately $6.3 \cdot 10^6$ voxels are visited in order to render a full ROI DRR. This is a 7 fold speedup compared to the result of [6] (Table I) with similar number of voxels rendered and normalized by the ROI. Unfortunately, there are no explicit results on sampled DRR rendering. The available ones are implicit and based on 2D to 3D registration performance [4].

The results have been validated against a publicly available gold standard data set for registration purposes [13] on a 580 GTX GPU. The characteristics of the method were nearly identical on the two data set not counting the execution time scaling. This is shown on figure 2 for the following thread numbers: 1024, 1536 and 2048. This validation indicates that our optimized method will behave similarly on bigger data set with different spacing values.

IV. LEVEL SETS

The basic model for the level set evolution is a simple one: $\frac{\partial \phi}{\partial t} + \vec{F} \cdot \nabla \phi = 0$ Where F can be any arbitrary function and the underlying curve is the zero level set of ϕ that is going to be denoted as the level set function (it is considered as negative inside the zero level set and positive outside). According to the considerations in [14] one might omit the exact solution of the underlying PDE and use a rule based approach. Let us assume that ϕ is defined over a domain D $\in R^K$ $(K \geq 2)$. One can define two sets namely L_{in}, L_{out} as follows:

$$L_{in} = \{\mathbf{x} | \phi(\mathbf{x}) < 0 \; and \; \exists \mathbf{y} \in N(\mathbf{x}) \; that \; \phi(\mathbf{y}) > 0\} \quad (1)$$

$$L_{out} = \{\mathbf{x} | \phi(\mathbf{x}) > 0 \; and \; \exists \mathbf{y} \in N(\mathbf{x}) \; that \; \phi(\mathbf{y}) < 0\} \quad (2)$$

$$N(\mathbf{x}) = \{\mathbf{y} \in D | \sum_{k=1}^{K} |y_k - x_k| = 1\} \; \forall \mathbf{x} \in D \quad (3)$$

Having these two sets, the motion of the zero level set can be obtained by investigating only the sign of the force field. In this way replacing elements from one set to the other the desired motion is received.

Table II
OPTIMIZED EXECUTION CHARACTERISTICS. COLUMNS 't_o' REPRESENT THE MEANS AND STANDARD DEVIATIONS OF OPTIMIZED EXECUTION TIMES OF DRR COMPUTING KERNEL. COLUMNS 'BS' SHOW OPTIMIZED BLOCK SIZES FOR DEVICE AND THREAD NUMBER PAIRS. COLUMNS 't_n' PRESENT MEAN OF EXECUTION TIMES WITH NAIVE BLOCK SIZE OF 256.

	8800 GT			280 GTX			Tesla c2050			580 GTX		
# of pixels	t_o (μs)	bs	t_n (μs)	t_o (μs)	bs	t_n (μs)	t_o (μs)	bs	t_n (μs)	t_o (μs)	bs	t_n (μs)
1024	1257±122	160	1404	547±66	96	882	417±39	10	1020	297±22	8	727
1536	1588±137	10	1978	826±110	160	1080	510±21	14	1006	342±18	16	709
2048	2128±183	14	2923	1144±150	12	1258	690±11	32	1000	391±17	16	697
3072	3016±223	8	4056	1480±112	8	1934	886±46	32	1705	550±21	192	677
4096	3922±338	10	6126	1922±149	10	2536	1159±81	64	1818	700±37	128	702
6144	5815±488	16	9270	2641±188	10	3992	1508±96	64	2478	1006±56	192	1307
8192	7300±637	32	7972	3424±222	10	4667	2222±242	128	2907	1290±80	256	-
full ROI	5269±575	128	7060	4545±425	32	5963	3989±628	128	4382	2666±600	128	3131

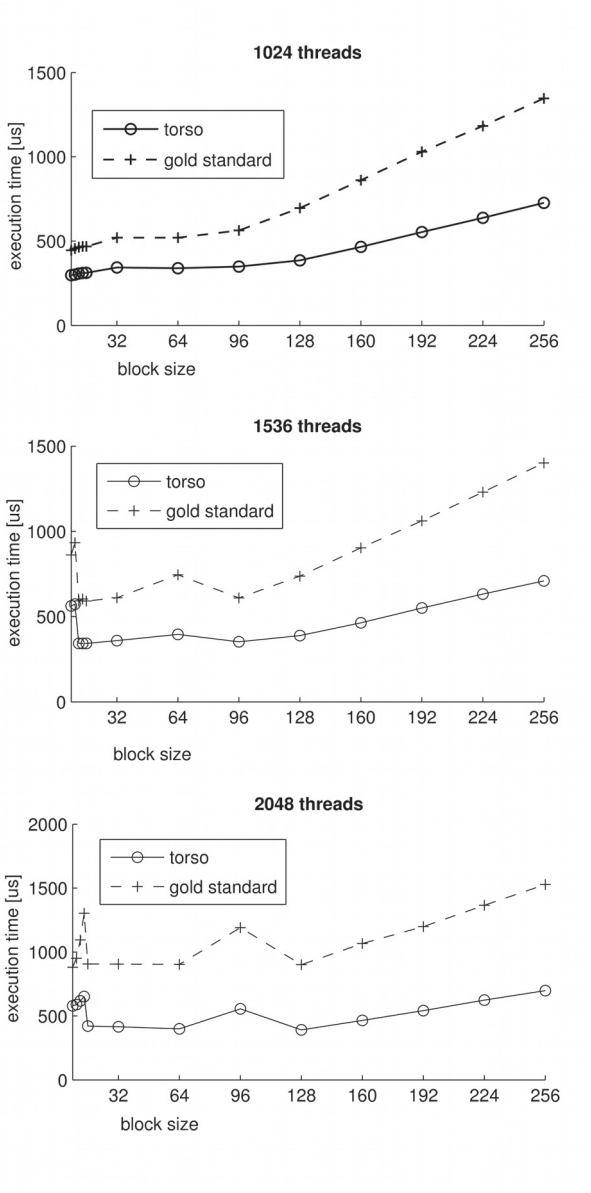
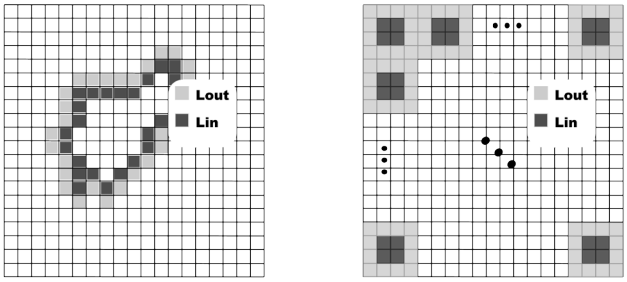


Figure 2. Block size dependence of average execution time of rendering 1024, 1536 and 2048 pixels from the 400×225 ROI on 580 GTX GPU from the 72 slice and the gold standard 825 slice CT respectively. The block size dependence is nearly identical in the two cases.



(a) sequential (b) parallel, manycore

Figure 3. Initial conditions (a) optimized for sequential machine and (b) for parallel machine. In the first case the size of the active front is relative small compared to the whole image. On the parallel one the whole image is an active front so the computing width of the parallel machine can be utilized. The reason is the difference in the $\max(\delta)$ value. On (a) it is proportional with the size of the width of the picture, but on (b) it is one. This makes the difference in the required number of iterations.

V. OPTIMAL INITIAL CONDITION

A. Theory

Definition 1. *Inner distance index (d_i^x) of a given pixel (x) is a positive number denoting the distance of the closest positive pixel according to the connectivity scheme.*

Definition 2. *Outer distance index (d_o^x) of a given pixel (x) is a natural number denoting the distance of the closest negative pixel according to the conectivity scheme.*

Definition 3. *General distance index (d_g^x) of a given pixel (x) is a natural number denoting the distance of the closest pixel of the opposite sign according to the connectivity scheme.*

Definition 4. *Distance index function, δ of a discretised region D and a given configuration (ϕ) is as follows. $\delta : (x \times \phi) \mapsto d_g^x$, where $x \in D$ and ϕ is the level set function.*

On manycore systems a greater area of active front can be investigated simultaneously. So the classical initial condition may be changed. This is illustrated in Fig. 3. A sequential machine optimized initial condition has a small active front relative to the image. On a parallel machine optimized one the whole area is an active front so the computing width be

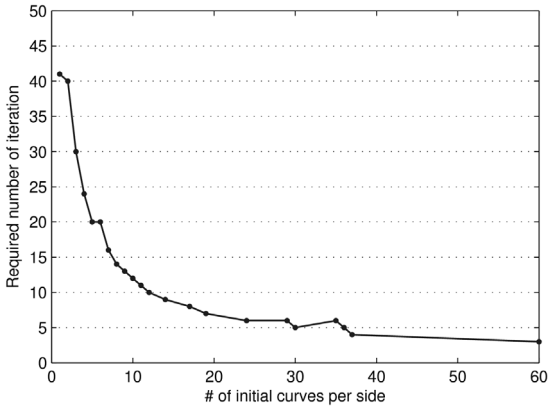


Figure 4. Required number of iterations depending on the initial condition on a 128×128 image.

(a) sequential (b) parallel, manycore

Figure 5. The first subfigure is the output if a sequential type initial condition is used while the second subfigure is the output if a 16×16 array of circles are used as initial front.

utilized.

B. Experiments

Figure 4 demonstrates on a 128×128 image how the required iteration decreases if the number of initial fronts are increased. Furthermore, this new initial condition succesfully finds disjoined parts of the same object. This can be seen on fig. 5. The first subfigure is the output if a sequential type initial condition is used while the second subfigure is the output if a 16×16 array of circles are used as initial front.

VI. Conclusions

To automatically register the content of an X-ray projection to a 3D CT, 20-50 iteration steps are required. For each iteration, 10-20 DRRs are computed depending on the registration procedure. On the whole this amounts to 200-700 DRRs to be rendered for a registration to converge. DRR rendering is the most time consuming part of the 2D to 3D registration. Following our implementation rules, the time requirements of a registration process can be decreased to 0.6-1 s if full-ROI DRRs are applied. If random sampling is used the time requirement of registration can be further reduced to *0.1-0.7 second* resulting in quasi real time operability.

References

[1] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 2000.

[2] D. Russakoff, T. Rohlfing, K. Mori, D. Rueckert, A. Ho, J. Adler, and C. Maurer, "Fast generation of digitally reconstructed radiographs using attenuation fields with application to 2D-3D image registration," *Medical Imaging, IEEE Transactions on*, vol. 24, no. 11, pp. 1441–1454, 2005.

[3] T. Rohlfing, D. B. Russakoff, J. Denzler, K. Mori, and C. R. Maurer, "Progressive attenuation fields: Fast 2D-3D image registration without precomputation," *Medical Physics*, vol. 32, no. 9, pp. 2870—2880, 2005. [Online]. Available: http://link.aip.org/link/MPHYA6/v32/i9/p2870/s1&Agg=doi

[4] W. Birkfellner, M. Stock, M. Figl, C. Gendrin, J. Hummel, S. Dong, J. Kettenbach, D. Georg, and H. Bergmann, "Stochastic rank correlation: A robust merit function for 2D/3D registration of image data obtained at different energies," *Medical physics*, vol. 36, p. 3420, 2009. [Online]. Available: http://dx.doi.org/10.1118/1.3157111

[5] M. F. Dorgham and S. Laycock, "Accelerated generation of digitally reconstructed radiographs using parallel processing," in *Proc. Medical Image Understanding and Analysis*, 2009, p. 14–15.

[6] J. Spoerk, H. Bergmann, F. Wanschitz, S. Dong, and W. Birkfellner, "Fast DRR splat rendering using common consumer graphics hardware," *Medical Physics*, vol. 34, no. 11, p. 4302, 2007. [Online]. Available: http://link.aip.org/link/MPHYA6/v34/i11/p4302/s1&Agg=doi

[7] A. Kubias, F. Deinzer, T. Feldmann, D. Paulus, B. Schreiber, and T. Brunner, "2D/3D image registration on the GPU," *Pattern Recognition and Image Analysis*, vol. 18, no. 3, p. 381–389, 2008.

[8] NVIDIA, "CUDA c programming guide," Sep. 2011. [Online]. Available: http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf

[9] "OpenCL," http://www.khronos.org/opencl/, Jul. 2011. [Online]. Available: http://www.khronos.org/opencl/

[10] NVIDIA, "PARALLEL THREAD EXECUTION ISA VERSION 3.0," Jan. 2012. [Online]. Available: http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/ptx_isa_3.0.pdf

[11] L. Zollei, E. Grimson, A. Norbash, and W. Wells, "2D-3D rigid registration of x-ray fluoroscopy and CT images using mutual information and sparsely sampled histogram estimators," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2, 2001, pp. 696–703 vol.2.

[12] L. Zollei, J. Fisher, and W. Wells, "A unified statistical and information theoretic framework for multi-modal image registration," in *Information Processing in Medical Imaging*, ser. Lecture Notes in Computer Science, C. Taylor and J. Noble, Eds. Springer Berlin / Heidelberg, 2003, vol. 2732, pp. 366–377. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-45087-0_31

[13] S. A. Pawiro, P. Markelj, F. Pernus, C. Gendrin, M. Figl, C. Weber, F. Kainberger, I. Nobauer-Huhmann, H. Bergmeister, M. Stock, D. Georg, H. Bergmann, and W. Birkfellner, "Validation for 2D/3D registration i: A new gold standard data set," *Medical Physics*, vol. 38, no. 3, p. 1481, 2011. [Online]. Available: http://link.aip.org/link/MPHYA6/v38/i3/p1481/s1&Agg=doi

[14] Y. Shi, "Object based dynamic imaging with level set methods," PhD, Boston Univ. College of Eng., 2005.

Estimation of Relative Direction Angle of Distant, Approaching Airplane in Sense-and-avoid

Tamás Zsedrovits
(Supervisors: Tamás Roska and Ákos Zarándy)
zseta@digitus.itk.ppke.hu

***Abstract*— Visual detection based sense and avoid problem is more and more important nowadays as UAVs are getting closer to entering remotely piloted or autonomously into the airspace. It is critical to gain as much information as possible from the silhouettes of the distant aircrafts. In our paper, we investigate the reachable accuracy of the orientation information of remote planes under different geometrical condition, by identifying their wing lines from their detected wingtips. Under the assumption that the remote airplane is on a straight course, the error of the spatial discretization (pixelization), and the automatic detection error is calculated.**

***Keywords* — UAV, See and Avoid, long range visual detection.**

I. INTODUCTION

According to many aviation experts, pilotless aircrafts are going to revolutionize air transport in the near future. As written in the cover story of December 2011 issue of IEEE Spectrum Magazine: "A pilotless airliner is going to come; it's just a question of when," said James Albaugh, the president and CEO of Boeing Commercial Airlines [1]. Surely, this final goal is expected to be achieved step-by-step.

One of the most important problems which has to be solved is the collision avoidance or sense-and-avoid capability. Provided that the size and the energy consumption of the Unmanned Aerial Vehicle (UAV) are limited, a camera based avoidance system would provide cost and weight advantages against radar based solutions [2], [3]. Furthermore near airfields, because of a great density of aircrafts and the limited frequency resources of air traffic controllers the camera-based approach seems to be more feasible then others. Today's kilo-processor chips allow us to implement complex algorithms in real time with low power consumption.

In [4], [5], [6] and [7] a camera-based autonomous on-board collision avoidance system and its implementation aspects on kilo-processor architectures are introduced. This sense-and-avoid system is capable of avoiding a single target as long as the lighting conditions are good, or the sky is nearly homogenous. If the intruder is far from our camera, less information can be obtained with image processing, but from a given distance the shape of the intruder is distinct, thus shape analysis can be used to get more information [8].

Provided that the intruder aircraft is close enough to our UAV its wing can be seen, the relative angle of attack can be obtained and can be used to estimate its trajectory. In this paper the automatic estimation process is introduced and the precision in miscellaneous situations are studied. The automatic solution is compared to the ground truth and to the theoretically computed values in each situation. For the measurements realistic images rendered by FlightGear flight simulator is used [4].

II. GEOMETRICAL DESCRIPTION

In this section the geometrical description of the studied situation is introduced. Let us assume that we have one intruder aircraft and it is approaching to our UAV and it follows a straight trajectory. In this case the position of the intruder on the image plane changes slowly (given no self motion).

This situation is unobservable with our Kalman-filter based estimation algorithm [5], which estimates the 3D position of the intruder from the change of the coordinates of the intruder in the image plane. Thus, additional information is required in order to determine the relative position of the intruder aircraft. For one thing, this information can be achieved with running an excitatory manoeuvre [9], which consumes fuel, which is a limited resource on a UAV.

On the other hand, if wingtips of the intruder aircraft can be distinguished on the image, the relative direction angle can be estimated.

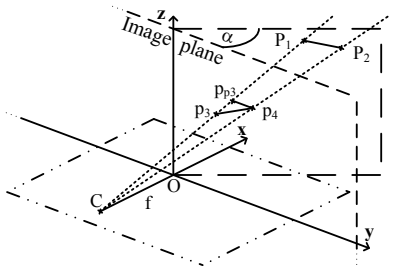


Fig. 1. Diagram of the relative direction angle (α) calculation: C is the camera centre; f is the focal length; O is the centre of the image plane (yz plane) and the origin; $\overline{P_1P_2}$. is the model of the wing of the intruder aircraft in space; $\overline{p_3p_4}$ is the wing in image plane; p_{p3} is the projection of p_3 to the horizontal line goes through p_4

Provided that the intruder is coming towards us, it grows in the image. In the beginning this growth is slow and later it accelerates. The relative bank angle of the intruder in the picture, namely the coordinates of the wingtips, is measurable.

As shown in Fig. 1. the wing of the intruder in the image plane is projected to $\overline{p_3p_4}$ and in space it is $\overline{P_1P_2}$. It is assumed that the wing of the intruder is horizontal, that is parallel with xy, assuming straight level flight. The centre of our coordinate system is the central point of the recorded image and the yz plane is the image plane. It is assumed, that the images are transformed to the NED frame [10].

If the intruder isn't in xy plane, therefore none of its image coordinates are 0 in the image coordinate system, the line going through the two wingtips includes an angle introduced by the z axis offset. Assuming p_4p_{p3} is parallel with y, from this $p_3p_4p_{p3}$ angle we would like to estimate the intruder's relative angle in 3D (α) that is its direction, which can be used to enhance the estimation. Consequently this $p_3p_4p_{p3}$ depends on the angle α and the subtended angle in which is seen.

If the intruder is on the xy horizontal plane, p_3 equals p_p3 and the α angle cannot be estimated with this algorithm. The altitude of our UAV can be easily changed with acceleration or deceleration, which consumes less fuel than the complex excitatory manoeuvre mentioned before.

The angle α can be calculated as follows [11]:

$$\cos\alpha = \frac{\langle p_{p3}-p_4; P_1-P_2\rangle}{\|p_{p3}-p_4\|\|P_1-P_2\|}.$$

In this model the instances rotated by 180° are equal and the $\alpha = \cos^{-1} X$ function gives good solution in $\alpha = [0°; 180°]$ range. The relative angle α should be in the $[-90°; 90°]$ range, so it is transformed according to the following rules. If $\alpha > 90°$, then $\alpha = 180° - \alpha$, if $\alpha < -90°$, then $\alpha = -180° - \alpha$. With these calculations the expected results are obtained consistently.

III. MEASUREMENT SITUATIONS

The accuracy of the calculation is studied with given image resolution and position. Three kinds of situations are examined:

1) With pinhole camera model, the given centroid point of the intruder is projected back from image plane to space to several distances. The wingspan of the intruder is 11m (36 ft 1 in), which is the wingspan of Cessna 172, a typical light aircraft that shares the airspace with our UAV. Thus the wing is represented by an 11m line segment and is rotated in the previously calculated point. The field of view and resolution of the camera and the distance along x axis is required for the calculation. The fuselage of the aircraft is neglected, which gives an initial error. With these calculations the lower bound of the error is approximated. Two kinds of points are used: a) calculated points without rounding to determine the error induced by the limited numerical precision; b) calculated points with rounding to determine the error induced by the discretization in space

2) With the calculated centroid points in space according to section 1) images are taken from FlightGear flight simulator. The wingtip coordinates are taken by a human expert from these simulated images and the angle values are calculated from these coordinates.

3) Similarly to the above, the intruder points are extracted from the simulated images rendered by FlightGear with our image segmentation algorithm [4]. After that, from intruder pixel coordinates the wingtip coordinates are calculated with the following simple algorithm. The wingtip coordinates are determined by the extremes of the y and z coordinates in the appropriate order. In order to reduce the error induced by the image formation, the calculated coordinates are refined according to the image pixel values with the following expression:

$$p_{corrected\_y} = \frac{\sum_{i=p_y-s}^{p_y+s} i * p_v^i}{\sum_{i=p_y-s}^{p_y+s} p_v^i}$$

where $p_{corrected\_y}$ is the refined coordinate value, p_y is the original coordinate value, s is the radius, p_v^i is the grayscale value of the i[th] point.

In this section the measurements are described in situations introduced in chapter III. The position dependence of the error and the effect of the discretization are shown.

B. *Pinhole camera*

First the pinhole camera model is used. Provided that the points are calculated without rounding, this approach should come close to the theoretical limits and the computation error has to be near zero.

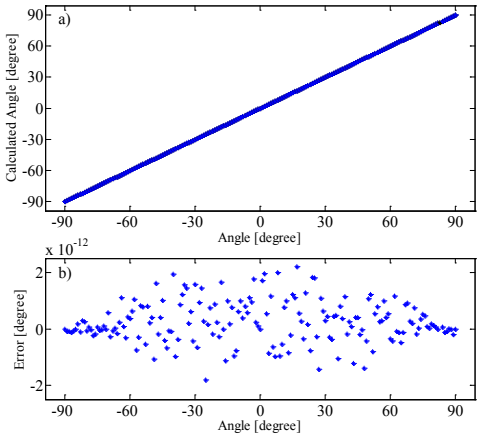


Fig. 2. α angles calculated from pinhole model without rounding and their error to ground truth; a) the original angles with black (covered by calculated angles) and the calculated angles with blue; on the bottom of the figure the error values for each calculated angle

The measurements are done with double precision and the error of the angles is in the range of picodegree as shown in Fig. 2, which is the range of the error introduced by the numeric representation. Indeed this error can be seen as zero in the point of the computation part.

In Fig. 2. a) the real rotation angles versus the calculated angel values are shown, and the part b) depicts the error of the estimated angle, which is the difference between the two angles. The distance along the x axis to the image plane is 2 km (1.24 miles) and the intruder is seen in 7° azimuth and elevation angle offset.

Let us assume that a typical HD camera is used to record the scene. This camera is calibrated and the recorded pictures are undistorted, thus the pinhole camera model can be a valid approximation. The difference between this measurement scenario and the one stated above is that here the image coordinates are discrete integer values and the image plane is finite.

According to the measurements, the precision of the estimation with a given camera depends on the subtended angle and the relative distance along the x axis. Undoubtedly, it isn't surprising because the larger the distance the smaller the intruder in the image and the bigger the altitude difference the more you observe the wing of the intruder.

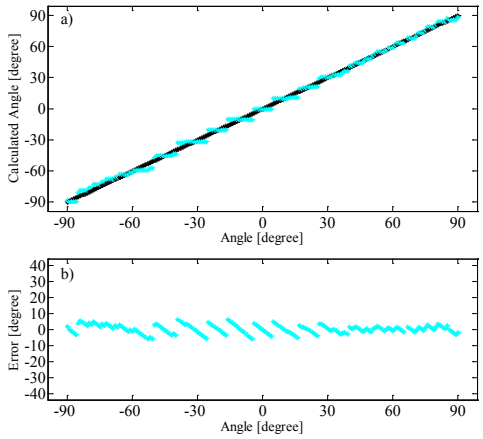


Fig. 3. α angles calculated from pinhole model with rounding and their error to original rotation angles; a) the original angles with black and the calculated angles with cyan; b) the error values for each calculated angle (max ±6°); the intruder is seen in (24°, 14°) direction and the distance along x axis is 1km

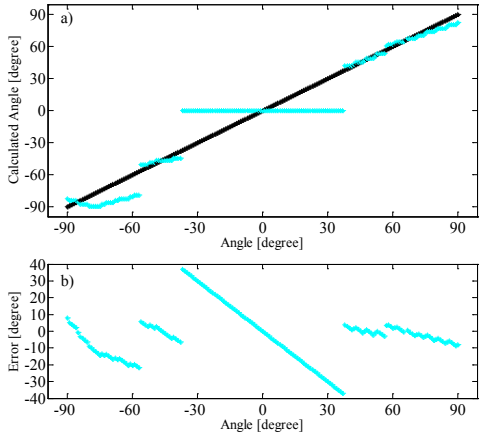


Fig. 4. α angles calculated from pinhole model with rounding and their error to original rotation angles; same as before, the subtended angle is (24°, 3.5°) and the maximum error is ±37°

The two figures (Fig. 3, Fig. 4) show examples where the relative distance along the x axis is 1 km (0.62 miles), the resolution is 1920x1080 pixels, the horizontal field of view is 50° and the pixels are squares. The wingspan of the intruder is 11m (36 ft 1 in), which is the wingspan of Cessna 172. The size of intruder in the image plane is between 15 and 20 pixels, depending on the rotation angle and the position. The intruder is seen in 14° and 3.5° elevation successively, and it is seen constantly in 24° azimuth.

The following figure (Fig. 5) shows the maximum error values in each subtended angle with constant azimuth of 24° and with changing elevation from -14° to 14°. In each position the intruder is rotated with angles from -90° to 90° and the maximum of the absolute of the error is chosen. This measurement shows the position dependence of the calculated α. Fig. 5. depicts that the initial error is ±6° and the closer the intruder is to the horizontal axis the bigger the error we get.

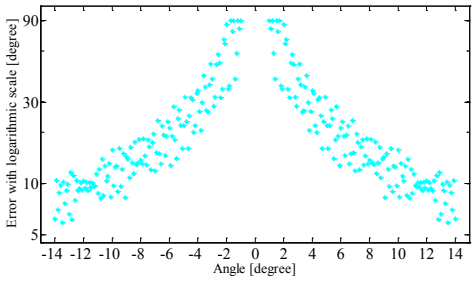


Fig. 5. Maximum of absolute value of the errors of the rounded α angles calculated with pinhole camera model in different relative vertical positions and from 1 km distance along the x axis; in the figure on the horizontal axis the vertical angle in which the intruder is seen; on the vertical axis the error in degree with logarithmic scale

C. *Points by human expert on simulated images*

In our simulation environment [6] pictures is taken and the wingtip pixel coordinates are selected by a human expert. The intruder is placed in space according to section III. 1) and in every position it is rotated by specific angles in the xy plane. The resolution is 1920x1080 pixels and the horizontal field of view is 50° and the pixels are squares, such as in the previous case B.

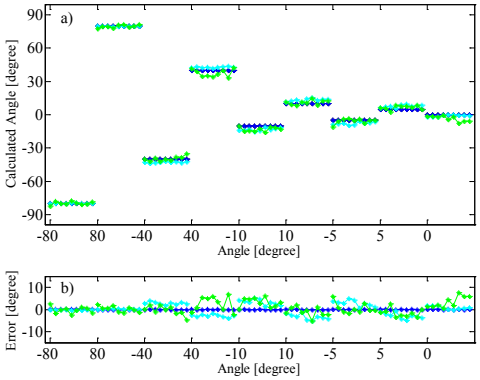


Fig. 6. α angles calculated from coordinates selected by a human expert on images generated by FlightGear simulator; a) angles in different vertical positions, on the vertical axis the angle values, on the horizontal axis the real rotation angles in 9 different positions; b) the error; original angles with black (covered), angles calculated from pinhole model with blue, angles calculated from pinhole model with rounding with cyan, angles calculated from coordinates selected by hand with green

In Fig. 6. a) the ground truth α values are with black (covered). The angles calculated from pinhole camera model

are shown with blue; the values calculated from rounded coordinates are shown with cyan and the angles calculated from points selected by hand are shown with green. On Fig. 6. b) the error values are shown and the colours are similar to previous. The figure depicts only the result of the measurement in one specific distance. The intruder was placed in 9 different positions and was rotated with 9 different angles (-80°, 80°, -40°, 40°, -10°, 10°, -5°, 5°, 0°). The other results obtained from another distances are similar to that are described previously in section *A*, thus the altitude difference is in inverse ratio to the error.

The measurements above shows that with good wingtip coordinates in realistic situation the error can be near to theoretical minimum.

D. Points by automatic algorithm on simulated images

The error of the automatic wingtip detection algorithm running on simulated images has been measured. The simple algorithm determines the wingtip coordinates from the segmented images. The extreme of y and z coordinates are used in appropriate order to get the coordinates

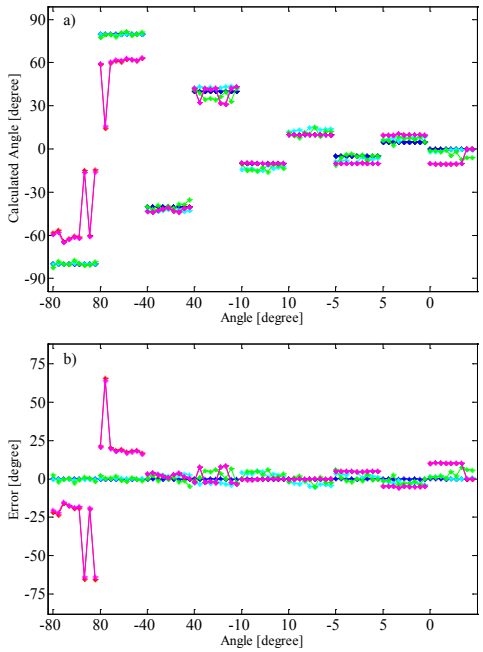


Fig. 7. α angles calculated from coordinates calculated by the automatic algorithm on images generated by FlightGear simulator; a) angles in different vertical positions, on the vertical axis the angle values, on the horizontal axis the real rotation angles in 9 different positions; b) the error; original angles with black (covered), angles calculated from pinhole model with blue, angles calculated from pinhole model with rounding with cyan, angles calculated from coordinates selected by hand with green, angles calculated automatically with red and the corrected values with magenta

Fig. 7. depicts one example, where similarly to section *B*, the intruder had been placed in a specific locations in space and then it was rotated with specific angles (same as before). In the figure the ground truth is with black (covered); the values from pinhole camera model are with cyan and blue; the values form points selected by human expert are green; the values from automatic algorithm are with red and the values calculated from corrected points are with magenta. In this case when the intruder had been rotated with 80° and with -80° angles, the error of the estimation is bigger, because the simple algorithm couldn't distinguish between the pixels of the wing and the pixels of the tail.

In contrast, in the mid-range the performance of this really simple algorithm is almost the same as the performance of the human expert.

IV. CONCLUSIONS

The reachable accuracy of the orientation calculation of visually detected remote airplanes was studied. The orientation calculation was based on the detection of the wingtips. As it turned out the relative orientation of the remote aircraft (depicted by α) can be calculated if it is on a straight course, and its level differs from the observer. Naturally, the orientation measurement is more accurate when the level difference is higher, and the airplane is closer. The exact reachable accuracy figures are shown in charts, and their calculation methods are given. The acquired measurements will be used to enhance the estimation accuracy of the currently existing EKF based sense and avoid system.

V. ACKNOWLEDGMENT

The ONR Grant (Number: N62909-10-1-7081) is greatly acknowledged.

VI. REFERENCES

[1] P. E. Ross, "When Will We Have Unmanned Commercial Airliners?" *IEEE Spectrum Magazine*, December 2011

[2] T. Hutchings, S. Jeffryes, and S. J. Farmer, "Architecting UAV sense & avoid systems," *Proc. Institution of Engineering and Technology Conf. Autonomous Systems*, 2007, pp. 1–8.

[3] G. Fasano, D. Accardo, L. Forlenza, A. Moccia and A. Rispoli, "A multi-sensor obstacle detection and tracking system for autonomous UAV sense and avoid", XX *Congresso Nazionale AIDAA, Milano*, 2009.

[4] T. Zsedrovits, Á. Zarándy, B. Vanek, T. Péni, J. Bokor, T. Roska, "Collision avoidance for UAV using visual detection", *ISCAS 2011*

[5] B. Vanek, T. Péni, T. Zsedrovits, Á. Zarándy, J. Bokor and T. Roska, "Performance Analysis of a Vision Only Sense and Avoid System for Small UAVs", *AIAA Guidance, Navigation, and Control Conference 2011*

[6] T. Zsedrovits, Á. Zarándy, B. Vanek, T. Péni, J. Bokor, T. Roska, "Visual Detection and Implementation Aspects of a UAV See and Avoid System", *ECCTD 2011*

[7] B. Vanek, T. Péni, J. Bokor, Á. Zarándy, T. Zsedrovits and T. Roska., "Performance Analysis of a Vision Only Sense and Avoid System for Small UAV", *European Workshop on Advanced Control and Diagnosis 2011*

[8] Pratt, W. K., *Digital Image Processing: PIKS Inside*, PixelSoft Inc., Los Altos, CA, 2001

[9] Hernandez, M. L., "Optimal Sensor Trajectories in Bearings-Only Tracking," Tech. rep., QinetiQ, 2004.

[10] Stengel, R., *Flight Dynamics*, Princeton Press, 2004.

[11] T. Zsedrovits, Á. Zarándy, B. Vanek, T. Péni, J. Bokor, T. Roska, " Estimation of Relative Direction Angle of Distant, Colliding Airplane in Sense-and-avoid and Tracking", *ICUAS 2012*

Retina inspired algorithms: Looming direction detection

Tamás Fülöp
(Supervisor: Dr. Ákos Zarándy)
fulop.tamas@itk.ppke.hu

***Abstract*— The direction of an approaching object is very important with navigation development or collision avoidance systems. The appropriate detection of the looming motion is a key initial step of an intelligent system, while the direction can add extra information for more precise decision strategy. The methods are built on the usual image processing chain, when the extraction of looming properties is the last step. We are proposing looming object detection methods, where these properties can be extracted in the first step. In this paper I intend to show the modified looming detection model, which helps to recognize the direction of an approaching motion.**

***Keywords*-looming detection; bio-inspired algorithm; image processing; retina**

I. INTRODUCTION

The detection of looming motion is important in Nature and also in information technology. As a prey animal, for example a mouse, can survive an approaching eagle attack, or simply can navigate and find its way. The detection of a looming motion is not concerned about what is coming: it focuses on the fact of approaching. Fast looming objects are usually dangerous, so the systems need to be capable of avoiding those situations.

Looming object detection is very important in robotics, and in the development of intelligent vehicles too. Similarly to biological systems, it also needs to avoid the potential risky situations. The active devices, like RADAR or LIDAR, are capable of providing exact information about looming. Despite the precise detection of these devices, these are very expensive and limited to be used everywhere (law, range). Camera based intelligent sensing systems cannot be used in all weather and light conditions, but a CMOS chip is cheap.

Most of the image processing tasks need a large computing capacity. The running cost is influenced by the quality requirements. Some class of object detection has got efficient methods (face, pedestrian, etc.), but usually these processes can run only in one system, with strong constraints. Efficient algorithms are therefore needed to recognize interesting situations with sufficiently low cost.

Animals are ready to recognize and fast avoid many kinds of situations, like approaching objects. Research results are available how this kind of detection works [1][2][3] and how to use as last minute warning system[4][5]. The architectural development of processors has led to the possibility of efficient biological model implementations.

II. LOOMING OBJECT DETECTION

Botond Roska and his research group found the Pvlab-5 ganglion cell in mouse retina. This type of cell reacts only to looming motion [1], when dendrite of a cell is connected to many cones in the retina. These collections are the receptive field, where the ganglion cell is capable of sensing the signals of an approaching object.

The basis of this model is to observe the silhouette change on the receptive field: when the looming motion occurs, the silhouette grows which causes a positive activity on the edge of the silhouette on the excitatory channels, while the inhibition is much lower. In a lateral motion, the excitation and inhibition is equal, which does not cause any reaction.

A. Improving on previous model

Referencing [6] discusses the basic model. The Pvlab-5 in mammalian retina only recognizes the 10^{o} of the whole space, so it is enough to recognize a small part of the space. To cover the whole retina needs many overlapped Pvlab-5 cells to detect where the looming motion takes place. The original model has got two strict limitations:

1. Only reacts to OFF direction changes (bright to dark changes)
2. Lateral motion is detected as sharp looming motion for a moment when object step in the receptive field

The OFF type reaction dependency causes a strict limitation of usage. The main question is that, how the OFF type reaction can be avoided. The second problem is solvable with some time filters.

III. ELIMINATE LIMITATIONS

The key initial step is how to eliminate the OFF-type dependency of the original algorithm, which helps to make a context-free algorithm. It comes from the mathematical model that the offsets and weights of inhibition and excitation parameters are changed equally, when the inhibit channel is working as an ON-type looming object detector. It follows that the absolute sum of two channel results is a change map. This change-map can also be produced as a simple frame difference, but the frame differencing cannot be as parameterized as the modified looming algorithm (for example: noise tolerance).

The resulting change-maps can be used as a looming model input, because it only contains the various layers, so the looming algorithm only needs to look to the change direction,

which is more reliable. We can formalize the following equations. Equation (1) shows the calculation of changes with simple frame differencing, while eq. (2.1)-(2.4) show an alternative frame differencing solution which is based on equations of the original looming model [6].

(notations: i: pixel in a picture (pic), timestamp in the lower index; w: weight vector of time window, s: time window length, t: timestamp, do/lo: dark/light offset of pixel channels, dw/lw: dark/light weight of a channel)

(1):

$$pic_{diff}(t) = |pic(t) - pic(t-1)|$$

(2.1):

$$pix_t = \sum_{j=0}^{s-1} i_{t-j}\, w_j$$

(2.2):

$$g(pix, do, dw) = \begin{pmatrix} (pix_t + do)dw, if\ (pix_t + do) > 0 \\ 0 \end{pmatrix}$$

(2.3):

$$h(pix, io, iw) = \begin{pmatrix} (-pix_t + lo)lw, if\ (-pix_t + lo) > 0 \\ 0 \end{pmatrix}$$

(2.4):

$$pix_{diff}(t) = g(pix, do, dw) + h(pix, lo, lw)$$

The absolute value in eq. (1) detects the absolute changes, while steps of eq. (2) show the sophisticated way to detect changes. Eq. (2.1) shows the weighted change of one pixel in the last *s* frame. Eq. (2.2) is capable of setting up the noise tolerance of dark to light changes, while eq. (2.3) sets-up the light to dark. This is useful when sensors of cameras do not handle the direction of image changes in the two ways and the offset settings can handle the noise of image acquiring. Resulting eq. (2.4) gives absolute difference of changes for each change direction.

The absolute difference of last frames can be used in eq. (3) as image input. It contains only the changes with their values. As the result of eq. (2) the modified looming algorithm works as a silhouette change detector. All of the interesting movements like looming motion are being interpreted as OFF-type changes and they can be an input of the original looming detector algorithm in eq. (3.1)-(3.5).

(notations: i: pixel in a picture (pic), timestamp in the lower index; w: weight vector of time window, s: time window length, t: timestamp, eo/ io: excitatory and inhibitory offset of a channel, ew/iw: excitatory and inhibitory weights of a channel)

(3.1):

$$exc_t = \sum_{j=0}^{s-1} pic_{diff_{t-j}}\, w_j \quad inh_t = -\sum_{j=0}^{s-1} pic_{diff_{t-j}}\, w_j$$

(3.2):

$$g(exc, eo, ew) = \begin{pmatrix} (exc_t + eo)ew, if\ (exc_t + eo) > 0 \\ 0 \end{pmatrix}$$

(3.3):

$$h(inh, io, iw) = \begin{pmatrix} (inh_t + io)iw, if\ (inh_t + io) > 0 \\ 0 \end{pmatrix}$$

(3.4):

$$f(exc, eo, ew, inh, io, iw) = g(exc, eo, ew) - h(inh, io, iw)$$

(3.5):

$$res = \sum_{all\ pixel} f(exc, eo, ew, inh, io, iw)$$

As we described in [6] these equations can detect looming motion in a receptive field. The *res* is the value of the change. The problem with this result is that, we cannot identify immediately the nature of the motion. The type identification needs at least two consecutive results. The relationship between the two results can describe it as eq. 4 defines. The final decision is an answer, where res(t) is the result of the 3.5 in time.

(4):

$$decision = \begin{Bmatrix} \text{"shrinking"}, if\ res(t) < res(t-1) \\ \text{"lateral/nothing"}, if\ res(t) = res(t-1) \\ \text{"looming"}, if\ res(t) > res(t-1) \end{Bmatrix}$$

IV. Determination of looming direction

The retina has got many Pvlab-5 cells. These cells lie overlapped on the whole retina and these cells have different responses. We have described in [6] what happens, when the looming motion is activated only one cell, but we have to investigate what happens when the looming and lateral motions exit the cell and enter another one.

A. Looming case

When the projected silhouette of looming motion exits the receptive field it causes the response of the ganglion cell to fall down suddenly. This falling down is the same as when the looming stops. The only difference is the response of some neighboring cells. When the looming motion stops, all of the responses of the neighboring receptive field fall down. When the silhouette change of looming motion exits from a receptive field, the neighbor fields' response are change. When the silhouette change of the looming motion enters to the receptive field, it starts the response.

B. Shrinking case

When changes of shrinking motion entering into a receptive field they cause response, same as at shrink start. When changes of shrinking motion exits receptive field, it causes decay answer.

C. Lateral case

We know that the lateral motion does not cause any effect in a cell, except two situations:

When the projected silhouette changes of an object motion enter the cell, it activates only the excitatory channels, while exiting activates only the inhibitory channels of the original model. When the whole object is on the cell it is being activated both channels, which equalizes the strength of the answer.

Figure 1 shows, what happens exactly on the overlapped receptive fields, when a lateral motion occurs (movement is right to left). The black disk stays in first and starts to make discrete steps to the left. The silhouette of the disk activates the excitatory pathways.

2nd time is shown that the second receptive field gets excitatory stimulations only, which causes positive "+" value of response. The fourth receptive field gets only inhibitory stimulations, it gives negative "-" responses. The third receptive field gets similar excitatory and inhibitory stimulus, so the value will be "0".

A potential value can be specified from inhibitory and excitatory pathway. The direction can be calculated from the potential value difference between activations, and speed can be also calculated, when the method knows the size or distance of object. The potential value difference can be written up from any receptive-fields, but we can use the locality and spreading of changes if we choose the neighboring cells for differencing.

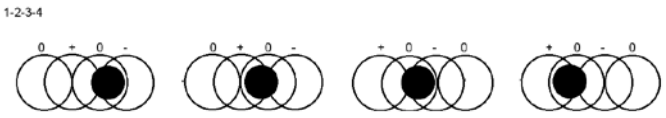


Figure 1. Lateral motion: Inhibitory and excitatory channels work. The sum of two channels make different activation patterns on receptive field in the different moments.

The Figure 2 shows the overlapped receptive-fields and the local difference. The linear combination of the potential differences gives the vector which shows a theoretical direction of object motion. When receptive field is small, it shows local changes, like optical flow too.

The relationships between receptive fields are shown in Figure 3. Analyzing the difference between opposite receptive fields and making an average of it shows a theoretical direction of motion. The eq. 5. is shown the equation.

When looming motion takes place, the vectors point to the center of the looming object, while in the case of a shrinking motion they points outwards from the center of the object. When a looming motions occur with lateral motions, they can also be calculated from the sum of two situations.

(5):

$$\begin{pmatrix} x_{size} \\ y_{size} \end{pmatrix}_{center} = \begin{pmatrix} (rR - rL) + \sqrt{(bR - bL)} + \sqrt{(gL - gL)} \\ \sqrt{(bR - bL)} + \sqrt{(gL - gL)} \end{pmatrix}$$

The variables from Figure 3: bL:blueLeft, bR: blueRight, rL: redLeft, rR: redRight, gL: greenLeft, gR: greenRight, center is gray.

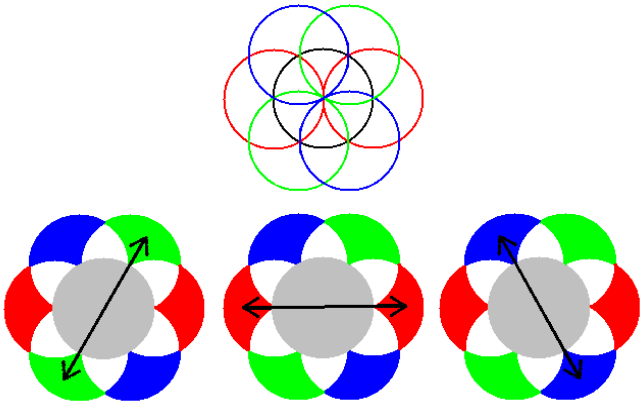


Figure 2. Comparable overlapped receptive fields. Arrows show the corresponding fields.

V. RESULTS

The result shows that the new method is capable of detecting looming motion of a light object with dark background too. The environment was the inverted grayscale picture from [6]. The concrete results in context free environment show in the following.

Validating the behavior of the model in real situations is shows that, the model works. The receptive fields were changed to squares for faster calculation in MATLAB, because the input picture was much larger than Eye-RIS picture 1280×1024. The square receptive field based model works as good as the circle based model in MATLAB, the results were compared. The new setup used eight plus one neighbor receptive fields.

As a test in real environment, we made recordings by car. The first results were disappointing, because every part of the picture gave a signal, and motion direction. The reason was simple, the car chassis bows on the move. This result shows that the algorithm is capable of detecting optical flows in real time with this algorithm.

Next, the algorithm got a simple image stabilization algorithm. One interesting result of the model is that it does not react to a faraway optical-flow, so it is possibly good for Region Of Interest detector algorithm. This algorithm is capable of detecting cars farther away 50m (moving cars, both of them operate 70km/h), which comparable with the most of the RADARs on the market. The sensitivity of detection depends on the background and the foreground objects, but a built-in gain can help. The method detected the local movement direction in many situations, so it is capable to annotate the object motion direction.

VI. SUMMARY

This article shows the modification of the looming object detector model, which is capable of eliminating the type dependency of the original model. The modification does not cause any processing time increase, and works properly in the calibrated environment.

The new algorithm was tested in real environment too, which shows encouraging results in following. The new method is capable to be implemented in real time environment, and in sensor processor systems, but it needs robust image stabilization.

ACKNOWLEDGMENT

The authors express their thanks to grants TÁMOP-4.2.1.B-11/2/KRM-2011-0002 and TÁMOP-4.2.2/B-10/1-2010-0014..

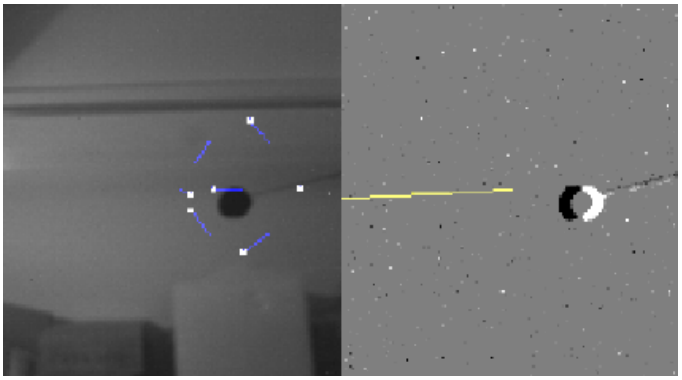


Figure 3. The local motion vectors with blue. The yellow arrow shows the average of vectors in clearly lateral motion (10cm/s)

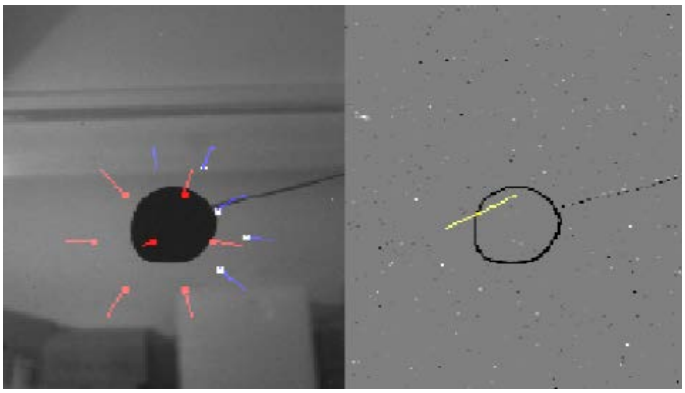


Figure 4. The local motion vectors with red. The yellow arrow shows the average of vectors, as local motion direction. (looming with 10cm/s and a very small lateral motion)

REFERENCES

[1] T.A. Münch, R. Azeredo da Silveira, S. Siegert, T.J. Viney, G.B. Awatramani, B. Roska,"Approach sensitivity in the retina processed by a multifunctional neural circuit", Nature Neuroscience, 12(10), pp. 1308-1316, 2009.

[2] O'Shea, M. Rowell, C.H.F. Williams, J.L.D, "The anatomy of a locust visual interneurone: the descending contralateral movement detector" J. exp. Biol. 60, 1-12. 1974.

[3] Wang, Y., Frost, B. J., Time to collision is signalled by neurons in the nucleus rotundus of pigeons,Nature, 1992, pp: 356, 236‑238

[4] Linan-Cembrano, G., Carranza, L., Rind, C, Zarandy, A., Soininen, M., Rodriguez-Vazquez, A, "Insect-Vision Inspired Colision Warning Vision Processor for Automotive", IEEE Circuits and Systems Magazine, Volume: 8, Issue: 2 On page(s): 6-24 2008

[5] Yue SG, Rind FC, Keil MS, Cuadri J, Stafford R., „A bio-inspired visual collision detection mechanism for cars: Optimisation of a model of a locust neuron to a novel environment" Neurocomputing Volume: 69 Issue: 13-15, AUG 2006, pp 1591-1598

[6] Á. Zarándy, T. Fülöp, "Implementation and validation of a looming object detector model derived from mammalian retinal circuit", Focal-plane sensor-processor chips, Á. Zarándy ed, Springer Science 2011, pp. 245-259.

Compressive Sensing in Digital In-line Holography

Péter Lakatos
(Supervisors: Dr. Szabolcs Tőkés and Dr. Ákos Zarándy)
lakatos.peter@itk.ppke.hu

***Abstract*— Compressive sensing (aka compressed sensing or sampling) is a novel signal reconstruction or sampling model, which enables significantly less measurement than reconstructed data for a class of signals. It also offers algorithmic solutions via the linear inverse problem. We use these models and algorithms to solve the reconstruction problem of digital in-line hologram of sparse or otherwise redundant images.**

***Keywords* - compressed sensing; compressive sensing; digital holography; in-line holography; holographic tomography; sparsity; inverse problem; linear inverse problem;**

I. INTRODUCTION

Compressive sensing is a novel signal reconstruction or sensing model which enables significantly less measurement than reconstructed data. Not in general but for some wide class of signals. It applies the fact that most of the signals are sparse or redundant in some way. For example most of the images can be represented in some wavelet basis with only a few significant coefficients.

Compressive sensing grew up from questions raised up by medical imaging techniques (like MRI [1]) and after some theoretical groundwork [2-4] it produces a lot of practical (mainly in different imaging techniques) or simply fun (single pixel camera, [5]) results.

In the second section we introduce compressive sensing with some theoretical foundations and the linear inverse problem which is essential in the practical usage of it. In the third section we take a fast look to digital in-line holography. In the fourth section we show how we can adopt the philosophy and practice of compressive sensing to holography.

II. COMPRESSIVE SENSING

There is lot of different aspect of compressive sensing. It can be introduce from the direction of signal sampling theorems, denoising functions [3] or random matrixes [2]. Here we will use a linear algebraic approach [15].

A. Linear algebra aproach of compressive sensing

In information theory and its related subjects almost every measuring or sensing process can be write in the form of a linear equation system:

$$g = \Phi f \tag{1}$$

where f is the subject of the sensing, Φ represents the sensing process and g is the outcome of the sensing. Here f and g are real (or complex) valued vectors with size N and M, respectively, and ϕ is an N by M real (or complex) valued matrix. M is the number of measures. We know ϕ and g and we are interested in f. If a measuring is not in this form, discretization, linear approximation or some other processes (tricks) usually can help.

Such a linear system is easily solvable if $M \geq N$, i.e. we have at least as many equations as variables. On the other hand, if $M<N$, there is impossible to solve the equation, because there is infinitely many solutions. Unless we have some additional information or constraints on the variables (f).

Compressive sensing is dealing with the case of $M<N$ when some redundancy or sparsity on the subject of the sensing (f) is assumed or a priori known.

In the sparse case we can formalize the problem as

$$\hat{f} = \underset{f}{\operatorname{argmin}} \|f\|_0 \quad \text{subject to} \quad g = \Phi f \tag{2}$$

where $\|f\|_0 = |\{i: f_i \neq 0\}|$ is the number of nonzero element of f. $\|f\|_0$ is also known as the l_0-norm of g (but in fact it is not a norm, because it is not scalable). So we search for the sparsest solution.

Redundancy in f means there is some basis (Ψ) in what f is sparse. Let α be the representation of f in this basis: $f = \Psi \alpha$. In this case we can formalize the problem as

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \|\alpha\|_0 \quad \text{subject to} \quad g = \Phi \Psi \alpha \tag{3}$$

and then take $\hat{f} = \Psi \hat{\alpha}$.

The problem with the above mentioned l_0-norm is that it is numerically hard to handle and extremely sensible to noise. Compressive sensing suggests that instead of the l_0-norm, we can recover f or α by using of the l_1-norm $\|\alpha\|_1 = \sum_{i=1}^{N} |\alpha_i|$. In this case we can formalize the problem as

$$\hat{\alpha}_1 = \underset{\alpha}{\operatorname{argmin}} \|\alpha\|_1 \quad \text{subject to} \quad g = \Phi \Psi \alpha. \tag{4}$$

Compressive sensing guarantees that the solution of problem (2) and problem (3) are the same (i. e. $\hat{\alpha}_1 = \hat{\alpha}$), if there is incoherence (dissimilarity) between the sensing and the sparsifying matrix, and the number of measures is not too small:

$$M \geq C \cdot \mu^2(\Phi, \Psi) \cdot K \cdot \log_{10} N \tag{5}$$

where C is a small positive constant, K is the maximal number of nonzero elements of $\hat{\alpha}$ and μ is the above mentioned similarity of Φ and Ψ, called mutual coherence:

$$\mu(\Phi, \Psi) = \sqrt{N} \cdot \max_{i,j} |\langle \Phi_i, \Psi_j \rangle| \tag{6}$$

where Φ_i and Ψ_j denote the i-th and j-th column vector of Φ and Ψ, respectively.

Notice that if $\mu(\Phi, \Psi)$ and K are not too big (and in a lot of theoretically or practically important cases they are not), then M is enough to be much smaller than N, unlike in the well known Nyquist-Shannon sampling theorem or in the linear algebraic considerations in the beginning of this section, where M≥N is required. It is not a contradiction since the redundancy or sparsity constraints.

B. The linear inverse problem

Compressive sensing states that we can solve problem (2) by solving problem (3). They can be reformulate as

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}}(\|g - \Phi\,\Psi\,\alpha\|_2^2 + \|\alpha\|_0) \tag{7}$$

$$\hat{\alpha}_1 = \underset{\alpha}{\operatorname{argmin}}(\|g - \Phi\,\Psi\,\alpha\|_2^2 + \|\alpha\|_1) \tag{8}$$

Both of them can be considered as a special case of the linear inverse problem:

$$\hat{x} = \underset{x}{\operatorname{argmin}}(\|y - K\,x\|_2^2 + \tau\,\varrho(x)) \tag{9}$$

where x and y are vectors, K is a matrix with proper size, τ is a nonnegative constant called the regularization parameter and ϱ is a $\mathbb{R}^N \to [0, \infty[$ function called the regularizer function. Commonly used regularizer functions are for example:

- the l_0-norm
- the l_1-norm
- the Euclidean or l_2-norm $\|x\|_2 = \left(\sum_{i=1}^{N} |x_i|^2\right)^{\frac{1}{2}}$
- the general l_p-norm $\|x\|_p = \left(\sum_{i=1}^{N} |x_i|^p\right)^{\frac{1}{p}}$
- if x represents an image the total variation norm $\|x\|_{TV}$ which we will introduce in the fourth section

One of the advantages of this reformulation that if we choose λ carefully, the effects of noise can be reduced [6].

For the solution of the linear inverse problem a lot of algorithms were developed recently thanks to the general interest for the compressive sensing. The best of them are the SpaRSA (sparse reconstruction by separable approximation, [7]), the IST (iterative shrinkage/thresholding [8]) and the TwIST (two-step IST, [9]). All of them are iterative algorithms usable effectively only with a group of regularizer functions, but with an important group.

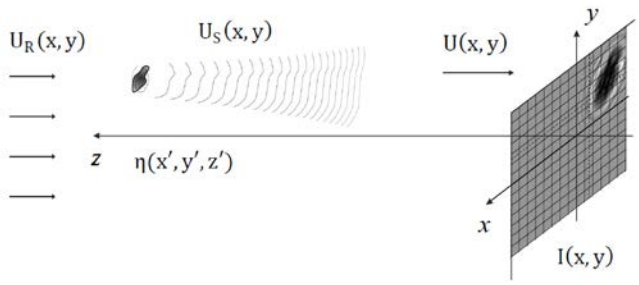


Figure 1: In-line hologram model

III. DIGITAL IN-LINE HOLOGRAPHY

Holography is an imaging technique based on the capture of coherent fields scattered from objects. It was introduced by Gabor in 1947 [10] and it became common after the development of the laser by Leith and Upatnieks in 1962. Gabor earned the Nobel Prize in Physics in 1971.

In holography [16] there is always a reference beam with complex amplitude $U_R(x,y)$ and an object beam scattered from the object $U_S(x,y)$, and we capture the interference $U(x,y) = U_R(x,y) + U_S(x,y)$ of them in a photographic plate or in digital photometric sensor. Both of these devices can capture the intensity of the field:

$$\begin{aligned} I(x,y) &= |U(x,y)|^2 = U(x,y) \cdot U^*(x,y) = \\ &= |U_R(x,y)|^2 + |U_S(x,y)|^2 + \\ &+ U_R^*(x,y) \cdot U_S(x,y) + U_R(x,y) \cdot U_S^*(x,y) \end{aligned} \tag{10}$$

If after the capture of I we light the photographic plate with the reference beam (or in the digital case simulate it), we get

$$\begin{aligned} U_R(x,y)I(x,y) &= U_R(x,y)(|U_R(x,y)|^2 + |U_S(x,y)|^2) + \\ &+ |U_R(x,y)|^2 \cdot U_S(x,y) + U_R(x,y) \cdot U_S^*(x,y) \end{aligned} \tag{11}$$

The first term is the reference beam with slightly modified amplitude. The second is the object beam, which forms a real image of the object. Finally the third term is called the "conjugate object beam" which forms an artifact called the "twin image".

There are plenty of holographic processes, but we can easily group them by the route of the reference beam compared to the scattered beam. In the off-axis holography the two beams are not parallel when they arrive to the sensor. In the on-axis holography the two beams are parallel, but this is achieved by a beam splitter. Finally in the in-line holography the two beams are also parallel and the reference beam arrives to the sensor among the scattering objects. The last one works only if there are a few and little objects in a transparent volume. It also suffers from the effect of the twin image, but it is easy and cheap to realize it.

In in-line holography we usually use a plane waves with high amplitude as reference beam, so it can be considered as constant $U_R(x,y) = U_R$ with high intensity compare to the scattered beam:

$$I(x,y) = |U_R|^2 + U_R^* U_S(x,y) + U_R U_S^*(x,y) \quad (12)$$

$$I(x,y) = |U_R|^2 + 2\,Re\left(U_R^* U_S(x,y)\right) \quad (13)$$

With the Born approximation the scattered beam can be considered as

$$U_S(x,y) = \iiint \eta(x',y',z') \cdot h(x-x', y-y', z-z')dx'dy'dz' \quad (14)$$

where η is the scattering density of the measured volume, z is the distance of the sensor and h is the point spread function (aka impulse response function). After discretization and consider the finite aperture we get

$$U_{S,n_x,n_y} = U_S\left(n_x \cdot \Delta p, n_y \cdot \Delta p\right) = \sum_{m_x}\sum_{m_y}\sum_{m_z} \eta\left(m_x \cdot \Delta y, m_y \cdot \Delta x, m_z \cdot \Delta z\right) \cdot h(m_x \cdot \Delta x - n_x \cdot \Delta p, m_y \cdot \Delta y - n_y \cdot \Delta p, z - m_z \cdot \Delta z) \quad (15)$$

where Δy, Δx and Δz are the size of a voxel (3D volume pixel) and Δp is the size of a pixel in the sensor [17]. We can rearrange (11) in the form of

$$U_S = H \cdot \eta \quad (16)$$

with the vectors U_S and η and the matrix H. With this we get

$$I = |U_R|^2 + 2\,Re(U_R^* \cdot H \cdot \eta) \quad (16)$$

which is, if H and U_R are real valued,

$$d = c \cdot 1 + H \cdot \eta \quad (17)$$

where d is the measured intensity data, the 1 is a vector containing only ones and c is a constant.

IV. Compressive Holography

In the previous section we saw that holographic imaging can be representing as a linear system. This observation clear the way for the compressive sensing. In this section we introduce two case studies which both realize a compressive sensing based reconstruction method of digital in-line holography. One of them is a classical 2D reconstruction, while the other is a 3D tomographic one, and they use different constraints.

A. Hologram reconstruction with sparsity constraints

In [11] Denis et al. used a 2D reconstruction, so it reconstructs objects from a fixed z depth in one run. However they ran their algorithm for a series of depths, one after other, just like the classical holographic reconstruction methods.

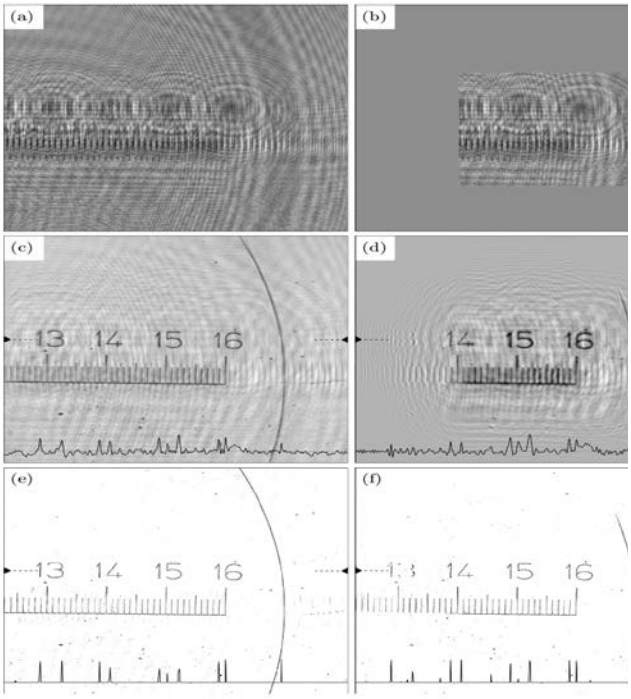


Figure 2: (a) hologram, (b) hologram with missing pixels on the side, (c-d) classical reconstructions, (e-f) compressed sensing reconstruction

Since the 2D reconstruction they used (15) without the last summation. They chose the Fresnel approximation as the point spread function:

$$h_z(x,y) = h(x,y,z) = \frac{1}{\lambda z}\sin\frac{\pi(x^2+y^2)}{\lambda z} \quad (19)$$

They assumed that the objects are located sparsely in the examined volume, so their regularizer function was the l_1-norm:

$$\hat{\eta}_1 = \underset{\eta,c}{\operatorname{argmin}}(\|c \cdot 1 + H \cdot \eta - d\|_2^2 + \tau\|\eta\|_1) \quad (20)$$

or after some modification of H and d:

$$\hat{\eta}_1 = \underset{\eta}{\operatorname{argmin}}(\|H \cdot \eta - d\|_2^2 + \tau\|\eta\|_1) \quad (21)$$

They used a modified version of the IST [9] to solve it with changing τ in every step.

They achieved axial resolution of 60 μm, and their algorithm worked well with objects outside of the image. Results showed in Fig. 2.

B. Hologram reconstruction with smoothness constraints

In [12] Brady et al. used the approach of the angular spectrum method [13] to reformulate the problem in the form of

$$d = H \cdot \eta \quad (22)$$

with $H = G_{-1}QG_1$. Here $G = \mathrm{bd}[F_{2D}, F_{2D}, \dots, F_{2D}]$, where bd stands for blockdiagonal and F_{2D} represents the 2D discrete Fourier transformation. G_{-1} means the same, but with the inverse DFT. Finally $Q = \mathrm{bd}[H_1, H_2, \dots, H_{M_z}]$, with

$$H_{l,n_x,n_y} = e^{ikl\Delta z} e^{il\Delta z\sqrt{k^2 - n_x{}^2\Delta k_x{}^2 - n_y{}^2\Delta k_y{}^2}} \qquad (23)$$

with $k = \frac{2\pi}{\lambda}$ wavenumber.

They handled the redundancy of the images not by finding a proper basis but by the minimalization of the total variation:

$$\hat{\eta}_{TV} = \underset{\eta}{\operatorname{argmin}} \|\eta\|_{TV} \quad \text{subject to} \quad d = H \cdot \eta \qquad (24)$$

with $\|\eta\|_{TV} = \sum_{m_x} \sum_{m_y} \sum_{m_z} \left| \nabla (\eta_{m_z})_{m_x, m_y} \right|$.

They used the TwIST algorithm [9] to solve the problem numerically. See results in Fig.3.

They give a wider overview of the subject in [14].

V. Conclusion and Future Plans

We presented the compressive sensing and its application for digital in-line holography, both in a theoretic way and via case studies. Our plan is to adapt and optimize these algorithms to the water quality measuring digital holographic microscopy [18]. Further plan is to extend these models to use multiple nonparallel detectors to reduce axial resolution.

References

[1] Michael Lustig, David Donoho, and John M. Pauly, Sparse MRI: The application of compressed sensing for rapid MR imaging, Magnetic Resonance in Medicine, 58(6), pp. 1182 - 1195, 2007

[2] David Donoho, Compressed sensing, IEEE Trans. on Information Theory, 52(4), pp. 1289 - 1306, 2006

[3] E. Candès, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Communications on Pure and Applied Mathematics, 59(8), pp. 1207-1223, 2006

[4] Emmanuel Candès and Terence Tao, Near optimal signal recovery from random projections: Universal encoding strategies?, IEEE Trans. on Information Theory, 52(12), pp. 5406 - 5425, 2006

[5] Abdorreza Heidari, D. Saeedkia, A 2D Camera Design with a Single-pixel Detector, Int. Conf. on Infrared, Millimeter and Terahertz Waves, Busan, South Korea, 2009

[6] R. Tibshirani, Regression shrinkage and selection via LASSO, Journal Royal Statistical Society B, vol 58, pp 267-288, 1996

[7] Stephen J. Wright, Robert D. Nowak, Mário A. T. Figueiredo, Sparse reconstruction by separable approximation, Journal IEEE Transactions on Signal Processing, Volume 57 Issue 7, Pages 2479-2493, 2009

[8] José M. Bioucas-Dias , Mário A. T. Figueiredo, Two-step algorithms for linear inverse problems with non-quadratic regularization, IEEE International Conference on Image Processing ICIP', 2007

[9] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Communications on Pure and Applied Mathematics, V 57, I 11, pp 1413-1457, 2004

[10] D. Gabor, "A new microscopic principle", Nature, 161, pp 777-778, 1948

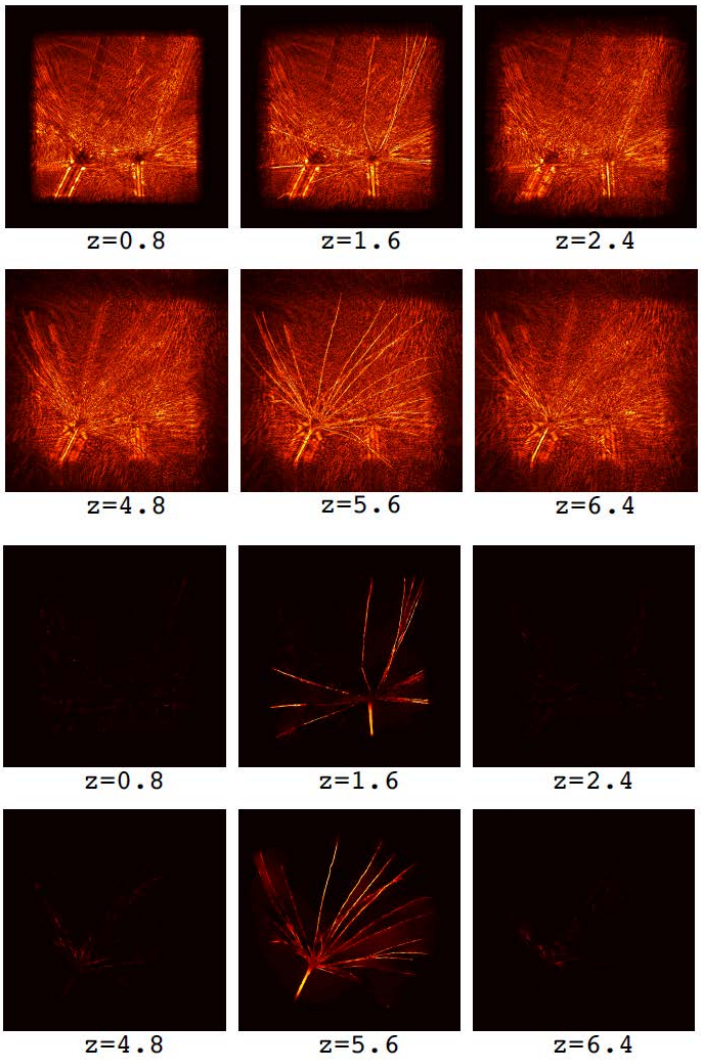


Figure 3: Classical and sparse reconstruction from a hologram of two dandelions at distance 1.5 and 5.5

[11] Loïc Denis, Dirk Lorenz, Eric Thiébaut, Corinne Fournier, Dennis Trede, "Inline hologram reconstruction with sparsity constraints.", Opt.Lett. 34, pp 3475-3477, 2009

[12] David J. Brady, Kerkil Choi, Daniel L. Marks, Ryoichi Horisaki, and Sehoon Lim, Compressive Holography, Optics Express, Vol. 17, Issue 15, pp. 13040-13049, 2009

[13] Kyoji Matsushima, Tomoyoshi Shimobaba, Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields, Opt. Express 17, 19662-19673, 2009

[14] Sehoon Lim, Daniel L. Marks, and David J. Brady, Sampling and processing for compressive holography, Applied Optics, Vol. 50, Issue 34, pp. H75-H86, 2011

[15] Yair Rivenson, Adrian Stern and Bahram Javidi, Compressive Fresnel Holography, IEEE/OSA Display Technology, Journal of , vol.6, no.10, pp.506-509, 2010

[16] J. W. Goodman, Introduction to Fourier optics, 3rd Ed., Roberts and Company Publishers, 2005

[17] Corinne Fournier, Loic Denis, Eric Thiebaut, Thierry Fournel, Mozhdeh Seifi, Inverse Problem Approaches for digital hologram reconstruction, Proc. SPIE 8043, 80430S, 2011

[18] Z. Göröcs, L. Orzó, M. Kiss, V. Tóth, Sz. Tőkés, In-line color digital holographic microscope for water quality measurements, Proceedings of the SPIE, Volume 7376, pp. 737614-737614-10, 2010

Self-referenced Digital Holographic Microscopy

Márton Zsolt Kiss
(Supervisor: Szabolcs Tőkés)
kisma1@digitus.itk.ppke.hu

***Abstract*—By developing a self referenced digital holographic microscope (Selfref-DHM) it becomes possible to record holograms and numerically reconstruct volumetric images of low coherence fluorescent objects such as (auto)fluorescent biological samples (e.g. algae). Our goal was to develop and construct a simple, compact portable device. In contrast to the common holographic approaches where there is a conventional reference beam, a reference beam should be produced together with the object beam from the same fluorescent source via imaging it by two separate optical paths (with near zero path length differences) to get interferences fringes. These interference forms separate holograms of all the point sources. The waves coming from the separate sources are mutually incoherent but have an inherent short coherence length. Initially we have tested the Selfref-DHM setup with test objects illuminated by LED light source that has similar spectral bandwidth as the fluorescence sources like chlorophyll. Digital reconstructions of the measured holograms need considerable processing. To accelerate the hologram processing a parallel implementation seems essential. Using GPU-s we were able to enhance the algorithm speed considerably, without the loss of the reconstruction accuracy.**

***Keywords*-digital holographic microscopy, fluorescent microscopy, self-referenced holography**

I. INTRODUCTION

Digital holographic microscopies have recently become fields of much interest because of the ability to numerically reconstruct holograms, which are captured by digital detectors, creating the high quality images with high lateral and sometimes extremely high axial resolutions. This technique makes it possible for example to produce lensless microscopes [1] or also to alter conventional microscopes to measure volumetric sample with one exposure without scanning [2]. At the field of material science and biology fluorescence microscopic applications were needed for such advantages as selectivity and sensitivity. Because of the need of fluorescence and volume detection many 3D fluorescence microscopic applications have already been used based on Fluorescent Coherence Tomography [3], Lens Free Fluorescent Imaging [4], Fresnel Incoherent Correlation Holography [5] and Three-Dimensional Holographic Microscopy [6]. In this article a Self-referenced Digital Holographic Microscope is presented. The final purpose of this setup is to detect freely floating, alive microbiological organisms and to create their fluorescent images within a volume. This solution is the combination of the FINCH method [5] and the modified Hariharan-Sen method [7] which means that after the objective the reference and object bands are created on an asymmetric triangular optical path, avoiding the need of any spatial light modulator. The main advantage of this setup is that it uses only inexpensive passive optical elements. Design concepts, the applied test targets and the first measurement results are presented in this article.

II. THE OPTICAL SETUP

A. *Self-referenced holographic setup*

The key idea of any digital holographic system is to create an interference pattern between two coherent beams, traditionally a reference beam and an information carrying object beam, and to capture this interference pattern with an image sensor. Reconstruction of these captured interference patterns are done numerically. The maximum optical path difference of an interferometer should be smaller than the coherence length of the used light in order to get the interference pattern. At self-referenced holographic setups the light source of the reference and the corresponding target beam is the same; and all the particles of the observed object itself that emit or reflect light are light sources. The self-referenced holographic setups divide the light coming from every object point into two parts to modulate them separately: generating the reference beam and focusing the target beam. The corresponding beams are reunified to create the hologram. Because of using fluorescent light, whose coherence length is usually only a few micrometers, the interferometer should have a nearly zero path length difference between the corresponding beams.

B. *Optical parameters*

The light originating from the sample is collected with a triplet lens with a focal length of 40.3 mm that makes an intermediate image. Because of this long focal length small object distance change makes short image distance change as well. The entrance of the asymmetric triangular is a pellicle beam splitter that directs 67% of the incoming light clockwise to produce the reference beam and 33% anticlockwise to produce the target beam. The triangular path contains a lens with focal length 150 mm moved from the symmetrical center plane by 10 mm. The reference and the target beams reach this different sides of this lens after taking a different path, and traveled different distances from the entrance of the asymmetric triangular path. As a result of this the lens creates two images from the intermediate image with different magnifications and image distances, so the radii of the wave-front curvature of the reference and target beams will be different. The pellicle beam splitter reunites these beams after they take the same length path in different directions and creates the interference pattern that is captured by a monochrome Lucam detector. The self-referenced holographic optical setup can be seen in Fig. 1.

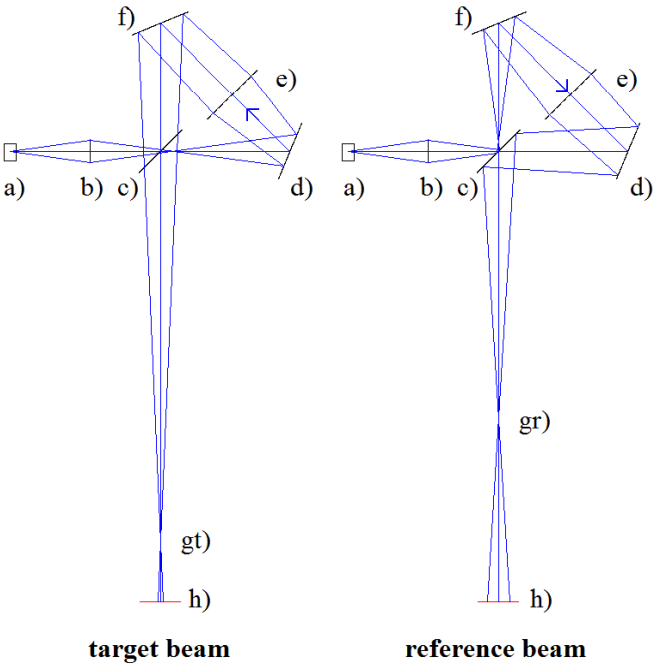


Figure 1. The path of a target beam and a corresponding reference beam from one object point, where a) is one illuminating point of the object, b) triplet, c) beamsplitter pellicle, wich is the entrance and the exit of the triangular path, d) mirror, e) lens (f=150), f) mirror, gt) image by object beam, gr) image by reference beam, h) detector that saves the holograms

III. THE SAMPLE

There are many fluorescent objects whose measurement with fluorescent imaging setups, e.g. fluorescently marked or autofluorescens living cell samples, is useful. Our aim is to detect and make images of algae. These usually contain chlorophyll-A or chlorophyll-B. Cyanobacteria, which can be toxic, contain Chlorophyll-A. Below in Fig 2. the emission curve of a cyanobacteria can be seen excited by 405 nm light. A maximum point of the curve is at wavelength 656 nm. The bandwidth is about 30 nm. Equation (1) shows the calculation of the coherent length (l) from the middle wavelength (λ_0) and the bandwidth ($\Delta\lambda$).

$$l=\lambda_0\text{^}2/\Delta\lambda \qquad (1)$$

The calculated coherence length of the fluorescently emitted light of the Cyanobacteria is about 14 [μm]. This coherence length can be augmented with a narrow laser line filter. This way we can increase the contrast of the interference pattern, but the available light is reduced.

We applied a test target in our setup, which is a black plate with six holes of 30 □m ◊
holes is 0.36 mm. This is back-lighted with a LED light source that emits light with the approximately the same coherence length as the chlorophyll-A, but its color was amber. This target tries to model the fluorescent objects.

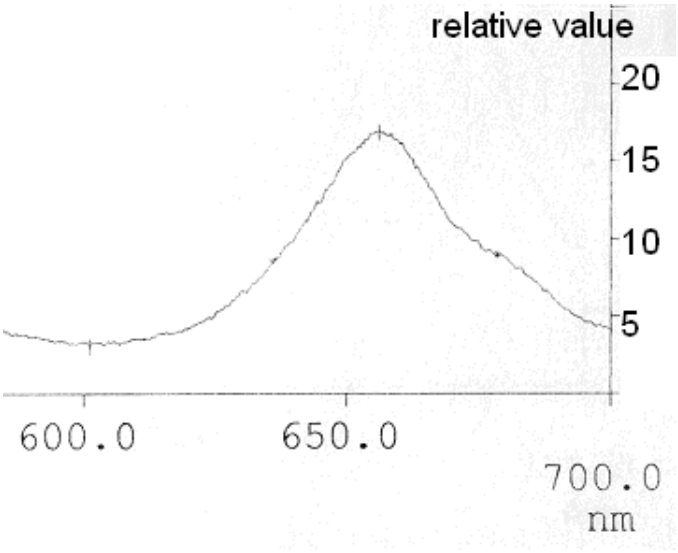


Figure 2. Emitted light of a Cyanobacteria excited with 405[nm]

IV. RESULTS

The self-referenced holographic setup was built and tested with the target that was placed in different working distances. The depth of the measured area was 1mm. The distances between of the reference (Fig. 1. gr) and the target images (Fig. 1. gt) changed between 112 and 117 [mm]. The magnifications of the target changed between 3 and 3.2 and the magnifications of the reference changed between 2.2 and 2.3. As in Fig. 3. can be seen, this self-referenced digital holographic setup is able to create self-referenced holograms with non-coherent light. This finding corroborate that there is a possibility to measure auto fluorescent live cells with this kind of self-referenced holographic setup. To implement the measurements on fluorescent samples a large f-number lens should be applied because the auto fluorescent light provides much less light than LED illumination.

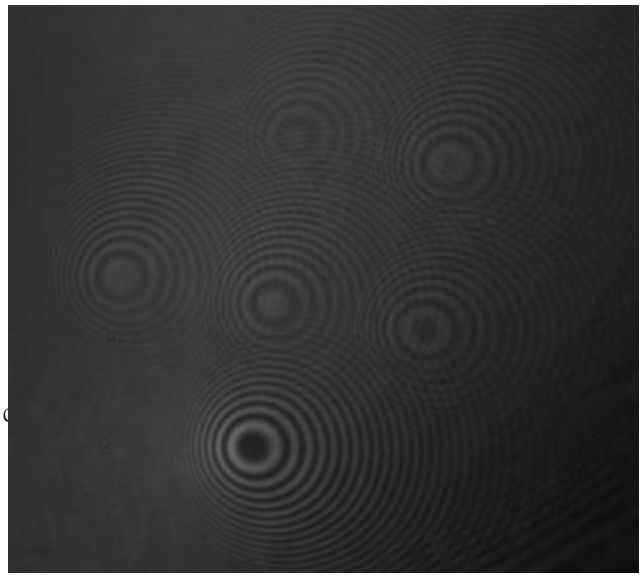


Figure 3. The self-referenced hologram of 6 pinholes back-lighted with LED light colored amber

REFERENCES

[1] O. Mudanyali, D. Tseng, C. Oh, S.O. Isikman, I. Sencan, W. Bishara, C. Oztoprak, S. Seo, B. Khademhosseini, and A. Ozcan. "Compact, light-weight and cost-effective microscope based on lensless incoherent holography for telemedicine applications", *Lab Chip*, 10(11):1417–1428, 2010.

[2] Zoltán Göröcs, László Orzó, Márton Kiss, Veronika Tóth, and Szabolcs Tőkés. "In-line color digital holographic microscope for water quality measurements" , *Proc. SPIE* Vol. 7376, 737614 (2010);

[3] Bilenca, A. and Ozcan, A. and Bouma, B. and Tearney, G. "Fluorescence coherence tomography", *Optics Express*, Vol.14, 7134-7143, 2006.

[4] Zhu, H. and Yaglidere, O. and Su, T.W. and Tseng, D. and Ozcan, A. "Cost-effective and compact wide-field fluorescent imaging on a cell-phone", *Lab Chip*, Vol.11, 315-322, 2011.

[5] Brooker, G. and Siegel, N. and Wang, V. and Rosen, J. "Optimal resolution in Fresnel incoherent correlation holographic fluorescence microscopy" *Optics Express*, Vol. 19, 5047-5062, 2011.

[6] Bradley W. Schilling and Ting-Chung Poon and Guy Indebetouw and Brian Storrie and K. Shinoda and Y. Suzuki and Ming Hsien Wu, "Three-dimensional holographic fluorescence microscopy", *Optics Letter*, Vol.22, 1506-1508. 1997.

[7] Hariharan, P. and Sen, D.," Triangular path macro-interferometer", *JOSA*, Vol.49, 1105-1105. 1959.

Using Particle Filters for Parameter Estimation in Quantized Gaussian Autoregressive Processes

András Horváth
(Supervisors: Dr Miklós Rásonyi, Dr Tamás Roska)
horan@digitus.itk.ppke.hu

***Abstract*—I propose a method for the calculation of the maximum likelihood estimator for the autoregression (AR) coefficient of a stable quantized Gaussian autoregressive process. Our method uses particle filters with resampling, suits ideally many-core architectures and can be implemented in a parallel way, which yields fast processing speed. The extension to multidimensional autoregressive-moving-average (ARMA) systems is straightforward.**
This note is based on: "Maximum Likelihood Estimation of Quantized Gaussian Autoregressive Processes using Particle Filters with Resampling " *NOLTA 2012*

I. Introduction

Quantization is one of the most commonly used nonlinear operations in communication [1], hence parameter estimation of quantized stochastic processes is often needed in practice.

Solutions based on statistical theory are computationally expensive and in real life problems (e.g. speech recognition/speaker identification) only methods with relatively fast (real-time) processing speed can be applied. I investigate a scalable, parallelizable method that can estimate the parameter of an AR(1) process in an efficient way. Our algorithm can be executed quasi real-time on a virtual machine and it suits ideally a multicore architecture (e.g FPGA), which would result in real-time execution.

I have selected the Gaussian ARMA processes because their versatility makes them a natural class for investigating the effect of rounding off. An extension of our examples to multidimensional (ARMA) systems is possible but it would lead to minor complications which we prefer to avoid in the scope of this article.

II. Problem statement

I consider a stable Gaussian AR(1) process given by

$$X_n = \alpha_* X_{n-1} + \varepsilon_n, \quad n \geq 0,$$

where ε_n are i.i.d. random variables with law $N(0, \sigma_*^2)$ and $X_{-1} \in \mathbb{R}$ is a deterministic initial value. I assume that $X_{-1}, \sigma_* > 0$ are known. The autoregression parameter α_* is unknown but it lies in the interval of admissible parameters $(-1, 1)$.

Only the rounded-off, quantized values $Y_n = q(X_n)$, $n \geq 0$ are observed, where the quantizer function q is defined by

$$q(x) = k, \quad \text{for } x \in [k - 1/2, k + 1/2), \ k \in \mathbf{Z}.$$

Our aim is to calculate the maximum likelihood (ML) estimator: $\hat{\alpha}_n$ for α_* based on the observation sequence $Y_n, n \geq 0$. Similar problems have already been investigated and discussed in [2], [3], [4] but these papers assumed independent X_n (i.e. $\alpha_* = 0$). Following the footsteps of these authors I will apply the expectation maximisation (EM) method. However, I will combine it with suitably designed particle filters (see sections III, IV) while in [2], [3], [4] a Markov chain Monte Carlo (MCMC) approach was used instead. I analyse our simulation results in section V. I do not know of any other papers dealing with the calculation of maximum likelihood estimates in the present AR(1) setting.

III. Motivation for using particle filters

Particle filters (PFs) are used for state and probability estimations in numerous applications. They work well with difficult and non-linear Markovian models where the Kalman filters can not be used. Particle filters without resampling proved not to be efficient enough for complex, practical problems. Particle filters with resampling have much better performance. They lose, however, the property of being suitable for estimating functionals depending on the entire trajectory, because I alter the distribution of the particles during resampling since this would require keeping alive all the particles along the whole trajectory.

Nevertheless, in the following sections I would like to show how particle filters with resampling can be used successfully as part of the EM method for parameter estimation of quantized AR processes. To have simple formulas, I assume $X_{-1} = 0$ and $\sigma_* = 1/2$, but the conclusions hold for arbitrary values of these parameters.

For the moment I fix n, the number of iterations and concentrate on determining the maximum likelihood estimate $\hat{\alpha}_n$. Introducing the notation

$$p(\alpha; y_0, \ldots, y_n) = \frac{1}{\pi^{(n+1)/2}} e^{-y_0^2 - \sum_{j=1}^{n} (y_j - \alpha y_{j-1})^2}$$

for the joint density of $(X_0, \ldots, X_n)$ when the true parameter is α, the estimate $\hat{\alpha}_n$ can be found by maximising

$$\int_{Q(Y_0, \ldots, Y_n)} p(\hat{\alpha}_n; y_0, \ldots, y_n) dy_0 \ldots dy_n, \tag{1}$$

in $\hat{\alpha}_n$ which amounts, by taking the derivative, to solving

$$\int_{Q(Y_0,\ldots,Y_n)} \left(\sum_{j=1}^{n} -y_{j-1}(y_j - \alpha y_{j-1}) \right) \cdot$$

$$p(\hat{\alpha}_n; y_0, \ldots, y_n) dy_0 \ldots dy_n = 0, \tag{2}$$

here $Q(Y_0, \ldots, Y_n)$ denotes the cube $[Y_0 - 1/2, Y_0 + 1/2] \times \ldots \times [Y_n - 1/2, Y_n + 1/2]$.

Since (2) is a complicated nonlinear equation, following [2], [3], [4], I apply the EM method when implementing the calculation of the ML estimate. In the present context this means setting an arbitrary value $\tilde{\alpha}_0$ and then determining $\tilde{\alpha}_l$ recursively as the root of

$$\int_{Q(Y_0,\ldots,Y_n)} \left(\sum_{j=1}^{n} -y_{j-1}(y_j - \alpha y_{j-1}) \right) \cdot$$

$$p(\tilde{\alpha}_{l-1}; y_0, \ldots, y_n) dy_0 \ldots dy_n = 0 \tag{3}$$

Based on empirical evidence and theoretical results in various model classes [5] one expects that, after a suitable number of iterations (i.e. l large enough), one can get close to $\hat{\alpha}_n$. The great advantage of equation (3) is that it is linear in α.

Calculating $\tilde{\alpha}_l$ still poses a significant challenge as it requires the estimate of a functional of a trajectory (the integral in equation (3)). Unlike in [2], [3], [4], where an MCMC approach is suggested, I apply particle filters for the determination of the integrals (3). Note also that in [2], [3], [4] test runs concentrated on the i.i.d. case (where $\alpha_* = 0$ is known) and the (unknown) expectation of ε_1 was estimated. We took $E\varepsilon_1 = 0$ for simplicity but α_* is non-zero which leads to far more difficult, non-i.i.d. dynamics for the processes X_n, Y_n and hence poses a much more complex problem.

Applying PFs without resampling seems hopeless, especially considering the fact that in the present case we have a non-discrete, infinite state space where probability distribution of the trajectories is continuous so even by discretization without resampling we need an extremely large number of particles to simulate these trajectories.

It turns out that, with a small alteration PFs with resampling (i.e. the standard bootstrap filter) can be successfully used for probability estimation. The changes in the algorithm require some extra computation, but the number of particles can be decreased drastically compared to the method without resampling.

IV. Estimation of functionals of a trajectory by particle filters with resampling

We explain a method to evaluate the integral in equation (1) (i.e. equation (4) below), calculation of the integral in (3) can be carried out much in the same way.

In practical problems we usually have a given trajectory y_t, $t = 0, 1, \ldots, T$ generated by the previous model with T iterations (I choose $T = 100$ or $T = 500$).

Our aim is to estimate the probability that from the given model we will get the given trajectory of observations, i.e.:

$$P(Y_t = y_t, Y_{t-1} = y_{t-1} ... Y_0 = y_0). \tag{4}$$

Sequential Monte Carlo methods such as particle filters are routinely used in the case of such complex, non-linear, stochastic models with non-linear observations. The present system, however, has been revealed to be more challenging than usual.

First we look at PFs without resampling. We start off with an initial guess $\tilde{\alpha}_0$ for α.

At the initial step ($t = 0$) we generate N particles, ξ_k^0, $k = 1, \ldots, N$, i.i.d. with the same distribution $N(X_{-1}, \sigma_*^2)$ (i.e. with the law of $X_0^{(\tilde{\alpha}_0)}$).

At step t we iterate the particles using the system dynamics, i.e.

$$\xi_k^{t+1} := \tilde{\alpha}_0 \xi_k^t + \epsilon_{t+1}^k$$

with ϵ_{t+1}^k i.i.d. Gaussian with mean 0 and variance σ_*.

Every particle represents one possible trajectory and after the last (Tth) iteration we can calculate how many particles are identical with our observation. Based on this, our estimate for the probability in (4) is by the relative frequency

$$\frac{\sum_{i=1}^{N} I(\xi_i^0 = y_0, \ldots, \xi_i^n = y_n)}{N},$$

where the numerator represents the number of particles which results the same observations for every time and the denominator N is the total number of particles used. In theory this fraction converges to the value (4) as $N \to \infty$, however, in practice, especially in case of long, multidimensional trajectories we have to use an extremely large number of particles, because after each step the number of matching trajectories will always be decreased by the same ratio in average.

However, PFs with resampling (based on our current observation) are more promising, especially when combined with importance sampling [6]. We can decrease the number of useless (not identical with the observation) particles, and increase the number of useful (matching the observations) ones. The main point is that, in this way, the number of particles required will not depend on the length of the trajectory. PFs with resampling work well for estimating the density of X_T conditional to the observations $Y_0, \ldots, Y_T$. But during the resampling procedure the ratio between the number of good and bad particles gets lost and the estimation of the probability for the whole observed trajectory does not work in the obvious manner.

Here we briefly describe how one can calculate the probabilities between state transitions: although at each step certain particles do not match the observation and hence drop out from the cohort (as they are resampled with probability 0). One can 'blow up' the remaining cohort resampling M 'matching particles' (respecting the overall distribution of particles) to size N again. Getting more into the technicalities, this involves a discretization of the state space $[Y_t - 1/2, Y_t + 1/2]$ using a suitable mesh size; then one has to calculate how many of the

matching particles fall into the respective intervals and from this one can create a new cohort of N following the same distribution as the previous M particles.

We can approximate the distribution of these particles by dividing the set of the possible observations into Q intervals $J_1, \ldots, J_q$ and calculate the number of matching particles (ξ_k^*) in every interval.

$$f(q) = \frac{\sum_k I(\xi_k^* \in J_q)}{N},\ q = 1, \ldots, Q. \tag{5}$$

The limit of $f(q)$ is the distribution of the hidden states as the number of particles and the number of subintervals approach infinity.

Let us note the approximated distribution of the particles after the resampling with $g(i)$. There are no known theoretical connection between $f(i)$ and $g(i)$. $g(i)$ can be calculated easily in the same way as in equation (5).

Our aim is now to preserve the distribution of the particles. We can introduce the following operator, which is how the resampling step alters the distribution of the cohort:

$$\nu(f(q)) = g(q) \tag{6}$$

That transform $f(q)$ into $g(q)$. It can be easily seen that ν is always invertible and we can use ν^{-1} to restore $f(q)$ from $g(q)$.

All we have to do is to set the weights in interval J_q to $\nu^{-1}(g(q))$. After this step the weighted sum in interval J_q will be equal to the probability that the hidden state will be an element of interval J_q.

$$f(q) = \sum_k w_k^t \xi_k^t I(\xi_k^* \in J_q) \tag{7}$$

where ξ_k^t are particles after the resampling step and w_k^t represents the weight of the particle.

The mesh size (Q) is another parameter to be optimized for concrete test runs. An optimal mesh size can be approximated offline for every known model. We also have to note that when the mesh size is $Q = 1$, we will have the simple method with resampling, the regular bootstrap filter, where we do not alter the weights of the cohort. We do not have to alter bootstrap filters to expand the number of good particles after a resampling (because this is what resampling does). If all the weights are zero outside the given interval, only N particles within the interval can be selected, however some particles will be selected repeatedly and they will be overrepresented in the current distribution (it is also possible that some particles will not be selected. They will be underrepresented). We will not alter the distribution of the particles, we will alter the weights and with this the distribution that the particles are forming. This way the distribution of the particles is the same as in case of the regular boostrap filter, whose convergence has been proved in [7].

In this way we can calculate probabilities of trajectories as well as integrals of (3) using PFs with resampling, which makes the number of particles to be used exponentially less.

From this point on we follow the EM method: having calculated the integral in (3) using $\tilde{\alpha}_0$ as a parameter, one may get $\tilde{\alpha}_1$ solving (3), then iterate the procedure to get better and better approximations $\tilde{\alpha}_l$ of $\hat{\alpha}_n$, where l is the number of iterations performed.

This way we can mix the advantages of probability calculations with the fast, parallel computation possibility of bootstrap filters.

V. Simulations and Results

We have created a virtual machine to test the usability of our method in practice. We have simulated different AR(1) processes and we have tried to estimate the parameters according to the previously described algorithm. We have repeated every measurement 10000 times and calculated the absolute error as an average to eliminate the noise of our measurements.

Our results can be seen in tables I and II. It can also be seen that the use of importance-sampling leads to a much lower error, especially in case of a large number of particles. These low errors (we can approximate α with error ± 0.001) may be accurate enough to be used in case of real life problems.

With this type of calculation one iteration of $\tilde{\alpha}_l$ with 1000 particles and over a 100 steps long trajectory could be calculated in 4.6 seconds on a single core architecture. On a four core architecture the execution speed can be further decreased: the same estimation can be calculated in 1.8 seconds on an Intel Q9600 CPU.

The main advantage of our method is that almost every step of the algorithm (including the time consuming resampling step) is parallelizable, hence on a proper architecture such as an FPGA, GPU or digital-CNN architecture the computation speed could be further decreased, see [8].

In case of the method without resampling, $\tilde{\alpha}_l$ can not be estimated with 1000 particles at all, because in most of the cases there are no trajectories that will match our observations. For the current model and for a trajectory of length 100 the PF with resampling could calculate the probability with 300 particles with the same accuracy as a PF without resampling with 10000000 particles. This shows how drastically resampling decreases the number of effectively used particles. While we keep all of the particles and the computational power efficient, for a longer trajectory the computation time will increase linearly (instead of the exponential growth of the normal method), because the number of the efficient particles is not decreasing.

The biggest problem in case of this algorithm is that we have no information about α. If the distance between the real value and our initial value is too large there will not be any particles matching the observation. To avoid this we can restart our algorithm with another parameter, however this takes an extra iteration. To avoid this extra calculation and further speed up our algorithm we can extend our model and add $\tilde{\alpha}_0$ as a parameter, this way we can start our algorithm with multiple values of $\tilde{\alpha}$ and the parameters not matching the observations will be filtered out during the iterations by the resampling step.

VI. Conclusion

This short note describes a method that performs probability calculations using the fast, parallel computation possibility of bootstrap filters. The processing speed of our method depends linearly on the trajectory length and its accuracy increases with it. In previous methods and solutions either the accuracy did not change without the exponential increase of computational costs or the running time depended exponentially on the length of the trajectory. Our method fits ideally multicore architectures such as an FPGA, GPU or digital-CNN hence one could create a device that can solve this difficult problem in real-time with low power consumption.

References

[1] H. Boche, U. J. Monich. Behavior of the Quantization Operator for Bandlimited, Nonoversampled Signals. *IEEE Transactions on Information Theory* 2433 - 2440, 2010

[2] L. Finesso, L. Gerencsér, I. Kmecs. A randomized EM-algorithm for estimating quantized linear Gaussian regression. *Proceedings of the 38th IEEE Conference on Decision and Control, Phoenix, Arizona, USA*. 5100–5101, 1999.

[3] L. Finesso, L. Gerencsér, I. Kmecs. A recursive randomized EM-algorithm for estimation under quantization error. *Proceedings of the American Control Conference, Chicago, Illinois.*, 790–791, 2000.

[4] L. Gerencsér, I. Kmecs, B. Torma. Quantization with adaptation – estimation of Gaussian linear models. *Communications in Information and Systems*, **8**, 223–244, 2008.

[5] Moon, T.K. The expectation-maximization algorithm IEEE Signal Processing 1996 vol 13-6 47-60, 1996

[6] Peter W. Glynn and Donald L. Iglehart Importance Sampling for Stochastic Simulations Management Science Vol. 35-11, 1367-1392, 1989

[7] D. Crisan, A. Doucet. A Survey of Convergence Results on Particle Filtering Methods for Practitioners. *IEEE Transactions on Signal Processing* 736-746, 2002

[8] A. Horvath, M. Rasonyi Topographic Implementation of Particle Filters on Cellular Processor Arrays. *ELSEVIER Signal Processing Submitted*

VII. Tables

NumP	1000	1000	1000	1000	2000
Div	10	20	50	50	20
TrajL	100	100	100	500	500
I1	0.1472	0.0682	0.0672	0.0103	0.0141
I2	0.1263	0.0680	0.0672	0.0085	0.0125
I3	0.1080	0.0678	0.0672	0.0054	0.0114
I4	0.0860	0.0675	0.0669	0.0031	0.0112
I5	0.0750	0.0673	0.0669	0.0027	0.0110
I6	0.0724	0.0672	0.0666	0.0025	0.0110
I7	0.0722	0.0666	0.0665	0.0025	0.0110
I8	0.0719	0.0666	0.0658	0.0025	0.0110

TABLE I

THE FIRST ROW CONTAINS THE NUMBER OF PARTICLES (N), THE SECOND ROW CONTAINS 'HOW FINE' THE INTERVAL OF THE OBSERVATION WAS DIVIDED (Q) FOR THE DISTRIBUTION ESTIMATION, THE THIRD ROW IS THE LENGTH OF THE TRAJECTORY (n) AND THE ROWS BEHIND THIS ARE THE ERRORS TO EVERY ITERATION OF THE EM METHOD (ITERATIONS 1–8). THESE EXPERIMENTS WERE DONE USING THE IMPORTANCE-SAMPLING METHOD.

NumP	1000	1000	1000	1000	2000
Div	10	20	50	50	20
TrajL	100	100	100	500	500
I1	0.0847	0.1152	0.1661	0.1363	0.2127
I2	0.0826	0.0991	0.1524	0.1022	0.2056
I3	0.0801	0.0976	0.1471	0.0782	0.1948
I4	0.0790	0.0958	0.1439	0.0475	0.1866
I5	0.0782	0.0951	0.1393	0.0189	0.1820
I6	0.0771	0.0948	0.1373	0.0091	0.1748
I7	0.0762	0.0910	0.1274	0.0085	0.1647
I8	0.0750	0.0908	0.1268	0.0084	0.1629

TABLE II

THE FIRST ROW CONTAINS THE NUMBER OF PARTICLES (N), THE SECOND ROW CONTAINS 'HOW FINE' THE INTERVAL OF THE OBSERVATION WAS DIVIDED (Q) FOR THE DISTRIBUTION ESTIMATION, THE THIRD ROW IS THE LENGTH OF THE TRAJECTORY (n) AND THE ROWS BEHIND THIS ARE THE ERRORS TO EVERY ITERATION OF THE EM METHOD (ITERATIONS 1–8). THESE EXPERIMENTS WERE DONE WITHOUT USING THE IMPORTANCE-SAMPLING METHOD.

Data-flow graph partitioning to design locally controlled arithmetic units in FPGAs

Csaba Nemes
(Supervisors: Zoltán Nagy, Péter Szolgay)
csaba.nemes@itk.ppke.hu

***Abstract*—In an interesting class of computationally intensive problems, where a partial differential equation has to be solved over the discretized points of a mesh, a dedicated arithmetic unit can be implemented in FPGA to accelerate computations. The overall performance of the accelerator highly depends on the operating frequency of the arithmetic unit, which can be limited by the long controlling signals. In the paper a data-flow graph partitioning algorithm is presented to design a locally distributed control to the arithmetic unit, which can be efficiently mapped to the FPGA and operate at high frequency. Partitioning and placement aspects of the design are investigated to obtain a high-speed circuit. The algorithm is demonstrated during the automatic circuit generation of a complex partial differential equation.**

***Keywords*-data-flow graph partitioning; FPGA; floorplan**

I. Introduction

Computational problems defined on a mesh can be efficiently accelerated by FPGAs. In this type of problems a mathematical expression has to be evaluated continuously and many times over different points of a mesh. The implemented circuit can be decomposed to a memory interface and an arithmetic unit. Our primary aim is to automatically map the given mathematical expression to the FPGA by replacing every mathematical operation with a standard Xilinx floating-point IP core. A mathematical expression related to a Computational Fluid Dynamics (CFD) [1] problem is used as a test case.

To reach high operating frequency the arithmetic unit shall be partitioned and a local control unit should be assigned to every cluster. Our approach significantly differs from low level partitioning techniques as we partition the IP components at system level to help the standard Xilinx synthesis and place and route tools to exploit the local connectivity of the floating-point execution units. In a previous work [2] it was demonstrated that the ideal partitioning minimizes the number of the cut nets and the number of I/O connections of the clusters is less than roughly 10. The latter constraint will guarantee that the signals of the control unit will have tolerable fanout and will not decrease the operating frequency of the rest of the circuit. In a previous work [3] a new placement aware partitioning strategy was proposed and compared to traditional algorithms. In this work an improved placement algorithm is proposed which can inherently avoid deadlock and handle more objectives.

II. Constraints and objectives

A. Minimize the number of cut edges and constrain the number of IO connections of each cluster

The number of cut edges can be minimized and the number of the I/O connections of each cluster can be constrained by common graph partitioning techniques [4] [3]. The number of cut nets is minimized to reduce the extra area requirements of the circuit while the number of I/O connections of each cluster is constrained to limit the fanout of the control signals. If a control has a small fanout, the desired operating frequency of the given cluster can be reached.

B. Minimize the length of the longest interconnection between the clusters

Usually the long or high fanout interconnections cannot meet the timing requirements during routing procedure. Clusters which are connected to each other via the synchronizing FIFO buffers shall be placed close to each other to avoid long interconnections. If long interconnections cannot be avoided the only way to keep the timing requirements is to split up the control signals by extra pipeline registers. Unfortunately, splitting the control signals increases the latency of the control and it has to be redesigned to operate on data bursts.

C. Deadlock

Naive partitioning techniques, which only minimize the number of cut nets, can create clusters which mutually depend on each other even if the original graph is acyclic. This problem does not arise if the control logic has an enable signal connected to every single floating point unit to halt the operation, however the usage of the enable signal is not favorable in high performance implementations due to its high fanout. In our architecture there is no enable signal and only the synchronizing FIFO buffers are monitored. A cluster only starts to operate if all of its inputs (and outputs) are ready, therefore the mutually dependent clusters will never start (deadlock). Using traditional partitioning techniques the deadlock-free partitions have to be manually selected from a number of trials.

D. Pipeline length

The overall pipeline length of the arithmetic unit can be computed by finding the longest directed path in the graph depicting the connections between the clusters. The pipeline length has effects on both performance and circuit area. In case

of large CFD problems, where the arithmetic unit has to execute a lot of operations during even one time step the effects on performance can be negligible. However effects on circuit area cannot be neglected because longer pipeline length requires longer FIFO buffers to guarantee continuous operation. Naive partitioning strategies cannot minimize pipeline length while in our framework this objective can be also considered.

III. Framework and Algorithm

The main idea of the algorithm is to combine the partitioning and the placement objectives in a two-step procedure. In the first step an initial and simplified floorplan of the floating point units is created with simulated annealing to minimize the distance between the connected floating point units. In the second step the floorplanned floating point units are partitioned by another simulated annealing to minimize the previously described partitioning objectives.

A. Preprocessing and layering

The mathematical expression to be implemented is described in a text file, where inputs, outputs, and internal variables are also defined. In the first step the input file is parsed and a data-flow graph representation of the mathematical expression is created. Every mathematical operator is represented with a vertex and has an associated delay which will be the pipeline length of the corresponding IP core in the implemented circuit.

In the next step a *layering* [5] is performed, in which the data-flow graph is converted to a special bipartite graph. In this bipartite graph every vertex is associated with a vertical level and each arcs in the graph directs immediately to the next level.

Definition 1: $L = \{l_1, l_2, ...\}$ layering is a partition of the V vertices of the $G(V, E)$ directed graph such that

$$\forall (u, v) \in E \quad and \quad u \in l_i \rightarrow v \in l_{i+1}.$$

A layering can be generated via a breadth-first-search in linear time [5] by splitting up the arcs, which span more than one level, with extra delay vertices. An example layering is shown in Figure 1. The layering of the graph is an artificial restriction on the placement and the partitioning of the graph. Vertices can only be moved horizontally during the placement, and the representation of the clusters also depends on the structure of the layers. Evidently, this restriction can leave out the optimal solution from the search space, however, it significantly decreases the representation costs of the placement and the partitioning and makes the application of simulated annealing possible.

Fortunately, the layering has several other benefits beside the simplification. First of all, it naturally comes from the observation that it is good to cut the graph horizontally if the vertices have nearly the same delay and the minimal pipeline length can be preserved. The second benefit of the layering is that a simple and sufficient criterion can be formulated to check the existence of deadlocks during the simulated annealing.

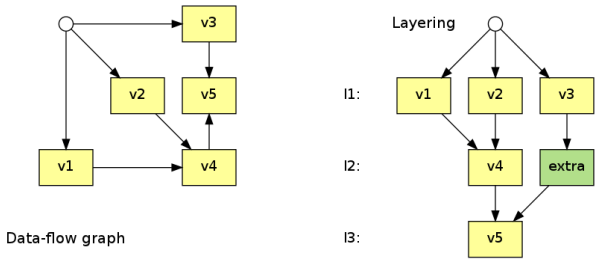


Fig. 1. A simple data-flow garph and its layered version

Finally, in physical implementation extra vertices are implemented as shift registers (extra vertices inside one cluster are joined) which hold the data for the proper number of clock cycles. From the aspect of performance it is advantageous because smaller interconnections help to keep timing requirements.

B. Floorplan with simulated annealing

In our current experiments a simplified homogeneous floorplan was used where every vertex has a unit width and the different types of resources are not distinguished. The blocks in one layer are represented by their sequence which is the 1D version of the famous sequence pair representation [5]. This representation is appropriate for ASIC floorplanning, however in case of FPGAs, where empty spaces inside the design are favorable some extensions are required to handle extra spaces. In our algorithm we introduce bubble vertices to hold empty spaces inside the design. Additional vertices increase the complexity of the problem, therefore the number of the bubble vertices on each layer is limited.

During each move of the simulated annealing algorithm the sequence of the vertices of one layer is perturbed. Layers are selected with probability proportional to the number of vertices(N_l) they contain. In the selected layer(l) one vertex is selected with probability $(1/N_l + 1)$ or a bubble is created with probability $(1/N_l + 1)$. If a normal vertex is selected, it will be swapped with one of its neighbor or if a bubble vertex is selected, it will be resized by one.

Initial sequences of the simulated annealing are computed with the Barycentre heuristic [6] which tries to minimize the edge crossing in a layered digraph.

During simulated annealing the linear combination of the following three objective functions were used:

1) Total squared distance (TSD) of the connected vertices: This objective minimizes the distance between the connected vertices. This objective automatically avoids long interconnections between the clusters. Distance between two vertices are determined according to their horizontal coordinates:

$$distance(A, B) := \begin{cases} (x_A - x_B)^2 & \text{if A and B are connected} \\ 0 & \text{otherwise} \end{cases}$$

where x_A and x_B are the horizontal coordinates of vertex A and B respectively.

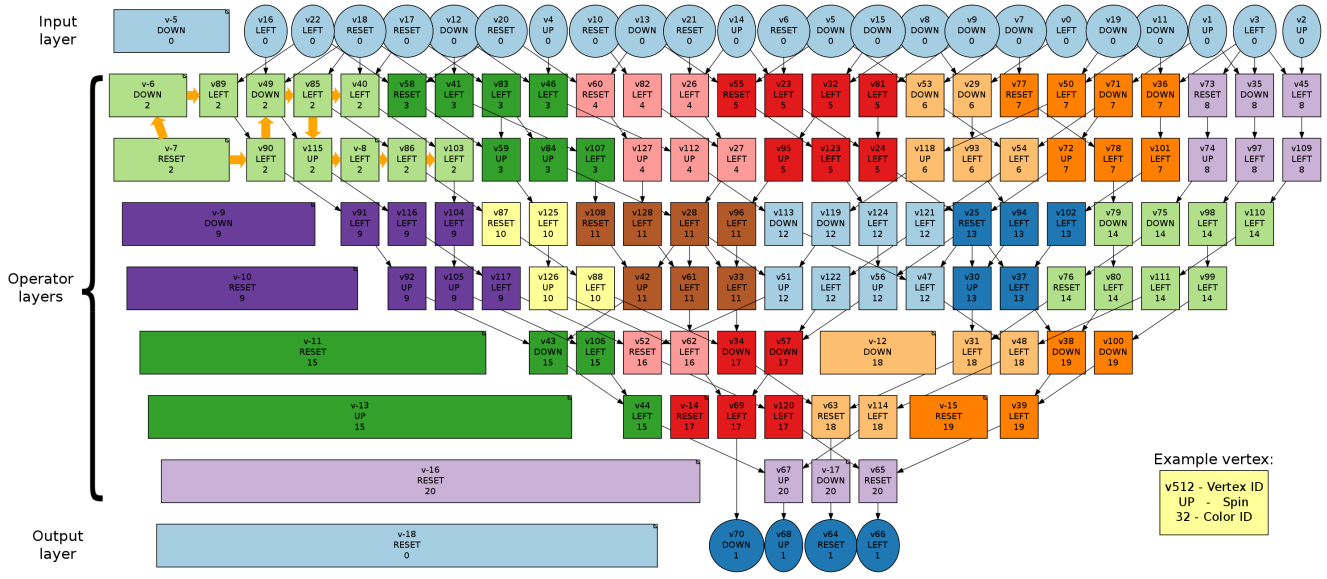


Fig. 2. Partitioning and floorplan of a layered graph generated from a mathematical expression related to CFD

2) Maximum distance (MD) between vertices which have a common input: This objective tries to place those vertices close to each other which have a common input. These cases are treated more carefully because the interconnecting signals have larger fanout.

3) Maximum interconnection length(MIL): In practice the length of the longest interconnection is more important than the TSD because usually the longest interconnection has a large delay explicitly limiting the operating frequency.

The result of the simulated annealing in case of a complex mathematical expression related to CFD is shown in Figure 2. In our experiments the following coefficients were used in the global objective function: TSD=0.2, LLI=6, MIL=3.

C. Partitioning

To use simulated annealing for graph partitioning a good representation is needed in which the cost of the partitioning and the quality of the search space are balanced. In the following, for simplicity, partitioning is represented by vertex coloring.

Restrictions: In our approach the layers of the graph are grouped to *belts* and each belt is colored separately; vertices of the same color must be on the same belt. This restriction significantly reduces the complexity of the problem and has the same advantages and disadvantages as layering. In our framework the height of the belts can be adapted by the user to the given mathematical expression.

One of the main benefits of this restriction is that two simple criteria can be formulated against deadlocks caused by partitioning.

Theorem 1: If the graph is layered, clusters are separated by the belt boundaries and the following two criteria hold, no deadlock is possible in the partition:

- Inside the belt only the neighboring clusters are connected.
- Inside the belt there is no mutual dependency between the neighboring clusters.

For a proof it shall be observed that no directed cycles can be present in the graph depicting the connections between the clusters.

Representation: The main motivation of the partitioning is to find a partition in which the clusters are locally connected and can be efficiently mapped to the FPGA. Hereby, we present a novel representation, which represents only *valid partitions*, in which clusters can be covered by continuous non-overlapping regions.

Validity of a partition can be guaranteed if every vertex inherits its color from one of its neighbors or is painted with a unique color. As the vertices have uniform sizes the direction of the inheritance can be described with a *spin* associated to every vertex. (See orange arrows and cluster 2 in Figure 2.)

The possible spins values are the following:

- LEFT : Vertex will inherit color from the left neighbor.
- DOWN : Vertex will inherit color from the bottom neighbor.
- UP : Vertex will inherit color from the upper neighbor.
- RESET : Vertex will be painted with a unique color.

Coloring: Each belt is colored separately. The algorithm starts from left and in every iteration one *column* of vertices is colored. Columns are computed based on the horizontal position of the vertices, therefore, in a worst-case scenario, if all the vertices have different position, the algorithm needs $O(N)$ steps to complete.

Vertices in a column are colored according to their spins from top to down. If no trivial coloring is possible, the color of the lowest vertex is reseted and the rest of the vertices in the column are revisited in a bottom-up order. With this

bidirectional visiting order both DOWN and UP spins can take effect. The results of the coloring is shown in Figure 2. Vertices with negative ID denote bubble vertices.

In each iteration the spin of one vertex of one layer is perturbed and the affected belt is re-colored. In our framework the linear combination of the following objectives is used to guide the simulated annealing.

- Number of cut arcs
- Maximum number of cut arcs which belong to the same cluster (maximum cluster IO)
- Number of clusters
- Number of connection between non-neighboring clusters which are on the same belt
- Number of mutual dependencies between neighboring clusters which are on the same belt

IV. Implementation results

The circuits were implemented on a Xilinx Virtex-6 FPGA (XC6VSX475T) with speed grade -1. FIFOs and floating-point units were generated by the Xilinx Core Generator [7]. As a reference an unpartitioned and a fully partitioned version of the arithmetic unit were also implemented. In the latter case every vertex formed a separate cluster. Partitioning results, resource utilization and operating frequencies are compared on Table I.

Pipeline length is 100 clock cycles in the reference cases, where the longest weighted path in the data-flow graph is not perturbed by the partitioning. In the optimized case unfavorable clustering of some of the vertices of the critical path increased the latency of the pipeline, however, it is still tolerable from the aspect of performance. The least FIFOs are needed in the unpartitioned case because only inputs and outputs have to be buffered. Despite the different pipeline lengths, the FIFO requirements do not differ significantly in the other two cases, which is favorable from the aspect of area requirements.

Latency differences of the inputs of a cluster can be absorbed in the FIFO buffers, however, inside the clusters there are no buffers and every data has to arrive precisely. During VHDL generation every cluster is checked and extra delays are added to equalize latency differences of the inputs of the vertices. Naturally, in the fully partitioned case no extra delays are needed because clusters contain only one vertex.

In Table I the cut arcs (*outside*) which connect operators with IO and the ones (*inside*) which connect operators to other operators are distinguished. In the unpartitioned case the number of cut arcs trivially equals with the number of inputs and outputs. Partitioning can cut input hyperarcs of the graph, which increases the number of the cut arcs outside the graph. The number of cut nets in the other two cases are roughly the same. This can be explained by the added extra delays, which significantly increased the size of the graph. According to these measurements the extra area requirements caused by the layering were compensated by the smart partitioning.

Finally, area requirements using the proposed algorithm are in the range of the fully partitioned case but the operating frequency is 15% higher.

TABLE I
Partitioning and implementation results of the CFD graph.

		Unpartitioned	Fully-partitioned	Optimized
Pipeline length		100	100	128
FIFO depth	32	23	68	60
	64	0	18	27
	128	4	6	3
	TOTAL	27	92	90
Extra delay vertex		26	0	33
Num. of clusters		1	44	19
Max cluster cut		27	6	10
Cut arcs	Outside	27	46	38
	Inside	0	46	52
	Total	27	92	90
Area	FF	41,664	48,591	50,775
	LUT	31,384	37,297	38,869
	DSP	244		
Frequency (MHz)		251,889	279,33	321,44
Improvement		100%	110,89%	127,61%
			100%	115,08%

V. Conclusion

It has been demonstrated that to design an arithmetic unit which consists of locally controlled groups of floating point units both partitioning and placement aspects have to be considered. To solve this problem a framework has been given, in which the partitioning is based on an initial floorplan of the vertices of the layered data-flow graph of the input mathematical expression. In the partitioning step a novel graph partitioning representation has been used to represent only valid clusterings, in which clusters can be covered by continuous and non-overlapping regions. Beside the valid clusters the algorithm can minimize the number of cut nets and guarantee deadlock-free partitions. The algorithm was tested by implementing a complex CFD related mathematical expression and 15%-27% performance gain was achieved compared to the non-partitioned or the fully partitioned versions.

References

[1] S. Kocsárdi, Z. Nagy, A. Csík, and P. Szolgay, "Simulation of 2d inviscid, adiabatic, compressible flows on emulated digital cnn-um," *Int. J. Circuit Theory Appl.*, vol. 37, pp. 569–585, May 2009.

[2] C. Nemes, Z. Nagy, M. Ruszinkó, A. Kiss, and P. Szolgay, "Mapping of high performance data-flow graphs into programmable logic devices," *NOLTA 2010. International symposium on nonlinear theory and its applications*, pp. 99–102, Sep. 2010.

[3] C. Nemes, Z. Nagy, and P. Szolgay, "Efficient mapping of mathematical expressions to fpgas: Exploring different design methodologies," in *Circuit Theory and Design (ECCTD), 2011 20th European Conference on*, aug. 2011, pp. 717 –720.

[4] G. Karypis and V. Kumar, "HMETIS 1.5: A Hypergraph Partitioning Package," Department of Computer Science, Tech. Rep., 1998, http://www-users.cs.umn.edu/~karypis/metis.

[5] A. Kahng, J. Lienig, I. Markov, and J. Hu, *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer, Jul. 2011.

[6] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for visual understanding of hierarchical system structures," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 11, no. 2, pp. 109 –125, Feb. 1981.

[7] "Xilinx product homepage," http://www.xilinx.com, 2010.

Electrophysiological Correlates of the Different Hierarchical Levels of Visual Word Processing

Balázs Knakker
(Supervisor: Dr. Zoltán Vidnyánszky)
bknakker@gmail.com

***Abstract*—The ventral occipito-temporal cortex plays a critical role in the perception of written words. Previous functional magnetic resonance imaging studies suggested that in the ventral occipito-temporal cortex (with a left hemisphere dominance), words are encoded through a posterior to anterior hierarchy of neurons tuned to increasingly more complex orthographic features, such as single letters, bigrams, and possibly whole words. In line with this, the results of neurophysiological research revealed that visually presented letters and words evoke a left hemisphere lateralized N1 component, which part/component of orthographic feature processing is reflected in N1 is still an unresolved question. Here we investigated this issue by measuring how word rotation – which is known to impair parallel letter processing and thus results in serial letter-by-letter reading – affects the N1 component. Our stimuli consisted of: 1. horizontal words composed of vertical letters (HW-VL, the control condition); 2. vertical words composed of horizontal letters (VW-HL); 3. horizontal words composed of horizontal letters (HW-HL). It was found that N1 amplitudes over the right hemisphere were strongly increased both in the HW-HL and VW-HL conditions. The latency of N1 was also affected by rotation: N1 peaks, especially over the right hemisphere, were delayed in the HW-HL condition compared to the other conditions. These results suggest that the left hemisphere N1 component of the ERP responses reflect both single letter as well as global, whole word orthographic processing, whereas the right N1 might be associated primarily with the visual processing demands at the stage of single letter processing.**

***Keywords*-visual word perception, reading, event-related potentials, hemispheric lateralisation**

I. INTRODUCTION

Reading requires the sophisticated coordination of different perceptual processes with cognitive and motor control. One basic component of reading is the brain's capacity of fast and effective recognition of common visual objects like faces or visual words.

What enables us to recognize objects is a hierarchical neural circuitry located in the occipital and inferior temporal cortex. For example, functional magnetic resonance imaging studies indicate that faces elicit a distinct pattern of neural activity in the so-called Fusiform Face Area (FFA) in the right hemisphere [1]. Regardless of the debated origin and nature of this specialisation, it is undoubtable that faces are subject to a high degree of perceptual expertise in humans. Given that the information processing capabilities of the visual system is finite, it is obvious to suggest that this kind of perceptual expertise must be more pronounced in case of usual realizations of these object classes. For example, it has been shown that inverting faces delay and impair recognition [2]. This inversion also alters the neural signature of face processing, i.e. delays and increases the negativity of the N170 component [3].

Similarly, words are also important and frequently encountered visual stimuli. Therefore, the visual system of the adult reader is capable of fast and efficient recognition of visual words of usual format. As clarified by imaging studies, the neural substrate of this ability is a posterior to anterior hierarchy of neurons tuned to increasingly more complex orthographic features, such as single letters, bigrams, and possibly whole words. This system is located in the fusiform gyrus as well – more specifically, in the so-called Visual Word Form Area [4] – but in contrast to faces, word-related neural activations tend to be left-lateralized [5]. The EEG correlate of orthographic processing is a left-lateralized temporo-occipital negativity in the 150-170 ms time window[6].

As in the case of faces, studying the effects of format alteration can refine our knowledge about the mechanisms and neural background of visual word recognition. As deviation from the usual format increases above a critical degree, word length effects emerge in reaction times and naming latencies, because the effective and fast neural mechanisms are disrupted and the visual system shifts towards a slower, letter-by-letter approach [7]. This study aims at exploring the electrophysiological signatures of these two processing modes by using rotated word stimuli.

II. MATERIALS AND METHODS

A. Subjects

17 healthy right handed young adults participated in this study. All of them had normal or corrected-to-normal vision, none of them had any history of neurological or psychiatric diseases. All participants gave informed consent.

B. Stimuli and Procedure

The stimuli were 4 and 5 letter Hungarian nouns from two semantic categories (living and non-living), presented centrally on a TFT screen using a monospace font (Courier New), so that they subtended about 2 degrees of visual angle along their

longer dimension. A small blue fixation dot was always present in the center of the screen. The background was middle grey.

In the control experimental condition, the stimuli were presented in the usual format – horizontally, with upright letters (Horizontal Words with Vertical Letters, HW-VL, see Figure 1A). In the second condition, the words were rotated counter-clockwise by 90° (Vertical Words with Horizontal Letters, VW-HL, see Figure 1B). In the third condition, the words were presented horizontally such that each individual letter in them was rotated around its own centroid by 90° counter-clockwise (Horizontal Words with Horizontal Letters, HW-HL, see Figure 1C).

The subjects were seated in a dark room, their head supported by a chin rest in a distance of 50 centimeters from the screen. The experiments were conducted in 5 runs, each lasting cca. 8 minutes, with some minutes of rest in between. Within runs, the orientation of the words was constant. The first four were HW-VL and VW-HL runs; in the last run, HW-VL and HW-HL trials alternated in a randomized way. In half of the trials of the first four runs, words were presented with flanking text – these trials are not included in this report.

In each trial, a word was presented for 800 milliseconds, and after a pause of 500 milliseconds, the fixation dot briefly flashed in green. This was a cue for the subjects to respond with a mouse button press, indicating which category the word they had seen belonged to. The response interval was maximized in 2 seconds. The length of the inter-trial-interval (ITI, starting from the time of the response or from the end of the response interval) was chosen from a uniform probability distribution between 1250 and 1750 milliseconds. After every third trial an additional 650 ms of pause was added with the fixation dot turning red, and the subjects were asked to try to blink only during this period.

Stimulus presentation and subject response registration was implemented in MATLAB using PsychToolbox version 3 [8], [9].

C. Electrophysiological and Behavioral Measurements

EEG was acquired using 64 electrodes (Brain Products ActiCap; amplifier: BrainAmp MR) mounted on an elastic cap according to the extended 10/20 system. The sampling rate was 500 Hz and the signal was digitized using an external D/A converter supplied by Brain Products and recorded by the Brain Vision Recorder software. Eye movements were recorded using IView X Hi-Speed (SensoMotoric Instruments) at a sampling rate of 240 Hz.

D. Analysis

Task performance of the participants was assessed with a one-way repeated measures ANOVA.

Preprocessing of the EEG signal was done in Brain Vision Analyzer. The signal was bandpass filtered (Butterworth zero-phase filter, 0.1Hz-30Hz, 12 dB/octave) and segmented. Segments containing artefacts were marked using amplitude,

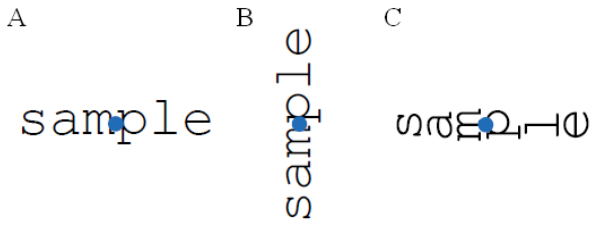


Figure 1. Sample stimuli from the three conditions . **A** Horizontal Words with Vertical Letters **B** Vertical Words with Horizontal Letters **C** Horizontal Words with Horizontal Letters

amplitude difference and voltage step thresholds and by visual inspection; these segments were not used in further analyses.

Data were imported to Matlab, and surface Laplacian approximations of the scalp current density was calculated using the CSD Toolbox[10] (spline flexibility m=4, smoothing $\lambda=10^{-5}$, maximal degree of Lagrange polynomials = 10). Artefact-free segments were baseline-corrected and averages for conditions of interest were computed for each subject. Channels P7, P8, PO9 and PO10 were used in the ERP analysis. On the subject averages, P1 and N1 component peaks were detected semi-automatically.

The amplitudes and latencies of the P1 and N1 component were entered into a repeated-measures ANOVA using within-subject factors CONDITION (3 levels: HW-VL, VW-HL and HW-HL), electrode SITE (2 levels: P7-P8 and PO9-PO10) and HEMISPHERE (2 levels: P7-PO9 for Left and P8-PO10 for Right). The Greenhouse-Geisser correction for violation of sphericity was applied where necessary. Post-hoc comparisons were conducted using Tukey's Honestly Significant Differences test. Simple lateralization indexes as differences of amplitudes between pairs of electrodes were compared in the same manner, excluding the HEMISPHERE factor.

III. Results

A. Behavior

Error rates were very small for all subjects (<1%) and did not differ across conditions.

B. Electrophysiology

The topography and lateralization of the N1 component differed significantly across conditions (CONDITION × HEMISPHERE × SITE: $F(2,32)=7.74$; $p=0.003$). Post-hoc comparisons revealed that both types of stimuli with unusual formatting (VW-HL and HV-HL) elicited significantly more negative N1 over the right hemisphere than normally formatted HW-VL stimuli ($p<0.001$ for all comparisons). This difference was non-significant over left-hemisphere sites.

The same post-hoc comparison showed significant lateralization effects, however, the site of lateralization was not entirely consistent. In the HW-VL condition, the N1 was significantly left-lateralized on the PO9-PO10 electrode pair ($p=0.001$). In contrast, the other horizontal word condition

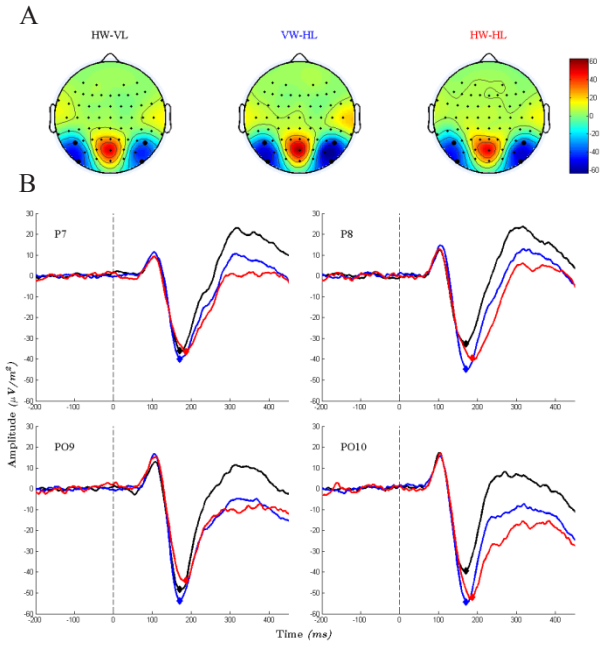


Figure 2. **A** ERP topographies at N1 latencies of each condition. **B** ERP time courses on electrodes of interest; the N1 (the time point of the corresponding topographies in A) is indicated on each curve. Marked electrodes on A and plots in B match in spatial arrangement.

(HW-HL), evoked right-lateralized N1 responses (PO9-PO10, p=0.004). In case of vertical words (VW-HL), the N1 was also right-lateralized, but significant differences only emerged on the P7-P8 electrode pair. Comparing lateralization indices across the three conditions yielded similar results, i.e. both unusual-format conditions elicited significantly more right-lateralized activity on PO9-PO10 than the normal-format condition (CONDITION × SITE: F(2,32)=7.74; p=0.003, post hoc p≤0.003 for the comparisons implied above).

Regarding the timing of N1, we observed significantly larger latency in the case of HW-HL stimuli compared to either of the two other conditions (main effect of CONDITION: F(2,32)=5.371; p=0.01, post hoc p<0.03). This difference was more prominent on the right (n.s.).

The P1 component was significantly smaller over SITE P7-P8 in the HW-HL condition compared to the VW-HL condition (CONDITION × SITE: F(2,32)=3.72; p=0.04, post hoc p=0.03). This difference was mainly evident on P7, still, this was not supported by a three-way interaction involving the HEMISPHERE factor.

IV. Discussion

The results described above show that altering the format of visual words can elicit modulations similar to the face inversion effect. All format alterations caused an increase in N1 response amplitudes over right hemisphere electrode sites, which also resulted in right-lateralization of these responses, as opposed to left-lateralized responses in the normal-format condition (Fig. 2).

The right-lateralization of most of the effects may be due to several factors. First, some authors argue that the fast and efficient, 'parallel' mode of visual word processing might only be available in the left hemisphere. In contrast, the neural substrate of letter-by-letter processing is present in both [11]. This asymmetry might be due to more direct access to phonological and semantic representations in the left hemisphere [12]. Nevertheless, in the case of central presentation, despite the bilateral distribution of the response in strictly retinotopic, earlier visual areas, the processing of visually familiar words can benefit from asymmetric relay to the left hemisphere [13]. This can account for the fact that only HW-VL stimuli evoked left-lateralized responses.

There are also more general theories of hemispheric lateralization that might be relevant. According to Dien [14], higher-level sensory areas in the left hemisphere respond more readily to predictable stimuli whereas the right hemisphere preferentially responds to unpredictable (novel, visually unfamiliar) stimuli. The results presented here are in line with this: the more peculiar the stimuli were, the more right-lateralized was the N1 in response to them.

The latency of the N1 response only differed for HW-HL stimuli, where, as opposed to the other two conditions, letter axes were parallel with the direction of reading. This N1 delay is similar to that found in the case of face inversion, where the effect is attributed to the disruption of first-order relations between constituent features, relations according to which we are able to quickly distinguish faces from non-face stimuli [2]. By analogy, this could possibly mean that the relation between word and letter axis would be an important cue of 'wordness' for the visual system. This is supported by the non-significant but visible trend (Fig. 2) that N1 responses on the left are equal or smaller for HW-HL compared to normally formatted (HW-VL) stimuli. However, clarifying this point would require further experiments.

V. Conclusion

These results support the notion that altering the format of words disrupts the highly efficient word-level visual processing based on perceptual expertise, and causes the visual system to resort to a less efficient letter-by-letter mechanism. We suggest that the N1 component of the ERP responses in the left hemisphere reflects both single letter as well as global, whole word orthographic processing, whereas the right N1 might be associated primarily with the visual processing demands at the stage of single letter processing.

Acknowledgment

This work was supported by grant from the Hungarian Scientific Research Fund to Z.V. (CNK80369).

References

[1] N. Kanwisher, J. McDermott, and M. M. Chun, “The Fusiform Face Area: A Module in Human Extrastriate Cortex Specialized for Face

Perception," *The Journal of Neuroscience*, vol. 17, no. 11, pp. 4302 – 4311, Jun. 1997.
[2] D. Maurer, R. L. Grand, and C. J. Mondloch, "The many faces of configural processing," *Trends Cogn. Sci. (Regul. Ed.)*, vol. 6, no. 6, pp. 255–260, Jun. 2002.
[3] S. Bentin, T. Allison, A. Puce, E. Perez, and G. McCarthy, "Electrophysiological Studies of Face Perception in Humans," *J Cogn Neurosci*, vol. 8, no. 6, pp. 551–565, Nov. 1996.
[4] L. Cohen, S. Lehéricy, F. Chochon, C. Lemer, S. Rivaud, and S. Dehaene, "Language-specific tuning of visual cortex? Functional properties of the Visual Word Form Area," *Brain*, vol. 125, no. Pt 5, pp. 1054–1069, May 2002.
[5] U. Maurer, B. Rossion, and B. D. McCandliss, "Category specificity in early perception: face and word n170 responses differ in both lateralization and habituation properties," *Front Hum Neurosci*, vol. 2, p. 18, 2008.
[6] U. Maurer, S. Brem, K. Bucher, and D. Brandeis, "Emerging neurophysiological specialization for letter strings," *J Cogn Neurosci*, vol. 17, no. 10, pp. 1532–1552, Oct. 2005.
[7] L. Cohen, S. Dehaene, F. Vinckier, A. Jobert, and A. Montavont, "Reading normal and degraded words: contribution of the dorsal and ventral visual pathways," *Neuroimage*, vol. 40, no. 1, pp. 353–366, Mar. 2008.
[8] D. H. Brainard, "The Psychophysics Toolbox," *Spat Vis*, vol. 10, no. 4, pp. 433–436, 1997.
[9] D. G. Pelli, "The VideoToolbox software for visual psychophysics: transforming numbers into movies," *Spat Vis*, vol. 10, no. 4, pp. 437–442, 1997.
[10] J. Kayser and C. E. Tenke, "Principal components analysis of Laplacian waveforms as a generic method for identifying ERP generator patterns: I. Evaluation with auditory oddball tasks," *Clinical Neurophysiology*, vol. 117, no. 2, pp. 348–368, Feb. 2006.
[11] A. W. Ellis, "Length, formats, neighbours, hemispheres, and the processing of words presented laterally or at fixation," *Brain Lang*, vol. 88, no. 3, pp. 355–366, Mar. 2004.
[12] L. Van der Haegen, Q. Cai, and M. Brysbaert, "Colateralization of Broca's area and the visual word form area in left-handers: fMRI evidence," *Brain and Language*, no. 0.
[13] L. Barca, P. Cornelissen, M. Simpson, U. Urooj, W. Woods, and A. W. Ellis, "The neural basis of the right visual field advantage in reading: an MEG analysis using virtual electrodes," *Brain Lang*, vol. 118, no. 3, pp. 53–71, Sep. 2011.
[14] J. Dien, "A tale of two recognition systems: implications of the fusiform face area and the visual word form area for lateralized object recognition models," *Neuropsychologia*, vol. 47, no. 1, pp. 1–16, Jan. 2009.

Representation of Facial Identity Information in the Medial and Anterior Temporal Lobe

Petra Hermann
(Supervisor: Dr. Zoltán Vidnyánszky)
hermann.petra@gmail.com

Abstract— **It has been shown that face processing involves several regions of the medial and anterior temporal lobe, including the perirhinal cortex and the temporal pole. It was suggested that these face responsive temporal lobe regions - located downstream from the well known fusiform face area - represent the highest stage of face processing and are critical for face identity discrimination and memory. An unresolved question is whether and to what extent facial identity information in these higher-level temporal lobe regions is represented in an abstract, feature-invariant manner. Here we addressed this question by measuring fMRI responses to intact and noisy (phase randomized) face stimuli. It was found that fMRI responses to the noisy face images were strongly reduced as compared to the intact faces in both the medial and anterior temporal regions, even though identity categorization was easy both in the case of intact and noisy stimuli. In contrast to this, fMRI responses to noisy images in the occipital face area and in the fusiform face area (OFA and FFA, two cortical regions involved in the structural processing of faces) were not altered or only moderately reduced, respectively. This excludes the explanation of reduced medial and anterior temporal fMRI responses in the case of noisy images based on a diminished input from OFA and FFA. Our results revealed that face-specific responses in the medial and anterior temporal lobe are highly sensitive to the quality of structural/contour information of the face stimuli and thus provide evidence against an abstract, feature invariant representation of the facial identity information in these regions.**

Keywords-face processing; fMRI; medial temporal lobe; temporal pole;

I. INTRODUCTION

Many face responsive areas – including different levels of face processing – have been already identified in humans and non-human primates [1–5]. However, we have no evidence whether and to what extent these face specific regions are sensitive to the quality of structural/contour information of face images. This sensitivity can be investigated with phase randomization resulting in the gradual elimination of facial features [6], [7]. Previous electrophysiological and neuroimaging studies revealed that phase randomization does not modulate neural responses to visual objects in the primary visual cortex [7–10]. On the other hand, noise sensitivity increases along the ventral pathway from V1 to the higher levels of visual hierarchy [9], [11], [12]. These results suggest that strongest sensitivity to phase noise, i.e. sensitivity to structural/contour information of images can be observed in regions which integrate information about the global structure of the stimulus [7], [13], [14]. A further study related to face processing in phase noise showed that the presence of phase manipulation does not affect the fMRI responses in face selective areas in inferior-occipital (OFA) and fusiform (FFA) cortex involved in the structural processing of faces [15]. However, we do not have enough information about the noise-induced modulation of neural activation in higher-order face responsive regions.

It was suggested that face specific regions in the medial and anterior temporal lobe - including the perirhinal cortex and the temporal pole - represent the highest stage of face processing and are critical for face identity discrimination and memory [2–5]. An unresolved question is whether these areas tolerate the damage to structural/contour information of faces and to what extent facial identity information in these higher-level temporal lobe regions is represented in an abstract, feature-invariant manner. Here we addressed this question by measuring fMRI responses to intact and noisy (phase randomized) face stimuli.

II. EXPERIMENTAL PROCEDURES

A. Subjects

Altogether 20 (one left-handed, nine females, mean ± SD age: 24 ± 4 years) subjects gave their informed and written consent to participate in the study, which was approved by the ethics committee of Semmelweis University. None of them had any history of neurological or psychiatric diseases, and all had normal or corrected-to-normal visual acuity.

B. Stimuli

In the fMRI session participants viewed images of human faces and household objects and performed a one-back memory task. Face stimuli consisted of front-view grayscale photographs of four male faces with neutral, happy and fearful expressions that were cropped to eliminate the external features (hair, etc.) (see Fig. 1). In their manipulated versions noise was added to the original images by decreasing their phase coherence to 45% (55% noise) using the weighted mean phase technique [6], which resulted in the gradual elimination of the facial cues important for accurate identity judgment. Object stimuli consisted of grayscale images of three different objects from four categories (cars, mugs, jugs, and fruits) chosen from the Amsterdam Library of Objects Images (ALOI) database [16]. All images were equated for luminance and contrast.

Figure 1. The four male identities with neutral expression used in the fMRI and psychophysics experiments.

In the psychophysics experiment these control and noisy faces were presented upright and upside-down (inverted) and participants performed an identity categorization task. To equate task difficulty between the control and noise conditions we decreased the identity difference between the four original faces in the control condition using a morphing algorithm (Winmorph 3.01). First we generated an average face from the four individual faces, into which each of them was warped along the identity axis to create intermediate images. Morph level and phase coherence were adjusted to achieve 65% accuracy and were based on pilot sensitivity measures. Morphed faces contained 75% of the average face, while the noise level was fixed at 55% (45% phase coherence).

In both experiments stimuli were presented centrally on a uniform gray background and subtended 3×4 visual degrees. Stimulus presentation was controlled by MATLAB 7.1. (The MathWorks Inc.) using the Psychophysics Toolbox Version 3 (PTB-3) [17], [18]. In the psychophysics experiment, visual stimuli were presented on a 26" LG LCD monitor at a refresh rate of 60 Hz and were viewed from 50 cm. In the case of the fMRI experiment, stimuli were projected onto a translucent screen located at the back of the scanner bore using a Panasonic PT-D3500E DLP projector (Matsushita Electric Industrial) at a refresh rate of 60 Hz. Stimuli were viewed through a mirror attached to the head coil at a viewing distance of 58 cm. Head motion was minimized using foam padding.

C. Procedure

The fMRI session included two block-design scans, each lasting 10 min. In each run 16-s-long epochs of control faces (CF), noisy faces (NF) and objects (O) were interleaved with baseline epochs, which contained only a fixation dot. Stimuli were presented with 0.5 Hz for 500 ms each. Blocks consisted of 6 control face, 6 noisy face, 6 object, and 19 baseline blocks, making a total number of 37 blocks per run. During the fMRI experimental session subjects performed a one-back task and reported the total number of one-back repetitions at the end of the run (see Fig. 2).

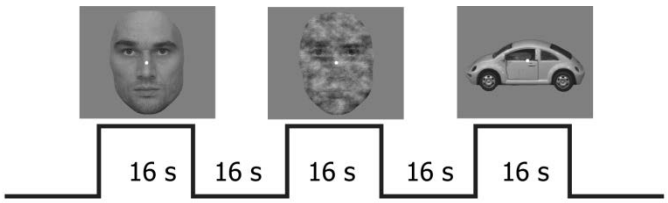


Figure 2. Experimental design in the fMRI session. 16-s-long epochs of control faces, noisy faces and objects followed each other in random order separated by baseline blocks.

In the psychophysics session subjects performed identity categorization in a four-alternative forced choice (4AFC) task: they were required to judge the identity of the face images as accurately and fast as possible, indicating their choice with one of the keyboard buttons. Before the experiment, each subject was given a practice session to get familiar with the task. Each trial began with a cue (1 deg) appearing just above fixation for 100 ms indicating the orientation of the upcoming stimulus (upright or inverted) and was followed by a 400-ms presentation of the face with a stimulus onset asynchrony (SOA) of 500 ms. The response window was a maximum of 2 s but was terminated when subjects responded. Trials were separated by an intertrial interval (ITI), which was randomized in the range of 1800–2200 ms. The fixation dot was present throughout the trial. Within a single run, the 2×2 conditions (control/noise and upright/inverted) were intermixed and presented in random order. Each participant completed two runs, yielding 72 trials altogether for each condition.

III. Data Analysis

A. Behavioral Data Analysis

Responses and reaction times (RTs) were collected during the experiments. Accuracy and RTs were analyzed with two-way repeated-measures ANOVA with condition (CF vs NF) and orientation (Up vs. Inv) as within-subject factors. *Post hoc* t-tests were computed using Tukey honestly significant difference (HSD) tests.

B. fMRI imaging and analysis

Data were collected at the MR Research Center of Szentágothai Knowledge Center (Semmelweis University, Budapest, Hungary) on a 3.0 tesla Philips Achieva scanner equipped with an eight-channel SENSE head coil. High-resolution anatomical images were acquired for each subject using a T1-weighted 3D TFE sequence yielding images with a 1×1×1 mm resolution. Functional images were collected using 31 transversal slices (4 mm slice thickness with 3.5 mm × 3.5 mm in-plane resolution) with a non-interleaved acquisition order covering the whole brain with a BOLD-sensitive T2*-weighted echo-planar imaging sequence (TR=2 s, TE=30 ms, FA=75°, FOV=220 mm, 64×64 image matrix, 2 runs, duration of each run = 610 s).

Preprocessing and analysis of the imaging data were performed using SPM8 (Wellcome Department of Imaging Neuroscience). The functional images were realigned to the first image within a session for motion correction and then spatially smoothed using an 8-mm full-width half-maximum Gaussian filter and normalized into standard MNI-152 space. The anatomical images were coregistered to the mean functional T2* images followed by segmentation and normalization to the MNI-152 space using SPM's segmentation toolbox. The resulting gray matter mask was used to restrict statistical analysis on the functional files. To define the regressors for the general linear model analysis of the data, a reference canonical hemodynamic response function was convolved with boxcar functions, representing the onsets of the experimental conditions. Low-frequency components were

excluded from the model using a high-pass filter with 128 s cutoff. Movement-related variance was accounted for by the spatial parameters resulting from the motion correction procedure. The resulting regressors were fitted to the observed functional time series within the cortical areas defined by the gray matter mask. Individual statistical maps were then transformed to the MNI-152 space using the transformation matrices generated during the normalization of the anatomical images. The resulting β weights of each current regressor served as input for the second-level whole-brain random-effects analysis, treating subjects as random factors. For visualization purposes, the contrast (CF > NF) (see Fig. 4A) was superimposed with $p_{unc}<3\times10^{-3}$ threshold onto the population average landmark and surface based (PALS-B12) standard brain [19] using Caret 5.62 [20]. Stereotaxic coordinates are reported in MNI space.

For the region of interest (ROI) analysis the face-selective areas were defined individually based on one of the two runs (subjects were randomly assigned to either of the runs, which was counterbalanced across subjects). Areas matching our anatomical criteria and lying closest to the corresponding reference cluster (i.e., clusters from the random-effects group analysis, $t_{(19)}>3.9$; $p_{unc}<5\times10^{-4}$) were considered to be their appropriate equivalents on the single-subject level. The location of the fusiform face area (FFA) and the occipital face area (OFA) were determined as areas responding more strongly to control faces than to objects ($t_{(280)}>3.9$; $p_{unc}<5\times10^{-4}$) or baseline (fixation dot) ($t_{(280)}>6.8$; $p_{FDR}<10^{-4}$). It was possible to define bilateral FFA (average MNI coordinates ± SD: 41±3, −50±5, −22±3 and −38±3, −49±5, −21±3 for right and left hemispheres, respectively) and bilateral OFA (42±3, −71±7, −13±3 and −40±5, −74±6, −13±3) in all 20 subjects. The superior temporal sulcus (STS) (61±7, −41±7, 7±4 and −58±5, −49±6, 5±5), the medial temporal lobe (MTL) (25±3, −8±3, −19±4 and -22±3, −10±5, −16±3) and the anterior temporal pole (ATP) (54±3, 5±5, −20±5 and -54±3, 2±5, −20±5) were defined as the areas that showed significantly stronger activation to control faces than to noisy faces ($t_{(280)}>3.9$; $p_{unc}<5\times10^{-4}$) and they could be identified in 18 subjects. For the remaining two subjects, the group-average coordinates were taken from the random-effects group statistics (see Table 1 for coordinates). A time series of the mean voxel value within a 7-mm-radius sphere around the local peak of the areas of interest was calculated and extracted (MarsBaR 0.38) [21] from the other run to insure independence and the same GLM was applied to the data as used in the whole-brain analysis. Beta values were estimated for each ROI and observer to characterize the magnitude of the signal change. We performed a two-way repeated-measures ANOVA for each area with hemisphere (R vs. L) and condition (CF vs. NF) as within-subject factors. In case the assumption for homogeneity of variances was not met, values were first rank transformed before being entered into the statistical test, which is noted by rANOVA (rank ANOVA) when detailing statistical results. *Post hoc* t-tests were computed using Tukey honestly significant difference (HSD) tests.

IV. Results

A. Behavioral Results

Accuracy rate in the one-back task was 84.5 ± 11.5% (mean ± SD), which indicated that the participants attended to the stimuli during the scanning session as instructed.

In the psychophysics session face identity discrimination performance did not differ significantly between the control and noise conditions (no main effect of condition: $F_{(1,19)}=0.17$; $p=0.68$), showing that task difficulty was similar in the two conditions. Moreover, the amount of face-inversion effect (main effect of orientation: $F_{(1,19)}=92.4$; $p<10^{-8}$) - a reliable marker for face-specific processing - was also comparable for the two stimulus types, indicating that noisy faces were also discriminated and processed face-like as opposed to discrimination based on low-level stimulus features. Reaction times did not differ significantly in either of the cases (see Fig. 3).

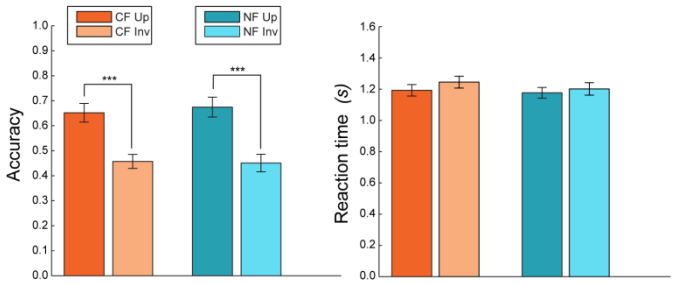


Figure 3. Behavioral results of a 4AFC identity discrimination experiment involving upright and inverted control (CF Up, CF Inv) and noisy faces (NF Up, NF Inv). The accuracy rate did not differ significantly between the control and noise conditions, indicating that task difficulty was equal in the two conditions. The robust face-inversion effect was also comparable for the two stimulus types, showing that noisy faces were also discriminated and processed face-like. Reaction times were similar in all conditions and did not differ statistically. Error bars indicate ± SEM (N=20, ***$p<0.0001$).

B. Results of the fMRI experiment

Whole-brain random-effects analysis of the fMRI data revealed that the presence of phase noise did not affect the fMRI responses in the inferior-occipital (OFA) and the fusiform (FFA) cortex. However, face-specific areas in the temporal lobe: bilateral superior temporal sulcus (STS), bilateral medial temporal lobe (MTL), and identity-sensitive patches in the right anterior temporal pole (ATP) showed markedly reduced fMRI activations in the noise compared to the control condition ($t_{(19)}>3.9$; $p_{unc}<5\times10^{-4}$) (see Fig. 4A and Table 1 for more details).

TABLE 1. Significant fMRI Clusters

MNI Coordinates	$t_{(19)}$ Value	Cluster Size	Area Label
26, -8, -18	5.85	156	Right MTL
-56, -48, 2	5.18	187	Left STS
-24, -8, -14	4.98	189	Left MTL
56, 2, -18	4.16	8	Right ATP
58, -36, 6	4.11	15	Right STS

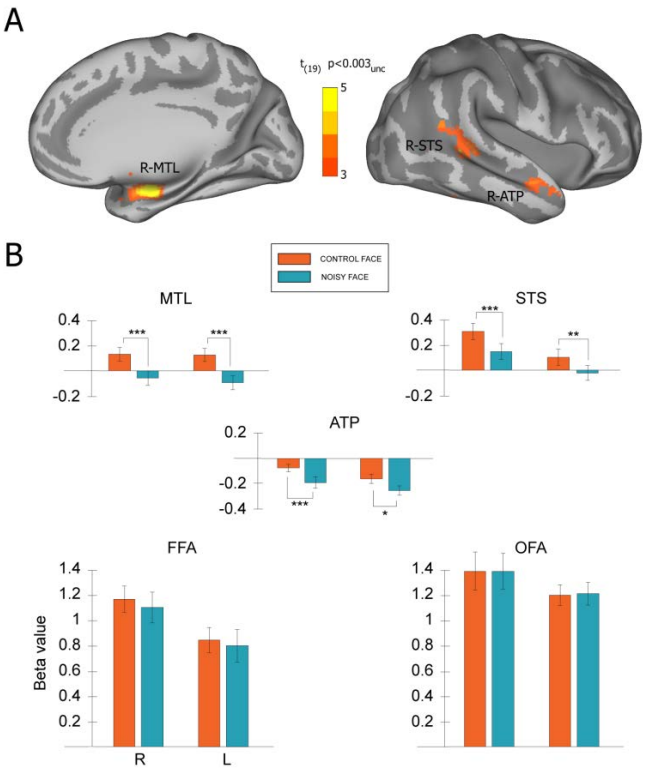


Figure 4. Results of the whole-brain and ROI analyses. (A) Areas which showed stronger fMRI responses to the control relative to the noise condition in the right temporal lobe. Maps are displayed with $p_{unc}<3\times10^{-3}$ on the PALS-B12 partially inflated brain [19]. (B) Based on an independent measurement, the ROI analysis, in agreement with the whole-brain analysis, showed significantly higher bilateral MTL, STS and ATP activation for control relative to noisy faces, while FFA and OFA did not show any significant modulation (***$p<10^{-4}$, **$p<10^{-3}$, *$p<10^{-2}$; R, right; L, left).

As previous individual brain analyses highlighted the large amount of interindividual variability in the location of the face-responsive areas in the ventral occipito-temporal cortex [1], we also performed an ROI-based analysis of the fMRI data in the face-selective regions (OFA, FFA, STS, MTL, and ATP) of the occipital and temporal lobes (for ROI definition, see Section III). In agreement with the results of the whole-brain random-effects analysis, the ROI analysis (Fig. 4B) suggested that the noise-induced modulation manifested itself only in the highest stage of face processing located downstream from the fusiform face area, namely in the perirhinal cortex and in the temporal pole. We found significantly higher activations in bilateral STS and bilateral MTL in the control relative to the noise condition (main effect of condition: $F_{(1,19)}=14.64$; $p<10^{-3}$ and $F_{(1,19)}=12.78$; $p<2\times10^{-3}$ for STS and MTL, respectively). In the case of ATP the effect of noise was more pronounced in the right hemisphere (condition×side interaction was marginally significant: $F_{(1,19)}=3.48$; $p<8\times10^{-2}$). However, the fMRI responses in the presence of phase noise were either not altered or only moderately reduced for face-specific areas in inferior-occipital (OFA) and fusiform (FFA) cortex, respectively (main effect of condition: $F_{(1,19)}=0.001$; $p=0.93$ and $F_{(1,19)}=1.02$; $p=0.32$ for OFA and FFA, respectively), which is in agreement with previous findings [15]. It was also found that the fMRI response was significantly higher in the right compared to the left hemisphere in the case of FFA, STS, and ATP, demonstrating the right-lateralization of face-processing (main effect of side: $F_{(1,19)}=1.96$; $p<2\times10^{-2}$, $F_{(1,19)}=14.70$; $p<10^{-3}$, and $F_{(1,19)}=6.33$; $p<2\times10^{-2}$ for FFA, STS, and ATP, respectively).

V. Conclusions

Our results revealed that face-specific responses in the medial and anterior temporal lobe are highly sensitive to the quality of structural/contour information of the face stimuli and thus provide evidence against an abstract, feature invariant representation of the facial identity information in these regions.

In contrast to this, fMRI responses to noisy images in the occipital face area and in the fusiform face area (OFA and FFA, two cortical regions involved in the structural processing of faces) were not altered or only moderately reduced, respectively. This excludes the explanation of reduced medial and anterior temporal fMRI responses in the case of noisy images based on a diminished input from OFA and FFA.

These findings also suggest that investigation of the perceptual deficits in the case of noisy (phase randomized) face stimuli might provide a unique opportunity to identify the specific role of medial and anterior temporal lobe in face processing.

Acknowledgment

This work was supported by grant from the Hungarian Scientific Research Fund to Z.V. (CNK80369).

References

[1] B. Rossion, B. Hanseeuw, and L. Dricot, "Defining face perception areas in the human brain: A large-scale factorial fMRI face localizer analysis," *Brain Cogn*, vol. 79, no. 2, pp. 138–157, Jul. 2012.

[2] D. Y. Tsao, S. Moeller, and W. A. Freiwald, "Comparing face patch systems in macaques and humans," *Proc Natl Acad Sci U S A*, vol. 105, no. 49, pp. 19514–19519, Dec. 2008.

[3] W. A. Freiwald and D. Y. Tsao, "Functional compartmentalization and viewpoint generalization within the macaque face-processing system," *Science*, vol. 330, no. 6005, pp. 845–851, Nov. 2010.

[4] E. B. O'Neil, A. D. Cate, and S. Köhler, "Perirhinal Cortex Contributes to Accuracy in Recognition Memory and Perceptual Discriminations," *J. Neurosci.*, vol. 29, no. 26, pp. 8329–8334, Jul. 2009.

[5] N. Kriegeskorte, E. Formisano, B. Sorger, and R. Goebel, "Individual Faces Elicit Distinct Response Patterns in Human Anterior Temporal Cortex," *PNAS*, vol. 104, no. 51, pp. 20600–20605, Dec. 2007.

[6] S. . Dakin, R. . Hess, T. Ledgeway, and R. . Achtman, "What causes non-monotonic tuning of fMRI response to noisy images?," *Current Biology*, vol. 12, no. 14, pp. R476–R477, Jul. 2002.

[7] G. A. Rousselet, C. R. Pernet, P. J. Bennett, and A. B. Sekuler, "Parametric study of EEG sensitivity to phase noise during face processing," *BMC Neurosci*, vol. 9, p. 98, Oct. 2008.

[8] C. A. Olman, K. Ugurbil, P. Schrater, and D. Kersten, "BOLD fMRI and psychophysical measurements of contrast response to broadband images," *Vision Research*, vol. 44, no. 7, pp. 669–683, Mar. 2004.

[9] B. S. Tjan, V. Lestou, and Z. Kourtzi, "Uncertainty and Invariance in the Human Visual Cortex," *Journal of Neurophysiology*, May 2006.

[10] H. E. Schendan and C. E. Stern, "Mental rotation and object categorization share a common network of prefrontal and dorsal and ventral regions of posterior cortex," *NeuroImage*, vol. 35, no. 3, pp. 1264–1277, Apr. 2007.

[11] G. Rainer, M. Augath, T. Trinath, and N. K. Logothetis, "Nonmonotonic noise tuning of BOLD fMRI signal to natural images in the visual cortex of the anesthetized monkey," *Curr. Biol.*, vol. 11, no. 11, pp. 846–854, Jun. 2001.

[12] G. Rainer, M. Augath, T. Trinath, and N. K. Logothetis, "The effect of image scrambling on visual cortical BOLD activity in the anesthetized monkey," *Neuroimage*, vol. 16, no. 3 Pt 1, pp. 607–616, Jul. 2002.

[13] B. Rossion, C. A. Joyce, G. W. Cottrell, and M. J. Tarr, "Early lateralization and orientation tuning for face, word, and object processing in the visual cortex," *NeuroImage*, vol. 20, no. 3, pp. 1609–1624, Nov. 2003.

[14] R. J. Itier and M. J. Taylor, "Inversion and contrast polarity reversal affect both encoding and recognition processes of unfamiliar faces: a repetition study using ERPs," *Neuroimage*, vol. 15, no. 2, pp. 353–372, Feb. 2002.

[15] É. M. Bankó, V. Gál, J. Körtvélyes, G. Kovács, and Z. Vidnyánszky, "Dissociating the Effect of Noise on Sensory Processing and Overall Decision Difficulty," *J. Neurosci.*, vol. 31, no. 7, pp. 2663–2674, Feb. 2011.

[16] J.-M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The Amsterdam Library of Object Images," *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 103–112, Jan. 2005.

[17] D. H. Brainard, "The Psychophysics Toolbox," *Spat Vis*, vol. 10, no. 4, pp. 433–436, 1997.

[18] D. G. Pelli, "The VideoToolbox software for visual psychophysics: transforming numbers into movies," *Spat Vis*, vol. 10, no. 4, pp. 437–442, 1997.

[19] D. C. Van Essen, "A Population-Average, Landmark- and Surface-based (PALS) atlas of human cerebral cortex," *NeuroImage*, vol. 28, no. 3, pp. 635–662, Nov. 2005.

[20] D. C. Van Essen, H. A. Drury, J. Dickson, J. Harwell, D. Hanlon, and C. H. Anderson, "An integrated software suite for surface-based analyses of cerebral cortex," *J Am Med Inform Assoc*, vol. 8, no. 5, pp. 443–459, Oct. 2001.

[21] Brett, J. L. Anton, R. Valabregue, and J. B. Poline, "Region of interest analysis using an SPM toolbox," *NeuroImage*, vol. 16, no. 2, p. 497, 2002.

Filtration of intravenous cardiopulmonary parasitic Nematodes using a cross-flow microfluidic Separator

András Laki
(Supervisors: Kristóf Iván Ph.D., Pierluigi Civera Ph.D.)
lakanjo@digitus.itk.ppke.hu

***Abstract*—A cross-flow microseparator device had been designed and fabricated to filtrate larvae of Dirofilaria species from intravenous blood samples applying monolithic polydimethylsiloxane (PDMS) microfluidic structures. The fabrication of our microfluidic device is based on soft-lithography techniques applying SU-8 epoxy based photoresist as molding replica. Since the optical refractive index of PDMS is close to that of glass a fast camera system, which is called Cellular Nonlinear Network (CNN) based camera, can be integrated into the device to count the number of nematodes population within liquid flow [1], [2].**

I. Introduction

A cross-flow microseparator device had been designed and fabricated to filtrate larvae of Dirofilaria species from intravenous blood samples applying monolithic polydimethylsiloxane (PDMS) microfluidic structures. Number of veterinarian dirofilariasis, which is caused by Dirofilaria (Nochtiella) repens and Dirofilaria immitis, increases significantly also in Europe. Every fifth pet contracts dirofilariasis in the Mediterranean region, Southern and Eastern Europe, North Africa and Asia. The vast majority of zoonotic filariae, which are commonly found in the subcutaneous tissues, pulmonary arteries and also within heart, can infect also humans. The advantage of a nematode-filtering device can be reached with collecting larvae from blood-stream without using parasiticides or anthelmintics. Additionally, due to the biocompatibility of the chosen materials and the shear-stress based separation a significant percent of blood sample can be recyclable without any kind of biological modification. Several diagnostic methods have been developed to explore the existence of nematodes or to determine the number of volumetric parasitic population from serological samples.

II. Types of microfluidic filtration

Our main goal of device construction was to design an effective microfilter that separates nematodes from the blood stream. The microfluidic separation and filtration is a common module of the Lab-On-a-Chip (LOC) devices. Due to the low Reynolds number within microchannels the blood stream is mostly laminar consequently an internal or external force is mandatory for particle filtration. First of all I would like classify the main filtration techniques with the applied force. The most common used separations are based on the application of shear-stress flows via hydrodynamic [3]–[5], electrokinetic [6], [7], magnetic [8], [9], acoustic [10], [11] or optical forces [12].

In our application the nematodes separation from the blood stream was solved by passive filtration that does not requires external forces. Form this point of view our decision was to use the shear-stress based filtration. Four main types can be classified by the following way: using membrane filters [13]–[16], weir-type [17], pillar-type [18], or cross-flow microstructures [19]–[21].

The following part shorty introduces the nowadays research on the field of microfiltration. The work of D. Choudhury et. al. represent a design and fabrication of a robust and scalable device capable of separating a heterogeneous population of cells with variable degree of deformability into enriched populations with deformability above a certain threshold. Their method is based on the dissimilar cytoskeletal architecture in diverse cell types induces a difference in their deformability that presents a viable approach to separate cells in a non-invasive manner. Using flow rates as large as $167\mu L/min$, throughputs of up to $2800 cells/min$ were achieved at the device output. A fluorescence-activated cell sorting (FACS) viability analysis on the cells revealed 81% of the population maintain cellular integrity after passage through the device. [22]

The research work of A. D. Browne et al. represents a new lab-on-a-chip for rapid analysis of low volume blood samples was designed, fabricated and demonstrated for integration of serum separation, hematocrit evaluation, and protein quantitation. Blood separation was achieved using microchannel flow-based separation. A novel method for evaluating hematocrit from microfluidic flow-separated blood samples was developed using gray scale analysis of a point-and-shoot digital photograph of separated blood in a microchannel. Protein quantitation was subsequently performed in a high surface area-to-volume ratio microfluidic chemiluminescent immunoassay using cell depleted serum produced by microfluidic flow-based separation of whole blood samples. All three steps were achieved in a single microchannel with separation of blood samples and hematocrit evaluation in less than 1 min, and protein quantitation in 5 min. [5]

Stefan H. Holm et. al represented a device that separates parasites from human blood using Deterministic Lateral Displacement (DLD). DLD-based particle separation technique,

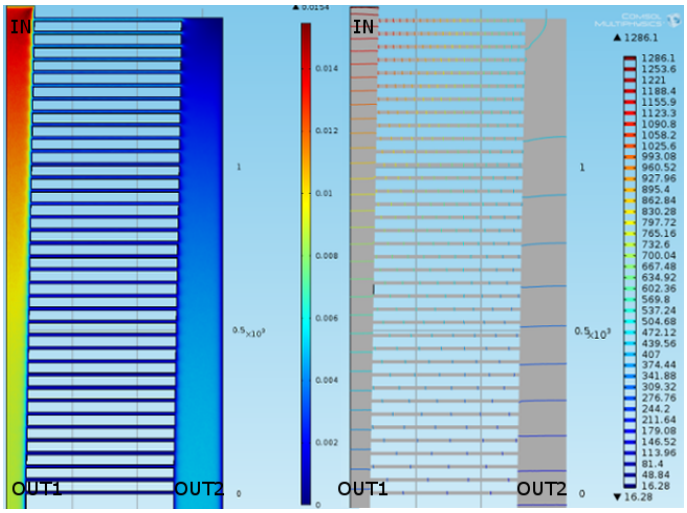


Fig. 1. Finite element simulations from the designed channel by Comsol Multiphysics. On the left side the flow velocity profile is represented inside a section of our microchannel in the range of $0m/s$ to $0.016m/s$, meanwhile on the right side the pressure profile is shown in the range of $16Pa$ to $1286Pa$.

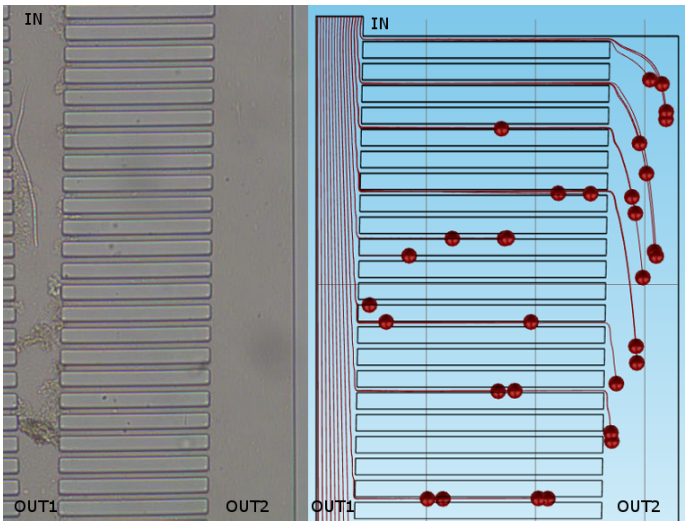


Fig. 2. Comparison between the finite element simulation and the validation test. On the left side the microfluidic nematodes filter is tested with hemolyzed intravenous blood that contains parasitic larvae. The membrane of RBCs could enter into the OUT2, but nematodes. On the right side finite element simulation represents the RBCs flow thought the lateral microcapillaries that was validated with experimental measurements.

which is easy to implement in cheap disposable plastic chips, was used to remove parasites from blood cells. The interaction of particles suspended, which is our case the whole blood consisting parasites, in a fluid with an ordered array of micropillars that the fluid is forced to flow through under low Reynolds number conditions. The array of circular posts divides the fluid into many narrow streams, the widths of which correspond to the cell diameter. If the particles are smaller than the critical size they are able to follow one such stream through the array whereas larger particles are forced. If the particles are not able to enter the collum system these particles change streams many times, always in the same direction, becoming laterally displaced [23].

After the comparison between the different methods we decided to create a cross-flow microfluidic filter to separates nematodes from the blood stream continuously. From the cental inlet channel the microcapillaries filter laterally to the collector channels that is described and represented in the following sections.

III. Finite element simulations

The Computational Fluid Dynamics (CFD) simulators calculate the flow models with solving the Navier-Stokes equations numerically. Generally the CFD solvers contain three logical steps starting with the refinement of mesh to the actual geometrical problem, continue with numerical solving and ends with postprocessing for the geometrical representation. The importance of the initial parameters are significant due to the computational stability and the computational time. The precision of the solution is meanly depends on the initial parameters and the boundary conditions.

In the first step the dimensions and the initial parameters have to be fixed such as the blood viscosity, which is $3.53 \cdot 10^{-3}$Pa s, the density of blood, which is $1060kg/m3$, the velocity of the inlet flow, which is $0.01m/s$, the pressure difference from the inlet on the outlets, which is $0Pa$ and also the cell diameter that is $5\mu m$.

The mesh size and the resolution of mesh is important in solving the partiality differential equations due the stability criterions and the computational time. The minimum width of our microchannels are just $10\mu m$, thus the maximum element size has to be around few microns. In our simulations a free tetrahedral mesh was built that was generated by the integrated mesh module.

Previous the construction finite element simulations had been made to rise the efficiency of the hydrodynamic separator. The central inlet (IN) of our cross-flow microfluidic separator is a rectangular cross-section microchannel, which tapers from $300\mu m$ width end to $20\mu m$ (OUT1) with the same $20\mu m$ depth. During the geometrical thinning of central inlet channel the hydrodynamic pressure rises that helps the biological sample entering and flowing through the $10\mu m$ wide and $20\mu m$ deep lateral microcapillaries into the collector outlet (OUT2). The efficiency of filtration is doubled by the symmetrical arrangement, thus in this case from the central inlet channel the biological sample can be filtered into the right (OUT2a) and the left (OUT2b) side collector outlets by perpendicular lateral microcapillaries. During the geometrical optimization the efficiency of microfluidic filter is the most important parameter that mainly depends on the width and the length of the perpendicular lateral microcapillaries and the width of the central outlet channel (OUT1) of our device.

Due to decrease the numerical calculation of CFD simulator the geometric symmetries were used thus the figure 1 demonstrates only the half part of the flow velocity profile and the pressure profile of blood stream. The calculated flow

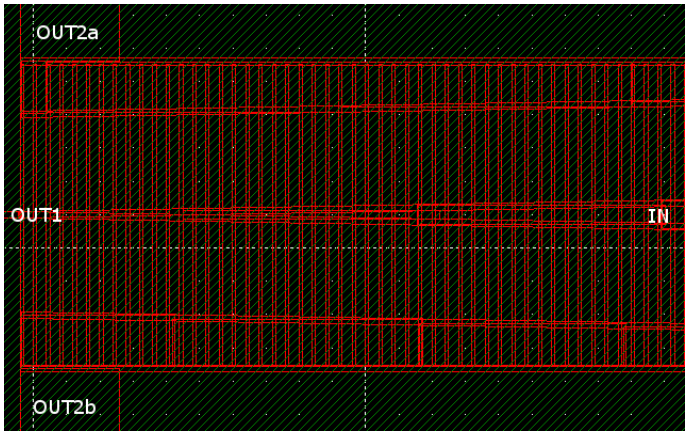


Fig. 3. A part of our chrome mask design, which is the terminating part of our microfluidic filter, is represented. IN is the continuation of central inlet channel, OUT2a is the output of the right collector channel, meanwhile OUT1b is the left one. OUT1 is the central outlet that contains the filtered parasites.

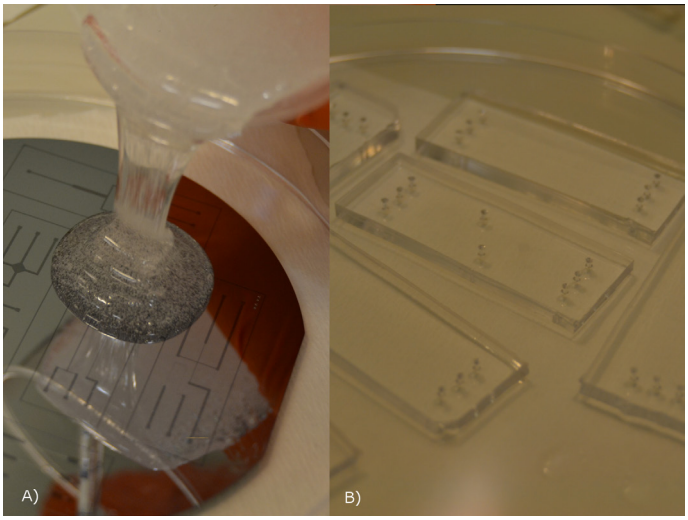


Fig. 4. PDMS layer preparation. A) Microfluidic mold was covered by prepared liquid PDMS layer with a thickness of $15mm$. B) Microfluidic device is lifted off and punched through in the inlet and outlet positions.

velocity profile is in the range of $0m/s$ to $0.016m/s$ that was acceptable also for the experimental measurements. After the $20\mu m$ thin output channel the optical detection was executed in a $100\mu m$ wide channel where the flow velocity was around $0.003m/s$ that was close to the approximated values. On figure 1 the pressure profile is also represented in the range of $16Pa$ to $1286Pa$.

Our devices was also tested by particle tracing module of Comsol Multiphysics that requires time-dependent solver. During the simulations the particle trajectories were calculated in the time range of $0s$ to $0.3s$ with time step of $0.001s$. The result is represented on figure 2 that also gives a comparison with the experimental test. The microfluidic filter was tested by hemolyzed blood sample containing nematodes. The membrane of the red blood cells (RBC) can flow thought the lateral capillaries, meanwhile the larvae remain inside the central channel. The efficiency of the filter has not been measured jet, and also the tests without hemolysis have not been done jet.

IV. Device fabrication

A low-cost, transparent, formable, bio-compatible organic elastomer, which is called polydimethylsiloxane (PDMS), was chosen to construct our system that is a common used material for Biomedical Micro-Electro-Mechanical Systems (BioMEMS) applications. Another propriety of PDMS, which lets the elastomer bind to glass surface, is significant due to corona discharge treatment ruptures the ligaments on the surface of the PDMS while PDMS changes from hydrophobic to hydrophilic. The following enumeration summarizes the main soft lithography steps that were used to construct our device.

Chrome mask design and fabrication: First of all the master mask, which is a fabricated chrome-covered glass slide, is required to make low cost replica. The precision of our microfluidic devices mainly depends on the resolution of this chrome master mask, the used wavelength of lithography and the type of applied photoresist. Figure 3 represents a segment of the designed chrome mask that has a resolution in space of $5\mu m$.

Fabrication of soft lithography molds: The PDMS layer was fabricated using a specific mold, which was produced on silicon wafer, via standard photolithography techniques in the MEMS Laboratory at Research Center for Natural Sciences of Budapest. To fabricate the microfluidic mold, SU8 photoresist was spincoated on a silicon wafer at a thickness of $20\mu m$ and baked at $95\,^{\circ}\mathrm{C}$ for 5 min. The photoresist on the silicon wafer was exposed under UV light through the designed chrome mask after washed the non-polymerized part.

Fabrication of PDMS layer: The PDMS, which belongs to a group of polymeric organosilicon compounds that are commonly referred to as silicones, samples were prepared by mixing the liquid pre-polymer and the curing agent in a ratio of 10:1. After mixing the two components the material is treated by 20 minutes of vacuum at 0.01 MPa. In the next step as it is shown in figure 4, the microfluidic mold was covered by this prepared liquid PDMS layer with a thickness of $15mm$. The polymerization time of the PDMS material at room temperature is around 48 h. Heating the wafer with the covering PDMS decreases the polymerization time but also can cause diameter variances. In order to avoid this effect the PDMS was polymerized at room temperature.

Corona discharge-treatment: When the polymerization is terminated as it is shown in figure 4, the microfluidic device is lifted off and punched through in the positions of inlet and outlets. The surface of PDMS was cleaned by alcohol and treated by 10 minutes of oxygen plasma. The covering glass slide is also treated by 10 minutes of oxygen plasma and the PDMS layer is aligned above the glass slide.

The fabricated microfluidic separator, which is shown in figure 5, is fabricated by PDMS-glass technique, thus the

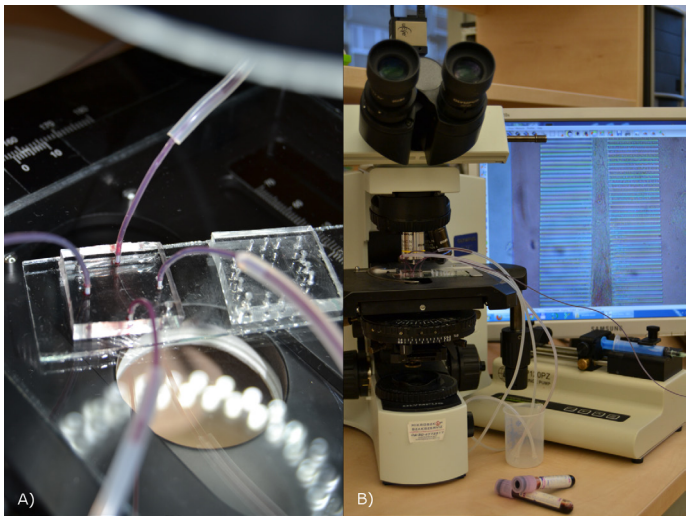


Fig. 5. The developed system. A) The fabricated microfilter during intravenous blood test. B) Components of developed system: syringe pump, microfluidic device with tubing, microscope and the workstation.

developed microfluidic chip is biocompatible, low cost and transparent. Since the optical refractive index of PDMS is close to that of the glass, a fast camera system, called Cellular Nonlinear Network (CNN)-based camera, can be integrated and used for example to count the number of nematodes inside the liquid flow.

V. Conclusion

A cross-flow microseparator device had been designed and fabricated to filtrate larvae of Dirofilaria species. Previously finite element simulations were calculate to increase the efficiency of microfilter by a Computational Fluid Dynamics (CFD) simulator. After the result of the optimization of flow velocity profile and the pressure gradient profile particle tracing simulation were calculated that were tested by experimental way.

VI. Acknowledgment

I would like to acknowledge my supervisor, Kristóf Iván for his kind help and his knowledge on this complex multidisciplinary field. Aslo I would like to thank Olga Jacsó for the biological samples and her kind help. I acknowledge Péter Fürjes and Zoltán Fekete for the fabrication of our devices and their kind help. I would like to kindly acknowledge Gergő Huszka, Tamás Pardy, Nóra Markia, Eszter Tóth, András Meleg and Péter Hajdu for their kind help. Finally I would like to thank my family and my friend for their love and help. Also, the support of TAMOP-4.2.1./B-10 and TAMOP-4.2.1./B-11 is kindly acknowledged.

References

[1] A. Laki, I. Rattalino, A. Sanginario, N. Piacentini, K. Ivan, D. Lapadatu, J. Taylor, D. Demarchi and P. Civera, "An integrated and mixed technology LOC hydrodynamic focuser for cell counting application," Biomedical Circuits and Systems Conference (BioCAS), 2010 IEEE, pp.74-77, 2010.

[2] A. Laki, I. Rattalino, F. Corinto, K. Ivan, D. Demarchi, and P. Civera, "An integrated LOC hydrodynamic focuser with a CNN-based camera system for cell counting application," in 2011 IEEE Biomedical Circuits and Systems Conference (BioCAS). IEEE, Nov. 2011, pp. 301-304.

[3] A. A. S. Bhagat, H. W. Hou, L. D. Li, C. T. Lim, and J. Han, "Pinched flow coupled shearmodulated inertial microfluidics for high-throughput rare blood cell separation," Lab Chip, vol. 11, no. 11, pp. 1870-1878, Jun. 2011.

[4] S. Choi, T. Ku, S. Song, C. Choi, and J. Park, "Hydrophoretic high-throughput selection of platelets in physiological shear-stress range," Lab Chip, vol. 11, no. 3, pp. 413-418, Feb. 2011.

[5] A. W. Browne, L. Ramasamy, T. P. Cripe, and C. H. Ahn, "A lab-on-a-chip for rapid blood separation and quantification of hematocrit and serum analytes," Lab Chip, vol. 11, no. 14, pp. 2440-2446, Jun. 2011.

[6] A. Valero, T. Braschler, A. Rauch, N. Demierre, Y. Barral, and P. Renaud, "Tracking and synchronization of the yeast cell cycle using dielectrophoretic opacity," Lab Chip, vol. 11, no. 10, pp. 1754-1760, May 2011.

[7] M. J. Hilhorst, G. W. Somsen, and G. J. de Jong, "Capillary electrokinetic separation techniques for profiling of drugs and related products," Electrophoresis, vol. 22, no. 12, pp. 2542-2564, Jul. 2001.

[8] A. I. Rodrguez-Villarreal, M. D. Tarn, L. A. Madden, J. B. Lutz, J. Greenman, J. Samitier, and N. Pamme, "Flow focussing of particles and cells based on their intrinsic properties using a simple diamagnetic repulsion setup," Lab Chip, vol. 11, no. 7, pp. 1240-1248, Apr. 2011.

[9] J. H. Kang, S. Choi, W. Lee, and J. Park, "Isomagnetophoresis to discriminate subtle difference in magnetic susceptibility," J. Am. Chem. Soc., vol. 130, no. 2, pp. 396-397, 2007.

[10] J. Nam, H. Lim, D. Kim, and S. Shin, "Separation of platelets from whole blood using standing surface acoustic waves in a microchannel," Lab Chip, vol. 11, no. 19, pp. 3361- 3364, Sep. 2011.

[11] T. Laurell, F. Petersson, and A. Nilsson, "Chip integrated strategies for acoustic separation and manipulation of cells and particles," Chem. Soc. Rev., vol. 36, no. 3, pp. 492-506, Feb. 2007.

[12] K. H. Lee, S. B. Kim, K. S. Lee, and H. J. Sung, "Enhancement by optical force of separation in pinched ow fractionation," Lab Chip, vol. 11, no. 2, pp. 354-357, Jan. 2011.

[13] H. Wei, B.-h. Chueh, H. Wu, E. W. Hall, C.-w. Li, R. Schirhagl, J. Lin, and R. N. Zare, *Particle sorting using a porous membrane in a microfluidic device*," Lab Chip, vol. 11, no. 2, pp. 238-245, Jan. 2011.

[14] R. Schirhagl, I. Fuereder, E. W. Hall, B. C. Medeiros, and R. N. Zare, "Microfluidic purification and analysis of hematopoietic stem cells from bone marrow," Lab Chip, vol. 11, no. 18, pp. 3130-3135, Sep. 2011.

[15] K. Aran, A. Fok, L. A. Sasso, N. Kamdar, Y. Guan, Q. Sun, A. ndar, and J. D. Zahn, "Microfiltration platform for continuous blood plasma protein extraction from whole blood during cardiac surgery," Lab Chip, vol. 11, no. 17, pp. 2858-2868, Aug. 2011.

[16] Y. Luo and R. N. Zare, "Perforated membrane method for fabricating three-dimensional polydimethylsiloxane micro uidic devices," Lab Chip, vol. 8, no. 10, pp. 1688-1694, Sep. 2008.

[17] V. VanDelinder and A. Groisman, "Separation of plasma from whole human blood in a continuous Cross-Flow in a molded micro uidic device," Anal. Chem., vol. 78, no. 11, pp. 3765-3771, 2006.

[18] D. R. Gossett, W. M. Weaver, A. J. Mach, S. C. Hur, H. T. K. Tse, W. Lee, H. Amini, and D. Di Carlo, "Label-free cell separation and sorting in microfluidic systems," Analytical and Bioanalytical Chemistry, vol. 397, no. 8, pp. 3249-3267, Aug. 2010.

[19] H. M. Ji, V. Samper, Y. Chen, C. K. Heng, T. M. Lim, and L. Yobas, "Silicon-based microfilters for whole blood cell separation," Biomedical Microdevices, vol. 10, no. 2, pp. 251-257, Oct. 2007.

[20] S. K. Murthy, P. Sethu, G. Vunjak-Novakovic, M. Toner, and M. Radisic, "Size-based microfluidic enrichment of neonatal rat cardiac cell populations," Biomedical Microdevices, vol. 8, no. 3, pp. 231-237, Sep. 2006.

[21] T. Morijiri, S. Sunahiro, M. Senaha, M. Yamada, and M. Seki, "Sedimentation pinched-flow fractionation for size- and density-based particle sorting in microchannels," Microfluidics and Nanofluidics, vol. 11, no. 1, pp. 105-110, Mar. 2011.

[22] D. Choudhury, W. T. Ramsay, R. Kiss, N. A. Willoughby, L. Paterson, and A. K. Kar, "A 3D mammalian cell separator biochip," Lab on a Chip, vol. 12, no. 5, pp. 948-953, Feb. 2012.

[23] . S. H. Holm, J. P. Beech, M. P. Barrett, and J. O. Tegenfeldt, "Separation of parasites from human blood using deterministic lateral displacement," Lab Chip, vol. 11, no. 7, pp. 1326-1332, Apr. 2011.

Analysis of protein folding and binding by simplified models

Dániel Györffy
(Supervisors: Dr. Péter Závodszky, Dr. András Szilágyi)
gydlacf@enzim.hu

***Abstract*—Proteins can undergo a remarkable conformational change when they bind to a partner. We investigated this conformational rearrangement upon binding by analyzing a Markov chain based on the 2D HP (hydrophobic-polar) lattice model of proteins and an advanced variant of the set of pull moves. A two-layer Markov model is developed to allow us to investigate dimers of chains of up to 8 beads. We show that the folding and binding of many sequences occurs asymmetrically, involving an asymmetric complex with one chain in the native conformation and the other in a different conformation.**

***Index Terms*—protein, folding, binding, HP model, Markov chain**

I. Introduction

Proteins often carry out their function as a part of a complex. They can bind other macromolecules such as proteins or nucleic acids. Smaller species can also bind to proteins, such as metal ions or small organic molecules, e. g. hormones. Binding is often accompanied by a conformational change of proteins ranging from side chain fluctuations through backbone motions to the partial or total folding/refolding of the molecule. This *coupled folding and binding* process is described by two different models in the literature. According to the *induced fit* model [2], the non-specific binding of the ligand can trigger a conformational change in protein, leading to the eventual formation of the bound conformation. The *conformational selection* model [3] predicts that a pre-existing conformational ensemble containing the bound conformation is present even in the unbound form.

In recent years, a group of proteins called intrinsically disordered proteins (IDPs, also called intrinsically unstructured, natively unstructured proteins etc.) has been intensively studied [4], [5]. IDPs, as free monomers, lack a well-defined spatial structure. However, when they bind to some target molecule or ion, a disorder-to-order transition can be observed [6], [7]. When the binding partner is relatively rigid, the binding process can be easily described by one of the above-mentioned models because the protein does not trigger a significant conformational change in the target. There is some difficulty, however, when the target also has a remarkable flexibility.

Our reseach focuses on homodimers, dimers of chains having the same sequence. This implies that there is an intrinsic symmetry in the system. The aim of our research is to reveal the mechanism of dimerization of homodimers, particularly to find out whether the symmetry in the sequences manifests itself in the dimerization mechanism, or the dimerization occurs asymmetrically. We also investigated the effect of the stability of the native conformation on the dimerization the mechanism.

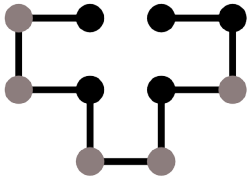


Fig. 1. The ground state conformation of the sequence **HHPHPPHPPH**.

II. Methods

A. 2D HP lattice model

In a 2D lattice model, proteins are modeled as self-avoiding walks on a square lattice (Figure 1), with beads on the lattice points. To be able to study the effect of the sequence on the folding and binding process in a simple way, we define two kinds of beads: a hydrophobic one – denoted as H – and a hydrophilic or polar one – denoted as P [8]. The systems we investigated consist of two chains in a square box where a periodic boundary condition is applied. A state of the system is defined by the conformations of the two chains and their relative positions and orientations. The energy of any state is calculated using an energy function. Two different energy functions were used. In the *distance based energy function*, each pair of hydrophobic beads not adjacent along the sequence contributes to the energy by a negative term proportional to the inverse square of the distance between the beads. In the *adjacency based energy function*, only the pairs of hydophobic beads not adjacent along the sequence but adjacent in space contribute to the energy by a non-zero term.

B. Pull moves

Proteins are dynamic systems so it is plausible to model them by a dynamic model. Dynamics is introduced by applying some moves from a predefined pool, called *move set*. Several move sets have been described in the literature. A good move set must fulfill three criteria: locality, ergodicity and reversibility. Locality means that a move only influences few beads and only causes a small displacement. Ergocity means that every conformation is reachable from any other by

a sequence of moves from the move set. In reversible move sets, if one conformation can be reached from another by one move then the other also can be reached by one move. Lesh and coworkers introduced a new move set, claiming that it meets all the three criteria [9] but we showed that it is actually irreversible. We used a fixed version of pull moves which is now reversible [2].

C. Exhaustive enumeration

The thermodynamics of a system can be characterized exactly if the density of states is known. In order to obtain the energy of each state, exhastive enumeration should be carried out. During exhaustive enumeration, all states are visited systematically and the energies of them are calculated. Obviously, this method can only be applied on systems of moderate size.

D. Markov chains

The state space of our two-chain systems can be modelled as a *Markovian kinetic scheme*, represented by a graph with nodes corresponding to states and edges corresponding to transitions between states [10], [11]. We applied, as transitions, pull moves to change the conformations, and translation and rotation to move the chains as rigid bodies. The weights of the edges are determined by the

$$p\left(\mathrm{A}\rightarrow\mathrm{B}\right)=\min\left(1,\frac{p_{ap}\left(\mathrm{B}\rightarrow\mathrm{A}\right)}{p_{ap}\left(\mathrm{A}\rightarrow\mathrm{B}\right)}\cdot\mathrm{e}^{-\Delta E/k_BT}\right)$$

Metropolis – Hastings criterion [12], where $p_{ap}\left(\mathrm{A}\rightarrow\mathrm{B}\right)$ and $p_{ap}\left(\mathrm{B}\rightarrow\mathrm{A}\right)$ are the a probability of getting to state B from state A and to state A from state B, respectively. The whole Markov model can be built by enumerating all states and all transitions between them.

III. Results

A. Probability of the native conformation

We investigated those $L=10$ sequences where the ground state of the dimer is unique. For these sequences, however, the monomers usually have multiple ground state conformations. The question we may ask here is whether the conformation present in the native dimer is the most probable in the presence of the other chain.

For simplicity, let us introduce some notations. Let N_D denote the conformation of a chain in the native dimer and $\{N_M\}$ the set of ground conformations in the absence of the other chain. Let $D=\{d_i\}$ be the set of sequences where $N_D\in\{N_M\}$ and $M=\{m_j\}$ the set of sequences where $N_D\notin\{N_M\}$.

Enumerating states where at least one of the chains is in N_D or some N_M conformation, the probabilities of these conformations can be compared. It is obvious that if only one chain is present, the probabilities of all N_M conformations are identical (because they have the same energy). For $m_j\in M$ sequences, the probability of N_D is lower than that of N_Ms, which follows from the Boltzmann distribution.

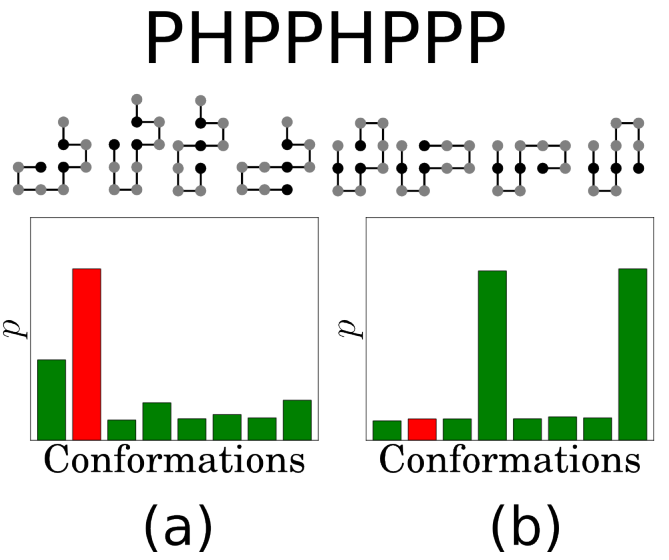


Fig. 2. The probabilities of N_M conformations calculated by exhaustive enumeration of states for the sequence **PHPPHPPPPH**. The probability of the N_D conformation is shown in red. The corresponding conformations are also shown. (a) There is no constraint on the chain conformations. (b) At least one of the chains is in N_D conformation.

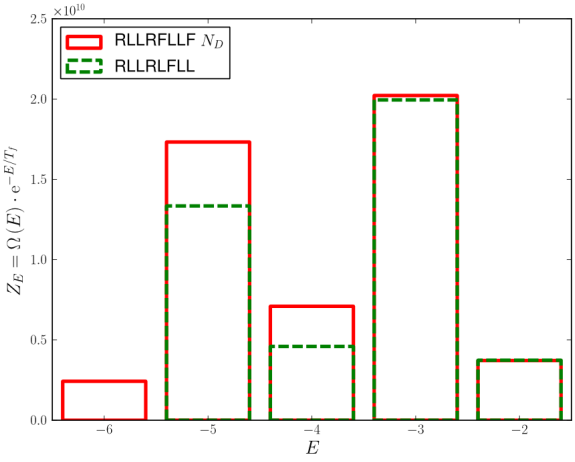


Fig. 3. Deviations of the partition functions calculated for particular energy levels for the N_D and the N_M conformation having the second highest probability. It can be seen that mainly the lowest energy levels are responsible for the differences. For the meaning of strings in the key, see Appendix A.

If, however, another chain is present then the underlying free energy surface changes, so the probabilities of the N_M conformations can differ. Because the energy of intra-chain contacts is unchanged, the interactions between the chains are responsible for the inequality.

The temperature was set to be the folding tempreture, the value where the probability of the native state is $1/2$, and the *adjacency based energy function* was used. For every investigated $d_i\in D$ sequences, the N_D conformation is more probable than any other N_M conformation (Figure 2/a) when another chain is present. If we investigate the partition

functions for any energy levels, we see that mainly the low-energy states of the two-chain system are responsible for the increased probability of the N_D conformation (Figure 3). If, however, one chain is in N_D conformation, the other chain is more likely to be in some $N_M \neq N_D$ conformation, rather than in the N_D (Figure 2/b).

For $m_j \in M$ sequences, although it was observed that the probability of N_D relative to N_Ms increased in the presence of the other chain, but not sufficiently to exceed them. It should be noted that the native state was excluded from these probability calculations.

B. Two-layer Markov model

The state space of a dimeric system of even relatively short chains is huge. Two chains of 8 beads have 20331 possible non-equivalent conformation pairs if the sequence is palindromic and 73984 if it is not. If we consider all possible positions in a square box of size $(2 * L)^2$ where L is the chain length, and four possible orientations for each chain then the number of all states turns out to be on the order of 10^7 (not considering the excluded volume).

For characterizing the dynamics of the whole system, we need to know all the motions transferring the system from one state to another. From a particular conformation, $2 \cdot 7 + 4 \cdot (L-2)$ moves can possibly originate which is 38 for $L = 8$. Even if only half of the moves can actually be carried out due to the excluded volume constraint, almost $800,000,000$ possibilities should be considered. If translation and rotation are also allowed, an additional factor of 8 (four possible translations and four possible rotations) should be applied. The size of this Markov model exceeds the memory capacity of our computer clusters.

To significantly reduce the size of the state space and thus also the number of possible transitions, we developed a two-layer Markov model. The model is based on the assumption that diffusional motions (translation and rotation) are much faster (occur on a much shorter time scale) than conformational changes. The model contains two distinct states for every non-equivalent conformation pair: an associated one, which corresponds to the minimum energy state for the given conformation pair, and a dissociated one where the interaction energy between the chains is assumed to be 0. Thus, the model is based on the assumption that translation and rotation are so fast that any associated states not corresponding to the minimum energy state of the given conformation pair will quickly move to the minimum energy state before the conformation of any of the chains could change.

We may visualize our Markov model as having two layers: associated states are on the bottom level and the corresponding dissociated states are on the top level. For each conformation pair, the associated and dissociated states are connected. Different conformation pairs related to each other by a pull move will be connected either on the bottom layer or the top layer depending on whether the move can be performed without dissociating the associated state. The first case corresponds to a conformational change without chain dissociation, while the second corresponds to a dissociation–conformational change–association process.

The Markov chain obtained by the above mentioned method and thus the dynamics of the system then can be treated analytically.

C. Folding before binding or binding before folding

A crucial property of the dimerization process is whether the binding occurs between already folded chains or the binding precedes the folding. The disorder-to-order transition of IDPs is a typical example where folding follows the binding of a partner.

The two-layer model offers an opportunity to study the behaviour of different sequences. If the folding occurs earlier than the binding then the largest flux to the node corresponding to the native state goes through the dissociated state of the two chains in their native conformations. If the largest flux leads through an associated state then the process can be considered *binding before folding*.

The Markov chain defines a time independent $\mathbf{T}$ transition matrix, where $T_{i,j}$ is the conditional probability that the system is in state j at time $t+1$ provided that the system was in state i at time t. Let $p(t) = (p_1, p_2 \dots p_n)$ denote the row vector of probabilities that the system is in state p_i at time t. We obtain the probabilities $p(t+1)$ at time $t+1$, if we multiply $p(t)$ by $\mathbf{T}$:

$$p(t+1) = p(t)\,\mathbf{T}$$

and from the properties of Markov chains

$$p(t+k) = p(t)\,\mathbf{T}^k.$$

Using this matrix operation, we monitor the changes of probabilities of particular states as time progresses.

In Figure 4, the time dependence of probabilities for the states connected with the native node summed over the associated and dissociated states are shown for sequences with different behaviours.

IV. Discussion

In biological recognition processes, macromolecules, mainly proteins, are usually invoked. Two types of the process are distinguished according to whether the recognition occurs between preformed, rigid surfaces or the binding of the partner itself induces the surface formation [3], [13], [14]. IDPs, the group of proteins of intensive recent interest are good examples for folding coupled with binding to a partner [7]. Being disordered in the unbound state has some advantages for the kinetics of complex formation [15], [16].

Some systematic studies have been carried out in order to reveal the distinctive properties of protein dimers belonging to different kinetic types of dimer formation [17]–[19]. In addition to the two extremes of rigid docking between already folded, stable monomers and the binding-induced folding of IDPs, some intermediate types also exist, where for example a dimeric intermediate can be observed [17].

Our studies on the highly simplified protein model proposed by Lau and Dill [8] revealed that dimeric intermediates play a

crucial role in dimer formation of a particular group of model sequences. If the conformation of the chain in the native dimer (N_D) is also a ground state conformation of the monomer then the presence of the other chain discriminatively stabilizes the dimeric native conformation, suppressing the other monomer native conformations (Figure 2/a). This phenomenon is attributable to the inter-chain contacts (Figure 3). In our kinetic simulations, the accumulation of states where just one of the chains is in N_D was also observed (results not presented here).

Furthermore, we found that if one state has already reached its N_D conformation then the N_D conformation of the other chain is only stable in the native dimer. This suggests a two-stage model where conformational selection [3] occurs in the first stage, with the N_D conformation getting selected by the presence of the other chain, and this is followed in the second stage by an induced folding of the other chain.

The dynamics of protein systems can be well studied by analyzing Markov chains [10], [20]. Exact calculations can be carried out for e.g. the folding time of a sequence, or the main folding pathways can be revealed. To build a complete Markov model with all of the possible states and transition between them, an exhaustive enumeration of states and moves taking the system from one state to another would be required. This is very expensive even for relatively short chains. To reduce the state space and thus the number of connections, we assumed that diffusional motions occur on a negligibly short time scale compared with conformational motions. By this simplification, we obtain Markov models for which exact calculations can be carried out for chain lengths up to eight.

In the future, further calculations can be carried out for the folding time of particular sequences, and the relationship between the rate of folding and other structural and dynamics parameters can be revealed. Analysis of pathways to the native state through associated states will give information about the free energy landscape of the vicinity of the native state, the functional landscape, or even for the whole state space. Our studies may also shed light on the role of alternating flexibility and rigidity in the binding process.

Flux studies revealed that, it has a crucial role in determining whether the dimerization occurs by a *folding before binding* or a *binding before folding* mechanism that how dense the subnetwork of associated states in the vicinity of the native state. Where this network contains a lot of nodes, and thus a lot of connections, there the major process will be the *binding before folding*. Less nodes and connections leads to the *folding before binding* mechanism (Figures 4).

APPENDIX A
CONFORMATION STRINGS

A conformation on a square lattice can be considered as a sequence of left and right turns and forward progress. This sequence can be represented as a string of the corresponding initial letters such as L (left), R (right) and F (forward). For example, the conformation in the Figure 1. is *LRRFRLRR*.

PUBLICATIONS

[1] A. Szilágyi, D. Györffy, and P. Závodszky, "The twilight zone between protein order and disorder." *Biophys J*, vol. 95, pp. 1612–26, 2008.

[2] D. Györffy, Z. P., and S. A., ""Pull moves" for rectangular lattice polymer models are not reversible," under review.

REFERENCES

[1] A. Szilágyi, D. Györffy, and P. Závodszky, "The twilight zone between protein order and disorder." *Biophys J*, vol. 95, pp. 1612–26, 2008.

[2] D. E. Koshland, "Application of a theory of enzyme specificity to protein synthesis." *Proc Natl Acad Sci U S A*, vol. 44, pp. 98–104, 1958.

[3] D. D. Boehr, R. Nussinov, and P. E. Wright, "The role of dynamic conformational ensembles in biomolecular recognition." *Nat Chem Biol*, vol. 5, pp. 789–96, 2009.

[4] P. Tompa, "Intrinsically unstructured proteins." *Trends Biochem Sci*, vol. 27, pp. 527–33, 2002.

[5] P. E. Wright and H. J. Dyson, "Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm." *J Mol Biol*, vol. 293, pp. 321–31, 1999.

[6] H. J. Dyson and P. E. Wright, "Coupling of folding and binding for unstructured proteins." *Curr Opin Struct Biol*, vol. 12, pp. 54–60, 2002.

[7] K. Sugase, H. J. Dyson, and P. E. Wright, "Mechanism of coupled folding and binding of an intrinsically disordered protein." *Nature*, vol. 447, pp. 1021–5, 2007.

[8] K. F. Lau and K. A. Dill, "A lattice statistical mechanics model of the conformational and sequence spaces of proteins," *Macromolecules*, vol. 22, pp. 3986–3997, 1989.

[9] N. Lesh, M. Mitzenmacher, and S. Whitesides, "A complete and effective move set for simplified protein folding," in *Proceedings of the seventh annual international conference on Research in computational molecular biology*, ser. RECOMB '03. New York, NY, USA: ACM, 2003, pp. 188–195. [Online]. Available: http://doi.acm.org/10.1145/640075.640099

[10] F. Noé and S. Fischer, "Transition networks for modeling the kinetics of conformational change in macromolecules." *Curr Opin Struct Biol*, vol. 18, pp. 154–62, 2008.

[11] J. H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, "Markov models of molecular kinetics: generation and validation." *J Chem Phys*, vol. 134, p. 174105, 2011.

[12] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57(1), pp. 97–109, 1970.

[13] A. P. Demchenko, "Recognition between flexible protein molecules: induced and assisted folding." *J Mol Recognit*, vol. 14, pp. 42–61, 2001.

[14] G. Schreiber, G. Haran, and H. X. Zhou, "Fundamental aspects of protein-protein association kinetics." *Chem Rev*, vol. 109, pp. 839–60, 2009.

[15] B. A. Shoemaker, J. J. Portman, and P. G. Wolynes, "Speeding molecular recognition by using the folding funnel: the fly-casting mechanism." *Proc Natl Acad Sci U S A*, vol. 97, pp. 8868–73, 2000.

[16] Y. Huang and Z. Liu, "Kinetic advantage of intrinsically disordered proteins in coupled folding-binding process: a critical assessment of the "fly-casting" mechanism." *J Mol Biol*, vol. 393, pp. 1143–59, 2009.

[17] Y. Levy, S. S. Cho, J. N. Onuchic, and P. G. Wolynes, "A survey of flexible protein binding mechanisms and their transition states using native topology based energy landscapes." *J Mol Biol*, vol. 346, pp. 1121–45, 2005.

[18] Y. Levy, P. G. Wolynes, and J. N. Onuchic, "Protein topology determines binding mechanism." *Proc Natl Acad Sci U S A*, vol. 101, pp. 511–6, 2004.

[19] B. Mészáros, P. Tompa, I. Simon, and Z. Dosztányi, "Molecular principles of the interactions of disordered proteins." *J Mol Biol*, vol. 372, pp. 549–61, 2007.

[20] J. H. Prinz, B. Keller, and F. Noé, "Probing molecular kinetics with markov models: metastable states, transition pathways and spectroscopic observables." *Phys Chem Chem Phys*, vol. 13, pp. 16 912–27, 2011.

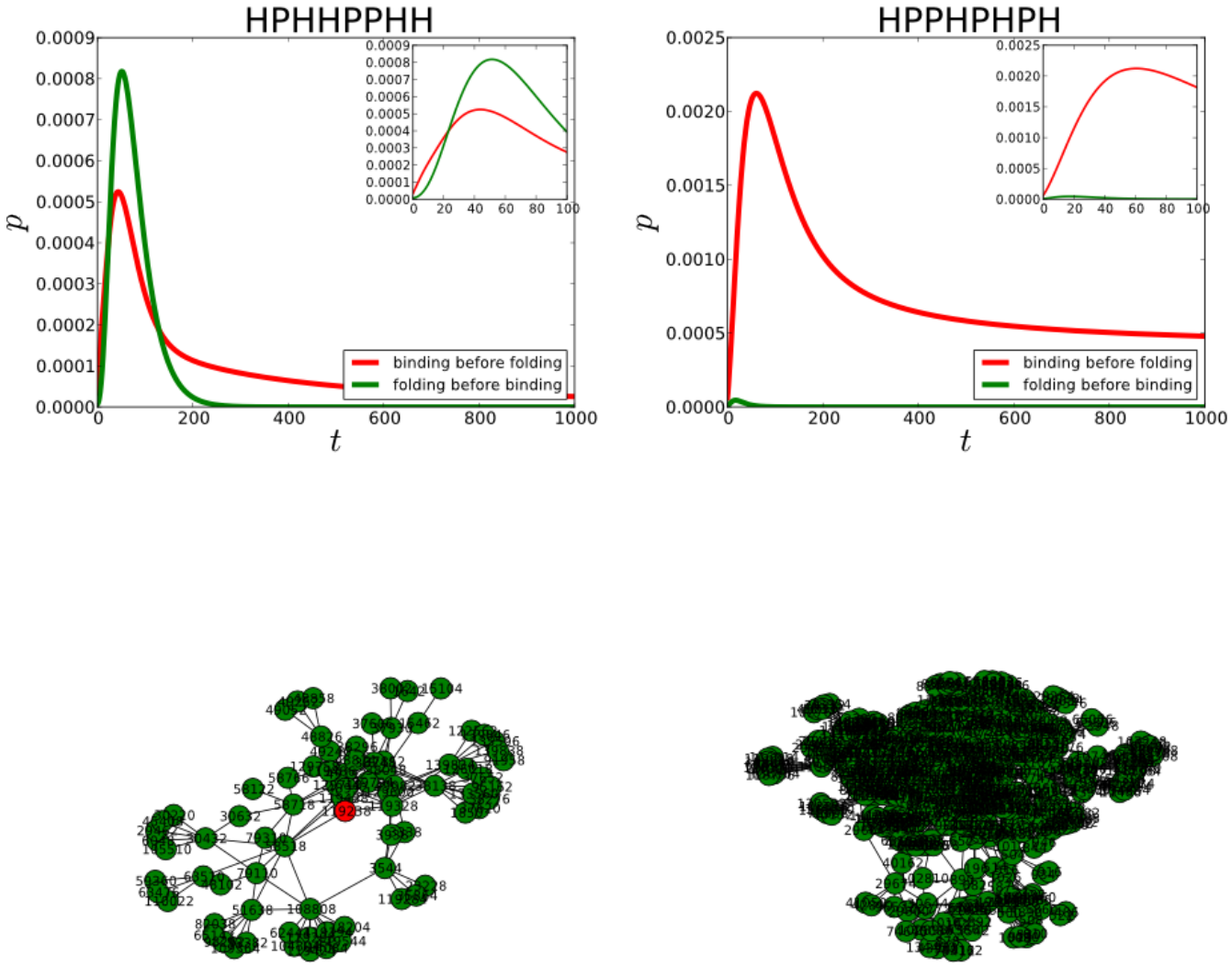


Fig. 4. The cumulative probabilities of probabilities that the system is in an associated or dissociated state, respectively, which is connected to the native state. For sequence **HPHHPPHH** the majority of the flux to native state passes through associated states while for the **HPPHPHPH** sequence the *binding before folding* process is more significant. On the bottom of the figure, the vicinities of native states, consisting of associated states, are shown for the corresponding sequences. It can be seen that the the network for **HPHHPPHH** is much more dense which can explain the predominance of the *binding before folding* mechanism.

Towards combining cortical electrophysiology, fMR measurements and 2-photon microscopy

Bálint Péter Kerekes
(Supervisor: István Ulbert)
bkerekes@cogpsyphy.hu

***Abstract*- The goal of our research is to make a system, which can be used in fMR-EEG experiments. First of all we need to know very well the functionality, and the behavior of a region of the brain to have a good principle in the planned experiments. We used multisilicon probes, microelectrode arrays with electronic depth control, and u- probes for these tests. We planned a data transmission system, which meets the fMR conditions, and tested it in MR environment. We measured the placement of the data transmission system, not to bother the MR imaging. We measured the MR artifacts on the data transmission system, and investigated some of the known artifact removing algorithms. For future experiments we started collaboration with the 2-photon microscope lab, and conducted in-vitro measurements on human brain tissue, to explore the neural mechanisms of cortical sharp wave oscillation.**

***Index Terms*- fMR-EEG; data transmission, 2- photon microscopy, sharp wave**

I. INTRODUCTION

In this study we are in the designing phase of an EEG system which will be compatible with fMR. There are many problems in this issue. We need to make a system which can work in a 3T magnetic field without any artefacts, or with artifacts what can be eliminated later in the data processing phase. I have looked through the existing systems in the scientific literature [1-8], but every one of these systems are struggling with some of these problems: the great magnetic field which can indicate unwanted currents in the system, the vibrations, the amplification and the digitalization needs to be near the electrodes, the power supplies of these systems, the synchronization of the fMR and EEG systems, and so one [9]. We are planning experiments in epileptic patients and in deep sleeping normal subjects [10-12], especially concentrating on the somatosensory cortex[13], but to know more of the oscillations, and the functions of this region we made animal tests in anesthetized rats to get familiar with the subject[14-15]. We investigated the cortico-cortical thalamo-cortical interactions, and made laminar analysis of the somatosensory cortex in slow wave sleep [16-21]. We made collaboration with the Two- photon Imaging Center for an experiment to connect the 2-photon microscope with our EEG data transmission system, and a special extracellular laminar electrode, dedicatedly modified for in-vitro studies

II. MATERIALS AND METHODS

A. Probes

- *Silicon probe* (Fig. 1.) [15]: The length of the silicon probe is 12mm, with a 7mm long part that can be inserted in the tissue, 280 µm wide, and 80 µm thickness. 24 square shaped and 100 µm spaced platinum recording sites were exposed at the end of the shaft. Bonding pads were designed at the other end of the device in the form of 200 µm x 200 µm SiO_2-Pt micro grids. The recording sites were electronically connected to the bonding pads via 4 µm wide and 300 nm thick conductive paths made out of Pt.

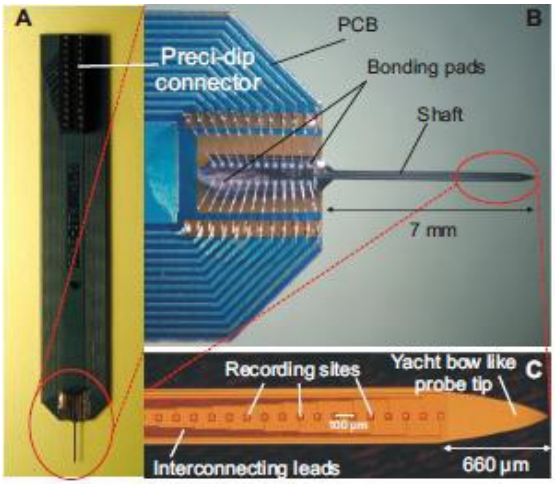


Figure 1. Outlook of the silicon probe [15]. A- The PCB with the 26 pole connector. B- The silicon chip with the Al bonding. C- The tip and the PT contacts of the shaft.

- *Microelectrode array* (Fig. 2.) [14]: A novel two dimensional silicon-based electrode array was developed in the framework of the NeuroProbes EU project. The electrode array is equipped with electronic depth control (EDC) system in order to select up to 32 active recording sites from the 2052 electrode sites without moving the array, the sites are separated by 40 µm in two rows. The array consists of four shanks (8mm long each) in a comb-like structure. The electrodes can be electronically switched to the eight output lines in 2x2 groups like in a tetrode configuration, any combination of two tetrodes can be selected on each shank. The complete system consists of the electrode array, switching matrix, front-end electronics, conditioning, multiplexing and interface electronics and the control software.
- *Laminar electrode* (Figure 3.) [28]: The extracellular laminar electrode is nowadays distributed by Plexon, Plextrode® U-Probe and we used the brain slice configuration of it. The 24 channel probe was connected to the mentioned data transmission systems head stage. We have fixed the electrode to the 2-photon microscopes built-in micromanipulator, to maneuver the probe above the slice.

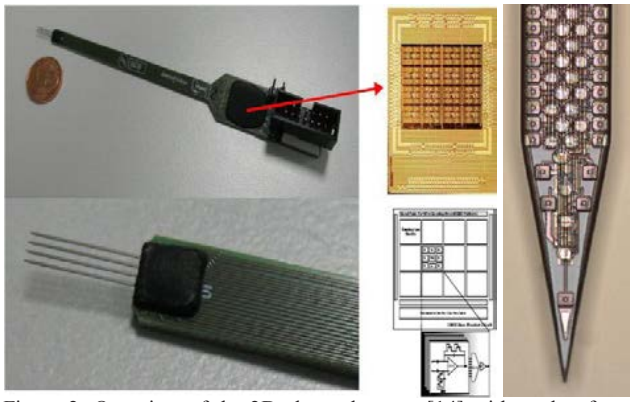


Figure 2. Overview of the 2D electrode array [14] with analog front-end. Analog front-end and structure on the right. Tip section of the shank of the 2D probe on the left.

B. *Data transmission*

In the fMR-EEG experiments we need a system which made of minimally magnetizeable materials. The A/D converter (Fig. 4), preamplifier (head stage), the amplifier, the data transmission and the power supply of these systems, optical USB link, fiber- ribbon- and USB cables.

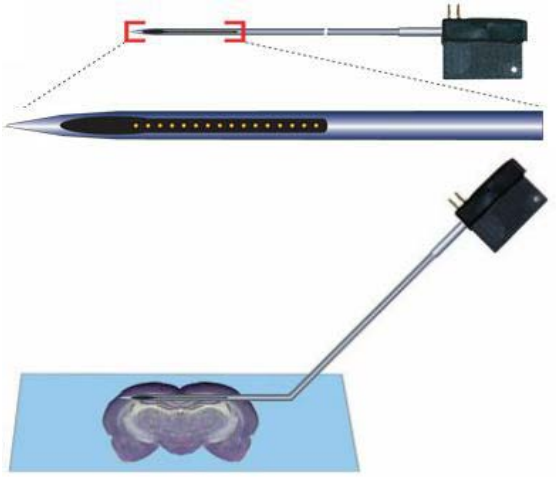


Figure 3. The scematics of Plextrode® U-Probe above, and the Brain slice configuration of it under. [28]

The planned MR compatible system (Fig. 5.) consists of a Head Stage (Gain 10x, Dc, 24 channel + reference), that connects with the Main Amplifier through ribbon cable (Gain 100x, 0.1 Hz-6kHz band, ±3.2 V Lithium battery). After the amplification the next step is the A/D conversion with the above mentioned device (24 channel, 16 bit/ 20-40kHz/ channel), the A/D converter is supplied by a +14.4 V Lithium battery. The A/D converter sends a +5 V power supply and the digital signals to the Optical USB links sender side (USB to optical conversion in the sender side, send the data through fiber optic cables and a back conversion on the receiver side). The optical USB receiver sends the signals to a computers USB port, and we can save the signals with a costume made LabWiev software (.cnt format). The systems elements from the point of the fiber optic cables will be out from the scanner room.

C. *2-photon microscope*

The 2-photon microscope (Figure 6.) what we used is a Femtonics Kft. design [30]. We are using this device for human in-vitro studies.

Figure 4. NI usb-6353 X series data acquisition device. [29]

D. *Human surgery*

We get the human cortex tissue, from cancerous or epileptic patient's surgeries. We are keeping it alive in a nearly frozen state solution (Fig. 7.), and after removing of the pia we cut it into 500um slices perpendicularly from the top of the cortex. The slices are transferred into an incubation chamber for an hour on 36C°. After the incubation they are taken one-by one to the 2-photon microscopes double flow chamber grid, and kept alive with 200ml recirculating oxygenized 36C° ACSF.

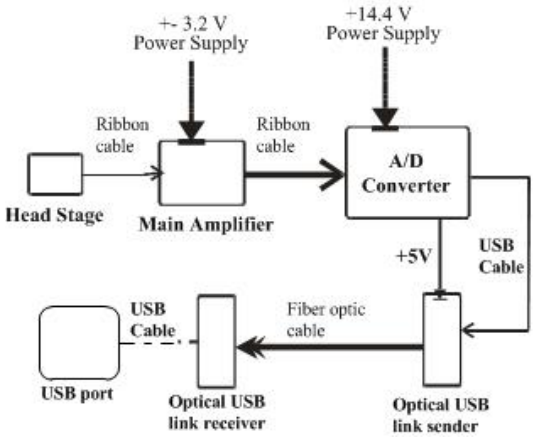


Figure 5. The sketch of the planned MR compatible data transmission system.

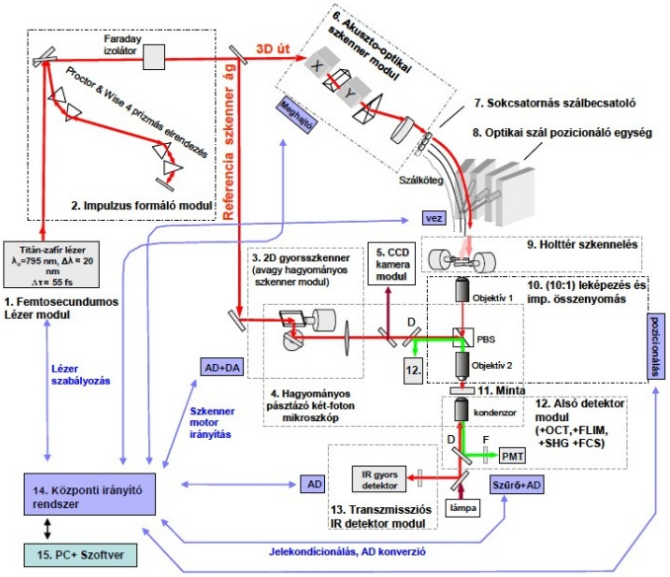


Figure 6. The schematics of the Femtonics 2-photon microscope [30]

III. RESULTS

A. *Silicon electrode*

The analysis of the somatosensory cortex was measured

SOLUTION	substance	molar mass g/mol	end concentration mmol/L	volume of the solution ml	how much to put in g	at the end add from 1 M stock solution (ml)	pH	osm
ACSF	*NaCl*	58	124	2000	**14.384**			314
	KCl	74	3.5	2000		**7**		
	NaHCO3	84	26	2000	**4.368**		7.485	
	D-Glucose	180.2	10	2000	**3.604**			
	CaCl2	110.8	1	2000		**2**		
	MgCl2	95.1	1	2000		**2**		
CUTTER / CARRIER	*Sucrose*	342	250	1000	**85.5**			336
	KCl	74	1	1000		**1**		
	NaHCO3	84	26	1000	**2.184**			
	D-Glucose	180.2	10	1000	**1.802**			
	CaCl2	110.8	1	1000		**1**		
	MgCl2	95.1	10	1000		**10**		
	Phenol Red					**1**		

Figure 7. The solution what used to carry and cut the brain tissue and the ACSF what used in the 2-photon microscopes chamber.

with laminar silicon probes. The probe provided good quality local field potential (LFP), multi-unit (MUA) and single-unit (SUA) activity recordings. The slow oscillation (SO) [16-27] rhythm was observed in anesthetized rats. The SO alternates between a depolarized („up-state") and a hyperpolarized („down-state") state. The LFP of the cortical „down-state" (characterized by neuronal silence) is negative superficially, and has a positive polarity in the deep cortical layers. From the experiments there are significant evidences of the SO-s cortical origin. We made offline analysis of the data we measured, for example current source analysis (CSD) to show the spatiotemporal changes of the transmembrane current sinks and sources, and because the probes design fits the needs (equidistant contact spacing, perpendicularly implanted to the laminar brain structures and the probe embrace the whole cortex) the CSD profile of the cortex in SO can be made.

B. Microelectrode array

We made comparisons between a passive electrode array, and the EDC microelectrode array. In this experiment a 4mm passive probe were implanted in the motor cortex, and the EDC probe into the S1 trunk region and the underlying thalamic nuclei (Fig. 8.).
We measured good quality LFP, MUA and SUA. The signals were analyzed forth offline. We wanted to have some infomations of the correlations of different regions of the brain. We made experiments with passive probes, and EDC probes to test the planned data transmission system, and it complied with the desired demands for these experiments. We tested some of the planned systems boundaries, and it works well on a doubled sampling rate than the old data acquisition system we used, even if its power supply were the lithium-ion batteries (for almost 6 hours).

C. 2-photon microscope and extracellular electrode

Our goal was to combine the two systems (2-photon microscopy and extracellular measuring) to make human cortex in-vitro studies, to understand more the phenomena beside the sharp wave oscillation [31]. First we scanned through the slice with the laminar electrode, and if we

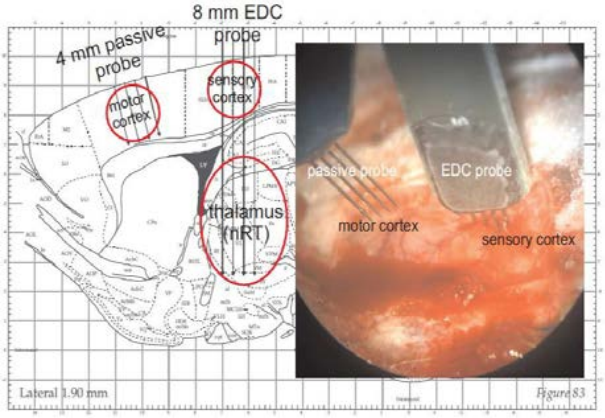


Figure 8. The targeted brain structures in the left, the implantation in the right

found good activity, or the oscillation what we needed, than changed into 2-photon mode, and made bolus loading (mixture of oregon green bapta1 and sulforodamin 101) into some sites in the cortex layers, where the activity appeared. After one hour the cells had taken up the bolus, and we could start to catch the Ca responses with line scans. We put in a LFP micropipette filled with ACSF to measure the field potentials, to correlate it later with the Ca responses, and after we found a good responding cell we tried to patch it, and measured the LFP, intracellular and Ca responses. The results were correlated in MatLab (Fig 9-10.).

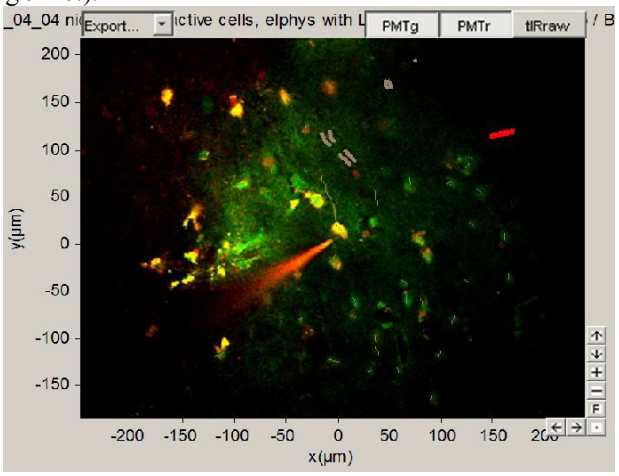


Figure 9. 2-photon picture of the loading site, with the patch pipette (orange), the LFP pipette is on the top right of the screen, the line scan is on the patches cells dendrite.

D. MR measurements for artifact rejection

We measured if the EEG data transmission systems elements placement influence on the MR imaging and found that if data transmission systems nearest element is
5cm away from the imaging area, than it doesn't bother the MR pictures. We need although to measure this after we get the MR compatible cap, and electrode system. We looked through the possible MR artifact rejection from the EEG recordings, and tested the Independent Component Analysis, to be a good solution for our studies.

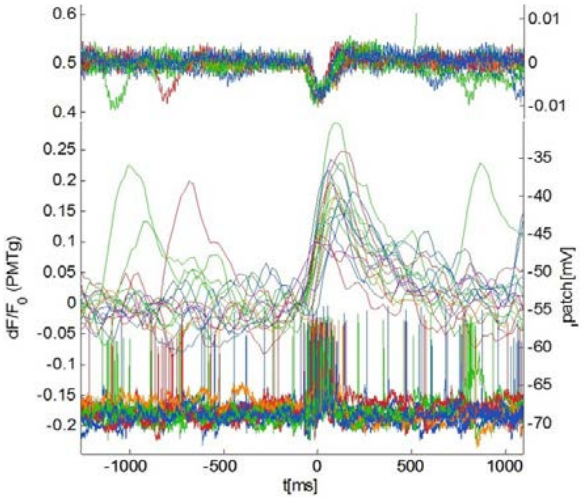


Figure 10. Sharp wave (above), cell burst (below), Ca response (middle) of one cell. The sharp wave was observed with the LFP micropipette, within 100um, the Ca response was observed with 2-photon line scan on the dendrite, and the cell was patched with a micropipette, to observe it's potentials.

IV. CONCLUSION

From the experiment we made in the animal models, we could say that now we have a good basis for the planned fMR-EEG experiments, what we can compare. We gathered the needed instrument for the MR compatible EEG data transmission system. I started to plan how to solve the two systems compatibility problems, read through the literature of the problems of the existing systems. I made tests to measure the placement of the EEG system to minimize the MR image artifact, and tests to certify the applicability of the EEG data transmission system in the MR environment, and tested the ICA algorithm on the measured data for artifact removal.

V. FUTURE PLANS

In the fMR-EEG experiments we will further investigate the quality of the ICA algorithm in the artifact removal. We will try to find the best possibility for the synchronization.

In the 2-photon experiments we will try some ACSF solutions for investigation of their applicability in our research, and make more experiments in the sharp wave oscillation study.

ACKNOWLEDGEMENT

The author wish to acknowledge Dr. György Karmos, Dr. István Ulbert, Richárd Fiáth, and Domonkos Horváth from the MTA-PKI for the assistance, and advice, Dr. Lajos Kozák from the MR Research Center Szentágothai Knowledge Center Semmelweis University for the study lectures in MR, and Rózsa Balázs, and the 2-photon team for the help in the 2-photon microscope lab, and OITI for the human surgeries.

REFERENCES

[1] M. Markus Ullsperger, PhD and P. Stefan Debener, Eds., *Simultaneous EEG and fMRI Recording, Analysis, and Application*. Oxford University Press, 2010, p.^pp. Pages.

[2] C. Mulert and L. Lemieux. (2010). *EEG–fMRI Physiological Basis, Techniqueand Applications*.

[3] M. Scott H. Faro and P. Feroze B. Mohamed. (2010). *BOLD fMRI A Guide to Functional Imaging for Neuroscientists*.

[4] R. B. Buxton. (2009). *Introduction to Functional Magnetic Resonance Imaging Principles and Techniques (2 ed.)*.

[5] P. D. Ingmar Gutberlet. (2009, Did you know ...? MR Correction.

[6] M. Filippi, Ed., *fMRI Techniques and Protocols*. Humana Press, a part of Springer Science, 2009, p.^pp. Pages.

[7] D. Weishaupt, *et al.*, Eds., *How Does MRI Work? An Introduction to the Physics and Function of Magnetic Resonance Imaging*. Springer, 2006, p.^pp. Pages.

[8] S. M. Mirsattari, *et al.*, "EEG monitoring during functional MRI in animal models," *Epilepsia,* vol. 48, pp. 37-46, 2007.

[9] B. Kastler and Z. Patay, *MRI orvosoknak A mágneses magrezonancia orvosi képalkotó eljárásként való alkalmazásának alapelevei*. Budapest
Udine: Folia neuroradiologica, 1993.

[10] M. Czisch, *et al.*, "Functional MRI during sleep: BOLD signal decreases and their electrophysiological correlates," *European Journal of Neuroscience,* vol. 20, pp. 566-574, Jul 2004.

[11] M. Czisch, *et al.*, "Acoustic Oddball during NREM Sleep: A Combined EEG/fMRI Study," *Plos One,* vol. 4, Aug 2009.

[12] E. B. Issa and X. Q. Wang, "Altered Neural Responses to Sounds in Primate Primary Auditory Cortex during Slow-Wave Sleep," *Journal of Neuroscience,* vol. 31, pp. 2965-2973, Feb 2011.

[13] A. Devor, *et al.*, "Coupling of total hemoglobin concentration, oxygenation, and neural activity in rat somatosensory cortex," *Neuron,* vol. 39, pp. 353-359, Jul 2003.

[14] H. P. Neves, *et al.*, "Multi-channel neural probes with electronic depth control," presented at the IEEE BioCAS 2010, 2010.

[15] L. Grand, *et al.*, "A novel multisite silicon probe for high quality laminar neural recordings," *Sensors and Actuators a-Physical,* vol. 166, pp. 14-21, Mar 2011.

[16] M. Steriade, *Neuronal Substrates of Sleep and Epilepsy*: Cambridge University Press, 2003.

[17] M. Steriade, *et al.*, "The Slow (4 Hz) Oscillation in Reticular Thalamic and Thalamocortical Neurons: Scenario of Sleep Rhythm Generation in Interacting Thalamic and Neocortical Networks," *The Journal of Neuroscience,*, vol. 13, 1993.

[18] M. Steriade, *et al.*, "Intracellular Analysis of Relations between the Slow (<1 Hz) Neocortical Oscillation and Other Sleep Rhythms of the Electroencephalogram" *The Journal of Neuroscience,* vol. 13, 1993.

[19] M. Volgushev, *et al.*, "Precise long-range synchronization of activity and silence in neocortical neurons during slow-wave sleep," *Journal of Neuroscience,* vol. 26, pp. 5665-5672, May 2006.

[20] M. Steriade, "The corticothalamic system in sleep," *Frontiers in Bioscience,* vol. 8, pp. D878-D899, May 2003.

[21] M. STERIADE and F. AMZICA, "Intracortical and corticothalamic coherency of fast spontaneous oscillations," *Neurobiology,* vol. 93, 1996.

[22] S. Chauvette, *et al.*, "Origin of Active States in Local Neocortical Networks during Slow Sleep Oscillation," *Cerebral Cortex,* vol. 20, pp. 2660-2674, Nov 2010.

[23] R. Csercsa, *et al.*, "Laminar analysis of slow wave activity in humans," *Brain,* vol. 133, pp. 2814-2829, Sep 2010.

[24] M. Massimini, *et al.*, "The sleep slow oscillation as a traveling wave," *Journal of Neuroscience,* vol. 24, pp. 6862-6870, Aug 2004.

[25] S. Sakata and K. D. Harris, "Laminar Structure of Spontaneous and Sensory-Evoked Population Activity in Auditory Cortex," *Neuron,* vol. 64, pp. 404-418, Nov 2009.

[26] T. J. Sejnowski and A. Destexhe, "Why do we sleep?," *Brain Research,* vol. 886, 2000.

[27] M. Steriade, *et al.*, "A Novel Slow (<1 Hz) Oscillation of Neocortical Neurons in vivo: Depolarizing and Hyperpolarizing Components," *The Journal of Neuroscience,* vol. 13, 1993.

[28] http://www.plexon.com/product

[29] http://sine.ni.com/ds/app/doc/p/id/ds-151/lang/en

[30] B. Rózsa, "Hippokampális interneuronok dendritikus Ca2+ szignalizációjának mérése 2-foton pásztázó mikroszkóp technológiával ", Semmelwies Egyetem Szentágióothai János Idegtudományi Doktori Iskola, Dokori értekezés.

[31] Rüdiger Köhling, *et al.*, " Spontaneous sharp waves in human neocortical slices excised from epileptic patients", *Brain*,vol. 121, pp. 1073–1087, 1998

Complex Electrophysiological Analysis of the Effect of Cortical Electrical Stimulation in Humans

Emília Tóth
(Supervisor: Dr. István Ulbert)
totem@digitus.itk.ppke.hu

***Abstract*— Electrical stimulation is frequently performed in concurrence with electrocorticogram recording for functional mapping (or electrical stimulation mapping-ESM) of the cortex and identification of critical cortical structures. In medically refractory epilepsy surgical candidates, intracranial electrodes are necessary to localize the epileptogenic focus prior to surgical resection. This electrodes are used to record the underlying brain activity and also for electrical stimulation of the cortex. Electrical stimulation mapping (ESM) is the gold standard for identifying functional and pathological areas of the brain. Although the procedure remains unstandardized, and limited data support its clinical validity nevertheless, electrical stimulation mapping for define language areas has likely minimized postoperative language decline in numerous patients, and has generated a wealth of data elucidating brain-language relations [3]. Our aim was to study another way of cortical stimulation, so called single pulse electrical stimulation (SPES) to map pathological and functional networks in the brain.**

***Keywords*-component; biomedical signal processing, electrodes, brain networks, electrocorticography, epilepsy, in vivo, human**

***Abbreviations*- ESM=electrical stimulation mapping; SPES=single pulse electrical stimulation; CT=computed tomography; DCES= direct cortical electrical stimulation; CCEP=cortico-cortical evoked potential; BA=Brodmann area; ROC curves=receiver operating characteristic curves**

I. INTRODUCTION

Mapping of functional areas in the human brain is crucial in epilepsy and tumor surgery. There are several non-invasive methods to identify eloquent cortices, such as functional Magnetic Resonance Imaging (fMRI) or Positron Emission Tomography (PET), but the gold standard is direct high frequency cortical electrical stimulation. In this study we used single pulse electrical stimulation evoked late responses to map language and motor networks and to better understand the electrophysiological mechanisms of the cortico-cortical evoked potentials.

Single pulse electrical stimulation is a new method to investigate the cortico-cortical connections in vivo in the human language, motor and sensory system which can provide insight into the mechanisms of higher-order cortical functions and the connections between functional areas [1]. When using a crown configuration, a handheld wand bipolar stimulator may be used at any location along the electrode array. However, when using a subdural strip, stimulation must be applied between pairs of adjacent electrodes due to the nonconductive material connecting the electrodes on the grid. Electrical stimulating currents applied to the cortex are relatively low, between 2 to 4 mA for somatosensory stimulation, and near 15 mA for cognitive stimulation. The functions most commonly mapped through DCES are primary motor, primary sensory, and language. The patient must be alert and interactive for mapping procedures, though patient involvement varies with each mapping procedure. Language mapping may involve naming, reading aloud, repetition, and oral comprehension; somatosensory mapping requires that the patient describe sensations experienced across the face and extremities as the surgeon stimulates different cortical regions.[2]

High frequency electrical stimulation is the gold standard in neurosurgery for mapping brain functions, but the exact mechanisms behind the effect and parameters used need to be further studied. There is also some risk associated with the

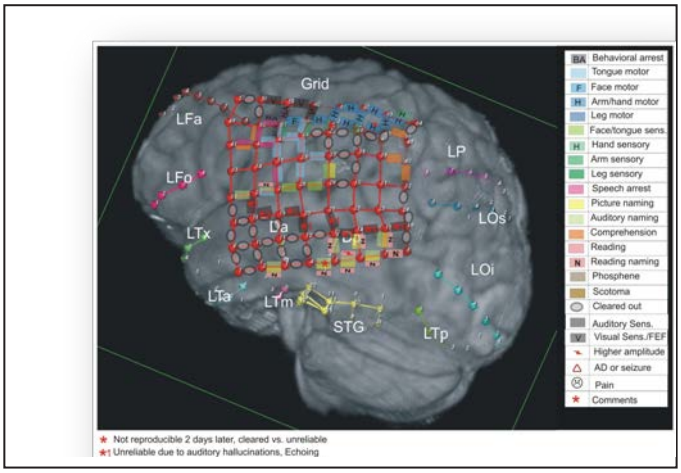


Figure 1. Reconstructed MRI picture with the implanted electrode array, colored lines represent the functions revealed with ESM.

stimulation, due to its proepileptic effect and the limits imposed by the fact that the cortex has to be exposed using some type of surgery.

During the development of epilepsy, the connections between nerve cells are also strengthen or weakening because of various reasons (neuronal cell death, proliferation, brain stem injury, etc). We hypothesized, this cause changes in the number of significant evoked potentials between the areas showing epileptic activity compared to other areas.

Our aim with this study was to find other ways to map functional networks in the brain, using a less invasive method and analyze the network features with this new approach. Single pulse electrical stimulation (0,5Hz) is much less

invasive in terms of seizure generation, and the distribution of the evoked potentials may reveal the intracortical pathways between cortical regions.

II. METHODS

A. Clinical electrodes and recordings

The electrode implantations and recordings, along with ESM and SPES took place at two well established epilepsy surgical centers in Budapest (National Institute of Neuroscience) and New York (North Shore-LIJ Health System). Patients were implanted with intracranial subdural grid, strip, and in some cases depth electrodes for 5–10 days. They were monitored to identify the seizure focus, at which time the electrodes were removed and, if appropriate, the seizure focus was resected. Continuous intracranial EEG was recorded with standard recording systems with sampling rates 1000 or 2000 Hz. The microelectrodes were implanted in eleven cases, perpendicularly to the cortical surface to sample the width of the cortex. This 24 contact laminar electrode has been described previously [4]. Differential recordings were made from each pair of successive contacts to establish a potential gradient across the cortical lamina.

B. Functional Stimulation Mapping

For localization of functional cortical areas, electrical stimulation mapping was carried out according to standard clinical protocol (bipolar stimulation: 2–5 s, 3–15 mA, 20–50 Hz). Areas were defined as expressive language sites when stimulation resulted in speech arrest. When stimulation resulted in a naming deficit based on auditory or visual cues, or an interruption in reading or comprehension, the area was deemed a nonexpressive language site. Sensory and motor areas were identified when stimulation caused movement or changes in sensation.

C. Cortical Electrical Stimulation and Cortico-Cortical Evoked Potentials.

Following implantation of intracranial electrodes, patients were monitored for epileptic activity and during this time, CCEP mapping was performed using single-pulse stimulation. Systematic bipolar stimulation of each pair of adjacent electrodes was administered with single pulses of electrical current (3 mA-15 mA, 0.5 Hz, 0.2-ms pulse width, 20-25 trials per electrode pair). The associated evoked responses (CCEPs) were measured at all other electrode sites. The current amplitude of 10 mA activated the maximal number of neuronal elements without epileptic afterdischarges or other clinical signs. The 2 seconds interstimulation interval was used to minimize the effect of overlapping evoked responses and to leave enough restitution time for the cortex. Patients were awake and at rest at the time of CCEP recording

D. Analysis of CCEPs.

Electrophysiological data analyses were performed using Neuroscan Edit 4.5 software (Compumedics) and own developed MATLAB scripts. Evoked responses to stimulation were divided into 2-s epochs (-500 ms to 1,500 ms) time-locked to stimulation pulse delivery. The CCEP consists of two usually negative peaks termed N1, timed at ~10–30 ms, and N2, which exhibits a broader spatial distribution and occurs between 70 and 300 ms [1]. To quantify the magnitude of the CCEPs in the time window of the N2, the data were low-pass filtered (30 Hz), and baseline correction (−450 to −50 ms) was performed. The SD was computed for each electrode separately using all time points in the -450 to -50 time window, CCEPs were considered significant if the N2 peak of the evoked potential exceeded the baseline amplitude by a threshold of ±6 SD as determined from the receiver operating characteristic (ROC) curves.

E. Electrode localization

To co-register the electrodes to anatomical structures, we used sophisticated imaging techniques, developed by our co-operational research team. We used intraoperative pictures and a postoperative CT scan to localize the electrodes in the skull. This was co-registered to a high resolution preoperative MRI where we could precisely localize the anatomical structures. Using these scans and freely available softwares (Bioimagesuite, Freesurfer, FSL, AFNI) we developed a semi – automated co-localizing each electrode to the underlying Brodman area of the brain. Determination of the seizure onset zone was performed by epileptologists [5].

F. Patients

Twenty patients (ages 6–53 years, 28±14.84, ten females) with medically intractable focal epilepsy were enrolled in the study after informed consent was obtained. These procedures were monitored by local Institutional Review Boards, in accordance with the ethical standards of the Declaration of Helsinki.

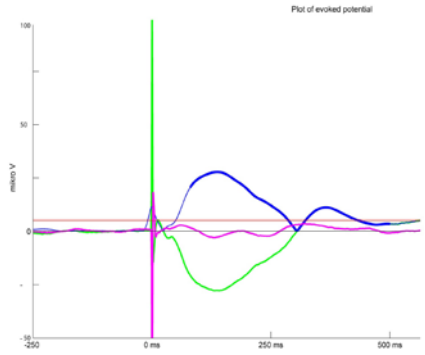


Figure 2. This figure shows averaged responses time locked to the bipolar stimulation artefact (-250-600ms). Green line is the significant response, blue line is the absolute value of the significant response, pink line is a non significant response and the red horizontal line is the threshold for the two responses.

G. Pathological and physiological networks

Neurologist defined pathological and non pathological electrode groups. Pathological electrodes were those which showed seizure onset, or early spread (in the first 10 s) The number of significant evoked potentials was divided into four groups, according to pathological or non pathological classification of the stimulate and the recording electrode.

In addition, two type of seizure spreading mechanism were distinguished, according to the consistency of seizure spread. Consistent seizure spread is when seizure starts always at same

places, inconsistent if there were more tha one typical seizure spreading mechanism.. Network connections were examining from these two aspects.

III. Results

III. Analysis of the significant signal features.

Due to the artifact caused by the stimulation we only focused on the N2 response, which seemed very reliable and reproducible. The variance of both time and amplitude of the N2 peak was high, but it the largest number of peaks occurred around 150 ms, and showed quasi-normal distribution, with two smaller deflections at around 180 -190 ms and 210-250ms. Analysis of 892 peaks, the average latency was 152.84 ms, with 58.7 ms standard deviation.

A. Create a graph.

A significant evoked response indicates the relationship between the electrodes which were stimulated and which showed the significant response. Significant CCEPs were converted to a distance matrix and transformed to a graph using multidimensional scaling (a toolbox from Matlab)

On the one hand the result shows that the functional areas which are close to each other are tightly connected (above somatosensory cortex BA40, BA3, BA2; visual cortex BA17, BA18, BA19 and motor cortex BA6, BA4). On the other hand, those regions which are physically more distant from each other seemed also connected, such as Broca's (BA 45) and Wernicke's (BA 21, BA20, BA22) area. Using this methodology we tried to map as many areas of the brain as possible, to be able to map all the connections between regions which were covered with electrodes.

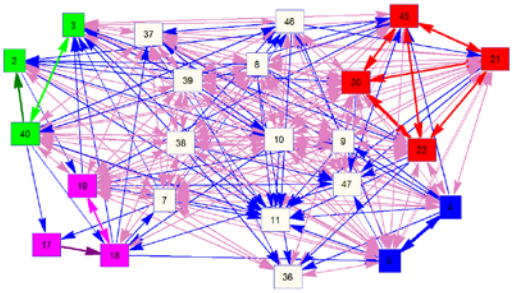


Figure 3. Significant CCEPs were converted to a distance matrix and transformed to a graph using multidimensional scaling. Numbers in squares represent Brodmann areas and lines represent connections. Functional networks are color coded: green sensory, pink visual, red language, blue motor. Lines color coded: thin light pink bidirectional, thin blue unidirectional, darker lines between the elements of functional networks is unidirectional, same color is bidirectional. Stimulating electrodes over Broca's area showed significant responses in electrodes part of the language network as defined with functional stimulation mapping. Responses to stimulation of the primary motor cortex revealed connections to major hubs involved in motor processing.

B. Analysis of changes taking place in cortical layers.

After processing the data from the laminar microelectrode and the implanted macroelectrodes, it can be concluded that after the stimulus, there is a decrease in the power of 15-100 Hz frequency band, and the stimulus elicit deactivation in the middle cortical (3th-5th) layers. This finding is in correlation with previous animal studies, which showed wide band decrease in oscillatory power after stimulation was induced.

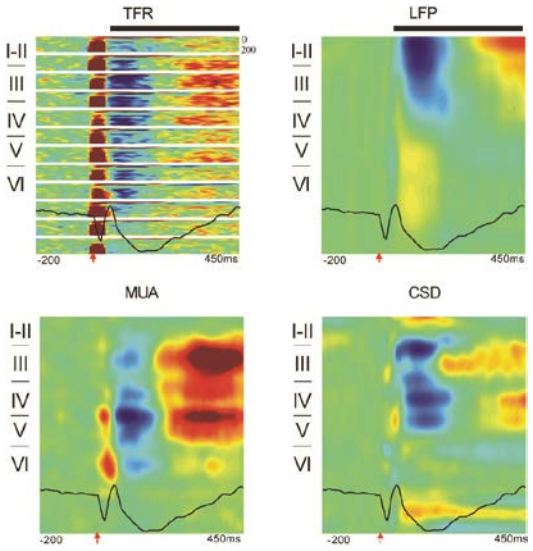


Figure 5. Time frequency analysis (upper left) of data derive from subdural electrodes, roman numbers indicate cortical layers. blue color indicate the power decrease in the 15-200 Hz frequency band, red color indicate increase. Upper right picture is the local field potential, which shows local inactivation in the first two cortical layers in the time of the N2 peak. Lower left picture is the multi-unit activity , lower right the current source densitiy, blue color signs source, red color signs sink.

C. Analysis of the pathological and physiological networks.

Two aspects of pathological and physiological networks were analyzed:

- Connections between pathological and non pathological electrodes, according to the stimulation-recording electrode distance. In this first case, number of significant evoked potentials were divided with the number of all possible responses, which is shown on the 5. Figure, displayed according to the stimulation-recording electrode distance, and consistent or inconsistent epileptic seizure spreading mechanism. The results show that consistent pathological-pathological connections are higher close to the epileptic focus than the inconsistent, but further away, this ratio is reversed. This could confirm, that in the case of inconsistent epilepsy, the seizure focuses are scattered and higher than normal connectivity exist between them.
- Connections between pathological and non pathological Brodmann areas, according to the number of incoming and outgoing connections. Connectivity between Brodmann areas was counted. Given Brodmann area outgoing connections are the number of significant evoked responses triggered by this Brodmann area, incoming connections are the number of significant evoked responses appear in this particular Brodman area. Results are shown in Figure 6.

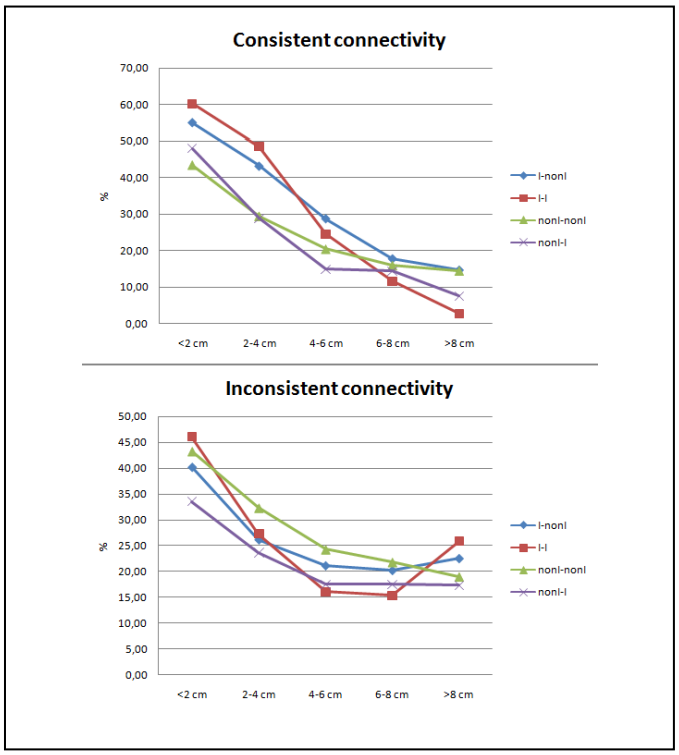


Figure 5. Connectivity line chart. 'I' means pathological electrodes, 'nonI' means non pathological electrodes. Pathological-pathological significant evoked responses (I-I signed red line) shows higher number near (0-4 cm) to the epileptic focus, than normal non pathological-non pathological (nonI-nonI signed green line), although this rate is turn over further (> 8 cm).

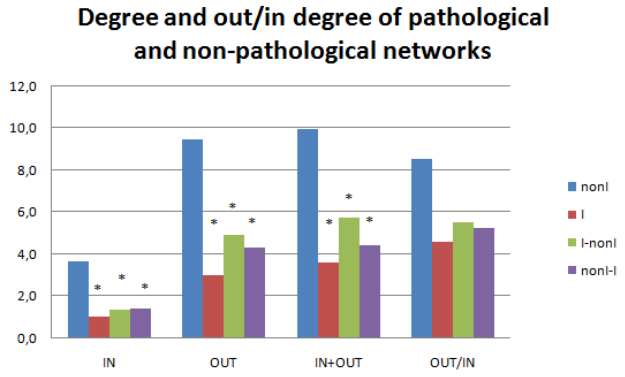


Figure 6. Number of incoming connections to non pathological (blue) Brodmann areas elicited from non pathological Brodmann areas, significantly (p=0.05) higher than which connected with pathological ones

It seems, that the pathological Brodmann areas (red bars) have significantly lower in and out connections than the non-pathological.

IV. Further Aims

We would like to make a pathological, non pathological network identifier algorithm to facilitate the neurologists work and take measurable the pathological or non pathological connections.

To verify these results, we need higher number of patients involved to increase the statistical significance of the study.

V. Conclusion

The results suggest that single pulse electrical stimulation evoked potentials may reveal connections of functional areas and functional networks of the human brain. Other studies also report that direct cortical stimulation has a suppressive effect on fast cortical activity and epileptic spikes [7], or can help to clarify the size of the area to be removed[8].

We conclude that single pulse electrical stimulation is a promising technique in delineating eloquent cortex and might be a useful tool to identify pathological networks.

References

[1] R. Matsumoto, D.R. Nair, E. LaPresto, I. Najm, W. Bingaman, H. Shibasaki, and H.O. Lüders, "Functional connectivity in the human language system: a cortico-cortical evoked potential study.," Brain, vol. 127, (no. Pt 10), pp. 2316-30, Oct 2004.

[2] L. Schuh and I. Drury, "Intraoperative Electrocorticography and Direct Cortical Electrical Stimulation.," Seminars in Anesthesia vol. 16, pp. 46-55, 1996.

[3] M.J. Hamberger, "Cortical language mapping in epilepsy: a critical review.," Neuropsychol Rev, vol. 17, (no. 4), pp. 477-89, Dec 2007.

[4] I. Ulbert, E. Halgren, G. Heit, and G. Karmos, "Multiple microelectrode-recording system for human intracortical applications.," J Neurosci Methods, vol. 106, (no. 1), pp. 69-79, Mar 2001.

[5] D. Kovalev, J. Spreer, J. Honegger, J. Zentner, A. Schulze-Bonhage, and H.J. Huppertz, "Rapid and fully automated visualization of subdural electrodes in the presurgical evaluation of epilepsy patients.," AJNR Am J Neuroradiol, vol. 26, (no. 5), pp. 1078-83, May 2005.

[6] C.J. Keller, S. Bickel, L. Entz, I. Ulbert, M.P. Milham, C. Kelly, and A.D. Mehta, "Intrinsic functional architecture predicts electrically evoked responses in the human brain.," Proc Natl Acad Sci U S A, Jun 2011.

[7] M. Kinoshita, A. Ikeda, R. Matsumoto, T. Begum, K. Usui, J. Yamamoto, M. Matsuhashi, M. Takayama, N. Mikuni, J. Takahashi, S. Miyamoto, and H. Shibasaki, "Electric stimulation on human cortex suppresses fast cortical activity and epileptic spikes.," Epilepsia, vol. 45, (no. 7), pp. 787-91, Jul 2004.

[8] A. Valentín, G. Alarcón, M. Honavar, J.J. García Seoane, R.P. Selway, C.E. Polkey, and C.D. Binnie, "Single pulse electrical stimulation for identification of structural abnormalities and prediction of seizure outcome after epilepsy surgery: a prospective study.," Lancet Neurol, vol. 4, (no. 11), pp. 718-26, Nov 2005..

A New Form of Humor − Mapping Constraint Based Computational Morphologies to a Finite-State Representation

Attila Novák
(Supervisor: Gábor Prószéky)
novak.attila@itk.ppke.hu

***Abstract*—MorphoLogic's Humor morphological analyzer engine has been used for the development of several high quality computational morphologies, among them ones for languages with complex agglutinating morphologies. However, Humor's closed source licensing scheme has been a limitation to making these resources widely available. Moreover, there are a few limitations of the rule based Humor engine: lack of support for morphological guessing and the use of frequency information or other weighting of the models. These problems were solved by converting the databases to a finite-state representation that allows morphological guessing and the addition of weights and has open source implementations.**

***Keywords*-morphological analysis, finite-state morphology**

I. INTRODUCTION

MorphoLogic's Humor ('High speed Unification MORphology') morphological analyzer engine [6] has been used for the development of several high quality computational morphologies. These include ones for languages with complex agglutinating morphologies, such as Hungarian and several other members of the Uralic language family: Komi, Udmurt, Northern Mansi and several Khanty dialects besides several Indo-European languages including Polish, English, German, French and Spanish. Most of these morphologies were created using a morphological description development environment that can generate the resources used by the analyzer engine from a feature-based high level human readable description that contains no redundant information and is thus easy to maintain [5].

However, one aspect of morphological processing is not covered by the original Humor implementation. The Humor lookup algorithm does not support a suffix based analysis of word forms whose stem is not in the stem database of the morphological analyzer and it can't be easily modified to add this functionality. Such a morphological guesser would be a very useful tool, because every corpus of natural language text contains a significant amount of words the stems of which cannot be expected to have been listed in the stem database of a general purpose morphological analyzer. Moreover, implementation of the rule based Humor engine does not make adding weights or frequency information of the morphological models possible. Thus integration and appropriate usage of frequency information as would be needed by data driven statistical approaches to text normalization (e.g. automatic spelling error correction or speech recognition) is not possible within the original Humor system. Being able to create a statistical model would be especially useful when building an unknown word guesser (a functionality missing from the original Humor implementation anyway), since this could provide a natural way of ranking weakly constrained guessed analyses. A third factor that can be mentioned here is that Humor's closed source licensing scheme has been a limitation to making these resources widely available.

This paper describes the approach that we followed to open up new possibilities for using the linguistic resources created using the Humor formalism by converting them to a representation that can be compiled and used by finite state morphological tools, among them ones with open source implementation. The finite state representation can be easily used for suffix based morphological guessing, it provides a natural means for introducing frequency data also making composition with weighted error models possible.

II. THE HUMOR MORPHOLOGICAL ANALYZER

The Humor analyzer performs a classical 'item-and-arrangement' (IA) style analysis, where the input word is analyzed as a sequence of *morphs*. Each morph is a specific realization (an *allomorph*) of a *morpheme*, an indivisible meaning or function bearing element of a word.

The word is thus segmented into parts which have (i) *a surface form* (that appears as part of the input string, the morph), (ii) *a lexical form* (the 'quotation form' of the morpheme) and (iii) *a category label* (which may contain some structured information or simply be an unstructured label). The following analyses of the Hungarian word form *Várnának* contain two morphs each, a stem and an inflectional suffix, delimited by a plus sign.

```
analyzer>Várnának
   Várna[S_N]=Várná+nak[I_Dat]
   vár[S_V]=Vár+nának[I_Cond.P3]
```

The lexical form of the stem differs from the surface form (following an equal sign) in both analyses: the final vowel of noun stem (having a category label [S_N]) is lengthened from *a* to *á*, while the verbal stem (having a category label [S_V]) differs in capitalization.

The lexical form and the category label together more or less well identify the morpheme of which the surface form (the morph) is an allomorph. In the case of stems, which are the core elements of words, the lexical form is called a *lemma*, which is the form of the word (stem) that is normally used to represent it in a dictionary. The category label of stems is their part of speech, while that of prefixes and suffixes is a mnemonic tag expressing their morphosyntactic function. In the case of homonymous lexemes where the category label alone is not sufficient for disambiguation, an easily identifiable indexing tag is usually added to the lexical form to distinguish the two morphemes in the Humor databases. The disambiguating tag is a synonymous word identifying the morpheme at hand.

The program performs a depth first search on the input word form for possible analyses. It looks up morphs in the lexicon the surface form of which matches the beginning of the input word (and later the beginning of the yet unanalyzed part of it). The lexicon may contain not only single morphs but also morph sequences. These are ready-made analyses for irregular forms of stems or suffix sequences, which can thus be identified by the analyzer in a single step, which makes its operation more efficient.

In addition to assuring that the requirement that the surface form of the next morpheme must match the beginning of the yet unanalyzed part of the word (uppercase-lowercase conversions may be possible) is met, two kinds of checks are performed by the analyzer at every step. On the one hand, it is checked whether the morph being considered as the next one is locally compatible with the previous one. On the other hand, it is examined whether the candidate morph is of a category which, together with the already analyzed part, is the beginning of a possible word construction in the given language (e.g. suffixes may not appear as the first morph of a word etc.). The global word structure check is performed on candidate morphs for which the local compatibility check has succeeded. Possible word structures (how morphemes may follow each other within words of the given language, i.e. *morphotactic* constraints) are described by an extended finite-state automaton (EFSA) in the Humor analyzer. Being based on a regular word grammar, the analyzer produces flat morph lists as possible analyses, i.e. it does not assign any internal constituent structure to the words it analyzes. Using a finite-state automaton is more efficient than having a context-free parser and it also avoids most of the irrelevant ambiguities a CF parser would produce.

As we have seen, the lexical database of the Humor analyzer consists of an inventory of morpheme allomorphs (surface realizations of morphemes), an extended finite-state word grammar automaton and data structures used for the local compatibility check of adjacent morphs. For the latter purpose, the Humor formalism uses two types of data structure: continuation classes and binary continuation matrices describing the compatibility of those continuation classes on the one hand, and binary properties and requirements vectors on the other hand. Each morph has a continuation class identifier on both its left and right hand side, in addition to a right-hand-side binary properties vector and a left-hand-side binary requirements vector. The properties vector is a single binary number. The requirements vector may also contain don't care positions, implemented as a combination of binary mask and a binary number.

Local compatibility check is performed as follows: a morph (typically a suffix (sequence)) may be attached to another morph (typically a stem) if right-hand-side properties of the stem match left-hand-side requirements of the suffix in the following manner:

1. A continuation matrix is selected by a masked subset of the binary properties vector of the stem. In most actual morphologies, two matrices are used: one for verbal stems and another for the rest, but some use more. In fact, a single matrix could also be used, but decomposition is motivated by the fact that at least half of the matrix would all be empty due to the fact that verbal suffixes can never be attached to nominal stems and vice versa.

2. The matrix code of the stem must be compatible with that of the suffix according to the selected binary compatibility matrix. Matrix rows are indexed by the suffix continuation matrix code and the columns by the stem continuation matrix code.

3. All the bits in the non-don't-care positions of the left-hand-side binary requirements vector of the suffix must match those in the right-hand-side binary properties vector of the stem.

The extended finite-state word grammar automaton used to check overall word structure is a deterministic automaton. It may have extra binary state variables in addition to its main state variable which can be used to handle non-local constraints within the word such as "a superlative prefix must be licensed by a subsequent comparative suffix or a lexically marked licensing stem", or "a verbal prefix in a non-monomorphemic word must be licensed by a subsequent verbal stem or verbal derivational suffix". The extra state variables make it possible that such constraints can be handled without an explosion of the state space of the automaton.

In addition to the previously mentioned data structures, the morphological database of the Humor analyzer also contains a mapping from right-hand-side property vectors to sets of possible morphological category labels which are used a terminal nodes in the word grammar. The mapping is actually defined from masked subvectors to labels. However, since several masked subvectors may match a specific vector, the mapping actually maps vectors to label sets. Arcs within the word grammar automaton are labeled by these morpheme category labels. The lookup and local compatibility check of each morph in the word form is followed by a move in the word grammar

automaton. The move is possible if at the current state of the automaton there is an outgoing arc labeled by one of the morphological category labels to which the right-hand-side property vector of the currently looked up morph is mapped. Whether the automaton is deterministic in the sense that there are no two outgoing arcs from any of the states the labels of which can match the same vector is checked at compile time.

The database would be difficult to directly create and maintain in the format used by the analyzer, because it contains redundant and low level data structures, thus it would be hard to read for humans, hard to keep consistent, hard to find and correct errors in it and hard to add new lexical entries. To avoid the need to maintain the database in a hard to read formalism, a development environment was created which facilitates the creation and maintenance of the morphological databases. All Humor morphologies built after the creation of the development environment were developed using this higher level formalism.

In the environment, the linguist creates a high level human readable description which contains no redundant information. The system transforms it in a consistent way to the redundant representations which the analyzer uses. One feature of the development environment is that the features used in the high level description are transformed to the format used by the analyzer using an encoding definition description, which defines how each of the features used in the morphological description should be encoded for the analyzer: certain features are mapped to binary properties while the rest determine the continuation matrices which are generated by the system dynamically. (Features can also be ignored to produce special-purpose versions of the analyzer and the generator which ignore certain restrictions and thus overanalyze/overgenerate).

III. FINITE-STATE MORPHOLOGIES

The most influential implementation of finite-state tools for morphological processing is the *xfst-lookup* combo of Xerox. *xfst* is an integrated tool that can be used to build computational morphologies implemented as finite-state transducers using various formalisms while lookup consists of optimized runtime algorithms to implement actual morphological analysis and generation using the lexical transducers compiled by xfst.

The formalism used for describing morphological lexicons in xfst is called *lexc*, originally an independent program of Xerox, the functionality of which later was integrated into xfst. The lexc formalism makes it possible to describe morphemes, organize them into sublexicons and describe word grammar using continuation classes. A lexc sublexicon consists of morphemes having an abstract lexical representation that contains the morphological tags and lemmas and usually a phonologically abstract underlying representation of the morpheme which is in turn mapped to genuine surface representations by a system of phonological rules.

The phonological rules can either be formulated as a sequential or a parallel rule system. Sequential systems of rewrite rules were extremely popular with phonologists from the end of the sixties until the beginning of the nineties and still any person trained in generative phonology knows how to read and write phonological rules of this sort. Sequential rule systems in theory introduce several levels of abstract representations between the surface and underlying lexical representations. The original functionality of xfst was just to implement algorithms to compile and compose rule systems of this sort and to compose them with lexical transducers compiled by lexc. It was a fundamental achievement of leading Xerox researchers to formulate these algorithms and implement them in an efficient way, because before that sequential rule systems were not usable for analysis in practice due to a combinatorial explosion of ambiguity on the intermediate levels of representation. All these intermediate levels along with the intractable amount of ambiguity can be eliminated by combining the whole rule system and the lexicon into a single transducer. The finite-state operation to combine sequential rule systems into a single transducer is composition.

In contrast to the sequential one, the alternative parallel rule formalism assumes only two levels of phonological representation. The two level rule system consists of simultaneous constraints on surface and underlying phonological representations. The finite-state operation to combine parallel rule systems into a single transducer is intersection. The Xerox implementation of the two level rule compiler is called *twolc*. The two level formalism is still popular in Finland.

An important feature of the Xerox finite-state transducer implementation is that it makes a factorization of the state space of the transducers possible in a manner similar to the extended word grammar automaton of the Humor analyzer. The construct is called *flag diacritics* and is implemented as a set of special epsilon arc types that trigger operations in the lookup algorithm such as setting/clearing of an extended state variable (called a flag), performing unification of the current value of the variable with the value specified on the arc or checking if the flag is set or cleared. If the latter value checking operations fail, exploration of the current path in the transducer is abandoned and backtracking occurs. Flag diacritics are single multicharacter symbols that label a single arc each. Being a kind of epsilon symbol, arcs labeled by flag diacritics are traversed without reading the input by the lookup algorithm. Regular non-epsilon multicharacter symbols are also often used in finite-state grammars e.g. to represent morphosyntactic tags as a single symbol. A greedy tokenizer is used to recognize these symbols in the input in the lookup algorithm.

Although handling of flag diacritics during lookup incurs some speed penalty, they are useful on the one hand because they can help prevent the size explosion of the transducer due to long distance dependencies in the morphology and, on the other hand, because they can also be used to describe constraints between adjacent morphemes in a linguistically expressive and easy to understand manner making the description of morphotactics much easier than it were using the continuation classes of the lexc formalism only. Flag diacritics expressing such local constraints can often be eliminated from the transducer without a significant transducer size penalty.

Although human languages differ greatly in how complex structures are expressed on the level of a single word form (i.e. in how complex their morphology is), the Xerox tools implement a powerful formalism that makes possible the description of even the most complex of those morphologies. This suggested that mapping of the morphologies implemented in the Humor formalism to a finite-state representation should have no impediment.

However, the Xerox tools, although they were made freely available for academic and research use in 2003 with the publication of [2]. Do not differ from Humor in two significant respects: a) they are closed source and b) cannot handle weighted models. Luckily, a few years later quite a few open source alternatives of xfst were developed. One of these open source tools, *Foma* [3], can be used to compile and use morphologies written using the lexc/xfst formalism. Another tool, *OpenFST* [1], is capable of handling weighted transducers and a third tool, *HFST* [4], can convert transducers from one format to the other and act as a common interface above the Foma and OpenFST backends.

IV. The Humor to lexc conversion

As the morphological models created with the Humor formalism contain a full description of the morphology including morphophonology, neither the sequential (xfst) or the parallel (twolc) rule component of the finite state formalism is needed for the conversion of the Humor grammars to a finite state representation.

The lexical form (the lemma in the case of stems) and category label of each morph is mapped to the lexical side of the lexc representation of the morpheme while its surface form is mapped to the surface side. In the case of morphologies transformed from the Humor representation, the latter is real surface form instead of the abstract underlying phonological representation that is common in usual lexc lexicon sources. Appropriate alignment of corresponding symbols in the lexical and surface representations is provided by the implementation of the lexicon converter. Tags are represented as single multicharacter symbols in the representation.

Local morph adjacency constraints represented as matrix codes, continuation matrices and binary properties and requirements vectors can be represented directly as lexc continuation classes. In fact the approach taken was to add a switch to the piece of code that generates the Humor encoding of high level property and requirements expressions in the development environment that, when present, triggers the code to regard all high level linguistic features (even the ones that were originally declared to be represented as binary vector elements) to participate in determining continuation matrices. This way the generated matrices alone completely describe all morph adjacency constraints.

As we have seen, morphotactic constraints, many of which are non-local, are represented by the word grammar automaton in the Humor model. The best way to handle such non-local constraints is using flag diacritics. The main state variable of the automaton is mapped one flag (called *St*) while the extended binary state variables to one additional generated flag each. An arc in the original automaton running from state s_i to s_j and checking the cleared state of variable v_k and setting state variable v_l is mapped to four flag arcs in the conversion process: one unifying the value of the St flag with s_i, one setting it to s_j, one checking the cleared state of flag v_k and one setting flag v_l. The exact set of flag diacritics arcs attached to the representation of each morph is determined by the word grammar category of the morph.

When generating the lexc representation of each morph, the sublexicon it will be member of is determined by its left matrix name and code, and its continuation class is determined by its right matrix name and code along with its word grammar category. Word grammar category determines flags as described above, while right matrix name and code hooks back to the morpheme lexicons through sublexicons directly encoding the compatibility relations encoded in the Humor matrices. Elimination of the word grammar state flag St is possible to improve the speed of lookup on the transducer, however it may result in a considerable growth of the state space.

The following is a brief comparison of a version of the Hungarian morphology containing about 144000 morphs in the original Humor compiled lexicon format and the converted version compiled by the Xerox xfst tool with and without the elimination of the *St* flag and run using the Xerox lookup tool.

TABLE I. Comparison of original Humor and xfst compiled equivalents of a 144000 morph Hungarian lexicon

	Humor lexicon	***lexc with St***	***lexc St eliminated***
runtime mem	3.3 MB	20.6 MB	38.5 MB
lookup speed	4700 w/s	12500 w/s	33333 w/s

Finite-state conversion results in a significant increase of the memory footprint (>11 times) of the morphological analyzer, however, it also yields a significant analysis speed benefit (>7 times). Elimination of further flags roughly doubles the size of the compiled lexicon for each eliminated flag. It also leads to an extremely long compilation time but it does not result in any significant speed benefit.

Acknowledgment

The original Humor tools were implemented at MorphoLogic by Miklós Pál based on ideas by Gábor Prószéky.

References

[1] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "Openfst: a general and efficient weighted finite-state transducer library," in Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007), 2007, pp. 11–23.

[2] Kenneth R. Beesley and Lauri Karttunen. Finite State Morphology. CSLI Publications, Ventura Hall, 2003.

[3] Måns Huldén, "Foma: a finite-state compiler and library," in Proceedings of EACL 2009, 2009, pp. 29–32.

[4] K. Lindén, E. Axelson, S. Hardwick, T.A. Pirinen and M. Silfverberg, “HFST - Framework for Compiling and Applying Morphologies” ;in Proc. SFCM, 2011, pp. 67–85.

[5] Attila Novák, “Language Resources for Uralic Minority Languages,” Proceedings of the SALTMIL Workshop at LREC 2008. Marrakech, 2008, pp. 27–32.

[6] Gábor Prószéky and Balázs Kis. “A Unification-based Approach to Morpho-syntactic Parsing of Agglutinative and Other (Highly) Inflectional Languages,” in Proceedings of the 37th Annual Meeting of the ACL, College Park, Maryland, USA, 1999, pp. 261–268.

Advances in full morphological disambiguation for less-resourced languages

György Orosz
(Supervisor: dr. Gábor Prószéky)
oroszgy@itk.ppke.hu

***Abstract*—This paper presents a new Hidden Markov model based morphological tagging method that mainly contributes on integrating a morphological analyzer and thus performs full disambiguated morphological analysis including lemmatization of words both known and unknown to the morphological analyzer. Applying the method we implemented a tool that is fast to train and use while having an accuracy on pair with slow to train Maximum Entropy or Conditional Random Field based taggers. Full integration with morphology and an incremental training feature make it suited for creating corpora for less-resourced languages. We show that the integration with morphology boosts our tool's accuracy in every respect – especially in full morphological disambiguation – when used for morphologically complex agglutinating languages. We evaluate the implementation of our method called PurePos on Hungarian data demonstrating its state-of-the-art performance in terms of tagging precision and accuracy of full morphological analysis.**

***Keywords*-part-of-speech tagging, morphological disambiguation, lemmatization, agglutinative languages, natural language processing**

I. Introduction

Tools called 'Part-of-speech (PoS) taggers' are widely used in current language processing systems as a preprocessing tool. Calling them PoS taggers is a bit misleading, since a number of tagging algorithms are capable of effectively assigning tags to words from a much richer set of morphosyntactic tags than just the basic PoS categories. However, morphosyntactic tagging is still just a subtask of morphological disambiguation: in addition to its tag, the lemma of each word also needs to be identified. Most of currently available taggers only concentrate on determining the morphological tag but not the lemma, thus doing just half of the job. For morphologically not very rich languages like English, and in situations where there is ample training material, a cascade of a tagger and a lemmatizer yields acceptable results. For morphologically rich agglutinating languages – such as Hungarian, Finnish or Turkish – and especially in cases where only a limited amount of training material is available, our results show that a closer integration of the tagger and the morphological analyzer (MA) is necessary to achieve acceptable results.

Halácsy et al. previously demonstrated [6] that a morphological analyzer can improve the accuracy of tagging. Improvement is considerable when disambiguating texts written in agglutinating languages. In our paper, we first introduce the main fields of application where we believe our new method could be applied more efficiently than others available. We then describe the process of creating our tool. Our implementation is mainly based on algorithms used in TnT [2] and HunPos [5] but the new tool is capable of morphosyntactic tagging and lemmatization at the same time thus yielding complete disambiguated morphological annotation[1]. In addition to the Hidden Markov model (HMM) used for the disambiguation of morphosyntactic tags, the tool includes an interface to an integrated morphological analyzer that not only greatly improves the precision of the tagging of words unseen in the training corpus but also provides lemmata. Finally we compare the performance of our implementation called PurePos with other tagging tools.

II. Need of an integrated morphological analyzer

A. Agglutinating languages

If we compare agglutinating languages like Hungarian or Finnish with English in terms of the coverage of vocabulary by a corpus of a given size, we find that although there are a lot more different word forms in the corpus, these still cover a much smaller percentage of possible word forms of the lemmata in the corpus than in the case of English. On the one hand, a 10 million word English corpus has less than 100,000 different word forms, a corpus of the same size for Finnish or Hungarian contains well over 800,000 [9]. On the other hand, however, while an open class English word has about 4–6 different word forms, it has several hundred or thousand different productively suffixed forms in agglutinating languages. Moreover, there are much more different possible morphosyntactic tags in the case of these languages (corresponding to the different possible inflected forms) than in English (several thousand vs. a few dozen). Thus data sparseness is threefold: *i*) an overwhelming majority of possible word forms of lemmata occurring in the corpus is totally absent, *ii*) word forms that do occur in the corpus have much less occurrences, and *iii*) there are also much less examples of tag sequences, what is more, many tags may not occur in the corpus at all.

The identification of the correct lemma is not trivial either, especially in the cases of guessed lemmata. But also for words that can be analyzed by the regular morphological analyzer, identifying the lemma can still be a problem. In Hungarian, for example, there is a class of verbs that end in *-ik* in their

[1] i.e. each word is annotated with its lemma and its morphosyntactic tag

third person singular present tense indicative, which is the customary lexical form (i. e. the lemma) of verbs in Hungarian. Another class of verbs has no suffix in their lemma. The two paradigms differ only in the form of the lemma, so inflected forms can belong to the paradigm of either an *-ik* final or an non-*ik* final verb and many verbs (especially ones containing the productive derivational suffix *-z/-zik*) have an ambiguous lemma.

B. Resource poor languages

A great proportion of resource poor languages (that lack annotated corpora) is morphologically complex. For these languages, to create an annotated corpus, an iterative workflow is a feasible approach. First, a very small subset of the corpus is disambiguated manually and the tagger is trained on this subset. Then another subset of the corpus is tagged automatically and corrected manually, yielding a new, bigger training corpus and this process is repeated. For this workflow to be in fact feasible, a fast turnaround time is needed for the retraining process. In terms of training time, Hidden Markov Model based taggers greatly outperform other tagger algorithms, like maximum entropy and conditional random fields, which take comparatively longer time to train. HMM thus fits better the iterative workflow sketched above.

III. Implementation

A. Background

Our goal was to develop a morphological disambiguation method that *i*) contains employs a morphological analyzer, *ii*) performs full morphological disambiguation including lemmatization, *iii*) can handle Unicode input, *iv*) can efficiently handle (including lemmatization of) both words unknown to the morphological analyzer and ones missing from the training corpus, *v*) and is fast to train.

Although there are a few tools (see the end of this section) that meet part of the above requirements, none of them satisfies all, therefore, we decided to build a morphological tagger. Choosing Java as the base of our implementation our tool has the advantage of universally representing characters of texts written in any language covered by Unicode. We modelled our implementation on HunPos, an open source HMM tagger written in OCaml. Enriching HunPos with the required functionality would probably have been the easiest way to attain our goal. However, the lesser known programming language in which it is implemented could be a hindrance to the portability, integration and further customization of the tool.

HunPos itself is an open source reimplementation and enhancement of Thorsten Brants' TnT. It is capable of *i*) using a generalized smoothed n-gram language model (while TnT has trigrams only), *ii*) context sensitive emission probabilities (another enhancement over TnT), *iii*) clever tricks like applying different suffix guesser models to capitalized and lower case words, *iv*) different emission models for ordinary words and special tokens that contain digits and other non-letter characters, and, *v*) in order to improve tagging speed, a beam search instead of an unconstrained Viterbi search. In addition, HunPos can load a morphological table at initialization time that can be used to emulate the operation of an integrated morphological analyzer if all the word forms in the text to be processed had been listed along with all their possible tags using the analyzer in an off-line manner before initializing the tagger. However, this kind of poor man's MA is hardly applicable in corpus annotaion scenarios as noted above. A special enhanced version of TnT had a similar table loading mechanism that was used by Oravecz and Dienes [9]. However, as far as we know, that version was not made accessible to the public.

A drawback of HunPos is that it can only process 8 bit input, it handles case distinctions incorrectly if we attempt to use it on UTF-8 text. There is a degradation of performance even when trying to use it for UTF-8 Latin text with accented letters. Moreover a model which is trained on 8-bit encoded text, cannot be correctly used for a text which has a different character encoding.

B. Reimplementation

We implemented the following new features in PurePos in addition to the integration of a morphological analyzer interface: *i*) a lemma guesser for words unseen in the training corpus and unknown to the MA and *ii*) incremental training, i.e. additional training data can be added to the tagger model without a full recompilation of the model.

Incremental training is made possible by the fact that the model generated during training is only normalized right before tagging (i.e. we serialize the model without normalizing it). The calculation of model parameters and normalization takes very little time when loading the model.

When revising the original HunPos code, we also discovered an implementation error in it. The authors use the special tokens lexicon erroneously: the lexicon is built up using names of special token classes, but in the tagging process, the algorithm searches for special tokens themselves in the lexicon instead of the name of the class to which they belong. This erroneous behaviour was fixed in PurePos.

C. Using the integrated morphological analyzer

The integrated morphological analyzer has a twofold role in the tagger. On the one hand, it helps to determine the correct tag for words unseen in the training corpus by eliminating false tag candidates generated by the suffix guesser. As Halácsy et al. [5] showed and as is clearly demonstrated in the evaluation section, the knowledge of possible tags for a particular token yields better accuracy during tagging, since the suffix guesser may have many false guesses that can be filtered out by using a morphological table. Keeping this feature in our implementation we also introduced an interface for a morphological analyzer, which is used in a similar way.

Since the identification of lemmata is part of our goal, we need to select the most probable lemma that corresponds to each selected tag and token. In the case of words missing from the lexicon of the integrated morphological analyzer, possible lemmata are generated by a morphological guesser.

This guesser gets its input from the training data, and learns [lemma transformation, tag] pairs (TTP) for the given suffixes. A transformation is represented as the difference between the word and the lemma. A reverse trie of suffixes is built in which each node can have a weighted list containing the corresponding TPPs. Once the training is finished, the lemma guesser is used along with the MA to calculate the possible lemmata. For tokens recognized by the MA, the analyzer supplies lemmata, while in the case of out-of-vocabulary (OOV) words, the lemma guesser is used to generate possible lemmata. Each guess contains a morphosyntactic tag with which it is compatible. Having the guesses for each token and the best tag sequence for the sentence, for each token the most probable compatible lemma (having the same morphosyntactic tag) is selected, where probability is estimated as the relative frequency of the lemma (given its main PoS category) in the training set.

The tagger uses the same rich tagset that is used in the training corpus and the MA without any mapping. In the case of the Szeged corpus this amounts to 1030 different tags.

IV. Evaluation

We compared our implementation with other state-of-the-art PoS taggers in terms of tagging accuracy, such as HunPos and the maximum entropy based OpenNLP tagger. The evaluation was done on a modified version of the Hungarian Szeged Corpus [3] in which the morphosyntactic annotation was converted into a form that is compatible with that of the HUMor [10] morphological analyzer that we interfaced with our tagger. Conversion was complicated by the fact that many closed class grammatical words like conjunctions, pronouns, postpositions and sentence adverbials are differently categorized in the two formalisms. At the same time, lemmata also had to be converted to be compatible with the used analyzer's output. HUMor can yield a richer analysis including a number of productive derivational affixes (e.g. participles, gerunds etc.) which results in a different lemma than the one in the original Szeged Corpus. Since this conversion was done automatically – that might contain errors in the case of lemmata – we had to create a handcrafted test set, on which the lemmatization tests were run. In addition to reporting accuracy of morphosyntactic tagging, we also present the results of lemmatization accuracy and a combined accuracy of morphological annotation of the tool (i.e. both the lemma and the tag must be identical to that in the gold standard for the annotation of the token to be accepted as correct).

Comparison of PurePos with HunPos and OpenNLP is only possible in terms of tagging accuracy as the latter perform no lemmatization. We know of only one freely available tool for Hungarian: magyarlanc [14] that performs full morphological annotation using a similar approach. Direct comparison of its performance with PurePos is not possible due to the difference in the set of tags and lemmata produced by the two systems, so, unfortunately, we currently cannot present a comparison of the accuracy of the two systems.

Since our algorithm is entirely modelled on that of HunPos, it doesn't really make sense to compare them in the regular way: using a morphological table of the text to be processed and using a built in MA to analyze the same words yield the same result. If we do this we get almost exactly the same results. The differences in the output of HunPos and PurePos mainly originate in having fixed a few relatively unimportant implementation errors in HunPos and another part of them are cases, where there is an indeterminacy in the exact set of paths to follow during beam search. While small bug fixes result in about less than 0.01% improvement (and only in those cases where the training set is small enough) the beam search indeterminacy accounts for even less difference.

	Guesser	Gue.+MT	Gue.+MA
Tagging acc.	98.14%	98.99%	98.99%
Lemmatization acc.	90.58%	91.02%	99.08%
Combined acc.	89.79%	90.35%	**98.35**%

TABLE I
Morphological disambiguation accuracy per token

	Guesser	Gue.+MT	Gue.+MA
Tagging acc.	75.08%	85.21%	85.21%
Lemmatization acc.	29.17%	30.74%	87.13%
Combined acc.	26.17%	28.05%	**78.11**%

TABLE II
Morphological disambiguation accuracy per sentence

	Accuracy
PurePos (with MA)	**98.99**%
PurePos (without MA)	98.14%
OpenNLP perceptron	97.16%
OpenNLP maxent	96.45%

TABLE III
Comparison of part-of-speech tagging accuracy

Table I and II present tagging, lemmatization and combined (full morphological disambiguation) accuracy of PurePos. The version termed "Guesser" in the tables uses only information learned from the corpus both for tagging unseen words and lemmatization of all tokens. The version termed "Gue.+MT" uses an off-line generated morphological table that lists all possible tags of all tokens in the test set that was generated using the HUMor morphological analyzer. However, this version still uses only the lemma guesser for lemmatization. "Gue.+MA" uses the integrated MA both for tagging and lemmatization. In this setup, the guesser is only used for OOV words. Table I shows token accuracy and table II shows sentence accuracy using 90% as training set and 10% as test set of the modified Szeged Corpus. The results clearly show that using only the lemma guesser yields mediocre results, but if the MA is applied for lemmatization for words in its vocabulary, accuracy significantly increases. Comparing PurePos (with and without MA) with OpenNLP (in table III) on the same training and test set (both with the Maximum Entropy and Perceptron learning package) shows that our system is competitive with nowadays popular systems.

In addition, we examined the learning curve of the system

modelling the incremental creation of an annotated corpus from scratch. For this we followed the iterative workflow described above. We used the following systems: *i*) the reimplemented HunPos algorithm with no analyzer integrated (as a baseline) using the trained lemma guesser to perform lemmatization, *ii*) the same with a constant morphological table containing the 100000 most frequent words from the Hungarian Webcorpus [7], [4] (an unannotated corpus independent of Szeged Corpus) and *iii*) full PurePos with the integrated analyzer that performs lemmatization as well.

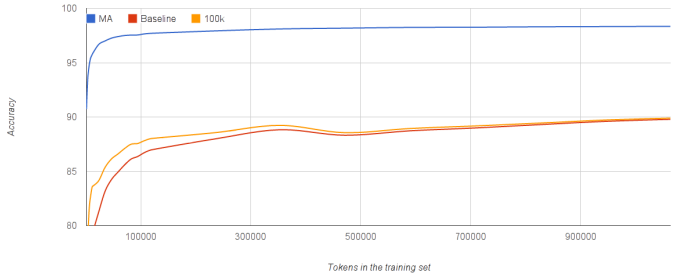


Fig. 1. Learning curve of full disambiguation accuracy

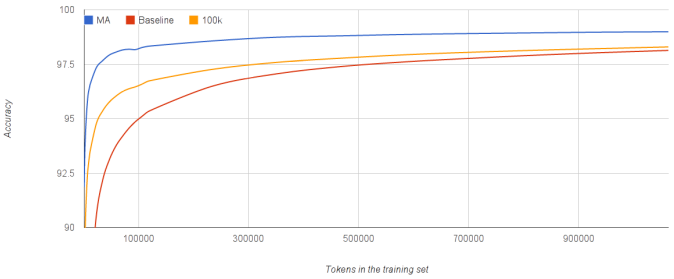


Fig. 2. Learning curve of PoS tagging accuracy

The integrated basic lemma learning algorithm – in the case of a training set of an adequate size – has a lemmatization accuracy of about 80–90% that can be used as a baseline. Its relatively low performance is due to overregularization of frequent irregular words. Figure 1 and 2 show that there is a clear advantage of having an integrated morphological analyzer to handle word forms missing from the training corpus and to do lemmatization. The advantage is striking at the initial phase of the corpus creation process. Although it becomes less pronounced as more and more training corpus is available, even with a 1 million word training corpus the tagger combined with the MA produces only about half as many tagging errors as the version containing no analyzer.

V. Conclusion

In our paper, we presented a new stochastic morphological tagger that has state-of-the-art morphological disambiguation accuracy using an interface for a morphological analyzer. The POS tagging performance of the tool is higher than that of OpenNLP and as high as that of HunPos, another open source tagger, however, it is easier to integrate and modify due to being implemented in Java. It can handle Unicode input and it performs full disambiguated morphological analysis, not just morphosyntactic tagging. It is fast to train and use thus it could be a tool of choice for corpus annotation projects for less resourced languages.

References

[1] Jason Baldridge, Thomas Morton, and Gann Bierner. The OpenNLP maximum entropy package. Technical report, 2002.

[2] Thorsten Brants. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the sixth conference on Applied Natural Language Processing*, number i, pages 224–231. Universität des Saarlandes, Computational Linguistics, Association for Computational Linguistics, 2000.

[3] Dóra Csendes, János Csirik, and Tibor Gyimóthy. The Szeged Corpus: A POS tagged and syntactically annotated Hungarian natural language corpus. In *Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora LINC 2004 at The 20th International Conference on Computational Linguistics COLING 2004*, pages 19–23, 2004.

[4] Péter Halácsy, András Kornai, László Németh, András Rung, István Szakadát, and Viktor Trón. Creating open language resources for Hungarian. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004.

[5] Péter Halácsy, András Kornai, and Csaba Oravecz. HunPos: an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 209–212, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[6] Péter Halácsy, András. Kornai, Csaba Oravecz, Viktor Trón, and Dániel Varga. Using a morphological analyzer in high precision POS tagging of Hungarian. In *5th edition of the International Conference on Language Resources and Evaulation*, pages 2245–2248, 2006.

[7] András Kornai, Péter Halácsy, Viktor Nagy, Csaba Oravecz, Viktor Trón, and Dániel Varga. Web-based frequency dictionaries for medium density languages. In Adam Kilgarriff and Marco Baroni, editors, *Proceedings of the 2nd International Workshop on Web as Corpus*, 2006.

[8] Attila Novák, György Orosz, and Indig Balázs. Javában taggelünk. In Attila Tanács and Veronika Vincze, editors, *VIII. Magyar Számítógépes Nyelvészeti Konferencia*, page 336, Szeged, 2011.

[9] Csaba Oravecz and Péter Dienes. Efficient Stochastic Part-of-Speech Tagging for Hungarian. In *Third International Conference on Language Resources and Evaluation*, pages 710–717, 2002.

[10] Gábor Prószéky and Attila Novák. Computational Morphologies for Small Uralic Languages. In *Inquiries into Words, Constraints and Contexts.*, pages 150–157, Stanford, California, 2005.

[11] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, volume 1, pages 133–142, 1996.

[12] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In Marti Hearst and Mari Ostendorf, editors, *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, number June, pages 173–180, Edmonton, Canada, 2003. Association for Computational Linguistics.

[13] Veronika Vincze, D. Szauter, A. Almási, G. Móra, Z. Alexin, and J. Csirik. Hungarian Dependency Treebank. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, number 1, pages 1–5, 2010.

[14] János Zsibrita, István Nagy, and Richárd Farkas. Magyar nyelvi elemző modulok az UIMA keretrendszerhez. In Attila Tanács, Dóra Szauter, and Veronika Vincze, editors, *VI. Magyar Számítógépes Nyelvészeti Konferencia*, pages 394–395, Szeged, 2009. Szegedi Tudományegyetem.

Automatic Structuring and Spelling Correction of Hungarian Clinical Records

Borbála Siklósi
(Supervisor: Dr Gábor Prószéky)
siklosi.borbala@itk.ppke.hu

***Abstract*— The first steps of processing clinical documents are structuring and normalization. In this paper we demonstrate how to compensate the lack of any structure in raw data by transforming simple formatting features automatically to structural units. Then we developed an algorithm to separate running text from tabular and numerical data. Finally we generated correcting suggestions for word forms recognized to be incorrect. Some evaluation results are also provided for using the system as automatically correcting input texts by choosing the best possible suggestion from the generated list. Our method is based on the statistical characteristics of our Hungarian clinical data set and on the HUMor morphological analyzer. The conclusions claim that our algorithm is not able to correct all mistakes by itself, but is a very powerful tool to help manually correcting Hungarian medical texts in order to produce a correct text corpus of such a domain.**

***Keywords*- spelling correction; clinical text mining; language models; morphology; agglutinative; biomedical corpora**

I. INTRODUCTION

In most hospitals medical records are only used for archiving and documenting a patient's medical history. Though it has been quite a long time since hospitals started using digital ways for written text document creation instead of handwriting and they have produced a huge amount of domain specific data, they later use them only to look-up the medical history of individual patients. Digitized records of patients' medical history could be used for a much wider range of purposes. Language technology, ontologies and statistical algorithms make a deeper analysis of text possible, which may open the prospect of exploration of hidden information inherent in the texts. However, the way clinical records are currently stored in Hungarian hospitals does not even make free text search possible.

Aiming at such a goal, i.e. implementing an intelligent medical system requires a robust representation of data. This includes well determined relations between and within the records and filling these structures with valid textual data. In this paper we describe how the structure of the medical records is established and the method of automatic transformation. Then, after the elimination of non-textual data, we demonstrate a basic method for correcting spelling errors in the textual parts with an algorithm that is able to handle both the language and domain specific phenomena.

II. REPRESENTATION OF MEDICAL TEXTS

We were provided anonymized clinical records from various departments, we chose one of them, i.e. ophthalmology to build the system that can be extended later to other departments as well. Due to the lack of a sophisticated clinical documentation system, the structure of raw medical documents can only be inspected in the formatting or by understanding the actual content. Besides basic separations - that are not even unified through documents - there were no other aspects of determining structural units. Moreover a significant portion of the records were redundant: medical history of a patient is sometimes copied to later documents at least partially, making subsequent documents longer without additional information regarding the content itself. However these repetitions will provide the base of linking each segment of a long lasting medical process.

A. XML structure

Wide-spread practice for representing structure of texts is to use XML to describe each part of the document. In our case it is not only for storing data in a standard format, but also representing the identified internal structure of the texts which are recognized by basic text mining procedures, such as transforming formatting elements to structural identifiers or applying recognition algorithms for certain surface patterns. After tagging the available metadata and performing these transformations the structural units of the medical records are the followings:

- content: parts of the records that are in free text form are further divided into sections such as header, diagnoses, applied treatments, status, operation, symptoms, etc.
- metadata: we automatically tagged such units as the type of the record, name of the institution and department, diagnoses represented in tabular forms and standard encodings of health related concepts.
- simple named entities: dates, doctors, operations, etc. The medical language is very sensitive to named entities, that is why handling them requires much more sophisticated algorithms, which are a matter of further research.
- medical history: with the help of repeated sections of medical records related to one certain patient, we have

been able to build a simple network of medical processes. Since the documentation of medical history is not standardized and not consequent even for the same patient, the correspondence is determined by partial string matching and comparing algorithms. Thus we can store the identifiers of the preceding and following records.

B. Separating Textual and Non-Textual Data

The resulting structure defines the separable parts of each record; however there are still several types of data within these structural units. Non-textual information inserted into free word descriptions are laboratory test results, numerical values, delimiting character series and longer chains of abbreviations and special characters. We filtered out these expressions to get a set of records containing only natural text. To solve this issue we applied the unsupervised methods of clustering algorithms. Since there are hardly any sentence boundaries in the classical sense, our basic units were lines (i.e. units separated with newline character) and concatenations of multiple lines where neighbouring lines were suspected to be continuation of each other. This continuation does not apply to the semantic content of the lines, rather to their behaviour regarding textual or non-textual form of information. Thus such short textual fragments were kept together with more representative neighbours avoiding them to be filtered out by themselves, since their feature characteristics are very similar to those of non-textual lines. Testing the efficiency of our feature set and clustering algorithm, a simple Naive Bayes classifier performed 98% accuracy on a data set of 100 lines. Portions considered to be textual information need to be normalized in terms of punctuation, spelling and the used abbreviations. A fault tolerant tokenization is applied to the running text that takes into account domain specific phenomena.

III. Spelling Correction

A. Language and Domain Specific Difficulties

In our case of processing clinical narratives, the agglutinating and compounding behavior of Hungarian language yields a huge number of different word forms and free word order in sentences render solutions applicable to English unfeasible. For the detailed characteristics of the distribution of Hungarian word forms see [7].

Moreover, medical language contains additional difficulties. Since these records are not written by clinical experts (and there is no spell-checker in the software they use) they contain many errors of the following types:

- typing errors occurring during text input mainly by accidentally swapping letters, inserting extra letters or just missing some,
- the misuse of punctuation,
- substandard spelling with especially many errors arising from the use of special medical language with a non-standard spelling that is a haphazard mixture of what would be the standard Latin and Hungarian spelling (e.g. tension / tenzió / tenzio / tensió). Though there exists a theoretical standard for the use of such medical expressions, doctors tend to develop their own customs and it is quite difficult for even an expert to choose the right form.

Besides these errors, there are many additional difficulties that must be handled in a text mining system, which are also consequences of the special use of the language. When writing a clinical record, doctors or assistants often use short incomplete phrases instead of full sentences. The use of abbreviations does not follow any standards in the documents. Assistants do not only use standard abbreviations but abbreviate many common words as well in a rather random manner and abbreviations rarely end in a period as they should in standard orthography. Moreover, the set of abbreviations used is domain specific and also varies with the doctor or assistant typing the text. In some extreme situations it might happen that a misspelled word in one document is an intentional abbreviation or short form in the other.

For the identification of an appropriate error model of the spelling errors, a corpus of corrected clinical records is needed. There is no such corpus at all for Hungarian medical language, thus we needed to create a corrected version of our real-life medical corpus. This was necessarily a partly manual process for a subset of the corpus, but we wanted to make the correction process as efficient as possible. Our goal was to recognize misspelled word forms and automatically present possible corrections in a ranked order. Additional algorithms with manual validation could then choose the final form, which is much easier than correcting the whole corpus by hand, moreover the baseline system might be easily extended to be able to carry out the whole process trained on the already corrected corpus.

B. Combination of Language Models

Aiming at such a goal, a simple linear model was built to provide the most probable suggestions for each misspelled word. We combined several language models built on the original data set and on external resources, that are the followings (the first two used as prefilters before suggesting corrections, the rest were used for generating the suggestions):

- stopword list: a general stopword list for Hungarian was extended with the most common words present in our medical corpus.
- abbreviation list: after automatically selecting possible abbreviations in the corpus, the generated list was manually filtered. Since we have not applied expert knowledge, this list should be more sophisticated for further use.
- list of word forms licensed by morphology: those word forms that are accepted by our Hungarian morphology (HUMor [10]) were selected from the original corpus, creating a list of potentially correct word forms. To be able to handle different forms of

medical expressions, the morphology was extended with lists of medicine names, substances and the content of the Hungarian medical dictionary. We built a unigram model from these accepted word forms.

- list of word forms not licensed by morphology: the frequency distribution of these words were taken into consideration in two ways when generating suggestions. Words appearing just a few times in the corpus remained as unaccepted forms. Those ones however, whose frequency was higher than the predefined threshold were considered to be good forms, even though they were denied by the morphology. Our assumption was that it is less possible to consequently use the same erroneous word form than being that form correct and contradicting our morphology.
- general and domain specific corpora: we built unigram models similar to that of the above described licensed word forms from the Hungarian Szeged Korpusz and from the descriptions of the entities in the ICD coding system documentation. We assumed that both of these corpora contains only correct word forms.

C. *Generating Possibly Correct Suggestions*

As the next step we filtered out those word forms that are not to be corrected. These were the ones contained in the stopword and abbreviation lists. For the rest of the words the correction suggestion algorithm is applied. For each word a list of suggestion candidates is generated that contains the word forms with one unit of Levenshtein distance [5] and the possible suggestions generated by the morphology. Then these candidates are ordered with a weighted linear combination of the different language models, the weight of the Levenshtein generation and the features of the original word form. Thus a weighted suggestion list is generated to all words in the text (except for the abbreviations and stopwords), but only those will be considered to be relevant, where the score of the best weighted suggestion is higher than that of the original word. At the end we considered the ten best suggestions.

IV. RESULTS

We investigated the performance of the system as a standalone automatic correcting tool, accepting the best weighted suggestion as the correction, but also as an aiding system that is only to help manual correction at this initial state. Since we did not have a correct test set, we had to create one manually by correcting a portion of our medical corpus. Our test set contained 100 paragraphs randomly chosen from the corpus. We used three metrics for evaluation:

- precision: measures how the number of properly corrected suggestions relates to the number of all corrections, considering the best weighted suggestion as correction
- recall: measures the ratio of the number of properly corrected suggestions and the number of misspelled word forms in the original text
- f-measure: the average of the above two

We investigated the result measures for several combinations of weighting the above described models and features:

- Models based on justification of morphology (VOC, OOV): since these models are the most representatives for the given corpus, these models were considered with the highest weight
- Models built from external resources (ICD, Szeged): these models are bigger, but they are more general, thus word forms are not that relevant for our raw texts. Our results reflect that though these models contribute to the quality of the corrections, they should have lower weights in order to keep the scores of medical words higher
- Original form (ISORIG, ORIG): the original forms of the words received two kinds of weighting: first whether if the word to be corrected is licensed by the morphology or not; second whether if the actual word is the original word form, regardless of its correctness. This was introduced so that the system would not "correct" an incorrect word form to another incorrect form, but rather keep the original one, if no real suggestions can be provided.
- Morphological judgment on suggestions (HUMor): each generated suggestion licensed by the morphology received a higher weight to ensure that the final suggestions are valid words.
- Weighted Levenshtein generation (LEV): when generating word forms that are one Levenshtein distance from the original one, we gave special weighting for more probable phenomena, such as swapping letters placed next to each other on the keyboard of a computer, improper use of long and short forms of Hungarian vowels, or mixing characteristic letters of Latin (e.g.: t-c, y-i).

TABLE 1. Results with best combination of weights

Model	**Weights**
OOV	0.05
VOC	0.25
SZEGED	0.15
ICD	0.2
HUMOR	0.15
PRECISION	**70%**
RECALL	**75%**
F-MEASURE	**72%**

The low numerical values in the table can be explained by several phenomena. The relatively small size of our test set does not reflect all types of errors. However manually creating a larger corrected text is very time and effort consuming. The system though provides great help in this task as well, so the evaluation of a generalized application will be much more accurate. Domain specific ambiguities also cause trouble at the time of evaluation. We allow the system to accept more than one correction as appropriate, but still there are several cases, where this is still a problem to decide. Thus the system might reject some correct forms while accepting other erroneous ones. The precise handling of abbreviations is still a problem, but is to be solved later on, thus it is unavoidable to fail on such fragments like "szemhéjszél idem, mérs. inj. conj, l.sin." or "Vitr. o.s. (RM) abl. ret. miatt". Human evaluation instead of the used metrics predicts much better results, which means that the readability of the texts has significantly improved.

Regarding the ranking of the suggestions, in 99.12% of the words of the test set, the 5 best suggestions contained the real correction. This means that using the system as an aiding tool for manual correction of medical texts is very powerful. An interactive user interface has been created to exploit the possibilities provided by such a feature, where the user can paste portions of medical texts, than the system highlights the words that it judged to be misspelled and offers the 5 best suggestions, from among which the user can choose. The scores are also displayed to give a hint to the user about the difference between each suggestions.

V. Further Plans

The system at its early phase has several shortcomings regarding the generation and weighting of suggestions. Several problems are discussed above, besides which two more problems are to be solved in the near future. The first is taking into account the context of a word. This could solve some ambiguous cases, where no decision can be made on the word level. The main difficulty for introducing this factor in the model is that a proper n-gram model is needed, which points back to the need of a correct corpus. The other important issue is that of multiple-word expressions. At its present stage the system is not able to correct such cases, when two words are written together without space between them, or vice versa. As our test set contains examples for all these unhandled appearances, the evaluation metrics would surely be improved if these problems were solved.

VI. Conclusion

The primary goal of developing our baseline algorithm was to aid the creation of a correct, reliable Hungarian medical text corpus. Having reached this goal, a more precise error model can be built to use for training a more improved system. As the results reflected, this motivation is fulfilled, since our correcting algorithm is quite efficient for such a basic aspect. The result of the system by itself could lead to several useful applications, such as at the background of a medical search engine, where both the query, and the actual result texts could be extended by other suggested forms of each word, making it possible to retrieve valuable information even if some misspellings are present on either side. The basic tagging and structuring described in the first part of this paper is also useful for storing, organizing and easier retrieving of the data. We demonstrated that the creation of an intelligent clinical system built on the knowledge lying in medical records is not trivial even in the preprocessing phase. However after some iterative application of the combination of automatic and manual work, a gradually improved corpus can be available, finally making the whole process automatic.

References

[1] Brill, E., Moore, R.C., "An improved error model for noisy channel spelling correction" Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, 2000, pp. 286–293

[2] Contractor, D., Faruquie, T.A., Subramaniam,L.V. "Unsupervised cleansing of noisy text." Proceedings of the 23rd International Conference on Computational Linguistics,2010, pp. 189--196.

[3] Farkas, R., Szarvas, Gy. "Automatic construction of rule-based ICD-9-CM coding systems", BMC Bioinformatics, 9., 2008.

[4] Heinze, D.T., Morsch, M.L., Holbrook, J., "Mining free-text medical records", A-Life Medical, Incorporated, pp.254—258., 2001.

[5] Levenshtein V. "Binary codes capable of correcting spurious insertions and deletions of ones". Problems of Information Transmission, 1965, 1(1): pp. 8—17.

[6] Mykowiecka, A., Marciniak, M. "Domain-driven automatic spelling correction for mammography reports." Intelligent Information Processing and Web Mining Proceedings of the International IIS: IIPWM'06. Advances in Soft Computing, Heidelberg, 2006.

[7] Oravecz, Cs., Dienes, P. " Efficient stochastic Part-of-Speech tagging for Hungarian" Third International Conference on Language Resources and Evaluation,2002, pp. 710—717.

[8] Patrick J., Sabbagh, M., Jain, S., Zheng, H. "Spelling correction in clinical notes with emphasis on first suggestion accuracy" 2nd Workshop on Building and Evaluating Resources for Biomedical Text Mining, 2010,pp. 2--8.

[9] Pirinen, T.A., Lindén, K. " Finite-state spell-checking with weighted language and error models – Building and evaluating spell-checkers with Wikipedia as corpus" SaLTMiL Workshop on Creation and Use of Basic Lexical Resources for Less-Resourced Languages, LREC 2010,2010, pp.13—18.

[10] Prószéky, G., Novák, A. " Computational morphologies for small Uralic languages" Inquiries into Words, Constraints and Contexts, 2005, pp 150—157.

[11] Rebholz-Schuhmann, D., Kirsch, H., Gaudan, S., Arregui, M., Nenadic, G. "Annotation and disambiguation of semantic types in biomedical text: a cascaded approach to named entity recognition". Proceedings of the EACL Workshop on Multi-Dimensional Markup in NLP,2005.

[12] Stevenson M., Guo, Y., Al Amri, A., Gaizauskas, R. "Disambiguation of biomedical abbreviations." Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, 2009, pp. 71.

Predicting Effective Drug Combination Using Protein-Protein Interaction Networks

Balázs Oláh
(Supervisor: Dr. Sándor Pongor)
olah.balazs@itk.ppke.hu

***Abstract*—Drug combinations are proved to be efficient in treating complex diseases such as cancer, diabetes, arthritis and hypertension. However designing such drug combinations is not an easy task due to its complexity. Most drug combinations were found in empirical way thus there is a lack of efficient computational methods. In this paper I will present a novel network analysis based method, which considers the complexity of the interaction between drugs - such as crosstalk and other network phenomena - through perturbation analysis (on the STRING protein-protein association network) generated by underlying drugs. The method also reveals the underlying mechanism of drug actions. Results show that those drugs can form efficient combinations, that have a large number of common perturbed proteins, even if their targets can be found in unrelated pathways.**

***Keywords*-drug combination;drug interaction, antagonism, synergy;**

I. Introduction

In the past few decades the number of novel marketed drugs have fallen much below the expectations despite the growing financial and other resources on this area [1], [2], [3]. Drugs designed by one drug - one target drug design strategy often fail at phase III or phase IV due to their side effects or the low therapeutic effect [1], [4] since the biological systems are robust against various kind of perturbations [3], [5] such as toxins, chemical compounds, mutations. The biological pathways are often redundant, diverse and modular. Furthermore they are rich in negative feedbacks, positive feedbacks, feed-forward and other regulatory loops that can compensate the effect of perturbations. Using multi-target drugs or drug combination can overcome that limitation since they modulate multiple proteins in the same time, thus they might be able to efficiently control the system. Ágoston et al. [6], [7], [8] showed that multiple partial knockout of targets is more efficient than single knockout. In addition drug combinations have lower toxicity and therapeutic selectivity [9]. Instead of developing highly selective compounds one should try to use multitarget drugs or drug combinations as a drug discovery paradigm and the available information about the complex biological system as well [10]. However finding efficient target and dose combinations, having proper therapeutic effect is hard since it is a combinatorical problem [11] and the complexity of the biological system also has to be considered. Still the number of approved drug combinations is increasing, most of them were found by experience and intuition [12], [13]. Several experimental methods, even high throughput methods, have been developed for measuring the efficiency of drug combinations, such as Bliss independence or Loewe additivity [14], [15]. This kind of exhaustive search is impractical so Wong et al. used a stochastic search algorithm and Calzoari et al. sequential decoding algorithms to find the best combinations [16], [17]. Yang et al. use differential equations to find a perturbation pattern that can alter the system to a normal state from a disease state [18]. Jin et al. wanted to understand how a synergism of two drugs arises using microarray data and a Petri net based model [19]. The common in these methods is that they require a large number of experiments or deep knowledge of the kinetic parameters of the pathways. Other methods use data mining algorithms to integrate pharmacological and network data [20], [21], [22]. Li et al. used the concept of network centrality and disease similarity to prioritize drug combinations [23]. Wu et al. used the microarray profile of the individual drugs for the predictions [24], while others use the concept of synthetic lethality and the available gene interaction data [25], [26]. In this paper I present a novel drug combination prediction algorithm which is based on the assumption that the perturbations generated by the drugs propagate through the possible interactions between proteins.

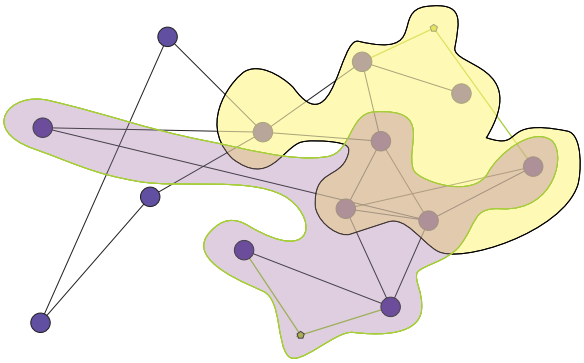


Fig. 1. *This figure shows a hypothesis how the synergism or the antagonism of two drugs can arise. The yellow pentagons represent the drugs, while the blue ones are the proteins. The drugs have some effect on their targets and this effect will propagate to the target neighbourhood through the network edges. The blue and yellow area are the subnetworks consisting of the most affected proteins. The proteins in the intersection affected by both drugs may explain the synergistic or antagonistic effect.*

II. Method and experimental design

The drug molecule has an effect on its targets for example by inhibiting or activating the function of the target. Since the proteins are linked and they can influence each others' function if one affects the target molecule then this also alters the proteins interacting with it and the disrupted neighbours also have neigbours thus the disruption starts to propagate. This phenomenon can be described by random walk or diffusion models and one can also exploit the most affected proteins and their function using gene ontologies, predifined functional

gene sets or graph clustering algorithms. When the drugs are administrated in combination then antagonistic or synergistic effect can occur. I assume that the proteins affected by the components simultanously can explain the synergism and antagonism phenomena (**figure 1**). The compounds alone maybe can not make a perturbation large enough to modulate the proteins or pathways causing a therapeutic effect, but together they can. My goal is to find compounds forming efficient drug combinations assuming that in the case of these good combinations several groups of proteins are affected by each component. For that purpose I used the famous PageRank algorithm, which is a simple random walk on a network and is succesfully used to prioritize disease candidate genes based on a similar hypothesis [27], [28], [29], [30]. In this paper I only consider two components combinations. The network is a graph $G(V, E)$ where V, E are the set of nodes and edges respectively. In this case the nodes represent genes or proteins, and the edges are the associations between them. The edges may have a weight, which can be interpreted as an association strength. Let A be the adjancency matrix of the graph. The element a_{ij} is the weight of the edge between node i and j, if there is no edge then it is 0. One could define a random walk on that graph by rescaling the edges to transition probabilites. Let the M be a stochastic matrix of the graph $G(V, E)$, then m_{ij} is the probability of going to node j from node i.

$$M = D^{-1}A$$

Where D is a diagonal matrix:

$$D = diag(d_1, d_2, \ldots, d_N) \tag{1}$$

where $d_i = \sum_{j=1}^{|V|} A_{ij}$.

$$P^{k+1} = M^T P^k = (M^T)^k P^0$$

Where P^k is a probability distribution, so p_i^k is the probability of being at node i in the step k. P^0 is the initial probability distribution vector, which are the probabilities of starting the random walk at a given a node.

a) PageRank: The PageRank with prior [31] is a modified random walk, where in each step the random walker jumps back to one of the initial nodes or continues the travelling with a certain probability.

$$P^{(i+1)} = (1-\alpha)\left(M^T P^{(i)}\right) + \alpha P^0 \tag{2}$$

$$p_i^0 = \begin{cases} \frac{1}{|N_T|}, & \text{if the protein } i \text{ is drug target} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where N_T is the number of drug targets.

A. Randomizations and the drug affected proteins (DAP)

In order to find the subset of the drug affected proteins a Monte Carlo simulation procedure was used. In protein interaction network there are nodes wich are more central or more important thus these are more likely to be reached by chance. To avoid this situation randomization procedure was applied to estimate statistical significance of each gene [32]. However the PageRank iteration converges (if some conditions are fulfilled) the null distribution depending on the number of steps (if k is small) taken by the random walker, on the parameter α and on the number of nonzero elements of P^0. Usually 50-100 thousand randomizations are required for a proper estimate [32], however I found that 10000 is enough. Since the randomization is a computationally demanding task parametric (Gaussian) and non-parametric (e.g. Kernel density estimation) estimation methods were tried, but only the brute-force solution was satisfying. If we have p-values then we can define the set of drug affected proteins (DAPs) as follows:

$$DAP = \{v_j | v_j \in V, p_j < 0.05\}$$

B. Measuring the interaction strength

I assumed that the sets of DAPs of the interacting drugs are largely overlapping, which is measured by the Jaccard coefficient. It is 1 if the two sets are identical and 0 if they are disjunct.

$$J(DAP_i, DAP_j) = \frac{|DAP_i \cap DAP_j|}{|DAP_i \cup DAP_j|}, \tag{4}$$

If two drugs share targets or target pathways, then the J is probably near to 1 but this is trivial.

C. Enrichment analysis of interaction causing proteins

My hypothesis is that the proteins which are affected by both drugs explain how synergistic or antagonistic effect arises at a system biology level. For that purpose I used the concept of enrichent analysis where the goal is to find a common pattern (e.g most of the genes are parts of the same pathway) which gives unifying schemes for the genes [33]. The standard approach is based on the using of hypergeomtic distribution. The problem with this approach is that we don't use the p-value information, which measures how the protein is affected by the drug only the fact that it is affected or not. Thus I used a modified version of the GSEA (Gene set enrichment analysis) [34], which is based on Kolmogorov - Smirnov statistics and predifined function sets. The original method was developed for microarrays where the correlation between the expression of a gene and a phenotype under study is measured. However the smaller the p-value is the more affected the proteins are so it can be seen as *anti correlation* measure, thus it should be converted into *correlation* like measure r_i.

$$r_i = \frac{\log(p_i)}{\log(\frac{1}{N_R})}$$

where N_R is the number of randomization ($\frac{1}{N_R}$ is the possible smallest non-zero p-value). For the enrichment analyis one has to order the N genes descending according to r_i-s. The enrichment score is:

$$ES = \max(|P_{hit}(S,i) - P_{miss}(S,i)||i = 1, 2 \ldots N)$$

where

$$P_{hit}(S,i) = \sum_{j \in S, j \leq i} \frac{|r_j|}{\sum_{k \in S} |r_k|}$$

$$P_{miss}(S,i) = \sum_{j \notin S, j \leq i} \frac{1}{N - N_H}$$

N_H is the number of genes that are not the members of a given S gene set. Note that the index i means the ranking position. $ES(S)$ is large if the list members have low p-values. To assign a statistical significance to the gene set a similar randomization procedure followed by a t-test was used. The gene sets were downloaded from the official website of GSEA. For the experiments I used the C2 - CP (canonical pathways) datasets. I also used the gene ontologies (biological process, cellular component, molecular function), which were downloaded from the official site of GO [35].

D. Description of the experiments

All the algorithms were implemented in MATLAB 2011a. The used network was STRING [36] and the drug combination was downloaded from the drug combination database [37].

1) String: STRING (Search Tool for the Retrieval of Interacting Genes) is one of the largest integrated protein interaction databases, which covers 66.9 Mio predicted and known interactions between proteins of 1100 organisms. The majority of the interactions (44.1 Mio) are predictions. The links between the proteins are some kind of associations (among them several indirect ones) - not only physical interactions. The evidence types for the associations are: neighborhood, gene fusion, cooccurence, coexpression, experiments, databases, text mining, homology. Each type of association has a confidence score, which is a probabilistic measure of the reliability of the link. The several types of links and their confidences can be combined into one association with one confidence score. In our experiment we used only combined interactions and confidences. Only the human proteins and there combined associations were considered.

2) Drug combination database (DCDB): The known drug combination dataset was downloaded from the drug combination database [37] in sql dump file format. However the dump file contains several errors, missing and truncated tables, entries with duplicated primary keys. Fortunately the tables also have been provided as text files so the missing and truncated tables were replaced based on that data. I also added a new entry for all drug combinations (DC) which describes whether the DC is intended to be used in cancer threapy, against parasites (in that case the drug targets are not human proteins) or other therapeutic effects. Only DC-s with two components were considered. Y. Lie et al. classify the drug combinations into two classes - pharmacodynamic and pharmacokinetic interactions - based on the underlying molecular mechanism on action [37]. Pharmacokinetic interactions are where one component affects how the other component(s) are absorbed, distributed, metabolized and excreted. Thus these kinds of interactions are divided into four categories:

1) positive or negative regulation of drug transport or permeation
2) enhanced or reduced drug distributions or localizations
3) drug metabolism interaction
4) drug elimination interaction

The pharmacodynamic interactions are where

1) all individual drugs act on the same target
2) individual drugs act on different targets in the same pathway
3) individual drugs act on different targets in related pathways
4) individual drugs act on different targets in cross-talking pathways
5) individual drugs act on different targets in pathways of yet unknown relations

Furthermore the database also contains information about the therapeutic effect of DCs due to synergism, antagonism or additism.

3) Measuring the classification performance: I used the AUC (area under roc curve) for measuring the ranking performance. The output is a ranked list. Whether a drug combination belongs to a positive (good combination) or negative class (not a good combination) depends on a variable treshold [38]. If the rank of the drug combination score is lower than the given treshold then it is considered to be positive otherwise negative, thus a FPR (false positive rate) and a TPR (true positive rate) can be determined. One could generate TPR, FPR for every possible ranking treshold, thus we got a ROC curve and an AUC value.

4) Experiments: Since a low number of true negative, unsuccesful DCs are available I used artificial or random drugs for control purposes. As it was mentioned earlier the drugs' DAP depends on the number of drug targets N_T, the propagation parameter α, and the number of steps taken by the random walker (k). In my experiment I chose $k = 2$ and $\alpha = 0.5$ based on the gene prioritization experience [28], [27], [29], [30]. For each parameter combination 300 random drugs and their corresponding $DAPs$ were generated and computed. Then the true combinations were compared to 100 random drugs (resampled from the 300 artifical drugs) and an AUC value was optained. For negative control I used random drugs against random drugs (the expected AUC is 0.5). I repeated this procedure 100 times and an average AUC and an average control AUC was generated.

III. Results and discussion

Table I shows the average result of the different DC categories. As it is expected the random generated AUCs are around 0.5. It is not surprising that if the targets of different drugs are in the same pathway or the same protein, then their sets of $DAPs$ are heavily overlapping. The most interesting part in the results is that drug combinations having different targets of unrelated pathways have a high score, which may implicate that the initial hypothesis is true. The high AUC value for the pharmacokinetical interactions is also surprising, however further inference should not be made due to the small sample size (avg number of DCs is 2.8).

DC type	action type	avg. AUC	avg. r. AUC	number of DCs
Pharmacodynamical	Different targets of related pathways	0.8774	0.4896	54
Pharmacodynamical	Different targets of the same pathway	0.9018	0.4920	5
Pharmacodynamical	Different targets of unrelated pathways	0.8885	0.5008	42
Pharmacodynamical	Same target	0.9790	0.4923	3
Pharmacokinetical	Distribution	0.9183	0.5032	5
Pharmacokinetical	Excretion	0.8312	0.5131	3
Pharmacokinetical	Metabolism	0.6755	0.4989	2
Pharmacokinetical	Reduced drug distribution or localization	1.0000	0.5281	1

TABLE I

The first two column show the pharmacological classification and the action type of the DCs, the avg. AUC and avg. r. AUC are the average AUC values that measures how much the sets of DAPs of DC components overlap. (for more detail look the Experiments section), the last columns show the number of drug combinations having the classification and action type

A. Limitation of the method

The main limitation of that method is that it is not possible to determine whether the interaction between two drugs is synergetic, antagonistic or additive, only the existance of the interaction can be presumed. It is neither possible to decide whether the pathways affected by DC components are upregulated or downregulated. Because of the scarness of the available information about the interaction between proteins such as the direction, interaction strenght, interaction type (inhibiton, activiation, binding, phosphorization, etc.) the above limitations will exist.

IV. Conclusions

In the paper I presented a novel network based strategy to predict efficient drug combinations based on the hypothesis that the compound generates a disruption which propagates through the network. Those drugs can make efficient combinations that have a large number of common perturbed proteins, that can be simply measured by the Jaccard coefficient. A modified gene set enrichment method was used for explaining how the therapeutic effect of the drug combination may emerge since the proteins which are affected by the individual components are known. For testing the hypothesis the drug combination database was used, which contains several hundreds of known drug combinations. However the method has some limitations for example it can not be predicted whether the interaction is synergetic or antagonistic due to the scarness of information about protein interactions.

Acknowledgment

This project was developed within the PhD program of the Multidisciplinary Doctoral School, Faculty of Information Technology, Pázmány Péter Catholic University, Budapest. Thanks are due to my supervisor, Prof. Sándor Pongor for his help and guidiance throughout the project. I am grateful to Gergely Lukács for his advices and help in creating and filtering the drug combination database.

References

[1] A. L. Hopkins, "Network pharmacology: the next paradigm in drug discovery," Nature Chemical Biology, vol. 4, 682–690, 2008.

[2] P. Imming, C. Sinning, and A. Meyer, "Drugs, their targets and nature and number of drug targets," Nature Reviews Drug Discovery, vol. 5, p. 821–834, 2006.

[3] K. Hiroaki, "A robustness-based approach to systems-oriented drug design," Nature Reviews Drug Discovery, vol. 5, p. 202–210, 2007.

[4] S. I. Berger, and R. Iyengar, "Network analyses in systems pharmacology," Bioinformatics, vol. 25, no. 19, 2466–2472, 2008.

[5] K. Hiroaki, "Biological Robustness," Nature Reviews Genetics, vol. 5, p. 826–837, 2007.

[6] V. Ágoston, P. Csermely, and S. Pongor, "Multiple weak hits confuse complex systems: A transcriptional regulatory network as an example," Physical Review E, vol. 71, no. 5, http://link.aps.org/doi/10.1103/PhysRevE.71.051909, 2005.

[7] P. Csermely, V. Ágoston, S. Pongor, "The efficiency of multi-target drugs: the network approach might help drug design," Trends in Pharmacological Sciences, vol. 26, p. 178–182, 2005.

[8] T. Korcsmáros, M. S Szalay, Cs. Böde, I. A Kovács, P. Csermely, "How to design multi-target drugs: Target search options in cellular networks," Expert Opinion on Drug Discovery, vol. 2, p. 1–10, 2007.

[9] J. Lehár, A. S. Krueger, W. Avery, A. M. Heilbut, L. M. Johansen, E. R. Price, R. J. Rickles, G. F. Short III, J. E. Staunton, X. Jin, M. S. Lee, G. R. Zimmermann, and A. A. Borisy, "Synergestic drug combinations tend to improve therapeutically relevant selectivity," Nature Biotechnology, vol. 27, p. 659–666, 2009.

[10] A-L. Barabási, N. Gulbahce, and J. Loscalzo, "Network medicine: a network-based approach to human disease," Nature Reviews Genetics, vol. 12, no. 1, p. 56–68, 2011.

[11] J. D. Feala, J. Cortes, P. M. Duxbury, C. Piermarocchi, A. D. McCulloch, and G. Paternostro, "System approaches and algorithms for discovery of combinatorial therapies," Wiley Interdisciplinary Reviews: Systems Biology and Medicine, vol.2, no. 2, p. 181–193, 2010.

[12] C. T. Keith, A. A. Borisy, and B. R. Stockwell, "Multicomponent therapeutics for networked systems," Nature Reviews Drug Discovery, vol. 4, p. 71–78, 2005.

[13] G. R. Zimmermann, J. Lehár, and C. T. Keith, "Multi-target therapeutics: when the whole is greater than the sum of the parts," Drug Discovery Today, vol. 12, no. 1-2, p. 34–42, 2006.

[14] W. R. Greco, G. Bravo, and J. C. Parsons, "The search for synergy: a critical review from a response surface perspective," Pharmacological Reviews, vol. 47, no. 2, p. 331–385, 1995.

[15] A. A. Borisy, P. J. Elliott, N. W. Hurst, M. S. Lee, J. Lehar, E. R. Price, G. Serbedzija, G. R. Zimmermann, M. A. Foley, B. R. Stockwell, and C. T. Keith, "Systematic discovery of multicomponent therapeutics," Proceedings of the National Academy of Sciences, vol. 100, no. 13, p. 7977–82, 2003.

[16] P. K. Wong, F. Yu, A. Shahangian, G. Cheng, R. Sun, and C. M. Ho, "Closed loop control of cellular functions using combinatory drugs guided by a stochastic search algorithm," Proceedings of the National Academy of Sciences of the United States of America, vol. 105, no. 13, p. 5105–10, 2008.

[17] D. Calzolari, S. Bruschi, L. Coquin, J. Schofield, J. D. Feala, J. C. Reed, A. D. McCulloch, and G. Paternostro, "Search algorithm as a framework for the optimization of drug combinations," PLoS Computational Biology, vol. 4, no. 12, http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2590660/, 2008.

[18] K. Yang, H. Bai, Q. Ouyang, L. Lai, and C. Tang, "Finding multiple target optimal intervention in disease-related molecular network," Molecular Systems Biology, vol. 4:228, Epub 2008 Nov 4.

[19] G. Jin, H. Zhao, X. Zhou, and S. T. Wong, "An enhanced Petri-net model

to predict synergistic effects of pairwise drug combinations from gene microarray data," Bioinformatics, vol. 27, no. 13, p. 310–316, 2011.
[20] X. M. Zhao, M. Iskar, G. Zeller, M. Kuhn, V. Noort, and P. Bork, "Prediction of drug combinations by integrating molecular and Pharmacological data," PLoS Computational Biology, vol. 7, no. 12:e1002323, Epub 2011 Dec 29.
[21] H. Fu-Yan, S. Jiangning, and Z. Xing-Ming, "Exploring Drug Combinations in a Drug-Cocktail Network, " IEEE International Conference on Systems Biology (ISB), Zhuhai, China, September 2 - 4, 2011, p. 382–387.
[22] X. Ke-Jia, S. Jiangning and Z. Xing-Ming, "A network biology approach to understand combination of drugs," The Fourth International Conference on Computational Systems Biology (ISB2010); 2010, p. 347–354.
[23] S. Li, B. Zhang, and N. Zhang, "Network target for screening synergistic drug combinations with application to traditional Chinese medicine," BMC Systems Biology vol. 5, Suppl 1:S10, 2011.
[24] Z. Wu, X. M. Zhao, and L. Chen, "A system biology approach to identify effective cocktail drugs," BMC Systems Biology, vol. 4, Suppl 2:S7, 2010.
[25] M. Cokol, H. N. Chua, M. Tasan, B. Mutlu, Z. B. Weinstein, Y. Suzuki, M. E. Nergiz, M. Costanzo, A. Baryshnikova, G. Giaever, C. Nislow, C. L. Myers, B. J. Andrews, C. Boone, and F. P. Roth, "Systematic exploration of synergistic drug pairs," Molecular Systems Biology, vol. 7:544. doi: 10.1038/msb.2011.71, 2011.
[26] J. Xiong, J. Liu, S. Rayner, Z. Tian, Y. Li, and S. Chen, "Pre-Clinical Drug Prioritization via Prognosis-guided genetic interaction networks," PLoS One, vol. 5, no. 11:e13937, 2010.
[27] S. Köhler, S. Bauer, D. Horn, and P. N. Robinson, "Walking the interactome for prioritization of candidate disease genes," The American Journal of Human Genetics, vol. 82, no. 4, p. 949–58, 2008.
[28] D. Nitsch, J. P. Gonçalves, F. Ojeda, B. de Moor, and Y. Moreau, "Candidate Gene Prioritization by Network Analysis of Differential Expression using Machine Learning Approaches," BMC Bioinformatics, vol. 11:460, 2010.
[29] O. Vanunu, O. Magger, E. Ruppin, T. Shlomi, and R. Sharan, "Associating Genes and Protein Complexes with Disease via Network Propagation," PLoS Computational Biology, vol. 6, no.1:e1000641, 2010.
[30] J. Chen, B. J. Aronow, and A. G. Jegga, "Disease candidate gene identification and prioritization using protein interaction networks," BMC Bioinformatics, vol. 10:406, 2010.
[31] W. Scott, P. Smyth, "Algorithms for Estimating Relative Importance in Networks," International Conference on Knowledge Discovery and Data Mining -Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, p. 266 –275, 2003.
[32] P. N. Westfall, and S. S. Young, *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*, Wiley, New York, 1993.
[33] D. W. Huang, B. T. Sherman and R. A. Lempicki, "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists," Nucleic Acids Research, vol. 37, no. 1, p. 1–13, 2009.
[34] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," Proceedings of the National Academy of Sciences of the United States of America, vol. 102, no. 43, p. 15545–15550, 2005.
[35] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene Ontology: tool for the unification of biology, " Nature Genetics, vol. 25, no. 1, p. 25–29, 2000.
[36] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, L. J. Jensen, and C. von Mering, "The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored," Nucleic Acids Research, vol. 39 (Database issue), p. 561–8, 2010.
[37] Y. Liu, B. Hu, C. Fu, and X. Chen, "DCDB: Drug combination database " . Bioinformatics, vol. 26, no. 4, p. 587–588, 2010.
[38] P. Sonego, A. Kocsor, and S. Pongor, "ROC analysis: applications to the classification of biological sequences and 3D structures," Briefings in Bioinformatics, vol. 9, no. 3, p. 198–209, 2008.

Automated corpus building and detection of the most frequent word sequences

István Endrédy
(Supervisor: Dr. Gábor Prószéky)
endredy.istvan.gergely@itk.ppke.hu

***Abstract*—Many areas of human language technologies need large corpora. The bigger is the better, but it is a rather expensive work to build a huge corpus. On the other hand, web is free and its content is growing every second. Crawling the web is a cost-effective way of compiling a big corpus. However, the same piece of text is often represented multiple times on various web pages, which may result in a statistically distorted representation of linguistic data. This paper describes a new web crawler that applies various techniques to identify and eliminate duplicate text and boilerplate data. In addition, an algorithm to compile a list of word sequences of outstanding frequency from the corpus that is capable of discounting sub-expressions is presented.**

***Keywords*-corpus building; web crawler; word sequences; boilerplate removal**

I. INTRODUCTION

There are various large Hungarian corpora available for the researchers. One of the biggest made 10 years ago at BME MOKK [1] relies on 600 million words. The Hungarian National Corpus [2] with its 190 million words is somewhat smaller, but its consciously selected content is fully POS-tagged. Nowadays we feel that there would be a need for a large, comprehensive, annotated and updated huge database.

Our aim is to download web pages, and extract the main content, the sentences they consist of. The next step is to put this information into a searchable and manageable IT model, and analyze it. Our main questions are: what are the most frequent words and word sequences, that is, word-n-grams? Do they represent the linguistically mostly described structures of the given language? Or: how do they change in time?

II. MAIN PARTS

The necessary tools to do this:

- a crawler, which surfs on the web all the time
- a text extractor, which gets the useful text (article) from the html content
- a database where we collect text

III. THE CRAWLER

The research has started in February 2012, and first I have tried to use existing crawlers. Their basic idea is to collect URLs (usually from search engine's URLs with input words) and download them.

I have tested the BootCat (Simple Utilities to Bootstrap Corpora And Terms from the Web) [5] text crawler's functionality. It needs an input file with words and terms, which are the focus of the corpus you want to build. Then it creates queries to search engines based on these, downloads them, and finally extracts the text from HTML with the help of BTE algorithm [6].

It can give a fast solution with limitations: it created a text corpus with size 150KB based on 4 words in 2 minutes.

We, however, want to download full domains, and not only single URLs.

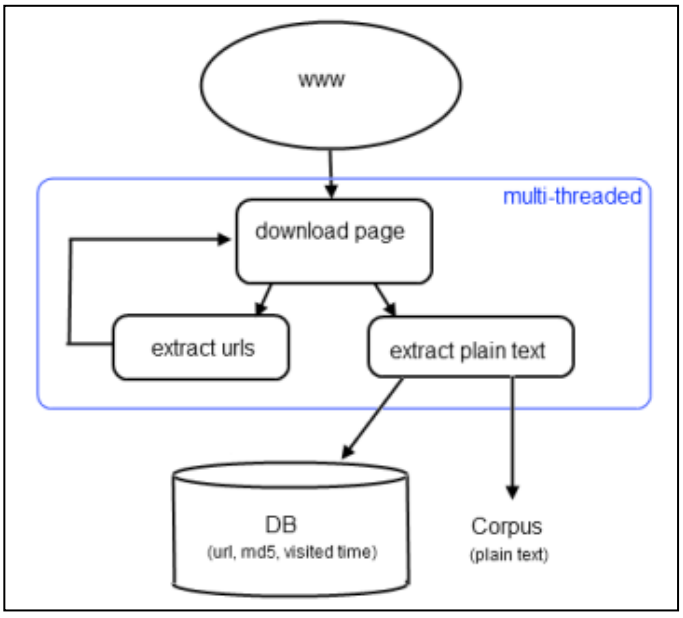


Figure 1. Main parts of the crawler

Our aim was to start with a simple crawler, which is able to download whole web sites, in a fast and storage friendly way, extracting the plain text from HTML pages. In addition, it is able to decide whether a page has been changed since the last time.

We decided to design and implement a crawler, which has a starting point, and it browses all the URLs it can. The strongest requirement was to remember the pages: was it changed or not.

We are focus to the main content: any changes in design, banners or menus are irrelevant, only the changes of the main part are interesting. It would be very expensive to store every content in a database, to be able to compare with its next versions. To be disk friendly it stores only a md5 hash about the text content of the web page. This way any irrelevant changes (any modification in boilerplate) will be ignored next time, and it is length is only 32 bytes.

Every visited page is stored in a database (now: sqlite or mysql) by these parameters:

- url (to be effective whole url is not stored, it would be redundant to store a domain million times, only <path without domain> and <domain_id>)
- md5 hash of the plain text content
- timestamp

Extracted URLs on a page are put in crawler's queue, each of them will be visited.

The main content is extracted, and plain text is written into a text file (each domain has an own output file).

My crawler does it slower than BootCAT, but with full domain: it can make a 2.6 GB corpus in 2.5 days, with downloading a full domain (origo.hu).

IV. TEXT EXTRACTION

Almost every web page has the same property: main content stands among non-informative parts: such as navigation menus, lists of links, headers and footers, copyright notices, advertisements, etc. These elements are called boilerplate in literature. They have usually the same structure (and same content) in many pages within a site, with some machine-generated parts.

It is not trivial how to remove boilerplate.

The quality of BTE algorithm (which is also shipped with BootCAT) was bad: advertisements or menus were tagged often as main content.

That's why I tried to create a new, own algorithm. Its basic idea was simple: if a term, word (text in HTML tag) is frequent, it must be boilerplate. Before it can remove boilerplate, it has to learn the site: it downloads some pages and makes the frequencies. It was easy to implement this in C++, and tested in many pages. But its quality was sometimes bad: if a site has an advertisement which always gives new ads, then this algorithm will think that it is part of the main content. (e.g. bookline is always advertising different books on index.hu, so it would be tagged as main content) So I evaluated this as a dead end.

I realized that there are various similar projects, so it was easy to study their results. The best work [3] compared the text extraction algorithms, and then he developed a very efficient, new algorithm: jusText [4]. It can extract articles even in an extreme situation, when the article is much smaller than the advertisements.

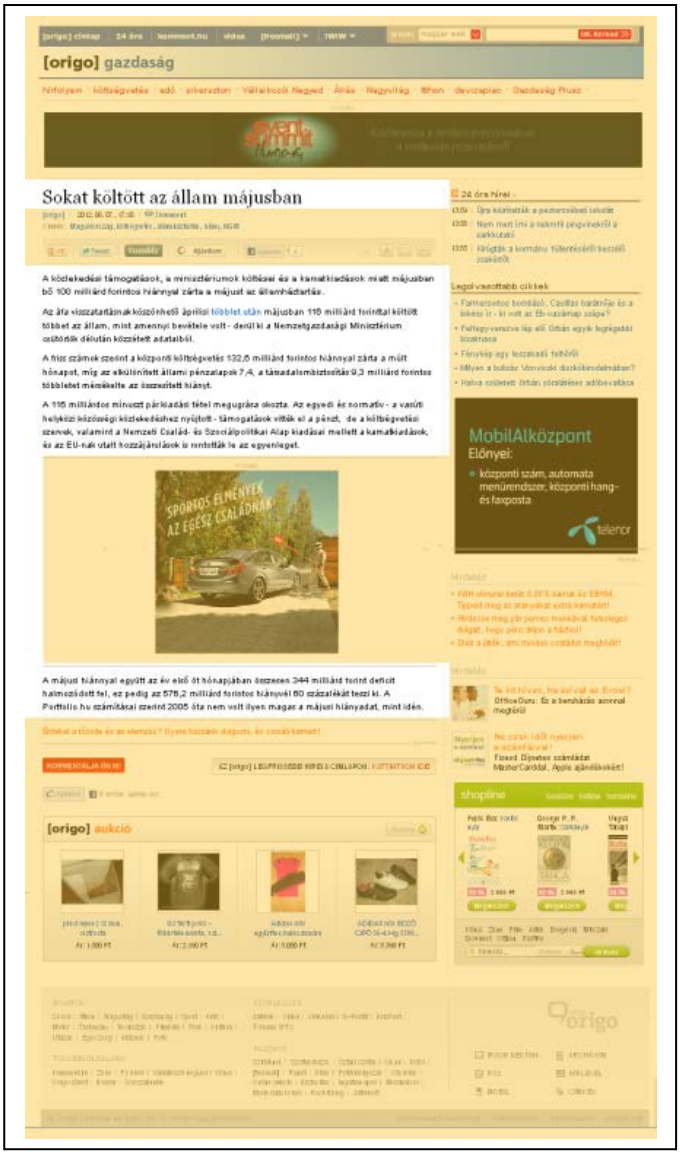


Figure 2. Main content (white) and boilerplate (yellow) in a web page

I've decided to re-implement this algorithm in C++, because originally it was written in Python), then I put it into our crawler. The result is a tool that can download many pages, and extract the articles on the fly. There is no need to store all the HTML-content, only the useful content: the article size is 20% of full HTML content, in average. (The algorithm, of course, can do that if we set the given parameter.)

The BTE algorithm is based the rate of text and tags frequency: if a text has many tags: it might be boilerplate.

(That's why it cannot handle tables.) It finds the longest part of a page with least tags. JusText has more parameters: it counts also the number of stop words (word which has great frequency in given language and is unimportant for a search engine: *the, is, at, and, or,* etc), number of links, length of text. Its quality is not perfect, but it is the best compared to the others [3].

V. WORD FREQUENCIES

The tools of this study are language independent, but our aim is to analyze mainly the Hungarian web. If we download it, even with extracted articles, it is only data, but we want to see the most frequent words and topics. A further idea would be to run this crawler regularly and see the difference which words get more popular, which ones less popular. If we, however, want to find the most frequent word sequences, we ran into various problems.

It's easy to count the occurrences of single words, it's a simple task. But if you want to count sequences as well, then it becomes a challenge. The source of the problem: a frequent sub-pattern may increase the frequency of an unfrequented pattern.

TABLE 1. COUNTING REAL FREQUENCIES OF THE WORD SEQUENCES

Word sequences	**Frequency**	
	Occurrences in text	***Independent occurrences***
"túl az Óperencián"	200	200
"túl az"	300	300-200
"túl"	1000	1000-500

If we want the occurrences when the given word sequence stands alone in text, then we should reduce the frequencies of the sub-sequences which it contains. This way we get a more realistic insight: how often it occurs in text.

The situation is complicated when the sequence is nested, which can not be simply subtracted. Let's assume the following input: "A B A B A" Table 2 shows the problem: frequency of "A" would be negative.

TABLE 2. COUNTING REAL FREQUENCIES OF THE INPUT "A B A B A"

Word sequences	**Frequency**		
	Occurrences in text	***Subtracted value***	***Correct***
"A B A"	2	2	2
"A B"	2	2-2=0	0
"A"	3	3-2-2=**-1**	**0**

I have created an algorithm (with the help of Attila Novák and Bálint Sass), which can count the exact number of every frequent word sequences. The algorithm has an input parameter: a limit, above which the sequence is interesting for us. We are looking for n-grams, longest length is n. The steps of the algorithm are the following:

- it reads first the whole file, counting all possible n-grams of every sentences
- it reads the text again by sentences
- if a n-length sample's frequency is above the limit: the count of all its sub-expressions will be decreased
- store its positions to avoid decreasing any words of that interval again later
- repeat last two steps with n-1, n-2, …, 1

Main parameter of the algorithm is the limit, this is the dividing line: if a sequence is above this: all its sub-patterns will be decreased.

First, we have played with small pieces text (to be able to check its precision), to detect the most frequent word n-grams. We observed the following:

- if the input is an article, it gives most probably the leading person, or main topic.
- if the input is a corpus, it gives the most typical word connections.
- if the input is a POS-tagged corpus, it can give the most frequent structures of that language

VI. RESULTS

Development of the crawler is in a test phase, it can download any domain in a few days, with 150 threads. Origo.hu was downloaded in 2.5 days, index.hu in 3.5 days. It seems to be stable, but it has large appetite for storage: it can fill the disk easily (especially when html content is also saved).

We made an interesting observation when applying the boilerplate removal algorithm to the two Hungarian news portal corpora we downloaded when testing the crawler. It is worth mentioning the difference in the size of the two corpora. Origo's net size is just 15% of that of index, although the ratio of number of pages is just 1:1.53. It has a simple reason: pages from origo contain much less main content than those from index. Its articles are much smaller. This clearly demonstrates the importance of the boilerplate removal: it plays an important role in improving the linguistic quality of the corpus.

TABLE 3. RESULTS OF THE CRAWLER

Domain	**Running time (in days)**	**Number of pages**	**Size of corpus (plain text)**	**Number of words**
origo.hu	2,5	1 090 000	2GB	950 000
index.hu	3,5	1 673 000	13GB	1 030 000

VII. CONCLUSIONS AND PLANS

The beauty of this project is its size: we would like to download the entire Hungarian web. That's why crawler needs a host with tons of free disk space. Now it visited only a few

domains, but it surprised us sometimes when it used all the free disk space of the host where it ran.

It will be interesting to run this crawler on a server with enough disk capacity, especially the 2nd, 3rd run, when only differences will be stored. Crawler's first benefit is that it provides an automated way to put a huge amount of human written texts into a corpus. But if we run this crawler regularly, we will be able to analyze the changes and observe trends in the web's content.

Our study is not corpus based, but corpus driven: when the corpus will be in our hands, its structure and sequences will suggest what the next step should be.

ACKNOWLEDGMENT

I would like to say thank you to György Orosz, who lent me his Linux server for playing with the crawler, to Attila Novák and dr. Bálint Sass for the good ideas, to Dr. Gábor Prószéky who helped me even from the hospital.

REFERENCES

[1] Halácsy, Péter; Kornai, András; Németh, László; Rung, András; Szakadát, István; Trón. Viktor, "Creating open language resources for Hungarian," In *Proceedings of LREC* 2004. Lisboa, 2004, pp. 203–210.

[2] Váradi, Tamás: The Hungarian National Corpus. In: *Proceedings of the 3rd LREC Conference,* Las Palmas, Spain, 2002, 385-389, http://corpus.nytud.hu/mnsz

[3] Pomikalek, Jan, *Removing Boilerplate and Duplicate Content from Web Corpora*, Masaryk University Faculty of Informatics, Brno, 2011.

[4] http://code.google.com/p/justext/ (May 20, 2012)

[5] Baroni, M.; Bernardini, S. "BootCaT: Bootstrapping corpora and terms from the web," *Proceedings of LREC 2004*. Lisboa, 2004, pp. 1313–1316.

[6] Finn, A.; Kushmerick, N.; Smyth, B. "Fact or Fiction: Content classifcation for digital libraries". In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

An Improved Methodology for POS-tagging Based on Advanced Statistical Models

László János Laki
(Supervisor: Dr. Gábor Prószéky)
laki.laszlo@itk.ppke.hu

***Abstract*—In this paper I examine the applicability of SMT methodology for part-of-speech disambiguation in Hungarian. Different methods and possibilities were used to improve the efficiency of the system. I also applied some methods to find a proper solution to handle out-of-vocabulary words. The accuracy of the best system is near 96%. The results show that such a light-weight system performs comparable results to other state-of-the-art systems.**

***Keywords*-SMT; POS-tagging; Lemmatization; Target language set; OOV**

I. Introduction

A wide spectrum of opportunities has been opened due to the fast development of information technology in almost all disciplines. This evolution could be detected on the field of computational linguistics as well. Processing of huge text materials has become easier, even the efficiency of these systems is increasing. Marking texts with syntactic and/or semantic information, or the morphological analysis of the language are really important tasks for computational linguistics. The task of part-of-speech (POS) tagging has not yet been perfectly solved, even though several systems have been implemented to achieve better results to this complex problem. The most popular ones are based on machine learning, in which the rules recognized by the systems themselves are based on different linguistic features. Further difficulties lie in determining the features, since these could be hardly formulated. Instead statistical machine translation (SMT) systems are able to recognize essential translation rules and features without any previous linguistic knowledge. [1]

Based on this assumption the application of SMT systems for text analysis could be successful. With the help of the standard frameworks and tools [2], [3] used for statistical machnie translation tasks, it is straightforward to handle complex POS structures. In this work I examine the applicability of these systems to solve the task of part-of-speech disambiguation.

II. Part-of-Speech Disambiguation

POS-tagging is the process of assigning a part-of-speech or other lexical class marker to each word in a corpus. However POS tagging is harder than just having a list of words and their part of speech, because some words can represent more than one part of speech depending on the context.

Most solutions apply analysis of the text based on pre-specified rule systems. The disadvantage of these methods is the huge cost of the creation of rules. Other frequently used approaches are based on machine learning, in which there are also some kind of rules used, however these are not of the same kind as linguistics rules, but are developed by the algorithms themselves based on relevant features. Further difficulties lie in the determining of these features, since these could be hardly formulated. It is very hard to determine and create a complete rule system that covers all the linguistic features and which can be processed by a computer.

III. POS-tagging as SMT problem

As described above both POS-tagging and lemmatization could involve huge amomunt of resources and complexity especially when applied to more complex languages, like Hungarian. In English a word can only have a limited number of forms, however in agglitunating languages this number is several orders of magnitude higher. Each affixum of a word contains some morphological information and also might produce a change in the lemma of the word. Therefore even more sophisticated algorithms are necessary to handle such a behaviour properly. These considerations deduce the application of the methods of statistical machine translation to POS-tagging. It has a great advantage over rule-based translation: only a bilingual corpus is needed to set up the training set of the system, but knowledge about the grammar of the language is not required to create the architecture of a baseline SMT system. In such a case POS-tagging is considered as a translation between sentences and their tagged versions.

Handling and analyzing out-of-vocabulary words that are not included in the training set (OOV words) has a significant influence on the success of a POS-tagging system. The type frequency of OOV words might vary in different languages. In English an OOV word will probably be a proper noun. In some other languages – such as Hungarian – OOV words would equally be nouns or verbs as well. This is due to the practically infinite number of word forms that might appear, thus it is impossible to have a corpus containing all forms of each word.

The benefit of this method is that the system is able to find rules without defining feature sets and it could do POS-tagging and lemmatization simultaneously. Another advantage is that it is a language independent method, where the the performance of the system only depends on the quality of the bilingual corpus used to train the system. Though my purpose was only

to test the system on Hungarian, in later works it might be extended to other languages easily.

IV. Framework

A. Corpus

In this study the Szeged Corpus 2 [4] was used as parallel corpus. This XML-based database contains both plain texts and their POS annotated version using the MSD-coding system. It is a rather small corpus containing 1.2 million words, which cover 155.500 different word-forms and 250.000 inter-punctuation signs. In contrast to natural language translation, where this size is unusably small, it is not such a relevant problem for POS-tagging, since the target language has a very limited vocabulary compared to any natural languages. For testing the system, 1500 randomly selected sentences of the corpus were used.

B. Training and decoding

Several methods of obtaining information from parallel corpora have been studied. Finally, I decided to use IBM models, which are relatively accurate, and the used algorithm was adaptable to the task. Based on these findings I decided to use the MOSES framework [2], [3], which implements the above mentioned IBM models.

C. Evaluation

To evaluate the efficiency of traditional SMT systems an automatic method is used. The BiLingual Evaluation Understudy [5] – BLEU score. The essence of this method is that the translations are compared with the reference sentences of the test set. BLEU score is calculated both to each n-gram lengths, and to a comulated average as well. Since POS-tagging is a one-to-one mapping between tags and words the most relevant measure gains from the case of 1-grams. Since BLEU score is not the usual method of evaluating a POS-tagger and lemmatizer, I also calculated the accuracy of the system to be able to compare the efficiency to other systems. This evaluation was used in sentence and in token level as well.

V. Handling OOV Words

The most obvious solution to reduce the number of OOV words is to increase the size of the corpus, so that all word forms would appear in it. Moreover it is important to have several occurences of each token in order to have a reliable statistics. Due to the agglutinative nature of Hungarian language, one stem could have many forms caused by the affixes; that is why an extremely large corpus would be needed to have all forms with the appropriate weight. This is an impossible requirement by itself, even more if a manually tagged corpus is expected. To eliminate this situation, I applied a method in which the system tries to find the appropriate tag for an unknown word based on the analyses of its context. All results will be compared with the system, which was trained with unmodified corpus. The source language corpus was created from the tokenized sentences without annotation. The target language corpus contained their POS tags (from now SMT_Baseline). The accuracy of the SMT_Baseline system is 91.46%.

A. In the original text

To examine the characteristics of frequent OOV words, a further investigation is needed during training and decoding. My basic assumption is to infer OOV POS-tags from the context. Though this is quite a simple method, however the complexity of the problem can also be reduced by limiting the possible POS-tags of an unknown word to some of the most probable ones. Thus at decoding time, the system has to choose only from these few tags.

To eliminate this problem I applied Guillem and Joan Andreu's method [6]. To achieve good results for Hungarian I used their results for English with some changes. A dictionary is created from words whose frequency in the training set is over a certain threshold value. The word frequency is calculated from the corpus. The words not included in this dictionary are changed to an optional expression (in this case "UNK"). The basic idea of the method is to change the less frequent words to the string "UNK".

Since OOV words are included in just a few word classes, therefore I assume that the annotation of the context of each OOV word is very similar. The SMT system performs the translation based on phrases, therefore the context of words and tags is taken into consideration already. By replacing the less frequent words to symbol "UNK", the annotation of the environment of these phrases will be more significant. Consequently the system can identify the POS tag for symbol "UNK".

The key question is the appropriate threshold value selection, since it determines the number of "UNK" symbols in the corpus. On one hand if this value is too high, too many tokens will be changed to "UNK" symbol; the probability of this symbol increases, therefore we will not receive correct annotation. On the other hand if the threshold is too small, too many rare words will be included in the dictionary causing that the advantage of the method could not be exploited sufficiently.

Therefore the system was trained with more threshold values (2, 4, 6, 8 and 10) to find the most appropriate one resulting in the best improvement of accuracy.

If the threshold is 1 the table gives us the result of SMT_Baseline system. That means none of the words were replaced with symbol "UNK". In the last column we can see the accuracy of the systems for each threshold value. For example: threshold 2 means that all words that appear in the corpus less than two times were changed to symbol "UNK". The second column of the table shows the percentage of words in the training set (of size 1 459 288) added to the dictionary. For example: in the case of threshold 2 almost 60% of the words became OOV words. The third column of the table contains the percentage of the words left original in the corpus.

From the above results it is straightforward that in the case of threshold value 2 the system achieved significant improvement compared to any of the previous ones. Only

38 words were not annotated against the 1697 in system SMT_Baseline1. If the threshold value is raised, it leads to a decrease in the accuracy of the system.

During deeper evaluation it turned out that this accuracy decrease is due to the lower rate of original words in the corpus (only small number of words are in the dictionary). In the case of threshold 2, symbol "UNK" was used for 6.13% of the words in the training set. This rate is 92% in the case of threshold being 10. This tendency matches with the theorem of Zipf's laws [7]. We have to note the 85.96% accuracy at threshold value 10. This result is quite good despite that only 8.09% of the training set was added to the dictionary. Table I shows an example from the output of the system SMT_OOV_token in the case of threshold 8.

TABLE I
AN EXAMPLE FROM THE OUTPUT OF THE SYSTEM SMT_OOV_TOKEN

System	Translation
Simple text:	ezt a **unk** és **unk** a diplomáciai **unk** kívül mindenekelőtt a magyarországi **unk unk** .
Reference annotation:	[pd3-sa] [tf] [x] [ccsw] [nc-sa] [tf] [afp-sn] [nc-pp] [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]
SMT annotation:	[pd3-sa] [tf] [nc-sa] [ccsp] [vmis3p—y] [tf] [afp-sn] [nc-pn] [st] [rx] [tf] [afp-sn] [nc-pn] [nc-sa—s3] [punct]

B. In case of lemmas

From the results of table I it can be seen that if the threshold value is too high, too many of the words become "UNK". Due to the agglutinative features of Hungarian language the original text contains different forms of nouns, verbs and adjectives of the same stem. This is the reason that the number of these different forms is under the threshold. Consequently in most cases nouns, verbs and adjectives are also replaced with "UNK" in the sentences of the corpus, which makes the decreases the accuracy of the system. My goal was to reduce the number of symbol "UNK" with replacing only really rare words in the text. Therefore the threshold was determined based on the frequencies of the lemmas and not on different word forms. The results of this system (called SMT_OOV_lemma) can be found in table II.

TABLE II
PERFORMANCE OF THE SYSTEM SMT_OOV_LEMMA.

Threshold value	Rate of words in the dictionary	Rate of words in the corpus	System's accuracy
SMT_Baseline	100%	100%	91.46%
2	70.00%	96.58%	92.57%
4	56.64%	94.50%	92.25%
6	50.18%	93.17%	91.81%
8	45.81%	92.13%	91.48%
10	37.08%	88.47%	91.10%

The results proved that calculating the threshold based on lemmas makes much fewer number of words to be marked as OOV (numerically only 3.42% of the words from the corpus).

We can observe that besides threshold 2, the best result of system SMT_OOV_lemma (92.57%) is worse than in the case of SMT_OOV_token (93.13%). The deep evaluation showed that the number of not annotated words increased (1015 cases) compared to system SMT_OOV_token, and 984 words were incorrectly analyzed.

C. Multiple thresholds

The above results have alraedy achieved high accuracy results of taggin Hungarian words, but still OOV words are included to several different POS types with quite high probability. In English such OOV words are mostly nouns. To have a more sophisticated method I applied numerical calculation of several threshold values that distinguish different POS-tags fo OOV words.

We can observe that the same thresholds in the above two systems divide the corpus in different proportions. Word forms with frequency values higher than the threshold are included in the dictionary of system SMT_OOV_token.; but if we determine the frequencies based on lemmas – such as in the case of system SMT_OOV_lemma. – the dictionary will contain more words. Thus a certain threshold divides the set of words to three parts. The first set contains the words, which are included in both dictionaries; these words are the most relevant ones. In the second set we can find those OOV words, which are really rare and were under threshold in both cases. The words, for which the word form is not frequent enough, but their lemma is over the threshold were included into the third set.

I examined the types of OOV words in each set. The results showed that adjectives and other types of OOV words mostly belong to the second category (under both threshold level), while verbs to the third set. Nouns can be found in both category roughly in a similar measure. Based on this observation another system was trained, which is able to distinguish OOV words, if they belongs to the second or third categories. The results of this system (called SMT_OOV_multi) are shown in table III.

TABLE III
PERFORMANCE OF THE SYSTEM SMT_OOV_MULTI.

Threshold value	System's accuracy
SMT_Baseline	91.46%
2	93.28%
4	90.65%
6	88.62%
8	87.40%
10	86.15%

The results reflect that the system with threshold 2 achieved the best performance (93.28%) of all the above systems. This improvement is caused by the fact that only 37 words were not analyzed. Furthermore in the case of incorrect analysis – numerically 1772 items – the error occurred mostly during the subanalysis of nouns.

According to the evaluation, the method of using multiple thresholds helped to distinguish adjectives and verbs; therefore lead to the improvement of the system.

D. Introducing postfixes

Based on the results of the previous systems it is straightforward to conclude that using multiple thresholds – three classes – are not enough to separate nouns, verbs and other types of words. Due to the wide range of affixes in Hungarian, one word could have many forms. Different POS types however have characteristic prefixes and postfixes (in case of Hungarian language mainly postfixes). Therefore previous methods were extended to use information based on the last characters of an OOV word to determine the type.

The best method would be to use a morphological analyzer to separate postfixes of a word with different lengths, but one of the purpose of my method is its simplicity, therefore I applyed a simple implementation for this task as well. To continue the idea of the previous sections in this section the last 2, 3 or 4 characters of the original OOV words were joined to the "UNK" symbol. The results of this system (called SMT_OOV_postfix) are shown in table IV.

TABLE IV
PERFORMANCE OF THE SYSTEM SMT_OOV_POSTFIX.

Threshold value	Systems accuracy Number of left characters 2	3	4
SMT_Baseline	91.46%	91.46%	91.46%
2	95.17%	95.83%	**95.96%**
4	94.17%	95.32%	95.90%
6	93.48%	94.97%	95.73%
8	92.94%	94.70%	95.60%
10	92.61%	94.55%	95.55%

This system significantly outperforms any of the previous ones. The worst result is better than the result of the SMT_Baseline. The best result was 95.96% which was achieved with threshold value 2 and 4-character-long postfixes of OOV words. The optimal length of the postfix might depend on the language, nevertheless in Hungarian most of the postfixes are 2 or 3 character long, that is why the system with 3-character postfixes and with the threshold value of 2 is above 95.83%. The slightly higher results in the case of four characters is due to the cumulative behaviour of suffixes.

Table V shows an example from the output of the system SMT_OOV_postfix in the case of threshold 8 similar to previous ones. We can see that this system made correct annotations for all words of the sentence in contrast to the above ones.

TABLE V
AN EXAMPLE FROM THE OUTPUT OF THE SYSTEM SMT_OOV_POSTFIX

System	Translation
Simple text:	ezt a **unk_erőt** és képességet a **unk_ciai** erőfeszítéseken kívül mindenekelőtt a magyarországi multinacionálisok adhatnák .
Reference annotation:	[pd3-sa] [tf] [x] [ccsw] [nc-sa] [tf] [afp-sn] [nc-pp] [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]
SMT annotation:	[pd3-sa] [tf] [nc-sa] [ccsw] [nc-sa] [tf] [afp-sn] [nc-pp] [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]

VI. CONCLUSION

In this paper applicability of the SMT system was examined for part-of-speech disambiguation and lemmatization in Hungarian. Based on my observations these tasks can be considered as translations from plain text to analyzed one. The accuracy of such systems can achieve results of up to 96% accuracy. Although the quality of the above presented systems is behind the state of the art systems – still compareble to those available for Hungarian –, but in my work an absolutely automated system was created which finds the rules itself and we do not have to determine any features for training either. On the other hand this system is able to perform annotation and lemmatization simultaneously.

Further significant improvement was achieved by handling out-of-vocabulary words using a method based on word frequencies.

Results showed that only statistical methods are not enough to solve the task of POS-tagging; some kind of hybridization is necessary to improve the quality of the system. The achieved results were encouraging and they pointed out that this way of research contains further possibilities.

REFERENCES

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, 2nd ed., ser. Prentice Hall series in artificial intelligence. Englewood Cliffs, NJ: Prentice Hall, Pearson Education International, 2009.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the ACL 2007 Demo and Poster Sessions*. Prague: Association for Computational Linguistics, 2007, pp. 177–180.

[3] P. Koehn, "Moses system," http://www.statmt.org/moses/.

[4] D. Csendes, C. Hatvani, Z. Alexin, J. Csirik, T. Gyimóthy, G. Prószéky, and T. Váradi, "Kézzel annotált magyar nyelvi korpusz: a Szeged Korpusz." in *I. Magyar Számítógépes Nyelvészeti Konferencia*. Szegedi Egyetem, 2003, pp. 238–247.

[5] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318. [Online]. Available: http://dx.doi.org/10.3115/1073083.1073135

[6] G. Gascó I Mora and J. A. Sánchez Peiró, "Part-of-Speech tagging based on machine translation techniques," in *Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Part I*, ser. IbPRIA '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 257–264. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72847-4_34

[7] G. Zipf, "Human behaviour and the principle of least-effort." Cambridge, MA: Addison-Wesley, 1949.

Numerical analysis of bacterial pattern formation

Dóra Bihary
(Supervisor: Dr. Sándor Pongor)
bihary.dora@itk.ppke.hu

***Abstract*—Members of bacterial communities communicate and cooperate via diffusible chemical materials: they emit into the environment, and at the same time, they also compete for nutrients and space. Agent-based models (ABMs) are useful tools for simulating the growth of communities containing multiple interacting microbial species. In this work we present numerical indices characterizing spatial distribution and the fitness of competing bacterial species in an ABM and we present data on how these indices can be used to visually summarize large scale simulation experiments. Preliminary results show bacterial agents utilizing different nutrients but sharing communication signals and public goods can form stable mixed communities in which the species grow faster than any of the single species alone.**

***Keywords*-quorum sensing; *Pseudomonas aeruginosa*; segregation index; relative fitness; heatmap**

I. Introduction

Bacteria are the most popular form of life on Earth. Some of them live in a strong relationship with human beings. They cause illnesses, people use them in industry for example in pharmacology, waste water cleaning or in food production. Despite of this, most bacterial species have not been characterized yet, many of them can not be analyzed in laboratories. Just because of this computer models have an important role in the examination of bacterial behavior and colony configuration.

The interaction between individual bacteria is based on diffusible signals, the best known example of which is a mechanism called quorum sensing (QS) [1], [2]. In this mechanism signaling materials are secreted by the bacteria and diffuse in the environment and the concentration of signals is correlated with the density of bacteria, or loosely speaking, with the size of the consortium. This is obviously true only in certain cases, however a small community can also be wedged in a small place where high signal concentration can be reached by a small community. From the computational point of view, bacterial swarming is when a group of bacteria is moving together with the aim of archieving a collective goal. In this swarming state species taking part in the swarming speed up, their metabolism and food consumption is growing and they secret chemical compounds frequently referred to as "public goods" or simply "factors" (e.g. surfactants, enzymes, siderophores), which facilitates movement and nutrient uptake.

According to our modeling experiments, colony patterns in multispecies communities can have two main types (see Fig. 1). In the first case (left) the different bacterial species are separated from each other, we call this segregated population. In the second case the population is mixed. This second pattern

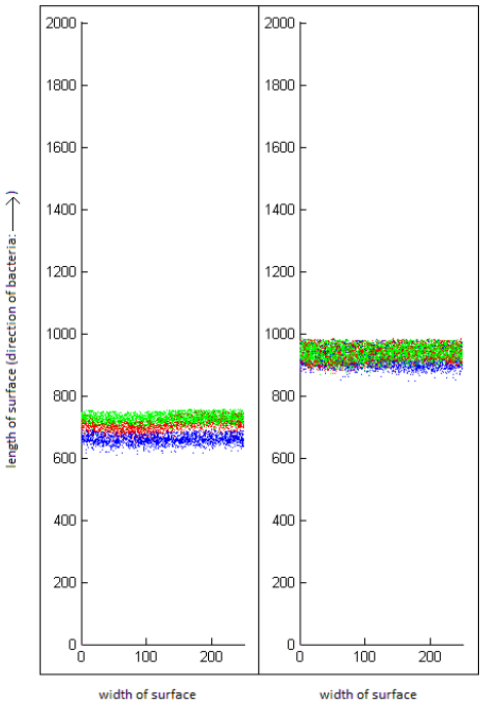


Figure 1. Simulation of segregation (left) and co-localization (right) of mixed, swarming communities.

makes it possible for species to communicate with each other, in the first case bacteria behave as individual species moving separately on the surface.

We can find large number of different model types in the literature describing this phenomena [3], [4]. Our simulation framework is based on a so called hybrid model, which is composed of two parts. Bacteria are represented by an agent-based model, while the concentration of food, signals and factors are represented by reaction-diffusion equations.

II. Biological background

The mechanism of quorum sensing is based on chemical compounds. Bacteria secret a basal amount of signals that, in an open environment, continuously diffuses away. If there is a sufficient number of bacteria present in a small environment, the concentration of the emitted signal can raise and eventually reach a threshold concentration. The bacteria sense this threshold concentration and change their metabolism: they increase the production of signals, start the production of factors. When

in turn the concentration of factors reaches a threshold in the environment, the bacteria increase their movement, food intake and their division rate. They enter an active state - they start swarming.

Our model organism was *Pseudomonas aeruginosa* an opportunistic patogen that can potentially cause death in patients of cystic fibrosis.

III. Simulation environment

Recent studies showed that hybrid models can reproduce certain well-known features of bacterial colonies, especially the formation of fractal-like patterns [4]. These models did not include

Our simulation model was built in order to prove that simple models can explain the specific behaviour of quorum sensing bacteria [5], [6]. It is a hybrid model which in our case has two parts. An agent-based model represents bacteria. They move randomly on a two dimensional surface, they eat, divide and produce materials - signals and factors. The amount of these materials in the surrounding of each agent determines its state - ground, active, or swarming. The change between these states is characterised by the amount of signal and factor. Ground state means that the concentrations of signal and factor are both below, while in swarming state they are both of above the respective thresholds. The metabolism, movement and secretion of agents changes according these states. The diffusion of materials - food, signal and factor - is described by reaction-diffusion equations. These equations have two parts: the diffusion shows how these materials diffuse in space and time on the surface (2D space), while the reaction part tells us about further changes. In our case this part describes how materials are consumed or produced by bacteria, and how materials decay with time.

The space in our model is represented by an infinite longitudinal two dimensional surface, open on one end and provided with periodic boundary conditions on its sides. We can best picture this longitudinal tract as a dendrite of a bacterial fractal-like bacterial colony. Cells of a species (individual cells endowed with a program describing the quorum sensing mechanism) are placed to the closed end of this surface and let to move randomly which will result in the formation of a dense agent community moving along the longitudinal track.

IV. Numerical representation of bacterial pattern formation

We studied populations consisting of two different species. In these binary populations it is not obvious whether and how the two species communicate and cooperate. Biological examples suggest that two different species may or may not share communication signals. If they do, they can for example react to the other's signals in a certain rate. If the two signals are totally different they will not have any effect on one another.

Our main question was to determine parameter settings with which a cooperation between species can be obtained. According to the above mentioned biological consideration we examined colonies where signals and factors were common in a certain rate. This means that one species can obtain its own materials, plus with a certain percentage it can obtain the other species chemicals. We had 11 cases - from 0 to 1 with steps of 0.1 - for signals and factors as well, which meant 121 parameter setting - 121 different simulations.

A. Segregation index

Multispecies bacterial colonies can form two main pattern types as shown in (Fig. 1). In a mixed population the members of the two species are homogeneously mixed - this is what we expect to happen if the two species can communicate and cooperate. In a segregated population - where species form separate colonies - we can not talk about interspecies communication. Our assumption is that a proper selection of parameters for sharing signals and factors may determine whether the species will form mixed or separate colonies.

For the quantification of segregation we introduce the so-called segregation index [7], [8]. This quantity was originally defined in such a way that a certain number of nearest neighbors were identified for all bacterial cells, and the largest percentage of a given species was determined for each cell. The segregation index was then calculated as the average of these materials. Calculation of this index is time consuming, especially since our bacterial communities are quite large.

In order to develop a segregation index that can be calculated in a more time efficient manner, we take advantage of the fact that space in our simulation is divided to cells that form a matrix-like lattice. In each row we can count the number of bacteria from each species. E.g. for three species we can have $n_1(i)$, $n_2(i)$, $n_3(i)$ in the i^{th} row. The sum of these numbers in the i^{th} row is $N(i)$. So the fraction of these values gives the ratio of each species in the population. If a population is segregated, this fraction is almost 1 for one of the species, and almost 0 for the two other species. For computing segregation index we average the maximal elements (the size of the largest species) in each row. We get a more representative measurement if we weight these values with the total number of bacteria in the actual row. By this step we avoid counting fractions for each row, we simply have to add the maximal numbers in each row, and divide this value with the total number of bacteria in the given step.

The number we get with this computation is in the range $[1/\text{number of species}; 1]$, so for later comparisons it is better to use the normalized version of the value where total segregation corresponds to 1.0 and no segregation/mixed population to 0.

Equation 1 and 2 show the computation of segregation index:

$$S = \frac{S_{sumPerRow}}{N_{allBacteria}}, \tag{1}$$

where $\max(n_i)$ is the cell number of the dominant species within the i^{th} space unit, the denominator is the total number of the population (including all species). For a randomly mixed community (such as shown in Figure 1, right), this quantity will approach the reciprocal of the number of species

present which allows us to construct a [0,1] numerical index as follows:

$$SGN = \frac{\left(S - \frac{1}{N_{species}}\right)}{\left(1 - \frac{1}{N_{species}}\right)}, \quad (2)$$

where SGN is the normalized segregation index calculated at a certain time step and $N_{species}$ is the number of agent species present.

B. Relative fitness

Cooperation between two species is meaningful if both or at least one species has some kind of an advantage from the colocalization. This advantage can be represented as the growth rate of a species within the equilibrium population. We can quantify this property using the concept of fitness [9].

In principle the fitness of a species can be computed from the size of the population taken at the beginning and at the end of simulation (eq. 3).

$$F = \frac{1}{\Delta t} \lg_2 \frac{N_{end}}{N_{start}}, \quad (3)$$

where F is the fitness value, Δt denotes the elapsed time, N_{end} and N_{start} are the size of the population at the beginning and end of the simulation respectively. By taking the logarithm of the fraction we get an expression that's sign depends on whether an increasing or decreasing population is there in the simulation. For increasing population the logarithm is positive, however for decreasing it becomes negative.

Fitness is a dimensionless quantity which is often represented on a relative scale, in comparison with the fitness of a reference species. Our reference population is a wild type species that can perform all phenomena of quorum sensing - they secret signals and factors and they respond to the above threshold concentrations of these materials. If we want to compare a mutant species with a wild type population we have to divide the fitness of the two species. This is called relative fitness. Equation 4 describes this computation.

$$F_{rel} = \frac{\lg_2(N_{end}/N_{start})}{\lg_2(N_{end,wt}/N_{start,wt})}, \quad (4)$$

where F_{rel} is the relative fitness, $N_{end,wt}$ and $N_{start,wt}$ are the reference values for wild type population. The Δt terms are cancelled by the division.

V. Results

To answer the above mentioned question - with what initial parameters in the signal-factor sharing space is it possible to get cooperating populations at two species colonies - we used a heat map representation. On these figures x axes denotes the sharing of signal and y axes the sharing of factor. The identical segregation index (or relative fitness) values are represented with identical colors.

Figure 2 shows results of an example. During the simulations signal and factor overlapping changed in a symmetric manner for the two species.

Top left figure (Fig. 2) shows the results of segregation index. Segregated regions are denoted by white, mixed regions by black color. There are some black dots on the diagram, but if we have to determine a securely mixed region it would definitely be the upper left region. This is where species understand each other's factors very well, and they share some signal as well. Signal overlapping is important, zero value never means mixed population but the rate of sharing must not be as high as we can see it at factors.

Top right figure (Fig. 2) shows results of relative fitness. A similar region can be obtained to the previous one. Here white regions mean fit and black regions unfit populations. It is a binary image where fit means that relative fitness is above and unfit means below 1.05. This value was calculated from $25-25$ simulations where we measured relative fitness values in segregated and mixed populations.

Our aim was to determine those regions of the parameter space where cooperation can happen. We defined this by having "good" relative fitness and being mixed population. So for analysing the original question we mapped these above mentioned two images to each other (fig. 2, bottom). This image shows with white those regions where relative fitness is above 1.05 and segregation index is below 0.5. These will be the cooperating regions.

VI. Conclusion

We attempt to describe multispecies bacterial colonies with agent based models in which agent species share secreted materials to varying extents. Our system contains two kinds of materials, signals and factors, that mediate communication and cooperation between agents, respectively. According to our preliminary results, the level of sharing signals and/or factors may determine whether two species will form mixed or segregated communities.

Acknowledgment

This project was developed within the PhD program of the Multidisciplinary Doctoral School, Faculty of Information Technology, Pázmány Péter Catholic University, Budapest. Thanks are due to my supervisor, Prof. Sándor Pongor and to Ádám Kerényi, who is the major developer of the agent-based simulation program.

References

[1] M. B. Miller and B. L. Bassler. Quorum sensing in bacteria. *Annu Rev Microbiol*, 55:165–99, 2001.

[2] V. Venturi and S. Subramoni. Future research trends in the major chemical language of bacteria. *HFSP J*, 3(2):105–16, 2009.

[3] K. Kawasaki, A. Mochizuki, M. Matsushita, T. Umeda, and N. Shigesada. Modeling spatio-temporal patterns generated by bacillus subtilis. *J Theor Biol*, 188(2):177–85, 1997.

[4] E. Ben-Jacob, I. Cohen, O. Shochet, I. Aranson, H. Levine, and L. Tsimring. Complex bacterial patterns. *Nature*, 373(6515):566–7, 1995.

[5] S. Netotea, I. Bertani, L. Steindler, A. Kerenyi, V. Venturi, and S. Pongor. A simple model for the early events of quorum sensing in pseudomonas aeruginosa: modeling bacterial swarming as the movement of an "activation zone". *Biol Direct*, 4:6, 2009.

[6] V. Venturi, I. Bertani, A. Kerenyi, S. Netotea, and S. Pongor. Co-swarming and local collapse: quorum sensing conveys resilience to bacterial communities by localizing cheater mutants in pseudomonas aeruginosa. *PLoS One*, 5(4):e9998, 2010.

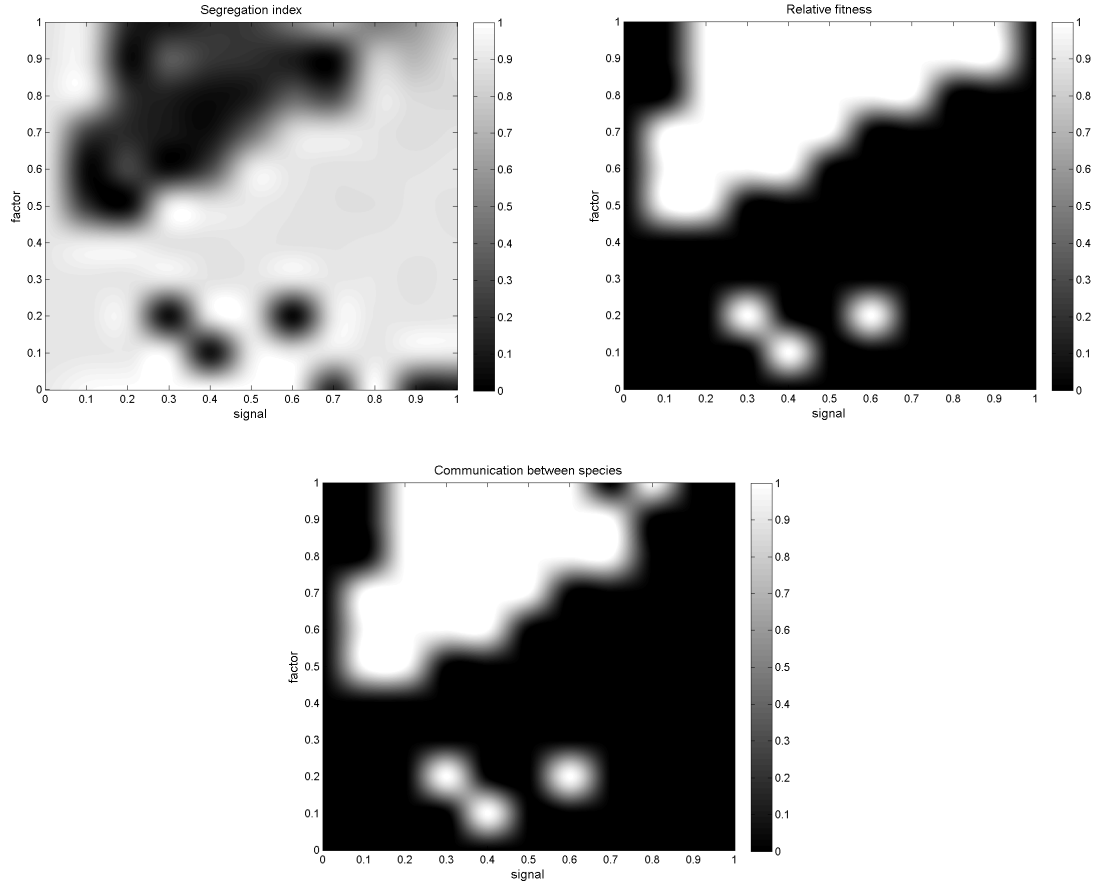


Figure 2. Top left image shows the normalized segregation index and top right the relative fitness in function of signal and factor sharing. Figure at the bottom shows regions on signal-factor sharing space where cooperation can be obtained.

[7] S. Mitri, J. B. Xavier, and K. R. Foster. Social evolution in multispecies biofilms. *Proc Natl Acad Sci U S A*, 108 Suppl 2:10839–46, 2011.

[8] C. D. Nadell, K. R. Foster, and J. B. Xavier. Emergence of spatial structure in cell groups and the evolution of cooperation. *PLoS Comput Biol*, 6(3):e1000716, 2010.

[9] H. A. Orr. Absolute fitness, relative fitness, and utility. *Evolution*, 61(12):2997–3000, 2007.

Classifying the Topology of AHL-Driven Quorum Sensing Circuits in Proteobacterial Genomes

Zsolt Gelencsér
(Supervisor: Prof. Sándor Pongor)
gelzs@digitus.itk.ppke.hu

***Abstract* — Virulence and adaptability of many Gram-negative bacterial species are associated with an N-acylhomoserine lactone (AHL) gene regulation mechanism called quorum sensing (QS). The arrangement of quorum sensing genes is variable throughout bacterial genomes, although there are unifying themes that are common among the various topological arrangements. A bioinformatics survey of 1,403 complete bacterial genomes revealed characteristic gene topologies in 152 genomes that could be classified into 16 topological groups. We developed a concise notation for the patterns and show that the sequences of LuxR regulators and LuxI autoinducer synthase proteins cluster according to the topological patterns. The annotated topologies are deposited online at http://bacteria.itk.ppke.hu/QStopologies/.**

***Keywords*; quorom sensing; N-AHL; topology; bacteria; proteobacteria;**

I. INTRODUCTION

Bacteria might be simple, single celled organisms but their social behavior means they can form complex communities and engage in coordinated behaviors. Many bacterial species use chemical signals to monitor their environment and regulate population density. the signal was identified as an acyl homoserine lactone (AHL) and the genetic circuit was then also identified and shown to be composed of a signal generator called the luxI/LuxI gene/protein and a response-regulatory protein designated as luxR/LuxR gene/protein. The name quorum sensing (QS) was then coined in the 1990s [1] to denote this cell-cell communication mechanism which is now recognized as a key trait governing bacterial community behavior [2-4]. The importance of quorum sensing has been clear for a number of years as it appears to be a promising target for controlling bacterial colonization and pathogenesis in human disease[5]. This could be of particular importance, as multidrug resistant strains of human pathogens continue to emerge. Potentially beneficial effects in relation to plant health can be attributed to QS in certain species including regulation of the production of anti-microbial compounds and induction of systemic resistance in plants [6].

One of the most well studied cellular and genetic mechanisms of QS is based on the N-acylhomoserine lactones (N-AHLs) [1]. Bacterial cells secrete and respond to autoinducers continuously to sense the surrounding environment, and respond to events as they occur. When an autoinducer reaches a critical level, the population of bacteria responds through a coordinated expression of specific target genes, which finally manifest in a particular behavior/phenotype. The N-AHL QS system is based on two proteins belonging to the LuxI and LuxR families[3, 7]. Luxl-family proteins are cytoplasmic enzymes and are responsible for N-AHL synthesis. After synthesis, the signal molecule moves freely across cell membranes and accumulates both intra- and extra-cellularly in proportion to cell density. When populations are low, N-AHL dissipates; when populations are high, N-AHL concentration increases. Above a critical concentration or cell density, N-AHLs interact with the LuxR-family which in most cases result in complexes (homodimers) that bind specific promoter DNA sequences (termed lux-boxes) located in the promoter region of target QS-regulated genes. This subsequently affects their expression resulting in particular phenotypes of the organism. Most often one of the targets of AHL QS is the luxI-family gene resulting in a positive feedback loop. Most N-AHL producing bacteria possess canonical QS circuits that broadly resemble the original principles of the design that were described in the nineties. However, a number of interesting variations have since been described in the AHL synthesis/AHL response gene layout and regulation of the QS genes. The goal of this work is to show, based on a new survey of the genomic databases, how these variations correlate with the topological arrangement of the QS regulatory genes.

II. A GENERALIZED REGULATORY FRAMEWORK OF N-AHL BASED QUORUM SENSING

In general terms, N-AHL-based QS signaling is often referred to as mere autoinduction, requiring only a synthase and a sensor/regulator protein. However, in the absence of regulation of QS, autoinduction would increase signal levels without limit. A down-regulation loop (Figure 1), which turns on at higher signal concentrations is the simplest way to limit and stabilize the signal levels. There are a variety of mechanisms that can play this role in QS systems. The DNA-binding negative regulator RsaL acts as a homo-dimer by binding on the bi-directional rsaL-luxI promoter [8]. RsaM is another small regulatory protein believed to be acting in a similar way to RsaL in many Burkholderia species [9]. The crucial role of these negative regulators is highlighted by the fact that their deletion leads to signal overproduction and a less virulent bacterial phenotype. Down-regulation can also be achieved by RNA-based mechanisms. For example, some QS systems act by activating in their ground state an RNA-binding

protein that inhibits transcription of synthases, and then

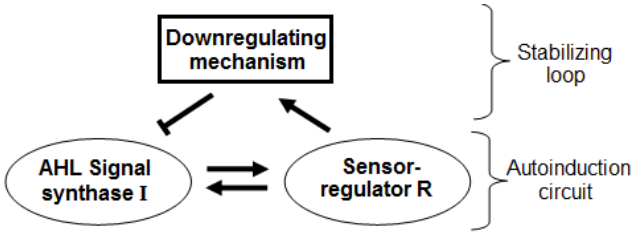


Figure 1. Regulatory outline of N-AHL-based QS signaling. Pointed arrows indicate activation, and the hammerhead arrow indicates inhibition.

gradually removing the inhibition when signal concentrations are increased. Meanwhile, another class of QS regulation system is believed to use sRNA species that considerably decrease the mRNA of the LuxR genes at low population density. For the sake of completeness we mention that downregulation can also be achieved by simple resource limitation where a reduction in number of QS cells would decrease of the QS signal concentration itself.

III. Topological Arrangements of QS Genes in Bacterial Genomes

Gene topology is a broad term that can include the arrangement of genes within chromosomes, with respect to the replication origin or other chromosomal elements. In this work we use the words topological arrangement" or briefly "topology" to denote the arrangement within a close neighborhood of the QS regulatory genes, to denote whether the genes are convergent, divergent, synthase upstream, receptor upstream, or synthase and receptor separated by other genes, etc. The preliminary overview of the searches showed a variety of topological types. To illustrate these, we developed a concise notation based on a PROSITE-like syntax[10]. The luxR, luxI, rsaL and rsaM genes were abbreviated as R, I, L and M, respectively, and X is used for all other genes. An arrow above each gene symbol then shows the direction of transcription. With this notation, for example, $\vec{R}\vec{I}$ denotes adjacent *luxR* and *luxI* genes transcribed in the same direction. $\vec{R}X(>5)\vec{I}$ denotes the same pattern with more than five genes between the *luxR* and *luxI,* without specifying the direction of transcription of the X gene. A summary of the topological categories found in our survey is given in Table 1. Somewhat arbitrarily, we divided the patterns into two groups: simple topologies and complex topologies. Simple topologies consist of luxI/luxR genes that are either vicinal or are separated by few genes. Such arrangements are characterized by typical patterns of transcriptional orientation that are conserved in many proteobacteria. This category makes up the majority of the observed cases in Table 1. and is the main subject of this review.

Within the class of simple topologies, the majority of the cases is made up of the R1 and the R2 topologies, that Goryachev termed type A and type B, respectively [11, 12]. In Table 1 we see combinations that are outside these two known

Table 1. Typical topological patterns found in completet bacterial genomes

ID	Pattern	Occurrence in complete genomes				
		Total	alpha	beta	gamma	delta
Short, conserved topological patterns						
R1	$\vec{R}\vec{I}$	96	71	14	11	0
R2	$\vec{R}\overleftarrow{I}$	53	2	2	46	3
R3	$\overleftarrow{R}\vec{I}$	11	1	3	7	0
R4	$\vec{I}\vec{R}$	2	2	0	0	0
L1	$\vec{R}\overleftarrow{L}\vec{I}$	15	0	7	8	0
M1	$\overleftarrow{R}\vec{M}\vec{I}$	30	0	20	10	0
M2	$\vec{R}\vec{M}\vec{I}$	1	0	1	0	0
X1	$\vec{R}\vec{X}\overleftarrow{I}$	1	0	0	1	0
X2	$\overleftarrow{R}\overleftarrow{X}\overleftarrow{I}$	2	2	0	0	0
X3	$\vec{R}\overleftarrow{X}\vec{I}$	4	0	2	2	0
X4	$\overleftarrow{R}\vec{X}\vec{I}$	1	1	0	0	0
X5	$\vec{R}\vec{X}\vec{I}$	2	1	1	0	0
Longer, unusual topological patterns						
M3	$\vec{R}X(>2)\vec{M}\vec{I}$	6	0	6	0	0
M3'	$\vec{M}\vec{I}$	2	0	2	0	0
X6	$\vec{R}\overleftarrow{X}(7)\vec{I}$	1	1	0	0	0
X7	$\vec{I}X(>7)\vec{R}$	5	5	0	0	0

categories. For instance, all 4 arrangements that are possible for two vicinal genes appear. In a characteristic subgroup of the simple patterns there is a single intervening gene between the *luxR* and *luxI* genes. These intervening genes show interesting commonalities in terms of function (Table 2). Out of 48 such single intervening genes 11 code RsaL and 29 code RsaM proteins that are both known to negatively regulate quorum sensing. In both cases of these proteins, typical topologies were observed: L1 and M1, respectively. RsaL (*L* in our notation), is a member of the tetrahelical superclass of H-T-H proteins[13] which are recognized as widespread QS repressors in bacteria, binding to DNA as dimers. For example, RsaL (which is predominant in pseudomonad genomes[14]) in *P. aeruginosa* prevents expression of the *R* gene by binding to DNA next to the *lux*-box [9]. In our analysis we also found that homologues of RsaL frequently occur outside QS circuits in various bacterial genomes (data not shown). The significance of this is not clear at this stage. In contrast, we found that RsaM (*M* in our notation), a protein of unknown structure that negatively regulates QS in *P. fuscovaginae*[9], seems to only occur in the context of QS circuits in our analysis. Even though RsaM seems to occur mainly in the $\overleftarrow{R}\vec{M}\vec{I}$ arrangement, it also appears in a number of other topologies (M1, M2 and M3;

Table 1). Of the rest of the X genes found in RXI topology, *mupX* of *P.fluorescens* NCIMB10586 is an amidase-hydrolase,

Table 2. Intervening genes in short, conserved topological patterns

Gene-types	No. found in genomes	Potential role
rsaL	11	Negative regulator
rsaM	29	Negative regulator
mupX	1	Negative regulator
Integrases/ transposases	2	DNA mobilization
Unknown function	5	?
LuxR-like regulator	1	?

which is able to digest/degrade the AHL signal of the same species [15]. MupX can therefore also can be considered as negative QS regulator. From the rest of the X genes, two are involved in DNA-mobilization (an integrases and a transposase) the rest are hypothetical proteins of unknown function.

IV. Taxonomic Distribution of QS Gene Topology Patterns

AThe number of complete genomes does not allow us yet to draw definitive conclusions about the preference of certain patterns to appear in various classes or species of bacteria. It is apparent however that the R1 topology is predominant in α-proteobacteria, while the R2 pattern is more frequent in γ-proteobacteria. Furthermore, the L1 and the M1 topologies seem to occur in the β and γ classes but not in α-proteobacteria. The question arises whether the known QS proteins, such as LuxI and LuxR, cluster simply according to the known taxonomy or rather according to the topological pattern.

An example in Figure 2 shows that LuxI proteins present in L1 patterns clearly separate from those present in R1 patterns within the same genome and cluster together with the respective genes of another (β or γ) class (Figure 2(A)). At the same time, the clustering of the RsaL proteins (Figure 2(B)) is identical to the clustering of their accompanying LuxI genes (Figure 2(A)). In other words, QS proteins seem to cluster according to gene topology at various taxonomic levels, which suggests that the luxR, luxI as well as the intervening genes may have evolved together. Gene neighborhoods are known to evolve via complex rearrangements, with different combinations of genes from a neighborhood fixed in different lineages [16, 17]. However, when the genes are part of fine-tuned regulatory networks, such as QS circuits, strict constraints may be imposed on the process of rearrangement.

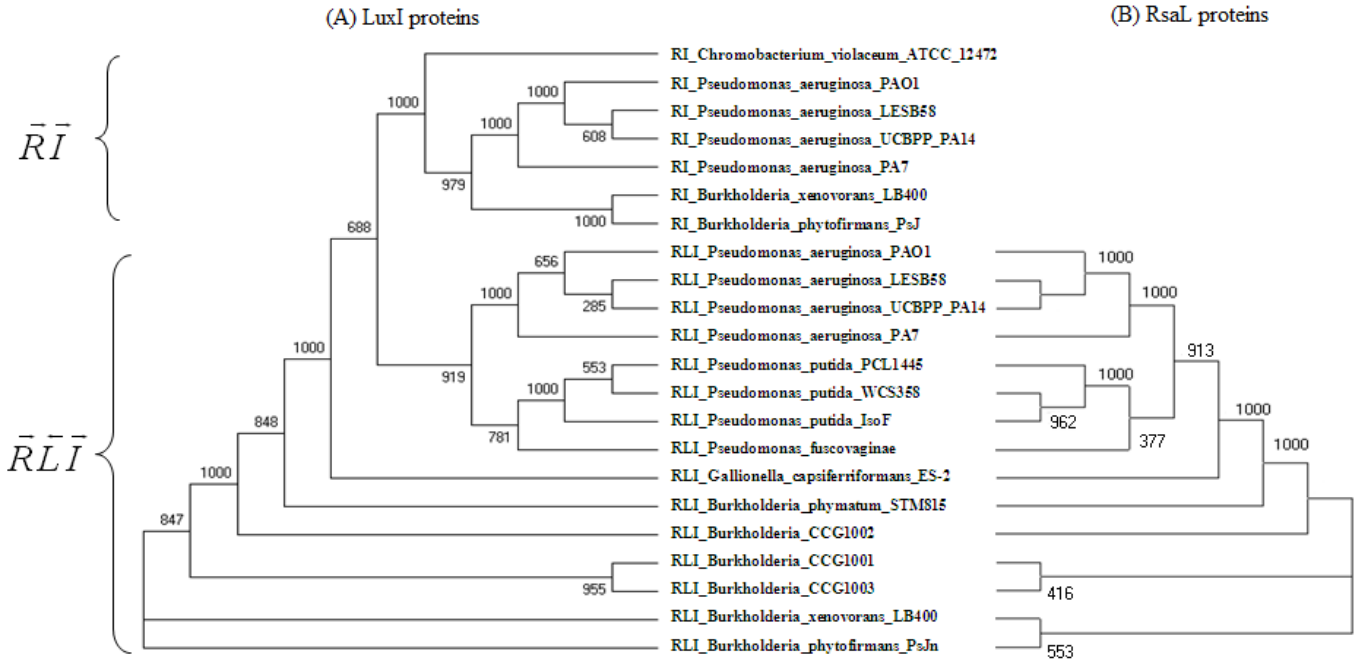


Figure 2. Clustering of LuxI and their negative regulators RsaL shows identical tendencies. (**A**) shows tree of LuxI proteins present RI and RLI pattern, and (**B**) shows RsaL proteins in RLI pattern. The sequences were taken from genomes harboring *rsaL* genes located between *luxR* and *luxI* genes. The clustering was carried out by the Phylip program package . The numerical value at each node indicates the bootstrap value supporting every split in the lineage (out of 1,000 bootstrap replicates).

V. Conclusion

We can conclude that the topological patterns in QS genes seem to follow a few basic rules—such as the conservation of genes between luxR and luxI genes, repeating patterns in certain taxa, etc.—but their variety is apparently greater than previously indicated. The clustering of QS genes suggests that the topological units might act as regulatory modules that evolve together. In order to further test these hypotheses, we plan to widen the scope of the analyzed databases so as to include in the survey draft genomes and individually sequenced genomic regions. We also plan to pinpoint conserved DNA motifs within the neighborhood of the QS circuits. Maintaining and manually curating such a collection is beyond the possibilities of a single research group. Rather, we consider using bioinformatic tools for updating the current collection in the form of an openly accessible Internet repository so that researchers of the field could add their own annotations in the future. This work is a preliminary, computational census of topological patterns found in complete proteobacterial genomes. It is expected that other kind of patterns may be found in published papers and in other sources, especially draft genomes, and we plan to develop computational protocols for finding these occurrences. On the other hand, there are a number of problems that are not touched upon in our work but may be answered as more detailed data become available. First, are the topological patterns associated with distinct biological functions? As QS circuits are involved in activating a large variety of genes in various bacteria, we speculate that there may be no simple correspondence between topology and the regulated functions. While we agree that these questions should be tackled by experiment, some supporting evidence can be gathered by comparing the gene neighborhood of QS circuits. The intervening genes (which are a conspicuous subgroup of neighborhood genes) appear to belong to only a few types, and in some genomes they are also found outside the immediate vicinity of QS genes. Another avenue would be to categorize the QS patterns according to the chemical nature of the signals produced by the luxI and/or sensed by the luxR gene. The present classification suggests that the patterns seem to be relatively conserved and their distribution among the various taxonomic groups is not random. This does not mean that one can make broad statements about the functional reasons of this apparent conservation. We tend to speculate that the known principles of gene expression (co-transcription, repression by proteins or RNA) can be combined into a finite set of topological patterns that allow stable positive autoregulation and control of linked genes. It would be tempting to speculate that such modules then can move between genomes but this problem, as well as further evolutionary questions are outside the immediate scope of the present paper.

Acknowledgment

This project was developed within the PhD program of Multidisciplinary Doctoral School, Faculty of Information Technology, Pázmány Péter Catholic University (Budapest).

I thank my supervisor Prof. S. Pongor (PPKE, ICGEB, Trieste) for his help and guidance throughout the project, for Kumari Sonal Choudharynak, Sanjarbek Hudaiberdievnek as well as Dr. Vittorio. Venturi and his group (ICGEB, Trieste) for their advice.

References

[1] W. C. Fuqua and S. C. Winans, "A LuxR-LuxI type regulatory system activates *Agrobacterium* Ti plasmid conjugal transfer in the presence of a plant tumor metabolite," *J Bacteriol,* vol. 176, pp. 2796-806, May 1994.

[2] W. C. Fuqua*, et al.*, "Quorum sensing in bacteria: the LuxR-LuxI family of cell density-responsive transcriptional regulators," *J Bacteriol,* vol. 176, pp. 269-75, Jan 1994.

[3] C. Fuqua*, et al.*, "Regulation of gene expression by cell-to-cell communication: acyl-homoserine lactone quorum sensing," *Annu Rev Genet,* vol. 35, pp. 439-68, 2001.

[4] C. M. Waters and B. L. Bassler, "Quorum sensing: cell-to-cell communication in bacteria," *Annu Rev Cell Dev Biol,* vol. 21, pp. 319-46, 2005.

[5] T. Bjarnsholt and M. Givskov, "Quorum sensing inhibitory drugs as next generation antimicrobials: worth the effort?," *Curr Infect Dis Rep,* vol. 10, pp. 22-8, Mar 2008.

[6] V. Venturi and S. Subramoni, "Future research trends in the major chemical language of bacteria," *Hfsp J,* vol. 3, pp. 105-16, 2009.

[7] N. A. Whitehead*, et al.*, "Quorum-sensing in Gram-negative bacteria," *FEMS Microbiol Rev,* vol. 25, pp. 365-404, Aug 2001.

[8] V. Venturi*, et al.*, "The virtue of temperance: built-in negative regulators of quorum sensing in Pseudomonas," *Mol Microbiol,* vol. 82, pp. 1060-70, Dec 2011.

[9] M. Mattiuzzo*, et al.*, "The plant pathogen *Pseudomonas fuscovaginae* contains two conserved quorum sensing systems involved in virulence and negatively regulated by RsaL and the novel regulator RsaM," *Environ Microbiol,* vol. 13, pp. 145-62, Jan 2011.

[10] L. Falquet*, et al.*, "The PROSITE database, its status in 2002," *Nucleic Acids Res,* vol. 30, pp. 235-8, Jan 1 2002.

[11] A. B. Goryachev, "Design principles of the bacterial quorum sensing gene networks," *Wiley Interdiscip Rev Syst Biol Med,* vol. 1, pp. 45-60, Jul-Aug 2009.

[12] A. B. Goryachev, "Understanding bacterial cell-cell communication with computational modeling," *Chem Rev,* vol. 111, pp. 238-50, Jan 12 2011.

[13] G. Rampioni*, et al.*, "The *Pseudomonas* Quorum-Sensing Regulator RsaL Belongs to the Tetrahelical Superclass of H-T-H Proteins," *J Bacteriol,* vol. 189, pp. 1922-30, Mar 2007.

[14] T. de Kievit*, et al.*, "RsaL, a novel repressor of virulence gene expression in *Pseudomonas aeruginosa*," *J Bacteriol,* vol. 181, pp. 2175-84, Apr 1999.

[15] A. K. El-Sayed*, et al.*, "Quorum-sensing-dependent regulation of biosynthesis of the polyketide antibiotic mupirocin in *Pseudomonas fluorescens* NCIMB 10586," *Microbiology,* vol. 147, pp. 2127-39, Aug 2001.

[16] T. Coenje and P. Vandamme, "Organisation of the S10, spc and alpha ribosomal protein gene clusters in prokaryotic genomes," *FEMS Microbiology Letters,* vol. 1, pp. 117-126, 2005.

[17] I. B. Rogozin*, et al.*, "Connected gene neighborhoods in prokaryotic genomes," *Nucleic Acids Res,* vol. 30, pp. 2212-23, May 15 2002.

Efficient bio-inspired shape description

Attila Stubendek
(Supervisor: dr. Kristóf Karacs and Dr. Tamás Roska)
stubendek.attila@itk.ppke.hu

***Abstract*— A megapixel image taken by a camera has large dimension. Despite the huge amount of data, human queries - meaning, details, of the image - usually cannot be answered directly. Generating low dimensional but meaningful features, as an intermediate level provide useful and simple data which can be compared to previously labeled examples, and used in classification.**

In this paper I investigate bio-inspired image and shape feature generators. I introduce two attempts for fast and efficient shape feature-generation based on the Projected Principal Edge Distribution. Finally I use and evaluate the developed descriptors.

***Keywords* – object recognition, shape description, bio-inpired technology, CNN**

I. INTRODUCTION

The difficulty of object recognition is that we want the computers to imitate the human perception. As we learn more about the retina and brain, the algorithms also develop to get closer to human behavior. Many core-computers provide new horizons in bio-inspired object detection, including shape classification.

In the Section I. a short overview is provided about feature generation algorithms (Section I.A), including bio-inspired methods (Section I.B.) and shape classification (Section I.C.). Then I introduce the Bionic Eyeglass (Section I.D.)

In the Section II. I introduce the H-MAX algorithm (Section II.A), and the PPED (Section II.B.) as efficient bio-inspired feature generators.

In the Section III. I investigate the possibilities to use the PPED as a shape description (Section III. A), then I introduce two attempts to modify and complete the PPED to achieve higher performance in shape recognition. (Section III. A. and B.)

In the Section IV. I present the results of the introduced shape descriptors in the banknote recognition task of the Bionic Eyeglass.

A. Object detection by dimension-reduction

While the image consists of plenty of individual pixels, the meaning of the image or answers regarding the contents is short. Answering human questions regarding the image about existence of some patterns, detection and recognition of objects recognition can be formulated as a mapping function from the image pixel values (color or gray-value) to the answers.

In the most cases the dimension reduction is done at least in two steps. First a feature extraction function is applied resulting in a smaller vector of features, describing specific property of the image. These features can be color, shape, texture, etc. Then the object detection is completed by analyzing the features, or by comparing to previously labeled features.

B. Bio inspired feature generators

The goal of the object detection is to find that kind of details in an image, that a human eye and brain would also detect. The difficulty of description design is that the human and computational methods are fundamentally different. Computational algorithms consist of simple mathematical functions and precise numerical operations, which can depict only basic image features. The human visual procession involves complicated parallel network of uncertain operations, allowing detection of higher-level features.

The meaning of the difference and matching is also ambiguous in the human perception. Two objects recognized as same by a human can be far away from each other in the feature space, and vice versa.

C. Shape classification

In a shape classification task the shape is given as a binary image, as a result of a segmentation or pattern extraction. The size of the image and also the orientation can be different. A shape description is considered to be efficient, if a) the representation is meaningful and compressed, b) the feature vector is adequate for comparison, c) the representation is invariant to small changes and noise, d) the description is scale-invariant and e) rotation invariant. The shape can be described in various ways. The basic features, as area, extent, orientation, eccentricity, etc. are highly compressed and meaningful features, but they are not enough for classification. The edge-based shape descriptors use the contour of the shape; area-based descriptors consider the shape as a two-dimensional binary function or a statistical set of points. [1]

D. The Bionic Eyeglass

The Bionic Eyeglass is a portable device helping visually impaired in their everyday life in the situations when only visual information is available. The tasks performed by the Bionic Eyeglass are defined based on consultations with potential users. Already realized functions, besides others are crosswalk detection, cloth color and texture recognition or banknote classification. [2][3]

Figure 1. Blind volunteer using the banknote recongition function of the Bionic Eyeglass

Since the tool is efficient only if it works real-time, has low power consumption, high computational power and robustness, it is necessary to employ the most efficient algorithms and special hardware.

II. BIO-INSPIRED IMAGE AND SHAPE DESCRIPTORS

A. H-MAX

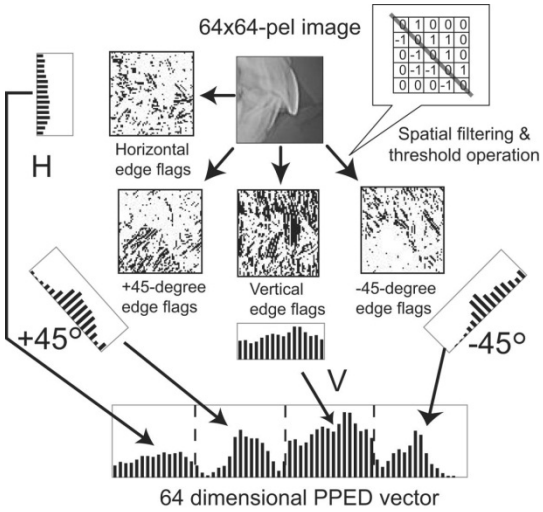
The H-MAX algorithm imitates the human visual cortex behavior starting from the orientation-specific cells to the complex, object-specified cells. The algorithm consists of four layer with different complexity, and between the layers only feed-forward connection is defined, from the simpler to the complex layer. The simple cells in the layer S1 detect the local orientation of the image region, then the complex cells in the layer C1 joins the same directions. In the Composite feature cells in the layer S2 local features are summed up, and finally in the Complex composite cells features are also joined to generate viewpoint-independent representation. [4]

B. The Projected Principal Edge Distribution

The Principal Projected Edge Distribution description is primary intended to describe 64x64 pixel square image regions. The image is scanned in overlapping windows, and the resulted feature vector for every window is compared to labeled templates.

The PPED feature vector for a 64x64 pixel window is generated in the following way. The edges of the image region are detected in vertical, horizontal, -45 and 45 degree directions respectively. For every pixel a threshold value is computed as the median of the differences of neighboring pixels. For every pixel location the maximal edge value is selected as principal and preserved only if exceeds the threshold value. The principal edge value flags are then projected in the same direction as the edges, resulting four edge vectors. After concatenation averaging and smoothing is applied to reduce local noise and increase the robustness. The final result for a window is a 64 long feature vector.

The method involving principal edge-detection in various directions, and adaptive threshold computation tries to copy the basic human visual processing. The PPED and its modifications (APED, CPED) was successfully used for detecting X-RAY image regions and also faces. [5]



. Figure 2. The generation of the PPED feature vector

III. BIO-INSPIRED SHAPE DESCRIPTORS

A. The PPED as a shape descriptor

Since PPED describes the edges of the image region, it offers the possibility to use it as a shape descriptor. However some modification, mainly simplifications can be applied.

To achieve scale-invariant shape analysis I resized the shape to a uniform 64x64 pixel, preserving the original aspect ratio. In previous works it showed that resizing to bigger size is unnecessary.

The locally adaptive thresholding is not the primary role of the shape description, but the preprocession resulting the binary shape image. Nevertheless the differences between neighboring pixel gray-values in a binary image are 0 or 1 (1 for in-shape pixels and 0 for others), consequently the median value is also 0 or 1. Though the results showed that it is better to use constantly 2 as the threshold value to select only the important edges.

The modified PPED as a shape descriptor shows good, but not perfect performance. Since the result feature-vector is relatively small, a large dimension-reduction is performed, I completed the feature-generator algorithm by other information about the shape. The goal was to find extra features that are orthogonal to the edge information.

Since the PPED can be considered as an edge-based descriptor, as orthogonal information I chose features that describe the whole area of the shape. In the Moments-Edge based Shape Descriptor presented in the section III.B. I involved eccentricity, extent and the first four moments; and in the Skeleton-Edge based description I used the skeleton of the shape as an extra data.

The deficiency of the PPED is also the lack of rotation-invariance. In the following two attempts to solve the rotation-

invariance I applied re-rotation by the orientation angle and rotation-variant template-vectors.

B. The Moments-Edge Shape Descriptor

1) Feature extraction

In the first attempt I completed the PPED vector by eccentricity and extent property; and also by the first four statistical moment of the shape. I used these two data in decision for selecting only similar candidate templates thus rejecting other samples. Finally the classification was made based on the PPED feature vector.

The eccentricity (in the case of shapes) is a single value between 0 and 1. The smaller the eccentricity is, the closer is the shape to a circle, while shape with eccentricity value of 1 is a line. The eccentricity is an appropriate property to categorize the templates as well the input. The area ratio, as the ratio of the area occupied by of the shape and the area of the minimal rectangle consisting the shape is also a simple and effective characterization property of a shape.

The two-dimensional statistical moments are frequently used as shape descriptors. Using more moments enables to describe the shape more in details, but loose the general recognition ability. [6] Since I used the statistical moments only as complement features I compute only the first four moments.

2) Rotation invariance by re-rotating

To achieve rotation invariance a characteristic angle had to be detected, to re-rotate the shape, and to have a normalized rotation. For shapes having small eccentricity (i.e. circles) the method is unambiguous, but shapes usually have higher eccentricity. As a characteristic angle I used the orientation property, more exactly the angle difference of the ellipse having the same first two moments as the shape.

3) Gradual decision

The shape description described below enables to make gradual decision.

First candidate templates are selected from the template set based on the eccentricity, extent and moment values. For each of these properties a threshold is determined based on preliminary measures. Only those templates become candidate templates, which distance of the first 10 elements of the input feature vector is within the limits. After the candidate selection the input is compared to each of the templates based only on the PPED vector. The nearest template is selected in the Euclidean space, but it is accepted only if the distance is smaller than the minimal acceptance threshold of the template sample described in the section III.B.4.

4) Rejecting non-class input

The deficiency of the Nearest Neighborhood decision, , as illustrated in Figure 3, is that normally it cannot reject those inputs, which do not belong to any class, because without specifying additional constraints there always will be a nearest element to every input vector, even if the distance is high. Defining a threshold is an obvious solution but the value of the threshold is dependent on the task and may vary for every template.

To determine the minimal acceptance threshold I used an automatic generation algorithm. In human recognition the classification is done by knowing which previously seen objects does not belong to the classes. Therefore I collected a huge database of shapes consisting various shapes called zero class examples. Then for every in-class template element I found the nearest zero-class example, considering the first 10 vector elements of basic properties and moments, and defined the minimal acceptance threshold as the half of the distance to it. The minimal acceptance threshold is automatically computed for every template shape.

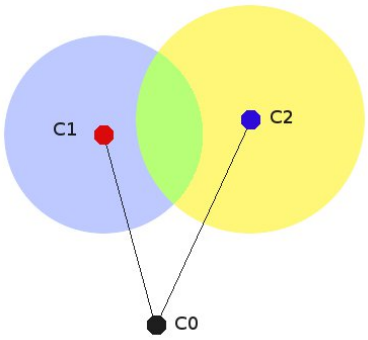


Figure 5. Automatic definition of minimal acceptance range. C1 and C2 represent two in-class elements, C0 a zero-class element. The minimal threshold is the half of the distance from the in-class element to the zero-class one.

C. Skeleton-Edge Shape Descriptor

1) Feature vector generation

From the aspect on the information source, shapes can be split into two groups. The symbolic shapes carry the meaning in their skeleton (letters of alphabets, etc). The area based shapes code the information in their contour (circle, siluette of a car, etc.). The PPED describes the edge of the shape, hence I chose skeleton, as the complement descriptor; and since the PPED encodes the spatial distribution of the shape in directional projections, to represent another spatial split, I chose square cell distribution of the skeleton. I defined 16, 16x16 pixel non-overlapping windows, form a 16 long vector from summing up the skeleton pixels in each window.

The advantage of employing skeleton distribution is that the skeleton can be computed iteratively using local operations on neighboring pixels. The algorithm of the PPED, detecting directional edges and summing up can also be performed locally. Hence the combined algorithm is feasible to implement paralelly, and is effective on multi core processors, including Cellular Neural/Nonlinear Network. [7]

2) Decision

The skeleton and the edge values represent different property of the shape. To use the feature vectors for classification I normalized and concatenated the two vectors, and computed the distance on the concatenated vectors.

IV. Experimental Results

1) *Moments-Edge Shape Descriptor*

I tested the Moments-Edge Shape Descriptor on the banknote-portrait shape database obtained from the Banknote Recognition task of the Bionic Eyeglass.[8] Images were taken by visually impaired during live tests, processed by a morphologic feature-extractor [9], resulting portrait candidate patterns. I selected 600 shapes of 6 classes corresponding to different denominations, and also 200 non-class shapes as training set.

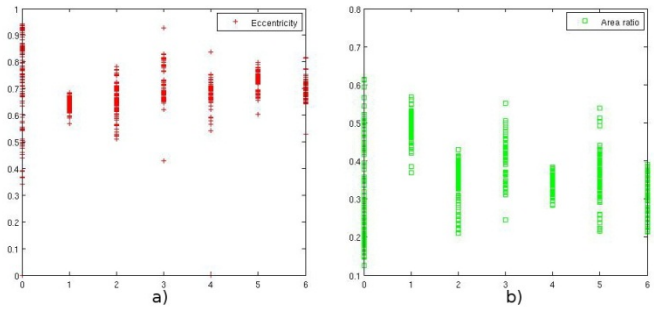


Figure 6. a) Eccentricity and b) Area ratio distribution of the 6 class and the zero-class train elements

The Figure 6. shows the distribution of the eccentricity and area radio values. The values are overlapping, but for many cases large number of templates, or even complete classes can be rejected.

I tested the model on database of 2700 portrait shapes. In average the feature extraction took 0.09 seconds on a single core CPU at 2.3 GHz. The accuracy of the classification was 99.1%.

2) *Skeleton-Edge Shape Descriptor*

I tested the Skeleton-Edge Shape Descriptor on a shape database consisting company logos. In the training set I put 25 different logos. In the test set I distorted the logos by rotating, aspect ratio change and adding random shape noise. I made rotations from -20 degrees to 20 degrees, aspect ratio change maximally by 30%, and added shape noise maximally 10% of the shape area to the image. Figure 7 shows sample images from the test set.

Figure 7. Sample shapes of the test set.

The global accuracy on the test set was 81%, the precision (the ratio of positive decisions and the total number of decision) and the cover (ratio of decisions and the positive elements) also. The Figure 8. shows the accuracy depending on the type and strength of distortion.

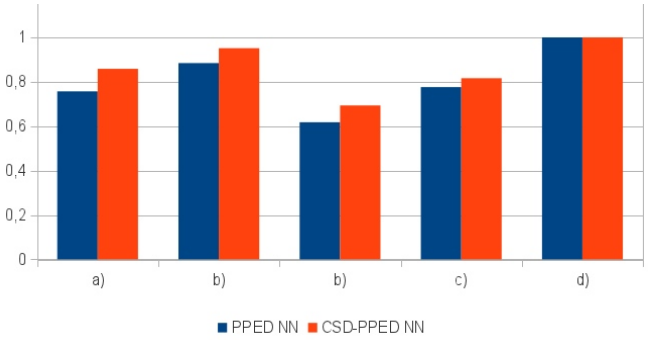


Figure 8. The robustness of the Skeleton-Edge Description in accuracy values, a) rotation by maximally 20, b) maximally 10 degrees, c) aspect ratio change by maximally 30%, d) maximally 10%, and 10% random shape noise.

Summary

In this paper I presented two approaches of bio-inspired shape descriptors and classification system. The results show that the descriptors are efficient both for classic computational architectures and for many-core processor architectures.

Acknowledgment

The help of University of Tokyo and Mr. Tadashi Shibata is gratefully acknowledged.

References

[1] R.C. Veltkamp, M. Hagedoorn, “State-of-the-Art in Shape Mathcing”, Principles of visual information retrieval, Pages 87 – 119, 1999.

[2] K. Karacs, A. Lázár, R. Wagner, D. Bálya, T. Roska, and M. Szuhaj, “Bionic Eyeglass: an Audio Guide for Visually Impaired,” in Proc. of the First IEEE Biomedical Circuits and Systems Conference (BIOCAS 2006), London, UK, Dec. 2006, pp. 190–193.

[3] K. Karacs, A. Lázár, R. Wagner, B. Bálint, T. Roska, and M. Szuhaj, “Bionic Eyeglass: The First Prototype, A Personal Navigation Device for Visually Impaired,” in Proc. of First Int’l Symp. on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2008), Aalborg, Denmark, 2008.

[4] M. Riesenhuber , T. Poggio, “Hierarchical models of object recognition in cortex”, Nature Neuroscience, pp. 1019-1025, 1999.

[5] Masakazu Yagi, Tadashi Shibata, "An Image Representation Algorithm Compatible with Neural-Associative-Processor-Based Hardware Recognition Systems," IEEE Trans. Neural Networks, Vol. 14, No. 5, pp. 1144-1161, September (2003).

[6] A. Stubendek: “Human like semantic models for object detection and classification”, PPKE – ITK PhD Proceedings 2011, Budapest.

[7] T. Roska and L. O. Chua, “The CNN universal machine: an analogic array computer,” IEEE Trans. Circuits Syst. II, vol. 40, pp. 163–173,Mar. 1993.

[8] Z. Solymar, A. Stubendek, M. Radvanyi, K. Karacs, "Banknote Recognition for Visually Impaired" in Proc. of the European Conference on Circuit Theory and Design (ECCTD’11), Linköping, Sweden, Aug 2011.

[9] M. Radványi: “Visual Feature Detection and Classification for Banknote Detection on Low Resolution Images”, PPKE – ITK PhD Proceedings 2011, Budapest.

Understanding Image Flows on Mobile Platform

Zóra Solymár
(Supervisor: Dr. Kristóf Karacs)
solymar.zora@itk.ppke.hu

Abstract— **Cell phones seem to be an adequate platform to house an assistive device, since people already carry them, and they have also learned how to use them. With my colleagues we have developed an intelligent algorithmic framework that can process important ambient visual information and provide the user with short and informative results. We consider two main use cases: either the cell phone is used on its own or an external computing unit is attached to it wirelessly to increase the available computing power. Results of tests with blind subjects are analyzed and compared to algorithmic results. Portable assistive devices are getting more and more widespread. Since most cell phones are nowadays equipped with cameras they can provide an easy way for blind and visually impaired people to obtain visual information about their environment. In this paper we present a banknote recognition system for our previously proposed mobile device, called the bionic eyeglass. We employ a recognition scheme based on the structural topology of banknotes that enables that the visual features detected are not joined in an ensemble classifier independently, but they are coupled via an excitation-inhibition model. We have shown that the technique outperforms classical ensemble classifiers.**

***Keywords*- bankote recognition, real-time video processing**

I. INTRODUCTION

Blind and visually impaired people have several difficulties performing daily activities. Dozens of solutions have been proposed and realized in order to help them and improve the quality of their lives, but only the most reliable and simple ones remained in use. Traditional aids such as white canes, guide dogs or tactile pavings are primarily intended to help blind and visually impaired pedestrians in navigating in urban environment, but are not feasible in common situations requiring visual input and more advanced visual understanding. Situations where traditional aids have limitations are very common in our daily lives such as recognition of banknotes, reading LED/LCD displays or avoiding chest and higher level obstacles.

The main approaches of vision restoration for blind and visually impaired people vary from invasive technologies - such as retinal implants or optogenetic researches - to non-invasive external, handheld, mobile devices. Functional rehabilitation through universal electronic devices are more widely accessible for blind and visually impaired people, since no special diagnose is needed for usage.

Assistive technologies that help the lives of visually impaired people get more and more widespread. Devices with single functionality solve specific problems of the target user group properly, but leading the users to carry and use several devices during their daily routine. There is a clear demand on having general purpose - universal - assistive devices. Mobile phones are ideal platforms to provide ambient intelligence functionality, since people do not need to buy and carry a new device. Taking into account that blind and visually impaired people are already confident using mobile phones, and that today's cell phones are equipped with a continuously expanding suite of embedded sensors - such as accelerometer, digital compass, gyroscope, GPS, microphone, and camera - cell phones are promising platforms for developing intelligent mobile devices. Mobile Internet connection provides the possibility to use remote central knowledge-base in navigation or object detection tasks, minimizing the local computing consumption of the device. Existing solutions using central knowledge base are equipped with general databases or task specific target databases to search in.

Since the computing power of mobile devices are continuously increasing, local computing can be accomplished more and more efficiently eliminating the need of a remote knowledge bases. From the number of sensor modalities incorporated into mobile devices, vision is the most commonly used on in assistive technologies for the blinds. Instead of forcing the user to take adequate still images of the target object or scene it is more convenient to process image flows at a few frames per second and to track detected features from frame to frame until a confident decision can be made.

In this paper I present the Bionic Eyeglass [1, 2] - a portable recognition device based on a smartphone. The rest of this paper is structured as follows. Section II deals with the system architecture. Section III elaborates on the writer's work within banknote recognition – including detection of interesting banknote patterns in the field of view and a corner estimation algorithm. Section IV gives an overview of the experiments performed and an analysis of the results is shown.

II. THE BIONIC EYEGLASS FRAMEWORK

A. System aspects

When designing a mobile navigation device for visually impaired people, we have to consider more aspects then for usual systems. Besides low consumption and mobility, the need for accessibility yields more constraints in selecting the proper hardware architecture.

The key challenge in creating such a device is to be able to process video flows in the selected situations in real time on a

mobile device, where mobile refers to the following requirements: (i) portable, (ii) small form factor, (iii) autonomy of at least 6-8 hours. In order to satisfy these requirements we need a computational platform that has high computing power as well as low power consumption.

We expect processes of high complexity (such as continuous monitoring of the environment and image processing) to run as long as possible, therefore high computational power at low consumption is a key issue.

An accessible user interface is essential. The basic options for input are tactile and auditory, where tactile can mean either buttons or touchscreens. Unfortunately for the blind, touchscreens have become more and more popular nowadays, yet it is impossible for them to feel where to point their fingers. The natural need for input interface for these people is buttons. Audio is less usable in general urban noise and cannot be applied in places demanding silence (e.g. theaters, conferences, etc.).

Considering the previous aspects it seemed useful to follow the smart-phone market with attention. In the past few years mobile Central Processing Unit (CPU) technology has developed fast giving base to a new generation of mobile navigation systems. The hardware requirements for the functions - e.g. camera, processing unit, mobility, low consumption are mostly available in smart-phones. But do the mobile phones available in the market have enough performance to run real time image processing functions for hours? Toytman and Thambidurai did banknote recognition tests on Android platform and their average results were about 30 seconds on Motorola Droid phone equipped with 800MHz CPU. [3] Instead, we expect that such function gives feedback several times within a second. Given the speed of development of mobile CPU it may become possible to achieve that time limit within 1-2 years, but other options are also can be considered.

Processing Unit (GPU) is also in the state of fast development. López et al. achieved remarkable acceleration in mobile image recognition by using Nokia N900 Mobile phone, included an ARM Cortex-A8 CPU and a PowerVR SGX535 GPU included in an OMAP3530 platform. [4]. Since the number of mobile phones with integrated GPU inside is increasing these days, GPU technology could grant a cellphone-based only architecture for our navigational system.

Embedding another high-performance computing device gives an option to increase speed, yet in this case, a communication protocol between the device and the mobile has to be built and can cause latency. Dedicated, task-special hardware devices are able to achieve the same power at lower consumption, compared to the mobile phone market. Field Programmable Gate Array (FPGA) technology is great example for such dedicated hardware. An FPGA board with Ethernet interface would be easily embeddable in the system by sending the image flow from the camera of the mobile to the FPGA either through wireless or direct connection. Ongwattanakul et al. showed that FPGA architectures are a good choice to implement morphological operators therefore the attached FPGA would be responsible for the image processing part.[5]

We see that there are basically two scenarios – one is the cellphone with an embedded special device other is cellphone-based only architecture. Since the functions are in the state of development, we tried to come up with a prototype that could be improved in both directions, and where the algorithms are easy to modify through a MATLAB developer interface, yet it can be tested by visually impaired people with the impression of a mobile navigation system.

B. Developer Interface

We built a communication protocol through Wi-Fi between an Android smart-phone and a laptop, where the cellphone sends the camera image flow captured by the camera to the laptop. (Figure 1.)

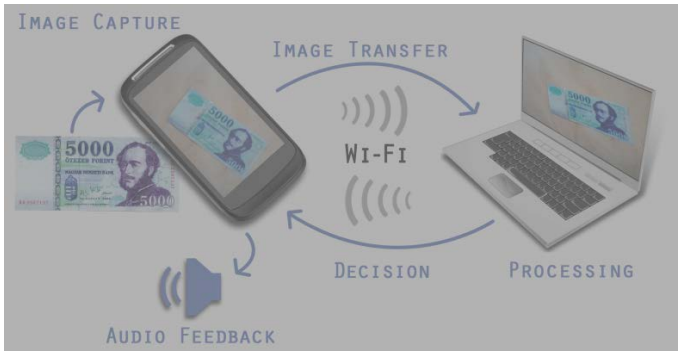


Figure 1. The Bionic Eyeglass prototype – a smartphone interacting with a laptop

The processing runs in MATLAB at a few frame per second, depending on the complexity of the actual task, and the answer is sent back to the mobile phone where auditory information is given to the user. Different recognition functions can be tested in the proposed way, where the user has to operate only with the cellphone.

III. BANKNOTE RECOGNITION

In countries where banknotes cannot be distinguished by their size people with visual impairments have real difficulties in determining their face values. Some currencies have been designed with tactile marks embossed on the banknotes, however, due to everyday usage they lose most of the tactile information in a short time and become practically unusable for the intended target group.

A. Tactile markers – detection and classification

Since our aim was to develop a side-dependent recognition algorithm, we had to focus on an extractable area that uniquely identifies the banknotes. Considering the Hungarian notes, one can find different symbols on the upper right corner of the backside:

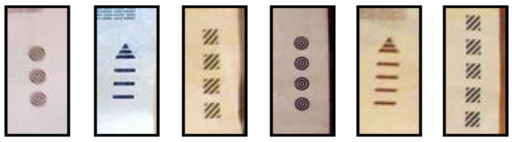


Figure 2. Markers on the upper right corner.

These embossed marks were originally designed for blind people but after a few weeks of usage they become hardly tactile, still the visual information remains. The right side of the banknotes, especially the upper region has light background and low density, therefore it is easy to extract. The following morphological steps are performed:

1. Binary conversion with threshold returned by the module discussed in section I. A.
2. Morphological closing using a 4 x 4 kernel
3. Remove objects with an area less than 350 pixels
4. Take XOR of the results of step 2 and 3
5. Remove objects with an area less than 10 pixels

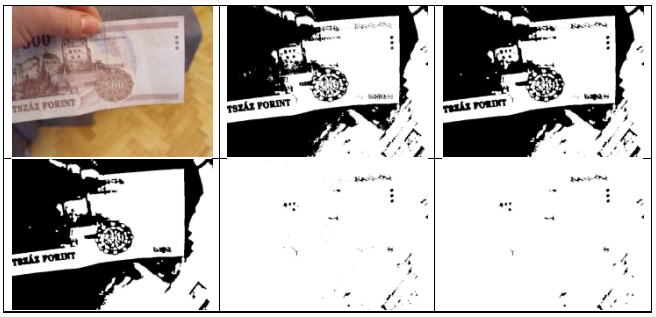


Figure 3. Steps to extract the tactile mark region on the back side of the banknote

Patches in the resulting binary image are grouped together into sets of 3, 4 or 5. Initial rules for pairing two sets are:

1. Distance between centers of mass are less than 35 pixels.
2. Difference between the areas is under 50%.

Further matching requires angle analysis. When forming a group of three sets, the bearing between the first two centers of mass and the bearing between the second and the third must not differ more than 15 degrees. The rules adapt locally as the sets grow, considering the local features of the actual patch sets. Adaptation means, that once we have a pair of two sets with a given d distance, we don't look for a third object with mass center within 35 pixels, but within d*1.15. For example, having two sets with distance 20 pixels, we look for a third object within 23 pixels. Same rules apply to degree and area difference. The parameters used for morphological steps and patch grouping are optimal for 640x480 images, at other resolutions they are modified proportionately.

Ideally the patches in the group corresponding to the region of interest have equal area and equidistant center of masses that lie on the same line. In the following step we select the group with minimal deviation with respect to these features. Once the decision is made, we have the presumed upper right corner of the patterns.

To classify the results, first all symbols are resized to a 10×10 window. The features extracted for the classifier are summarized in Table I.

TABLE I. *FEATURES FOR TACTILE* MARK CLASSIFICATION

Name of the Feature	Description
Number of Objects	The number of connected components in the region of interest
Axis Ratio	The ratio of Major Axis Length to Minor Axis Length
Extent	The ratio of pixels in the region to pixels in the total bounding box
Distance from Center	The mean distance between 4 extreme points and the mass center point
Distance and Major Axis Ratio	The ratio of mean distance between the centroids of the symbols and the Major Axis Length
Solidity	Scalar specifying the proportion of the pixels in the convex hull that are also in the region.

Using a test set of 5000 backside banknote image reference feature vectors were calculated for each class using K-means algorithm. During classification a simple Euclidean distance is calculated between the actual feature vector and the references, and the output is the class belonging to the smallest distance.

B. Corner estimation

In certain cases the morphological detection algorithm returns acceptable results on detecting tactile marks, but the classifier returns without having definite answer. This happens typically with the denomination 2000 and 5000, when the sizes of the detected markers are too small for accurate shape comparison. In such cases even though no acceptable decision is made, the position and orientation information of the detected markers can still be useful for other detection modules. For instance assuming the threshold returned by the adaptive binarization algorithm was not suitable for detecting the number patches, corner coordinates of banknotes can be determined, and it can yield to better result in number detection. Thus corner coordinates have to be estimated from the presumptive position and orientation of the tactile markers. During grouping tactile marks together – described before – the orientation of the set is already determined, and the mean distance between the mass centers is also calculated. Let d denote the mean distance between centers in the rest of the paper. Based on the center point and orientation of the group a coordinate system related to banknote can be defined. Center point becomes the origin, and the ordinate points to the direction defined by the orientation of the marker group. The ordinate – y axis – is than parallel to the shorter edge of the banknote, leading us to detect the upper right corner easily, by simply going among the y axis. In order to define which direction (y– or y+) the upright corner appears along the ordinate, one has to consider the following. On every Hungarian banknote a thick, darker line appears on both the top and the bottom part, and since the top line is always closer to the tactile markers, by analyzing small ROIs in both y+ and y– directions, upright corner can be estimated. Experimental

results showed that selecting the ROI to be a d*d square on a 4*d distance from the origin gives the best results. Once the upper right coordinate is calculated, the rest of the corners can be determined as well, based on the available information and the aspect ratio of the Hungarian banknotes. Fig. 5 shows an example of corner estimation for banknote.

Figure 4. Corners detected based on the upper right tactile markers

visualization. The outer rectangle indicates the estimated position of the banknote, the inner rectangle corresponds to the ROI it was derived from.

C. Ensemble Classification

Having multiple weak learners in the banknote recognition task – namely portrait, number, tactile marker, color and orientation classifiers – we had to develop an ensemble decider to name a denomination. Since dealing with an image flow, decision making is a spatial-temporal event where we can use priori hypothesis to predict the actual class.

$$y_i = argmax_{c \in C} s_i \ (1),$$

where y_i is the predicted class, C is the set of all possible classes, s_i is the summed vote vector (2).

$$s_i = s_{i-1} a + v_i w_i \ (2),$$

where $s_{i-1}$is the summed vote vector for the previous frame, a is the attenuation (the elements are equally 0.8 and $\dim(a) = \dim(C)$), v_i is the sum of the output vectors of each classifier ($\dim(v) = \dim(C)$) and every class has a probability for the actual frame), w_i (3) is a weight that amplifies v_i at the maximum argument depending on the previous five predictions.

$$w_i = \sum_{j=1}^{5} k_j \ (3),$$

where $k_j = 1$ if y_j equals y_i and 0 otherwise.

IV. Experimental Results

We performed experiments with five visually impaired subjects who were given a few basic instructions including hints on good measurement techniques and 14 Hungarian banknotes including all denominations. Their task was to name the note they were holding using the mobile banknote recognition system. With each subject two or three series of tests were carried out yielding 11 series altogether.

The ensemble decider made a decision 392 times during the eight tests, and 382 of them were correct, which corresponds to an algorithmic accuracy of 97.45%. Based on these votes, the subjects could correctly identify the banknote in 109 of the 112 cases. Typical causes for errors were covering the region of interest with one or more fingers, or placing the previous banknote too close, making it show up in consecutive rounds. We have noticed that as time went by during the tests, the ratio of errors reduced, and subjects have mastered the use of the device in less than half an hour.

References

[1] K. Karacs, A. Lázár, R. Wagner, D. Bálya, T. Roska, and M. Szuhaj, “Bionic Eyeglass: an Audio Guide for Visually Impaired,” in Proc. o the First IEEE Biomedical Circuits and Systems Conference (BIOCAS 2006), London, UK, Dec. 2006, pp. 190–193.

[2] K. Karacs, A. Lázár, R. Wagner, B. Bálint, T. Roska, and M. Szuhaj, “Bionic Eyeglass: The First Prototype, A Personal Navigation Device for Visually Impaired,” in Proc. of First Int’l Symp. on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2008), Aalborg, Denmark, 2008.

[3] I. Toytman and J. Thambidurai, "Banknote Recognition on Android Platform", EE368 Digital Image Processing, Spring 2011, Stanford University.

[4] Bordallo López M., Nykänen H., Hannuksela J., Silvén O. and Vehviläinen M., "Accelerating image recognition on mobile devices using GPGPU," Parallel Processing for Imaging Applications, Proc. SPIE, Volume 7872, 2011.

[5] Songpol Ongwattanakul, Phaisit Chewputtanagul, David Jeff Jackson, Kenneth G. Ricks, ”A Programmable Logic-Based Implementation of Ultra-Fast Parallel Binary Image Morphological Operations” in proc of the ISCA 18th International Conference Computers and Their Applications (ISCA 2003), Honolulu, Hawaii, USA, March 26-28, 2003.

Multi-joint Coordination in Manual Tracking

Bence Jozsef Borbely
(Supervisors: Dr. Jozsef Laczko, Prof. Dr. Andreas Straube, Dr. Thomas Eggert)
borbely.bence@itk.ppke.hu

***Abstract*—In the human motor control system there are different types of movement patterns and control methods. Continuous target tracking is one of the movement patterns that needs online control of the end-effector device (e.g. hand or pointing finger). Similar movements can be executed using different joint configurations of the arm because the human arm is a redundant kinematic manipulator in the 3D space. The main goal of this study is to evaluate whether target path predictability (as an internal constraint) has any influence on the control of the human arm as a multi-joint system in manual tracking movements using the redundancy-based Uncontrolled Manifold concept.**

***Keywords*-manual tracking; variance; uncontrolled manifold**

I. Introduction

The synergystic control of arm movements is an important property of the human nervous system. Reaching and grasping are often used to evaluate motor control processes because they represent an essential part of human movements [1]–[7]. In addition, effects of external modulation constraints (e.g. target shape) have been analyzed recently in reaching [8], [9], however other arm movement types are not so well examined, especially from the aspect of internal modulation constraints (e.g. path predictability in a tracking task).

Reaching and tracking are different from the control point of view, because in reaching there is a stationary target with a known position in our reference frame, therefore the nervous system needs to solve a feed-forward type problem to reach the target with the hand. Contrarily, in a tracking task there is a moving target that one wants to follow as accurately as possible. This involves countinuous error calculation and for good performance, tracking error minimization should be one of the goals of the nervous system which leads to a feed-back type controller method.

The main goal of this study is to evaluate whether target path predictability (as an internal constraint) has any influence on the control of the human arm as a multi-joint system in manual tracking movements. The effectiveness of control is described by means of joint angle variabilities. In addition to joint angle variances, manual tracking performance was analyzed by calculating constant and variable errors, and eye gaze was measured during movement execution. In the first stage of the study eye movement data was not analyzed.

Experimental setup design, measurements and data analysis have been performed at DFG Research Training Group 1091 ”Orientation and motion in space”, Munich, Germany.

II. Materials and methods

A. Subjects

Movements of 9 healthy subjects were measured and analyzed during the experiment. All of them had normal or corrected to normal vision to the viewing distance (40 cm in average). 7 subjects had right hand dominance, 2 of them were left handers. The male/female ratio was 7/2. The subjects had given informed consent prior to participation.

B. Experimental setup

During movement execution 3 properties were measured by a custom measurement system containing the following devices:

- An ultrasound-based movement analyzer system developed by Zebris Medical GmbH to record the spatial position of the subject's arm using ultrasound-emitting active markers attached to the anatomical landmarks of the arm. [8]–[10]
- A Wacom digitizer tablet capable of displaying VGA content on its screen to present the developed visual stimulus and record hand movements using a special pen device [11].
- A binocular VOG device (EyeSeeCam) developed at the University Hospital Munich Grohadern to record movements of the eyes during movement execution.
- A chin rest to prevent trunk and head movements during the experiment (trunk movement is highly reduced if the subject has to keep his/her head in a stable position, therefore no belts or any other fixing devices had to be used).

All measurement devices had their own control computers and those were connected to a main recording computer running a real-time operating system (QNX). Using this setup, real-time storage of all measurement data was achieved. Fig. 1. shows the experimental setup.

C. Visual stimulus and procedure

Specially designed visual stimuli were presented to the subjects on the Wacom digitizer tablet to make them perform the desired manual tracking movements. To deal with the trade-off between path randomization and repetition (variance calculation requires many trials with the same conditions), we

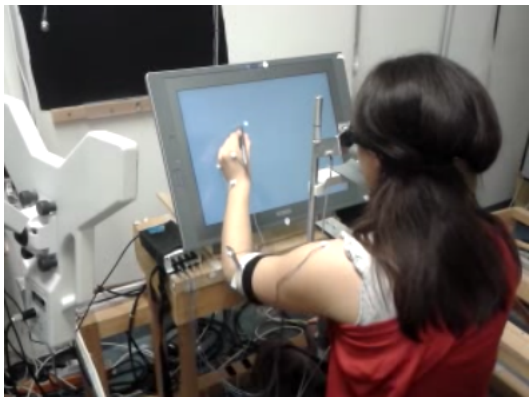


Fig. 1. Experimental setup

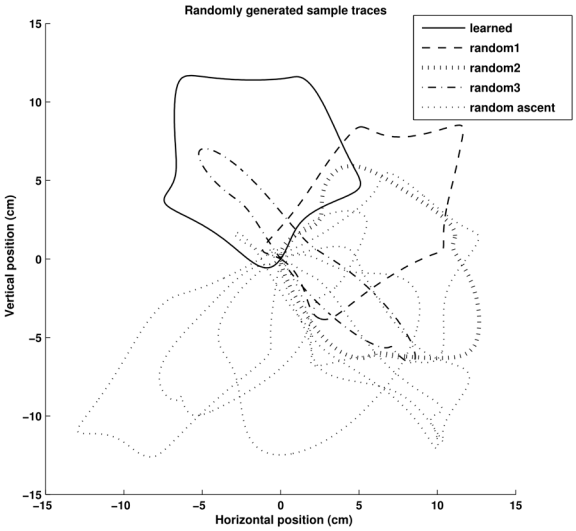


Fig. 2. Examples of L, R_1, R_2, R_3 and R_i traces

decided to generate 4 periodic random traces (Fig. 2.), and later on categorized one of them as the predictable or learned part (L), and the rest as random parts (R_1, R_2, R_3). Using this categorization we were able to build the stimulus paths as follows:

- Both predictable and unpredictable paths had 4 times the length as the generated traces.
- The predictable path was constructed by appending the so-called learned part (L) 4 times after each other, so in total we had exactly the same motion pattern on the screen 4 times for the visual target ($PATH = [L|L|L|L]$).
- The unpredictable path was constructed by appending a randomly generated introductory part (R_i) and the previously generated random parts, in a permuted order (e.g. $PATH = [R_i|R_2|R_1|R_3]$).

Having 3 different random parts to be permuted, an unpredictable block of these path variations was generated containing all 6 possible combinations of R_1, R_2 and R_3 following the R_i section, which was randomly regenerated in every case. This resulted in 6 paths, each having a real random introductory section ($1/4^{th}$ of total length) and a randomly permuted second section presented as one continuous path ($3/4^{th}$ of total length). To prevent the subjects from learning these "random" paths, the predictable and unpredictable paths were mixed up in an alternating way, so the final block structure looked like the following:

$$[L|L|L|L]$$
$$[R_i|R_1|R_2|R_3]$$
$$[L|L|L|L]$$
$$[R_i|R_1|R_3|R_2]$$
$$\ldots$$

Each of these blocks contained 12 paths determining individual trials, and 5 of these blocks were used in the experiment. Because the main interest of the study is the steady-state response, the first part of each path was left out from data analysis, so in sum 90 repetitions of the learned trace (L) and 30 repetitions for each of the random traces ($R_1 - R_3$) were analyzed per subject.

It is important to note that at this point the predictable part (L) would be just as unpredictable to the subject as the others ($R_1 - R_3$). To assure predictability, a learning phase took place before the measurement of the 5 blocks. This phase had its own block, containing those paths that were constructed using the L part only. 10 repetitions of these paths were used to make the subject familiar with the L trace, making him/her to be able to predict the movement of the visual target on its trace.

The instruction to the subjects was to follow the target with the Wacom pen on the screen as accurately as possible.

D. Data analysis

Recorded pen position data was transformed into centimeters and the constant and variable tracking errors were calculated. The constant error is the mean distance between pen position and target center, while variable error was defined as the variability of pen position direction with respect to the instantaneous tangential of the target path.

The recorded anatomical landmarks were transformed into 7 consecutive Cardan angles to express arm movements in the space of joint angles (for details see [9]). If subjects execute a movement task with identical measurement conditions repeatedly (e.g. in trials), total variability per DoF across trials can be calculated by computing the normalized variance of joint angles per total DoF. This variability, however, contains the sum of both kinds of joint angle variances: those that affect the variability of the task variable (in this case endpoint of the kinematic chain) and those that do not. A more detailed view of arm control can be achieved by decomposing the space of joint angle variances into 2 orthogonal subspaces (V_{ucm} and V_{ort}) by means of the uncontrolled manifold (UCM) concept. This decomposition is based on the Jacobian matrix of the instantaneous mean joint configuration (across trials). This matrix contains the partial derivatives of hand

position cordinates with respect to the joint angles, therefore it defines a linear relationship between angular changes of the arm and spatial movement of the endpoint. By calculating the null space of this linear mapping we can get the basis vectors spanning a subspace in which any element (e.g. a particular joint-configuration change) will not result in the spatial movement of the endpoint. This subspace is called the uncontrolled manifold and variability in this subspace is V_{ucm}. It is often referred as "good variance" because it does not have to be controlled or minimized by the nervous system to achieve the movement goal. Conrarily, computing the UCM's orthogonal subspace we get the basis vectors spanning a subspace in which any element (e.g. a particular joint-configuration change) will result in the spatial movement of the endpoint, therefore variability in this subspace (V_{ort}) should be minimized by the nervous system to reduce movement noise. Therefore, this variability is often referred as "bad variance". The UCM analysis was performed using the QR-decomposition of the actual Jacobian to get its null space (UCM) and the orthogonal space (ORT), and the variance values of V_{ucm} and V_{ort} later on. During variance per DoF calculation a dimension reduction was made in Vort, because the task variable (pen position) had only 2 active dimensions ($dim_{tot} = 7$, $dim_{ucm} = 4$, $dim_{ort} = 2$).

It is important to note that the UCM decomposition is based on the Jacobian of a virtual arm position (mean across trials) to get V_{ucm} and V_{ort}, therefore the method described here is only appropriate for descriptive statistics, but in other applications the principle can be used for real-time kinematic chain control (e.g. obstacle avoidance), too.

III. Results

The analysis of the recorded data showed the following results:

- the constant tracking error was significantly larger for the unpredictable path condition (Fig. 3. top),
- the variable error showed almost no difference between the two conditions (Fig. 3. bottom),
- V_{ort} was significantly smaller than V_{ucm} in each condition, but
- no difference was found between the conditions in V_{ort}, nor in V_{ucm}.

IV. Conclusion

The constant pen error results show that the subjects were able to learn the predictable trace because they produced larger errors in the unpredictable (random) condition, while they showed the same variable errors during the tracking. This could be caused by a mixed-mode operation of the nervous system: during the learning phase subjects "preprogrammed" a feed-forward controller to memorize the trace of the moving target - as a result, they produced better accuracy in that condition - while the fine tuning of the tracking was executed by a feed-back controller during real-time operation.

The UCM analysis results show that the human arm is highly redundant for the examined movement task, because

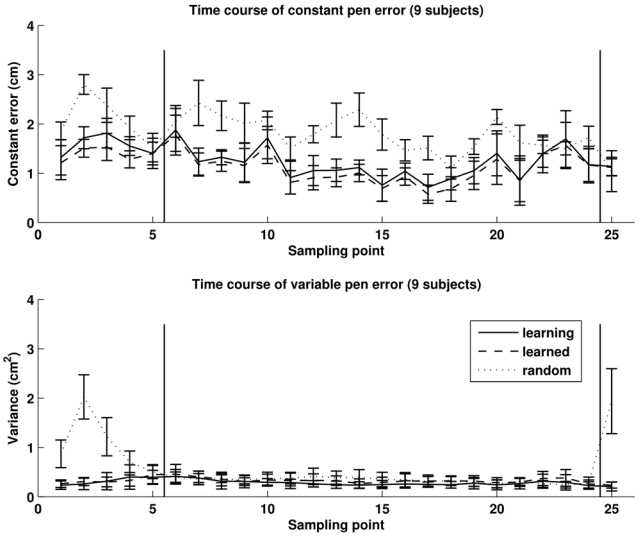


Fig. 3. Time courses of constant and variable pen errors. Vertical lines denote the borders of the analyzed period. This was needed because the larger pen errors in the beginning of the traces can be caused by the directional mismatch of the $R_1 - R_3$ parts.

V_{ort} ("bad variance") is significantly smaller than V_{ucm} ("good variance"). This means that most of the joint-configuration variability remained in "standby" mode - it did not induce noise in the task variable (endpoint) - resulting in a synergystic movement control scheme. It is assumed that using this additional variability the nervous system can handle additional external perturbations during movement execution. Comparing the experimental conditions the results show that target path predictability does not affect the measured movement. This can mean that the synergystic control of continuous target tracking movements is not affected by prior knowledge of the end-effector path but it is more closely related to the real-time feed-back principle.

Acknowledgment

This work was financially supported by the Research Training Group 1091 "Orientation and motion in space" of the German Research Foundation (DFG).

References

[1] M. Desmurget, C. Prablanc, Y. Rossetti, M. Arzi, Y. Paulignan, C. Urquizar, and J. C. Mignot, "Postural and synergic control for three-dimensional movements of reaching and grasping." *Journal of neurophysiology*, vol. 74, no. 2, pp. 905–10, Aug. 1995. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/7472395

[2] L. B. Bagesteiro, F. R. Sarlegna, and R. L. Sainburg, "Differential influence of vision and proprioception on control of movement distance." *Experimental brain research*, vol. 171, no. 3, pp. 358–70, May 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16307242

[3] P. Boulinguez, V. Nougier, and J. L. Velay, "Manual asymmetries in reaching movement control. I: Study of right-handers." *Cortex; a journal devoted to the study of the nervous system and behavior*, vol. 37, no. 1, pp. 101–22, Feb. 2001. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/11292156

[4] K. Y. Haaland, J. L. Prestopnik, R. T. Knight, and R. R. Lee, "Hemispheric asymmetries for kinematic and positional aspects of reaching." *Brain : a journal of neurology*, vol. 127, no. Pt 5, pp. 1145–58, May 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15033898

[5] K. Ohta, M. M. Svinin, Z. Luo, S. Hosoe, and R. Laboissière, "Optimal trajectory formation of constrained human arm reaching movements." *Biological cybernetics*, vol. 91, no. 1, pp. 23–36, Jul. 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15309545

[6] R. L. Sainburg and D. Kalakanis, "Differences in control of limb dynamics during dominant and nondominant arm reaching." *Journal of neurophysiology*, vol. 83, no. 5, pp. 2661–75, May 2000. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10805666

[7] R. A. Scheidt, M. A. Conditt, E. L. Secco, and F. A. Mussa-Ivaldi, "Interaction of visual and proprioceptive feedback during adaptation of human reaching movements." *Journal of neurophysiology*, vol. 93, no. 6, pp. 3200–13, Jun. 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15659526

[8] M. Krüger, T. Eggert, and A. Straube, "Joint angle variability in the time course of reaching movements." *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, vol. 122, no. 4, pp. 759–66, Apr. 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21030301

[9] M. Krüger, B. Borbély, T. Eggert, and A. Straube, "Synergistic control of joint angle variability: Influence of target shape." *Human movement science*, Jan. 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22244105

[10] R. Tibold, G. Fazekas, and J. Laczko, "Three-dimensional model to predict muscle forces and their relation to motor variances in reaching arm movements." *Journal of applied biomechanics*, vol. 27, no. 4, pp. 362–74, Nov. 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21896947

[11] J. Drever, A. Straube, and T. Eggert, "A new method to evaluate order and accuracy of inaccurately and incompletely reproduced movement sequences." *Behavior research methods*, vol. 43, no. 1, pp. 269–77, Mar. 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21287122

Increasing signal to noise ratio in terahertz imaging

Domonkos Gergelyi
(Supervisor: Péter Földesy, Phd.)
gerdo@digitus.itk.ppke.hu

***Abstract*—By applications of terahertz imaging the acquisition speed is a critical question. On the other hand the sources are weak that limits the achievable signal to noise ratio (SNR). Therefore it is a great challenge to find a good compromise between these two competing goals. In the followings I describe the possibilities of compressed sensing (CS) in relation with CMOS technology based terahertz detection.**

***Keywords*-compressed sensing; CMOS based terahertz detection**

I. INTRODUCTION

In the past years terahertz imaging has developed a lot and expanded to new fields. Surveillance [1] and material engineering [2] was followed by quality control, chemical imaging, and diagnostics [3][4]. However the high price of the different measurement configurations [5] hindered the technology from becoming more wide spread in real industrial applications.

CMOS based detectors are very competitive solutions regarding price and responsivity [6], but it is hard to fulfill application specific requirements. In this paper I investigate the increase of the resulting SNR by exploiting some domain specific information. This type of sensors can be 20 times faster than pyroelectric detectors and thermopiles. By forming small arrays they are a tradeoff between the complex, electro-optical setups and pure scanning [8]. Today kpixel sensor chips are already available. Due to their small area the terahertz beam can be maximally focused to increase the SNR. Hence they have considerable advantages in terahertz microscopy.

II. IMAGING SETUP

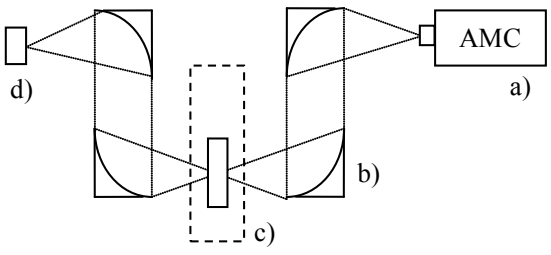


Fig. 1. Overview of the imaging setup: a, the terahertz source b, parabolic mirror c, the observed specimen within a high precision manipulator d, the focal plane sensor array.

The transmissive imaging configuration is depicted in Fig. 1. We built an active imaging setup with a YIG oscillator powered amplifier multiplier chain (AMC). The AMC is a purely semiconductor based waveguide multiplier that emits with its actual VCO compound about 1-0.01 mW power within the range of 0.15-0.52 THz respectively. For parallelizing and focusing the beam off-axis parabolic mirrors are utilized. The targets are placed into a high precision mover that operates in x, y, z direction with approximately 0.05 mm precision.

We tried out 3 different type of measurement configuration, namely transmissive, reflective and tomographic. The radiation is polarized and coherent. In this configuration we don't record any phase information, however this extension could be possible to exert more information about the observed object.

III. CS IN THZ MEASUREMENTS

In terahertz measurements even the windowing function plays an important role. Both constant offset and linear drift have to be canceled. Considering white, 1/f and $1/f^2$ noise power spectral densities (PSD) the "on-off-off+on" measurement scheme seems to be a standard.

As the terahertz sources are weak the SNR is critical in most of the cases. CMOS based detectors are fast. They give useable data within milliseconds and near 8 bit precision is achievable by 100 ms integration. These properties are competitive regarding other technologies, but terahertz imaging is still time critical. This gives grounds to apply compressed sensing.

A. Brief summary of the used CS notation

The compressive sampling of a discrete signal $\boldsymbol{x}$ is written as the multiplication:

$$\boldsymbol{\Phi x} = \boldsymbol{y} \tag{1}$$

where $\boldsymbol{\Phi}$ is an M by N random matrix containing only binary values with equal distribution. $\boldsymbol{x}$ is the signal vector and $\boldsymbol{y}$ is the vector of measurements. The signal can be reconstructed if a basis exists, in which the representation of the observed signal is sparse:

$$\boldsymbol{\Psi a} = \boldsymbol{x}. \tag{2}$$

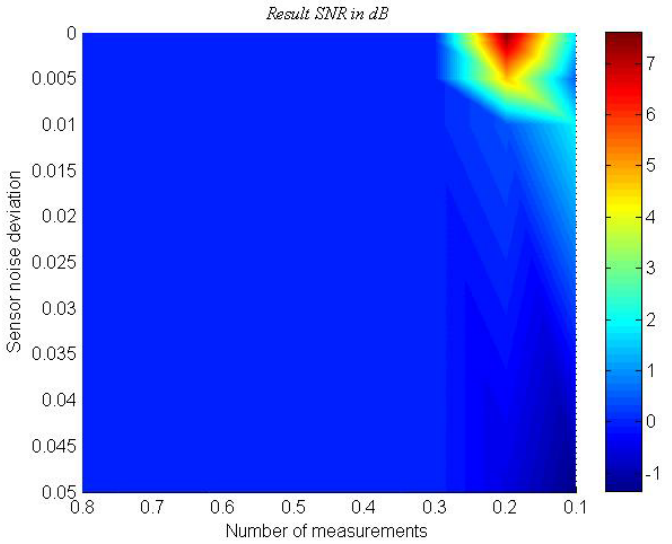


Fig. 5. Gain of an alternate projecting algorithm over L_2 minimalization. On the vertical axis the sensor noise deviation is given relative to a fixed maximal signal value in y. The horizontal axis shows the number of measurements relative to the total number of pixels. This test was performed on moderately structured images that are more close to the real measurements in tissues. This example makes obvious that for this type of application the classical CS based algorithms have exploitable advances only in a restricted region.

Here $\boldsymbol{\Psi}$ is the matrix of the special basis and a is the sparse representation of the signal $\boldsymbol{x}$. Exploiting this extra information one can minimize according to the L_0 norm, that is finding an a vector with the most zero components. Substituting (2) into (1), the problem formulated as:

$$a^* = \arg\min_a \|a\|_0 \; s.t. \; \boldsymbol{\Phi}\boldsymbol{\Psi}a = y. \qquad (3)$$

To solve the combinatorial problem above several techniques were examined. For instance the L_2 and L_1 optimization [9] and alternating projection methods based on smoothed zero norm functions [10][11].

These algorithms tolerate noise relatively good. However if the problem size is small, the measurements are noisy, and the imaged scene is less clearly structured then the problem became harder to solve. This is demonstrated on Fig. 3 and Fig. 4.

If one investigates the space determined by the noise variance, the size of the image, the M/N ratio and the entropy of the target texture it can be observed that the space where computationally more intensive methods yield considerable gain against L_2 minimization is small. This is partly visualized on Fig. 5. Here an obvious case is shown of a less structured object. This example was performed with smoothed L_0 minimalization algorithm. I have to emphasize that in typical applications of terahertz imaging the scene consists of mainly moderately structured features. The sensor SNR is approximately 40 dB at free space. However it drops rapidly either in transmissive or reflective configuration by scanning a specimen that has greater spatial extension or includes

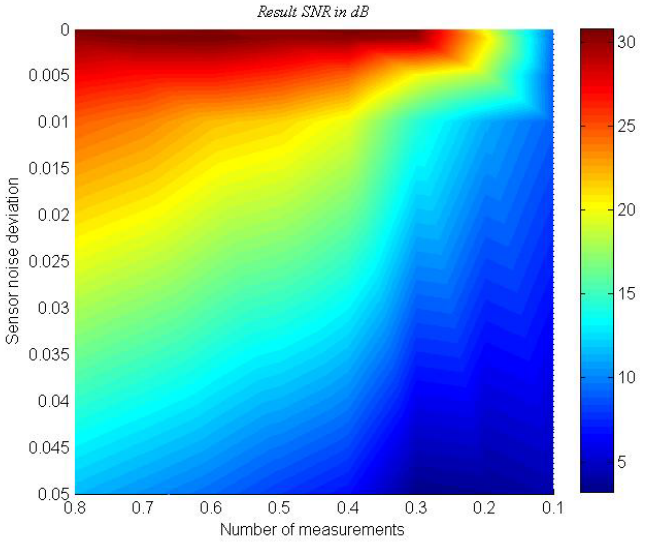


Fig. 3. It shows the usual performance of a CS algorithm on a structured target. On the vertical axis the sensor noise deviation given relative to the maximal signal value in y. The horizontal axis shows the number of measurements relative to the total number of pixels.

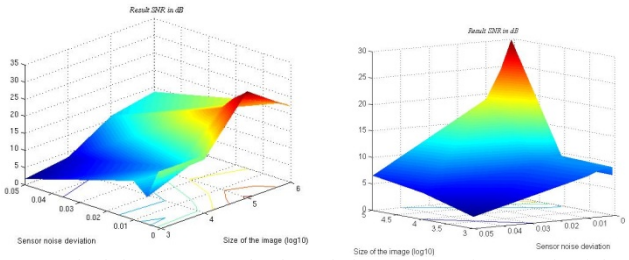


Fig. 4. On the left one can see the data of Fig. 3. as a surface. On the right the resulting SNR is depicted of the same algorithm on an image with much lower entropy indicating a richer surface texture. It can be seen even on the small images that the latter became much sharper.

dispersive layers. In addition the size of the practically solvable problems is also limited both physically and computationally. Due to the intrinsical properties of CS, I found that a sensor SNR between 31-36 dB is the lower limit of applying classical CS. Under these circumstences it provides practically no gain over L_2 minimalization.
These facts imply that in our case oversampling has to be considered as well. Although, to keep the samples count low whereas increasing the resulting SNR, compressed measurements are combined with cross validation technique.

B. Cross validation for increasing SNR

Cross validation (CV) has several applications within CS and has rigorous mathematical background. See [12] and [13] for details. However these works focus mainly on the approximation error estimation, and aiding the choice of parameters like the number of measurements or the sparsity. In this paper I investigate cross validation as a tool for increasing the SNR of the resulting images.

By usual compressed sensing imaging thorough cross validation is too expensive. Yet in our case N is relatively small, meaning that the arising computational overburden is

tractable despite of the O(L*N*log(N)) or in best case O(L*N) algorithms (here L is the number of CV runs).

By cross validation our measured data set is divided asymmetrically into two subsets. Then the optimization is performed on the bigger set. These two steps are repeated by new partitions of the original image. With these efforts an increase of 2-5.5 dB was achieved regarding the image SNR of a single L_1 optimization.

IV. NOISE MODEL

Two different type of noise is defined: image noise ($\eta_{i)}$ and sensor noise (ν_k) With these the measurements take the following form:

$$Y_k = \sum(\phi_{ki} * \eta_i) + \nu_k$$

Here η_i denotes independent Gaussian random variables at each pixel with c_k mean and σ standard deviation.

It is obvious that the image noise will scale linearly with the square root of the image size. Hence the tolerance will intrinsically improve with the increase of the problem size.

Its effect on the reconstruction can be the appearing of low frequency components in the sparse representation. In our case this kind of noise is neglectable.

The other type of noise has a much greater effect on the reconstruction. In relatively small problems it has approximately one order of magnitude stronger influence on the sensitivity. It never can be neglected because by real measurements the SNR of the sensor is always limited.

By inspecting living tissues in small scale the situation can be even worse concerning that the observed scene is only moderately structured or with other words it represents rich content. From our viewpoint these samples can be characterized by low sparsity and high entropy.

The achievable gain is always limited by the sensor SNR and the entropy of the imaged object.

V. CONCLUSION

Both the one dimensional measurements and the two dimensional vector based signal reconstruction schemes prove that it is possible to achieve significant increase in resulting image SNR by utilitzing compressed sensing based image processing.

VI. ACKNOWLEDGMENT

The work is supported by the Hungarian Scientific Research Fund - National Office for Research and Technology OTKA-NTKH CNK-77564 project. In addition the support of the grants TÁMOP-4.2.1.B-11/2/KMR-2011-0002 and TÁMOP-4.2.2/B-10/1-2010-0014 is gratefully acknowledged as well.

VII. REFERENCES

[1] R. Appleby, R.N. Anderton, “Millimeter-Wave and Submillimeter-Wave Imaging for Security and Surveillance”, Proceedings of the IEEE , vol. 95 , issue: 8 , pp. 1683 – 1690, 2007

[2] B.B. Yang, J.H. Booske, “Measurement of surface roughness effects on conductivity in the terahertz regime with a high-Q quasi optical resonator“, IEEE International Conference on Plasma Science (ICOPS), 26-30 June 2011

[3] K. Ajito, Y. Ueno, “THz Chemical Imaging for Biological Applications”, IEEE Transactions on Terahertz Science and Technology, vol. 1 , pp. 293 - 300. September 2011

[4] Y.D Taylor, R. S. Singh, D. B. Benett, P. Tewari, C. P. Kealey, N. Bajwa, M. O. Culjat, A. Stojadinovic, H. Lee, J. P. Hubschman, E. R. Brown, and W. S. Grundfest, „THz Medical Imaging: in vivo Hydration Sensing”, IEEE Transactions on Terahertz Science and Technology, Vol. 1., september 11., 2011.

[5] F. Friederich, W. von spiegel, M. Bauer, F. Meng, M.D Thomson, S. Boppel, A. Lisauskas, P. Hils, V. Kroyer, A. Keil, T. L0ffler, R. Henneberger, A. K. Huhn, G. Spickermann, P. H. Bolívar, and H. G. Roskos, „THz Active Imaging Systems With Real-Time Capabilities”, IEEE Transactions on Terahertz Science and Technology, Vol. 1., september 11., pp. 183-200.

[6] Z. Popovic and E. N. Grossman , “THz metrology and Instrumentation”, IEEE Transactions on Terahertz Science and Technology, Vol. 1., september 11., pp. 133-144.

[7] H. Sherry, J. Grzyb, Z. Yan; R. Al Hadi, A. Cathelin, A. Kaiser, U. Pfeiffer, “A 1kpixel CMOS camera chip for 25fps real-time terahertz imaging applications”, Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 19-23 Feb. 2012, pp. 252 – 254.

[8] F. Friederich, W. von spiegel, M. Bauer, F. Meng, M.D Thomson, S. Boppel, A. Lisauskas, P. Hils, V. Kroyer, A. Keil, T. L0ffler, R. Henneberger, A. K. Huhn, G. Spickermann, P. H. Bolívar, and H. G. Roskos, „THz Active Imaging Systems With Real-Time Capabilities”, IEEE Transactions on Terahertz Science and Technology, Vol. 1., september 11., pp. 183-200.

[9] D. Donoho, “Compressed sensing”, IEEE Trans. on Information Theory, Vol. 52, Issue 4, pp.: 1289-1306, April 2006.

[10] G. Hosein Mohimani, M. Babaie-Zadeh, C. Jutten, “A fast approach for overcomplete sparse decomposition based on smoothed l_0 norm”, IEEE Transactions on Signal Processing, Vol. 57, No. 1, Jan. 2009, pp. 289-301.

[11] L. Mancera, J. Portilla, “L0-norm-based Sparse Representation through Alternate Projections”, 13th International Conference on Image Processing (ICIP), October 2006.

[12] R. Ward, “Compressed Sensing With Cross Validation”, IEEE Transactions on Infromation Theory, Vol. 55., Issue 12, December 2009

[13] D.M. Malioutov, S. R. Sanghavi, A. S. Willsky, “Sequential Compressed Sensing”, IEEE Selected Topics in Signal Processing, Vol. 4, No. 2, April 2010

Cellular Particle Filter on GPU

Anna Horváth
(Supervisor: Dr. Cserey György, Dr. Karacs Kristóf)
horan3@digitus.itk.ppke.hu

***Abstract*— This paper presents the GPU implementation of the Cellular Particle Filter (CPF) algorithm. Though Particle Filters represent an attractive solution for Hidden Markov Model based problems, due to their computational requirements parallelization is inevitable for real time applications. Compared to distributed particle filters, our algorithm preserves the local connectivity of the particles, therefore it achieves the accuracy of the original filter, however with a running time less than 12 milliseconds at 16 thousand particles per state.**

Keywords- Bootstrap particle filter; paralellized resampling; locally connected representaion

I. INTRODUCTION

GPGPU grants us efficient parallel implementation of the suitable algorithms. The main challenge lies in remodeling an algorithm to fit it to the architecture. Particle filter can be considered as an extension of the Kalman filter, and therefore would be a great tool in many applications, such as image processing, robotics, stock market forecasts. The original algorithm has high running time due to the resampling step, thus it was considered unsuitable for parallelization.

Although there have been some implementations to parallel architectures [4], these cannot maintain the local connections of the particles. In [4] the authors admit, that information loss is around 25% compared to the original, locally connected algorithm.

Cellular particle filter [5] offers a solution for the local connectivity issue as well. This parallel version of the algorithm uses the structure of the CNN architecture for representation. This enables to model and utilize the connection of the particles.

However the cellular representation is not optimal for a GPU architecture. Therefore we made some further modifications to the algorithm to make it fully suitable and efficient. One of the cardinal issues is the random number generation, which is the key point of the modified algorithm.

II. THEORETICAL BACKGROUND

A. Hidden Markov Models

Hidden Markov Models (HMM) consist of a hidden stochastic process determined by Markovian dynamics (Equation 1), and its stochastic observation (Equation 2) in time space. The observation is dependent on the hidden state sequence, but we have less limitation than in case of the Kalman filter. Both sequences are non-linear, and noise (e_1 and e_2) needs not to be Gaussian, only its distribution has to be known.

$$x_{t+1} = \varphi(x_t, e_1(t+1)) \tag{1}$$

$$y_t = \psi(x_t) + e_2(t) \tag{2}$$

For more information about Hidden Markov models see [1].

B. Particle Filter

Particle filter is a tool for estimating the hidden states based on the observation. It is not an analytical calculation, but using a set of particles at each time step that follow the model dynamics. However there are many alternations, the basic particle filter algorithm is built up from three main steps.

The first step is error calculation between the current particle set, and the current observation value (Eq.3), where L is the likelihood value, for $i = 1, \ldots, N$ particles.

$$L_t^i = l(y_t - \psi(x)) \tag{3}$$

In equation (3) we use the density function of noise of the hidden process (e_1). Using the corresponding fitness value we calculate likelihood based on the density function.

Each particle is assigned a weight based on this likelihood, by normalizing over the whole particle set (Eq.4).

$$w_t^i = L_t^i \tag{4}$$

$$w_t^i = \frac{w_t^i}{\Sigma_{j=1}^N w_t^j} \tag{5}$$

Due to this normalization all of the particle weights are not only in the [0, 1] interval, but they sum up to 1.

The second main step is resampling. We choose a new particle set from our current particles, by uniform random sweepstake, using particle weights. The resampled set is used for the current estimation.

The last main step is the iteration, when we use the model, to generate the next time step's initial particle set.

More information about particle filters can be found in [2][3].

C. Cellular Particle Filter

In the resampling step for each retake we have to use the whole particle set, which is highly time consuming, and for a long time was considered not parallelizable. Cellular particle

filter offers a solution for this problem, and in contrast to other distributed particle filters [4], it maintains local connectivity, which ensures a better approximation.

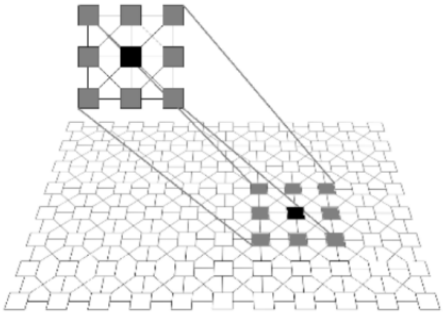


Figure 1.: Cellular architecture for particles, which enable parallelization of resampling step.

The basic modification is a change of representation. The set of particles are organized in a CNN type architecture. For each particle we perform the resampling on a neighborhood instead of the whole grid, see Figure 1. Therefore the filtering steps now can be simultaneously performed for each particle for each time step, which makes the algorithm suitable for parallel architectures. More details of the CPF can be found in [5].

III. GPU ADAPTION OF THE CELLULAR PARTICLE FILTER

A. About the architecture

We decided to implement the CPF to GPU architecture, as the devices' computational efficiency is high, the price of a device is relatively low. Also GPUs are spreading fast thanks to games industry, and development has quite fast speed, therefore computational effort is increasing by leaps and bounds. We used NVIDIA CUDA, see [6] for notations and details.

B. Main considerations of modification

Memory usage is critical issue on GPU architecture. To achieve a high throughput, on-chip memory (shared memory) should be used. The main computational tasks are performed block-wise, in the shared memory of the blocks, and synchronization is performed in each time step through global (off-chip) memory between the blocks.

Although the CPF presents a two dimensional representation, the shared memory throughput can be optimized for a one dimensional array type representation. The local connectivity is preserved by choosing each shared memory array size higher than the thread number exactly with the size of the required neighborhood. Each thread with index i can obtain its k neighbors at indexes $i-1, i-2, \ldots, i-k$.

C. Random numbers for resampling

Random number generation on CPU hardly arise any problems thanks to the high number of libraries which we simply include and are ready to use. On GPU this issue is one order harder.

Due to memory and communication optimization it is senseless to transfer random numbers from CPU. The random values should be generated on the device to avoid memory transfer between the main memory of the system and the global memory – which would add huge delay. The particle filter is sensitive on the resampling step, therefore if the random number distribution is not appropriate, the result will be biased.

NVIDIA SDK provides an implementation of Mersenne twister [7], which on the first sight would be a solution. We found that the generated distribution is inappropriate for a small set of numbers (hundreds or thousands). The Mersenne Twister's output is admissible for around 2 million numbers and above. As mentioned before, the main operations are performed in shared memory, and shared memory size is usually around a few hundreds. This means, that the original NVIDA SDK Mersenne Twister is not feasible for our implementation. Figure 2 shows the histogram of the MT generated numbers (gray line) compared to the histogram of a same number of random values from Matlab *rand()* (black line).

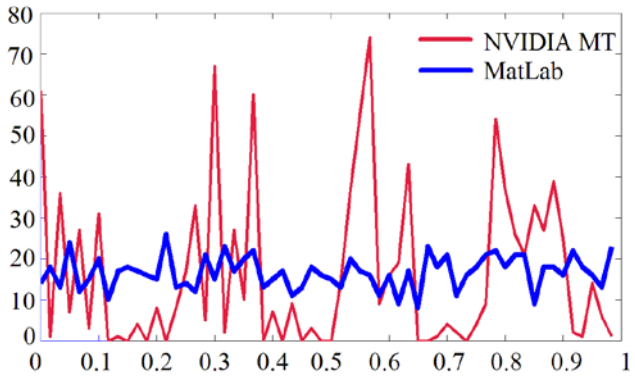


Figure 2.: Histogram of 1000 random numbers with 60 bins. Both Matlab and Mersenne Twister numbers were generated as uniform distribution.

In our first approach we implemented a Linear Feedback Shift Register Pseudo Random Bit Sequence generator. In every iteration a binary value is generated, by a single shift and XOR of the shift register values. Then we write the generated bit to the first position of the register. To avoid periodicity in the pseudorandom sequence a 24 bit shift register was implemented. We analyzed the auto- and crosscorrelation of the generated numbers for different threads, and compared to MATLAB generated values. Also histogram was checked, and all criteria were fulfilled to use the LFSR PRBS generator in our implementation.

However, we decided to review Mersenne Twister as it is a more sophisticated and commonly accepted method. To achieve an adequate distribution for the resampling, we made the modifications as listed below:

- The SDK implementation defines the degree of recursion 19, which was changed to the originally defined value: 397 [8].
- The SDK implementation defines the middle term 9, which was changed to the originally defined value: 624 [8].
- In the tempering transformation we used the originally defined masks, with hexadecimal values in the bitwise operations.
- we changed shift value *u* to 11 from the SDK value 12.
- for each thread the first element of the state array is calculated with a thread and current time based seed value
- the final value is calculated with initializing on the first element of the bit vector for each thread.

With the above modifications, we achieved an acceptable uniform distribution from the Mersenne Twister, which is illustrated on Figure 3.

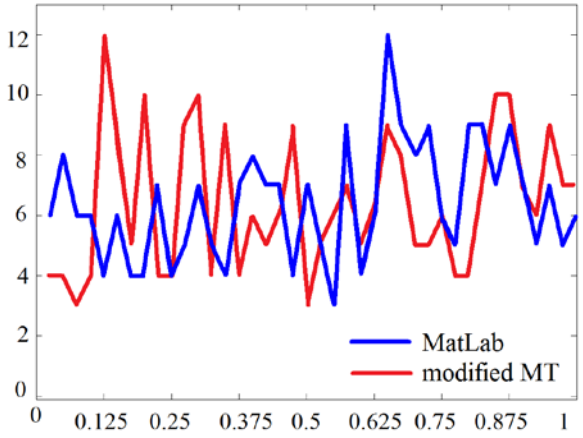


Figure 3.: 1024 random number generated with modified implementation of Mersenne Twister compared to Matlab, on 40 bins.

D. Implementation of CPF

In the following we specify some implementation details. We assume to allocate global memory arrays for the observation sequence (Y), for the state estimation (X), and for the particles ($x_{particles}$). The number of particles is denoted by N, the size of the neighbourhood by r. The number of threads in each blocks were set constant 256. All arrays are one dimensional.

In each time step we copy the particle values from the global memory to the shared memory by overlapping spilt. We load 256 values respectively to each shared memory, to the particle's array (x_{shared}), but sparin the first r elements of it. These positions are filled with the r neighbours in the global memory of the first element in x_{shared}. For the very first element, we use a circular approach, by taking the values from the end of the global memory array. There are three kernel calls for each estiated values to provide full synchronization.

The main kernel performs the following operations in each t time step, where global and local thread IDs are defined as follows:
i_{gl} = *blockDim.x*blockSize.x + threadId.x,* i_{loc} = *threadId.x.*

1. Initialization
   - $x_{shared}[i_{loc}+r] \leftarrow x_{particles}[i_{gl}]$
   - if $i_{loc} < r$ then $x_{shared}[i_{loc}] \leftarrow x_{particles}[i_{gl} - r + i_{loc}]$
   - if $i_{gl} < r$ then $x_{shared}[i_{loc}] \leftarrow x_{particles}[N - r + i_{loc}]$
2. Error calculation
   - $L_{shared} \leftarrow l\,(Y[t] - x_{shared}[i_{loc} + r])$
   - if $i_{loc} < r$ then $L_{shared}[i_{loc}] \leftarrow l\,(Y[t] - x_{shared}[i_{loc}])$
3. Resampling
   - if iloc == 0 refresh seed value for the block
   - fill U_{shared} with uniform random numbers
   - $w_{shared} \leftarrow L_{shared}[i_{loc}] + \ldots + L_{shared}[i_{loc} + r]$
   - select a particle from the neighbourhood depending on $U_{shared}[i_{loc}]$.
4. Iteration on particles
   - fill U_{shared} with new uniform random numbers
   - apply Box-Muller transform on pairs of uniform samples (n_t, stored in thread level)
   - $x_{shared}[i_{loc}] \leftarrow \varphi(x_{shared}[i_{loc}], n_t)$
   - $x_{particles}[i_{gl}] \leftarrow x_{shared}[i_{loc}]$

The estimation is performed by two kernel calls. The first kernel calculates the sum for each shared memory to a global memory array. The second kernel takes the average value of these sums with respect to the number of particles.

The following optimizations has been applied to the initial version, and we achieved a three-times speedup:

- random number generation, resampling, average calculation, Gaussian transform for the iteration: these operations are performed only on the relevant part of the shared memory to spare time;
- if statements have benn converted to inline;
- decreased the number of shared memory arrays: some are used twice (in cases where this poses no threats).

IV. MEASUREMENTS

During the measurements we used the following benchmark model (Equations 6,7), which is widely used due to its high complexity:

$$x_{t+1} = \frac{x_t}{2} + \frac{25x_t}{1+x_t^2} + 8\cos(1.2t) + n_t \qquad (6)$$

$$y_t = \frac{x_t^2}{20} + u_t \qquad (7)$$

This model is nonlinear, and the *n* and *u* noises are IID Gaussian sequences, n_t~*N(0, 10)* and u_t~ *N(0, 1)*.

Measurements were done on a PC with Intel i5 4 CPU with 4 GB system memory running Ubuntu Linux 11.04. We used an NVIDIA GeForce GTX 550 Ti GPU with 1 GB GDDR memory with CUDA toolkit 4.1. Running times were measured with the official profiler provided by the toolkit. The quality of estimation was measured by the RMSE of 1000 estimation for each particle number – neighborhood size pair of 100 time step long HMMs.

V. RESULTS

Table 1 shows some measurement data of the kernel running times of the reviewed implementation. *Table 2* shows some measurement data from the MatLab simulation. Comparison shows, that the speedup is impressive.

Reviewed CPF on GPU running times			
Configuration	*Main kernel(ms)*	*Estimation kernels (µs)*	*RMSE*
512 particles 256 neighbours	0.1352	6.62	50.51
1024 particles 256 neighbours	0.1369	6.73	50.00
4096 particles 256 neighbours	0.3001	7.82	49.70
8192 particles 256 neighbours	0.5789	12.04	49.68
16384 particles 256 neighbours	1.1387	18.33	49.71

Table 1.: Measurements of the reviewed implementation

MatLab simulation of CPF		
Configuration	*Time(s)*	*RMSE*
64 particles 25 neighbours	1.03	69.13
128 particles 25 neighbours	2.02	65.49
256 particles 25 neighbours	4.05	64.20
1024 particles 25 neighbours	16.16	63.14
4096 particles 25 neighbours	64.33	62.78

Table 2.: Measurements of the MatLab simulation

Beside running time root mean square error is an appropriate tool to measure the goodness of the estimated sequence compared to the hidden states. Based on our measurements we can say, that RMSE of the GPU implementation is satisfying, compared to the MatLab simulation results, and to [5].

VI. CONCLUSION AND FURTHER PLANS

The implemented algorithm based on the modification of Cellular Particle Filter to suit GPU architecture solves the problem of parallelization of the computationally expensive steps of the algorithm on a locally connected array. The measurements show that a system with a large number of particles can be processed in real time.

As further work, we would like to examine if a single kernel call could be implemented to spare kernel call overheads. Also, from the measurement data it can be seen that RMSE does not improve significantly. The reason is the type of the solved model; therefore we would like to implement more complex (e.g. 3D) models, to emphasize the effect of high particle numbers' on the precesion of the estimation.

ACKNOWLEDGMENT

NTP-OTKA for the financial support, and NVIDIA Corporation for the hardware support are gratefully acknowledged. I would like to thank Gábor Tornai, and András Horváth their help and suggestions.

REFERENCES

[1] Y. Ephraim, N. Merhav, Hidden markov processes, Information Theory, IEEE Transactions on 48 (2002) 1518-1569.

[2] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, Signal Processing, IEEE Transactions on 50 (2002) 174 – 188.

[3] A. Doucet, A. M. Johansen, A tutorial on particle filtering and smoothing: fifteen years later, Oxford Handbook of Nonlinear Filtering (2008) 4 – 6.

[4] M. Bolic, P. M. Djuric, és S. Hong, Resampling algorithms and architectures for distributed particle filters, Signal Processing, IEEE Transactions on, vol. 53, sz. 7, o. 2442–2450, 2005.

[5] A. Horvath, M. Rasonyi, Fast computation of particle filters on processor arrays, 12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA) (2010) 409 – 428.

[6] NVIDIA CUDA programming guide, http://developer.nvidia.com/object/cuda.html, 2010.

[7] Victor Podlozhnyuk, Parallel Mersenne Twister, June, 2007, NVIDIA http://developer.download.nvidia.com/compute/cuda/2_2/sdk/website/projects/MersenneTwister/doc/MersenneTwister.pdf

[8] Makoto Matsumoto, Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator, Keio University, and the Max-Planck Institute für Mathematik, Takuji Nishimura, Keio University

Component-Based Object Detection with Structural Dependencies

Mihály Radványi
(Supervisor: Dr. Kristóf Karacs)
radmige@itk.ppke.hu

***Abstract*— In this paper I present an algorithmic framework that takes advantage of a component-based object detection model, using specific spatial relations determined on canonical viewpoints. This method let us to investigate object detection as a combination of high and low level description of target objects.**

***Keywords*-canonical viewpoint, component-based, object detection**

I. INTRODUCTION

The goal of object detection is to determine whether a specific target object is present on the input image flow or depending on the task even the class of the target object should be identified.

Objects can be viewed from various distances and angles, each of which providing information of the target item. Since image acquisition is a projection form the real 3-dimensional world into a 2-dimensional image plane, measurable object properties are highly dependent on the actual viewpoint of the target object. The one(s) delivering the most information of a given object, or alternatively the ones that people are more likely to recognize/name the target of are called canonical viewpoints (CPOV or POV in this paper). Most object detection algorithms are highly dependent on specific/canonical viewpoints of the target object.

Object that are to be recognized are usually decomposed into smaller components, where components are easier to detect even in case of partial occlusion. This approach makes the detection less sensitive on processing losses or environmental difficulties. In my work I defined spatial relations between the components to describe target objects in specific (canonical) viewpoints and – similarly to text recognition where a corpus of valid character combinations (words) is given in order to increase the detection rate – relevant 2-dimensional models are defined for object detection tasks. During detection, candidate features are validated through model matching, and classified via weak learners. I designed the detection algorithm to be compatible with Cellular Neural/Nonlinear Network – Universal Machine (CNN-UM) and the underlying Cellular Wave Computing principle. [1]

Object detection examples discussed in Section 4. are related to the Bionic Eyeglass project, that aims to help blind and visually impaired people in everyday's navigation and orientation by developing a mobile navigational device with diverse functionalities. [2]

The paper is structured as follows: Section 2 introduces component-based object detection, Section 3 shows the algorithmic framework used for the experiments and Section 4 brings two sample tasks. Section 5 concludes the paper.

II. COMPONENT-BASED OBJECT DETECTION

There exist two main approaches in object detection: 1) one considers the target object as a whole to be detected – called global feature based model, 2) while the other decompose the target object into smaller, specific and distinctive regions (components) that are easier to detect. Fig. 1. illustrates the difference between considering the US dollar bill as a global object or defining unique components which can be detected even in case of occlusion.

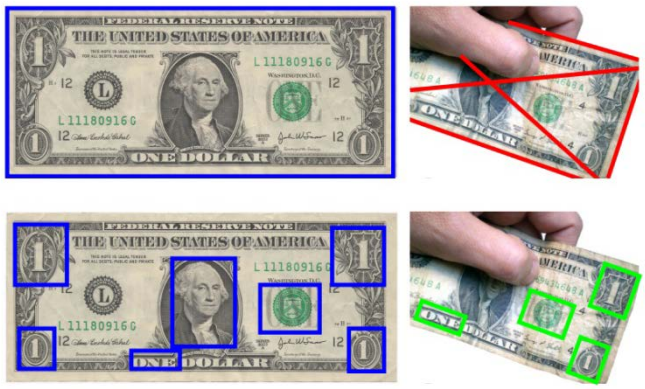


Figure 1. First row illustrates global feature based model and second row shows component-based model. Global feature based model uses the overall target object as a whole for recognition, while component-based approach divides the object into small features/components. In case of occlusion a set of components can still be detected, while global object is hard to detect.

A. Component selection

In order to obtain the best detection/classification results, we have to clarify whether we are interested in detecting the presence of a given type of object – cars, banknotes, mugs, etc. – or if we need to be more specific by distinguishing classes – van, truck, sport car – within the object types. Components can be selected to be more class-specific or to be similar across different classes, depending on the actual task and needs. Components that are similar across the classes help detecting a specific type of object - such as banknotes -, whereas class-specific ones serve the base for classification of objects - for instance determining the denominations of banknotes. Since

recognition of an object does not necessarily require the detection of all the components, component-based approach has the advantage of being less sensitive to partial occlusions or processing losses.

Suppose that each – or most – component has a unique detector algorithm. Relevant components can be grouped into sets based on the Point of View (POV) they appear on, providing mainly two advantages: 1) in case there is acceptable/small overlap between the component sets, by detecting only a few components the actual POV can be identified narrowing the number of steps needed for further processing; and vice verse 2) if the actual POV is known, detection steps of components related to other POVs can be ignored. The same method can be applied on various classifiers related to the different components as well. This scheme can interpret as a simple inhibitory/excitatory feed forward network, as it can be seen on Fig. 2.

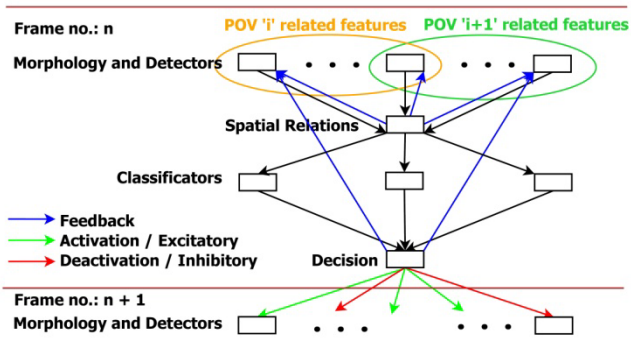


Figure 2. Schematic view of the algorithmic flow. Features are grouped into sets regarding to the Point Of View they appear on. Spatial relations between detected components are analized in order to validate the actual set of components and to determin the POV. Once the actual POV is determined the corresponding detectors are activated for the processing of the next frame and others are ignored.

B. *Feature description*

Both global and component-based models have several methods to describe the features. There exists line/edge based descriptions [3][4], interesting point and blob based descriptions as well. Most of these approaches generate feature vectors for each point or region of interest for training as well as for matching. Harris corners [5][6] are commonly used for interesting point detection and the method proposed by Lowe - Scale Invariant Feature Transform (SIFT) – is often used for interesting point description [7]. SIFT became common in object recognition tasks until a much faster alternative of SIFT was introduced by Bay et al. called Speeded-Up Robust Features (SURF) [8]. SURF generates scale and rotation invariant interesting point descriptions, with high repeatability, distinctiveness and robustness. SURF features are used for component based object detection in [9].

Shape descriptors characterize regions obtained from segmentation and morphological extraction based on simple properties such as area, perimeter, eccentricity or orientation of the shape. Using these properties candidate features of different component classes can be selected. The set of component classes that are to be detected can be tightened by information on actual POV given by previous processes.

C. *Spatial relations*

Similarly to the human face where eyes, eyebrows, nose, mouth and other parts has an anatomically predefined pattern of appearance, spatial relations between components of common objects can be determined. Spatial relations can be defined pair-, triple, or arbitrary n-wise, ensuring the system to be less sensitive to partial occlusions and detection errors/losses. A possible set of topological descriptions is:

- Normalized distances of center points
- Angle between major axis of components, or between vectors connecting different components
- Shape defined by center points – or other specific points – of components
- Ratio of areas (perimeters) of components.

Spatial relations are used before final classification and decision in order to determine whether the extracted component set forms a valid/possible configuration or further processing is needed. In case the extracted features do not form a valid configuration, detectors are applied again with modified parameters, otherwise classification of components and final decision is made.

III. ALGORITHMIC FRAMEWORK

In the proposed framework image flows are processed frame by frame extracting as many components as possible. Once the actual POV is identified, corresponding detectors are activated/exhibited and others are inhibited. Detected components are analyzed, and the ones fitting predefined requirements of their shape and size are selected as candidate features and primarily classified (presclassification). Besides simple classification spatial relations between candidate features are investigated taking into account the topological arrangement of objects as well. In case the requirements of spatial relations do not meet, the confidence of selecting false candidate features increase. However it does not mean that all the candidate features were selected incorrectly. Features that can be verified by their spatial relations are kept, and false candidates are ignored. In order to be able to select false candidates, confidence intervals of feature descriptors and detectors should be known. Once weakest features are ignored, a few classes might remain without candidates. Missing candidates can be detected based on the position of verified features. Verified (true) candidates can define Region Of Interest (ROI) for those missing features and ROI-s are given to a next iteration of the algorithm in order to detect missing features on the same frame if possible. This process can be repeated until processing time allows us to stay within real time performance. Based on properly detected and verified features of frame i, the actual canonical POV can be identified and this information can be feed forward to frame i+1. Once the POV is known, the corresponding set of feature classes is selected/excited and outlier classes inhibited (see Fig.2. for

details). This method allows us to execute only a minimal set of detectors/classifiers corresponding to the determined POV.

After processing a given number of frames, primary votes - voting vectors – are formed and evaluated in order to make a final decision on the appearance or recognition of the given target object.

IV. SAMPLE TASKS / EXAMPLES

In order to demonstrate the usability of the previously proposed method two sample tasks are presented. Both examples are parts of the Bionic Eyeglass Project as it was mentioned before. Table 1. summarizes the different definitions applied in banknote recognition and crosswalk detection tasks using the proposed framework.

TABLE I. OVERVIEW OF SAMPLE TASKS

	banknote recognition	***crosswalk detection***
Global object	banknote	crosswalk
POVs	frontside, backside	front and side view
Components	portrait, crest, number, tactile signs, backside graphics	stripes
Spatial relations	ratio of areas, distance of center points, angle between major axis	orientation of major axis, center points fitting the same line
Output classes	denominations	binary or confidence values

A. Banknote recognition

In banknote recognition task the global object to be detected is defined as the whole banknote itself. Considering Hungarian banknotes, there are six different denominations (classes) and two canonical Point Of Views, such as front side and back side of the banknote. Each POV has unique features as well as common ones, and similarly, each denomination class have unique versions of components as well as similar ones. Considering actual POVs, components are divided into three categories: 1) front side only elements: portrait, crest of Hungary, logo of the National Bank, 2) back side only objects: tactile marks, background image, 3) POV independent elements: numbers. Across the different classes of denominations crest is the only one that appear on each denomination unchanged, the logo of the National Bank might vary depending on the edition date, and the rest of the features are denomination/class dependents (Fig. 3.).

A principal feature detection algorithm extracts portrait or backside image objects respectively to the actual POV, and a number detector returns objects that are more likely to be numbers. Structural dependencies and valid configurations are defined based on the different POVs of the Hungarian banknotes. Experimental results showed that the angle enclosed by the orientation of the major axis of principal features

Figure 3.Different components of Hungarian banknotes are shown. Dashed bounding box indicates static objects and continuous the dynamic ones. Static objects appear on each denominations, while dynamic ones change regarding to the actual denomination. Objects within green frames are capable for banknote recognizing tasks, while reds are not due to detection difficulties.

(portrait or backside graphics) and number features or the normalized distance between the centroids of the two features can be detected with high confidence, thus can be used for candidate feature validation and POV determination. The process works as follows: candidate are extracted for principal components and numbers, than properties such as orientation, normalized distance of features are determined. In case the values lie within a predefined confidence interval specified for front side or similarly for backside, components are validated and the POV can be determined (Fig. 4.). Confidence intervals are determined based on results obtained on learning set. If the

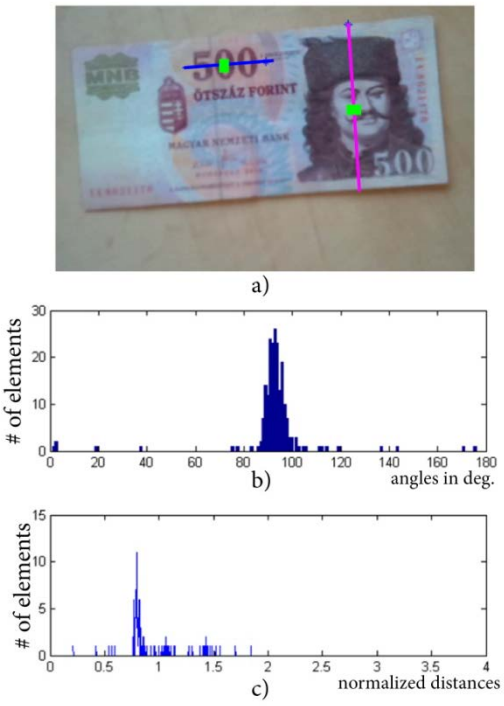


Figure 4.Spatial dependecies applied on a 500HUF banknote. a) shows the major axis and the center point of number (MAoN) and portrait (MAoP) components; b) shows the distribution of angles between MAoN and MAoP over a 300 element training set of 500 HUF images; c) shows the distribution of normalized distances of centroids on the same training set.

extracted values are outliers, candidate features cannot be validated (nor POV), requiring further processing.

Suppose that the candidate features are validated by their

spatial relations, the following step is to classify the components one by one using Projected Principal Edge Distribution [3] or applying a number recognition process on number candidates. The classification results (votes on denominations) are weighted based on the confidence values of the different classifiers and stored in order to accumulate results over a few frames and to make deliberate decision. Results are further detailed in [10].

B. Crosswalk detection

In case of zebra type crosswalks the global object to be detected is a set of pavement marks or the bounding box of the crosswalk itself. Considering pedestrian users, crosswalks have one canonical POV, containing parallel pavement marks orienting quasi-horizontal. Furthermore we do not distinguish different classes of crosswalks, but since crosswalks can easily be occluded partially by traffic, we decided to return confidence values of detection instead of strict decisions. In case a binary decision is needed the confidence value can easily be cut through a well defined threshold. Components that are to be detected are the stripes themselves defined as the lighter holes within road surface. Spatial relations for candidate feature validation are determined as objects with: similar orientation, quasi-horizontal - due to perspective distortion a minimal variance is accepted - shrinking size and finally, centroids lie on the same line. Once a minimal set of candidate features are extracted and verified by relative positions, a confidence value of the detection is calculated. The confidence value is defined by the number of components, the aspect ratio and area of the components. Once stripes of zebra type crosswalks are detected, a final decision is made by considering the estimated length and density of crosswalk stripes.

Fig. 5.shows extracted crosswalk stripes projected back on the original input images. Output masks of the crosswalk detection algorithm indicating orientations and centroids of the stripes and the histogram of the orientations of stripes in a given frame are shown on Fig. 6. One can quickly realize, that the originally parallel lines appear to be transformed on the image flow due to perspective distortion. This explains the need of a tolerance interval to be determined for component validation.

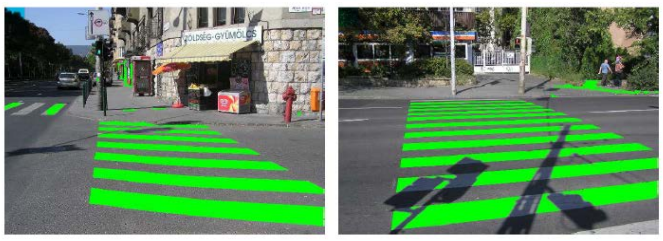


Figure 5. Images show crosswalk patterns detected by the Bionic Eyeglass.

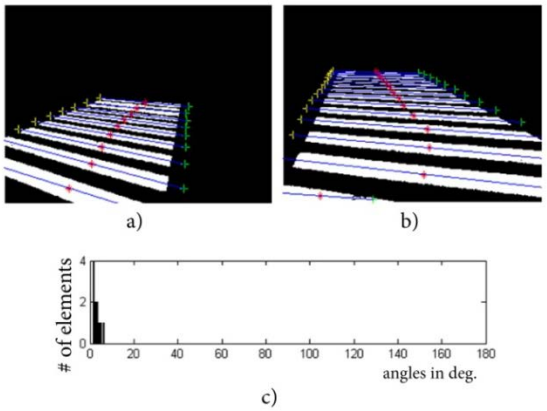


Figure 6. Detected crosswalk stripes are shown on a) and b) indicating the center points by red crosses and the major axis of components by blue lines. c) shows the distribution of angles of major axis of crosswalk stripes.

CONCLUSION

In this paper I presented an algorithmic framework that takes advantage of a component-based detection scheme. Further on, it uses spatial relations of components in order to identify valid configuration of extracted components generating excitatory or inhibitory feed-forward connections for classification and detection modules of other components. Final decision is made only in case the extracted features form a valid configuration, speeding up the overall detection/recognition process.

REFERENCES

[1] T. Roska and L. O. Chua, “The CNN universal machine: an analogic array computer,” IEEE Trans. Circuits Syst. II, vol. 40, pp. 163–173,Mar. 1993.

[2] K. Karacs, M. Radványi, "A Prototype for the Bionic Eyeglass”12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), 2010

[3] H. Zhu, P. Zhao, T. Shibata,”Directional-edge-based object tracking employing on-line learning and regeneration of multiple candidate locations” in proc. of IEEE International Symposium on Circuits and Systems (ISCAS) 2010: 2630-2633

[4] E. Tekin, J. Coughlan, H. Shen, “Real-Time Detection and Reading of LED/LCD Displays for Visually Impaired Persons” in proc of IEEE Workshop on Applications of Computer Vision 2011 (WACV11), Kona, Hawaii 2011.

[5] C. Harris and M. Stephens, “A combined corner and edge detector”, The Fourth Alvey vision Conference, Manchester, UK, 1988.

[6] C. Harris, “Geometry from Visual Motion”, Active Vision, MIT Press,1993.

[7] D. Lowe, "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157. doi:10.1109/ICCV.1999.790410.

[8] H. Bay, A. Ess, T. Tuytelaars, L. Gool "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008

[9] F. M. Hasanuzzaman, X. Yang, Y. Tian, “Robust and Effective Component-based Banknote Recognition by SURF Features”in proc. of Wireless and Optical Communications Conference (WOCC), 2011, Newark, New Jersey, USA

[10] Z. Solymar, A. Stubendek, M. Radvanyi, K. Karacs, "Banknote Recognition for Visually Impaired" in Proc. of the European Conference on Circuit Theory and Design (ECCTD’11), Linköping, Sweden, 2011.

A Dynamic MRF Model for Foreground Detection on Range Data Sequences of a Multi-Beam Lidar

Dömötör Molnár
(Supervisor: Tamás Szirányi)
molnar.domotor@itk.ppke.hu

***Abstract*—In this report, a probabilistic approach will be introduced for foreground segmentation in 360°-view-angle range data sequences, recorded by a rotating multi-beam Lidar sensor, which monitors the scene from a fixed position. To ensure real-time operation, we project the irregular point cloud obtained by the Lidar, to a cylinder surface yielding a depth image on a regular lattice, and perform the segmentation in the 2D image domain. Spurious effects resulted by quantification error of the discretized view angle, non-linear position corrections of sensor calibration, and background flickering, in particularly due to motion of vegetation, are significantly decreased by a dynamic MRF model, which describes the background and foreground classes by both spatial and temporal features. Evaluation is performed on real Lidar sequences concerning both video surveillance and traffic monitoring scenarios.**

***Keywords*-rotating multi-beam Lidar, MRF, motion segmentation**

I. Introduction

Foreground detection and segmentation are a key issues in automatic visual surveillance. Foreground areas usually contain the regions of interest, moreover, an accurate object-silhouette mask can directly provide useful information for, among others, people or vehicle detection, tracking or activity analysis.

Range image sequences offer significant advantages versus conventional video flows for scene segmentation, since geometrical information is directly available [1], which can provide more reliable features than intensity, color or texture values [2], [3]. Using Time-of-Light (ToF) cameras [1] or scanning Lidar sensors [4] enable recording range images independently of the outside illumination conditions and we can also avoid artifacts of stereo vision techniques.

Rotating multi-beam Lidar systems (RMB-Lidar) provide a 360° FoV of the scene, with a vertical resolution equal to the number of the sensors, while the horizontal angle resolution depends on the speed of rotation. For efficient data processing, the 3-D RMB-Lidar points are often projected onto a cylinder shaped range image [4], [5]. However, this mapping is usually ambiguous: On one hand, several laser beams with slight orientation differences are assigned to the same pixel, although they may return from different surfaces. As a consequence, a given pixel of the range image may represent different background objects at the consecutive time steps. This ambiguity can be moderately handled by applying multi-modal distributions in each pixel for the observed background-range values [4], but the errors quickly aggregate in case of dense background motion, which can be caused e.g. by moving vegetation.

On the other hand, due to physical considerations, the raw data of distance, pitch and angle provided by the RMB-Lidar sensor must undergo a strongly non-linear calibration step to obtain the Euclidean point coordinates [6], therefore, the density of the points mapped to the regular lattice of the cylinder surface may be inhomogeneous. To avoid the above artifacts of background modeling, [5] has directly extracted the foreground objects from the range image by mean-shift segmentation and blob detection. However, we have experienced that if the scene has simultaneously several moving and static objects in a wide distance range, the moving pedestrians are often merged into the same blob with neighboring scene elements.

In this report, we propose a hybrid approach for dense foreground-background point labeling in a point cloud obtained by a RMB-Lidar system, which monitors the scene from a fixed position. This method solves the computationally critical spatial filtering steps in the 2D range image domain by an MRF model, however, ambiguities of discretization are handled by joint consideration of the true 3D positions and the 2D labels. Using a spatial foreground model, one can significantly decrease the spurious effects of irrelevant background motion, which is mainly caused by moving tree crowns. We provide evaluation versus three reference methods using our new 3D point cloud Ground Truth (GT) annotation tool.

II. Problem formulation and data mapping

Assume that the RMB-Lidar system contains R vertically aligned sensors, and rotates around a fixed axis with a possibly varying speed. The output of the Lidar within a time frame t is a *point cloud* of $l^t = R \cdot c^t$ points: $\mathcal{L}^t = \{p_1^t, \ldots, p_{l^t}^t\}$. Here c^t is the number of point *columns* obtained at t, where a given column contains R concurrent measurements of the R sensors, thus c^t depends on the rotation speed. Each point, $p \in \mathcal{L}^t$, is associated to sensor distance $d(p) \in [0, D_{\max}]$, pitch index $\hat{\vartheta}(p) \in \{1, \ldots, R\}$ and yaw angle $\varphi(p) \in [0, 360^\circ]$ parameters. $d(p)$ and $\hat{\vartheta}(p)$ are directly obtained from the Lidar's data flow, by taking the measured distance and sensor index values corresponding to p. Yaw angle $\varphi(p)$ is calculated from the Euclidean coordinates of p projected to the ground plane, since the R sensors have different horizontal view angles, and the angle correction of calibration may also be significant [6].

The goal of the proposed method is at a given time frame t to assign each point $p \in \mathcal{L}^t$ to a label $\omega(p) \in \{\mathrm{fg}, \mathrm{bg}\}$ corresponding to the moving object (i.e. foreground, fg) or background classes (bg), respectively.

For efficient data manipulation, we also introduce a range image mapping of the obtained 3D data. We project the point cloud to a cylinder, whose central basis point is the ground

position of the RMB-Lidar and the axis is prependicular to the ground plane. Note that slightly differently from [5], this mapping is also efficiently suited to configurations, where the Lidar axis is tilted do increase the vertical Field of View. Then we stretch a $S_H \times S_W$ sized 2D pixel lattice S on the cylinder surface, whose height S_H is equal to the R sensor number, and the width S_W determines the fineness of discretization of the yaw angle. Let us denote by s a given pixel of S, with $[y_s, x_s]$ coordinates.

Finally, we define the $\mathcal{P} : \mathcal{L}^t \to S$ point mapping operator, so that y_s is equal to the pitch index of the point and x_s is set by dividing the $[0, 360°]$ domain of the yaw angle into S_W bins:

$$s \stackrel{\text{def}}{=} \mathcal{P}(p) \text{ iff } y_s = \hat{\vartheta}(p),\ x_s = \text{round}\left(\varphi(p) \cdot \frac{S_W}{360°}\right) \quad (1)$$

III. Background model

The background modeling step assigns a fitness term $f_{\text{bg}}(p)$ to each $p \in \mathcal{L}^t$ point of the cloud, which evaluates the hypothesis that p belongs to the background. The process starts with a cylinder mapping of the points based on (1), where we use a $R \times S_W^{\text{bg}}$ pixel lattice S^{bg} (R is the sensor number). Similarly to [4], for each s cell of S^{bg}, we maintain a Mixture of Gaussians (MoG) approximation of the $d(p)$ distance histogram of p points being projected to s. Following the approach of [7], we use a fixed K number of components (here $K = 5$) with weight w_s^i, mean μ_s^i and standard deviation σ_s^i parameters, $i = 1 \dots K$. Then we sort the weights in decreasing order, and determine the minimal k_s integer which satisfies $\sum_{i=1}^{k_s} w_s^i > T_{\text{bg}}$(we used here $T_{\text{bg}} = 0.89$). We consider the components with the k_s largest weights as the background components. Thereafter, denoting by $\eta()$ a Gaussian density function, and by $\mathcal{P}^{\text{bg}}$ the projection transform onto S^{bg}, the $f_{\text{bg}}(p)$ background evidence term is obtained as:

$$f_{\text{bg}}(p) = \sum_{i=1}^{k_s} w_s^i \cdot \eta\left(d(p), \mu_s^i, \sigma_s^i\right),\ \text{where } s = \mathcal{P}^{\text{bg}}(p). \quad (2)$$

The Gaussian mixture parameters are set and updated based on [7], while we used $S_W^{\text{bg}} = 2000$ angle resolution, which provided the most efficient detection rates in our experiments. By thresholding $f_{\text{bg}}(p)$, we can get a dense foreground/background labeling of the point cloud [4], [7] (referred later as *Basic MoG* method), but as shown in Fig. 2(a),(c), this classification is notably noisy in scenarios recorded in large outdoor scenes.

IV. DMRF approach on foreground segmentation

In this section, we propose a Dynamic Markov Random Field (DMRF) model to obtain smooth, noiseless and observation consistent segmentation of the point cloud sequence. Since MRF optimization is computationally intensive [8], we define the DMRF model in the range image space, and 2D image segmentation is followed by a point classification step to handle ambiguities of the mapping. As defined by (1) in Sec. II, we use a $\mathcal{P}$ cylinder projection transform to obtain the range image, with a $S_W = \min(\hat{c}, S_W^{\text{bg}}/2)$ grid with, where $\hat{c}$ denotes the expected number of point columns of the point sequence in a time frame. By assuming that the rotation speed is slightly fluctuating, this selected resolution provides a dense range image. Let us denote by $P_s \subset \mathcal{L}^t$ the set of points projected to pixel s. For a given direction, foreground points are expected being closer to the sensor than the estimated mean background range value. Thus, for each pixel s we select the closest projected point $p_s^t = \text{argmin}_{p \in P_s} d(p)$, and assign to pixel s of the range image the $d_s^t = d(p_s^t)$ distance value. For pixels with undefined range values ($P_s = \emptyset$), we interpolate the d_s^t distance from the neighborhood. For spatial filtering, we use an eight-neighborhood system in S, and denote by $N_s \subset S$ the neighbors of pixel s.

Next, we assign to each $s \in S$ foreground and background energy (i.e. negative fitness) terms, which describe the class memberships based on the observed $d(s)$ values. The background energies are directly derived from the parametric MoG probabilities using (2):

$$\varepsilon_{\text{bg}}^t(s) = -\log\left(f_{\text{bg}}(p_s^t)\right).$$

For description of the foreground, using a constant ε_{fg} could be a straightforward choice [2] (we call this approach *uniMRF*), but this uniform model results in several false alarms due to background motion and quantitization artifacts. Instead of temporal statistics, we use spatial distance similarity information to overcome this problem by using the following assumption: whenever s is a foreground pixel, we should find foreground pixels with similar range values in the neighborhood. For this reason, we use a non-parametric kernel density model for the foreground class:

$$\varepsilon_{\text{fg}}^t(s) = \sum_{r \in N_s} \zeta(\varepsilon_{\text{bg}}^t(r), \tau_{\text{fg}}, m_\star) \cdot k\left(\frac{d_s^t - d_r^t}{h}\right),$$

where h is the kernel bandwidth and $\zeta : \mathbb{R} \to [0, 1]$ is a sigmoid function:

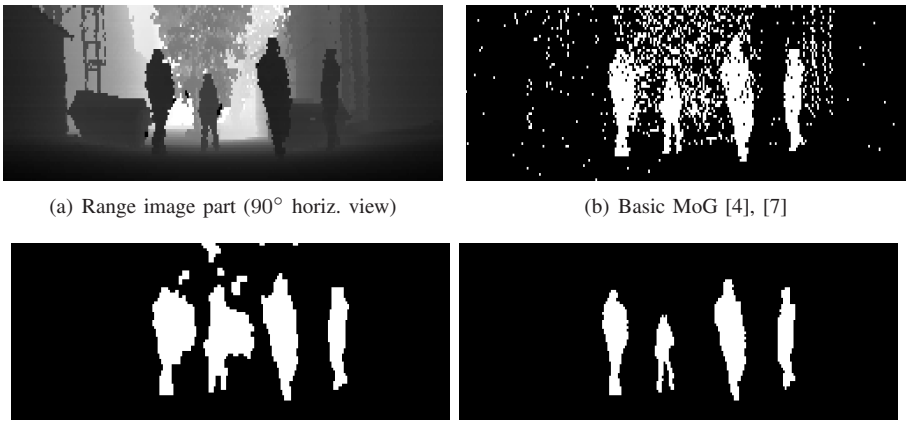
$$\zeta(x, \tau, m) = \frac{1}{1 + \exp(-m \cdot (x - \tau))}.$$

We use here a uniform kernel: $k(x) = \mathbf{1}\{|x| \leq 1\}$, where $\mathbf{1}\{.\} \in \{0, 1\}$ is the binary indicator function of a given event.

To formally define the range image segmentation task, to each pixel $s \in S$, we assign a $\omega_s^t \in \{\text{fg}, \text{bg}\}$ class label so that we aim to minimize the following energy function:

$$E = \sum_{s \in S} V_D(d_s^t | \omega_s^t) + \sum_{s \in S} \underbrace{\sum_{r \in N_s} \alpha \cdot \mathbf{1}\{\omega_s^t \neq \omega_r^{t-1}\}}_{\xi_s^t} + \sum_{s \in S} \underbrace{\sum_{r \in N_s} \beta \cdot \mathbf{1}\{\omega_s^t \neq \omega_r^t\}}_{\chi_s^t}, \quad (3)$$

where $V_D(d_s^t | \omega_s^t)$ denotes the data term, while ξ_s^t and χ_s^t are the temporal and spatial smoothness terms, respectively, with $\alpha > 0$ and $\beta > 0$ constants. Let us observe, that although



(a) Range image part (90° horiz. view) (b) Basic MoG [4], [7]

(c) uniMRF [2] (d) Proposed DMRF segmentation

Fig. 1. Foreground segmentation in a range image part with three different methods

the model is dynamic due to dependencies between different time frames (see the ξ_s^t term), to enable real time operation, we develop a causal system, i.e. labels from the past are not updated based on labels from the future.

The data terms are derived from the data energies by sigmoid mapping:

$$V_D(d_s^t|\omega_s^t = \mathrm{bg}) = \zeta(\varepsilon_{\mathrm{bg}}^t(s), \tau_{\mathrm{bg}}, m_{\mathrm{bg}})$$

$$V_D(d_s^t|\omega_s^t = \mathrm{fg}) = \begin{cases} 1, & \text{if } d_s^t > \max_{\{i=1\ldots k_s\}} \mu_s^{i,t} + \epsilon \\ \zeta(\varepsilon_{\mathrm{fg}}^t(s), \tau_{\mathrm{fg}}, m_{\mathrm{fg}}), & \text{otherwise.} \end{cases}$$

The sigmoid parameters τ_{fg}, τ_{bg}, m_{fg}, m_{bg} and $m_\star$ can be estimated by Maximum Likelihood strategies based on a few manually annotated training images. As for the smoothing factors, we use $\alpha = 0.2$ and $\beta = 1.0$ (i.e. the spatial constraint is much stronger), while the kernel bandwidth is set to $h =$ 30cm. The MRF energy (3) is minimized via the fast graph-cut based optimization algorithm [8].

The result of the DMRF optimization is a binary foreground mask on the discrete S lattice. The final step of the method is the classification of the points of the original $\mathcal{L}$ cloud, considering that the projection may be ambiguous, i.e. multiple points with different true class labels can be projected to the same pixel of the segmented range image. With denoting by $s = \mathcal{P}(p)$ for time frame t:

- $\omega(p) = \mathrm{fg}$, iff one of the following two conditions holds:
  (a) $\omega_s^t = \mathrm{fg}$ and $d(p) < d_s^t + 2 \cdot h$
  (b) $\omega_s^t = \mathrm{bg}$ and $\exists r \in N_r : \{\omega_r^t = \mathrm{fg}, |d_r^t - d(p)| < h\}$
- $\omega(p) = \mathrm{bg}$: otherwise.

The above constraints eliminate several (a) false positive and (b) false negative foreground points, projected to pixels of the range image near the object edges.

V. Evaluation

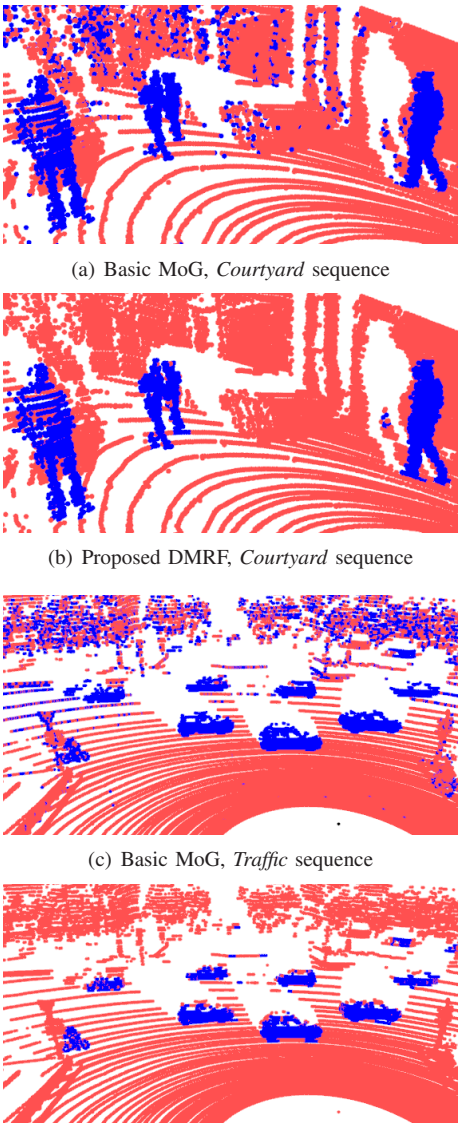
We have tested our method in real Lidar sequences concerning both video surveillance (*Courtyard*) and traffic monitoring (*Traffic*) scenarios (see Fig. 2). The data flows have been recorded by a Velodyne HDL 64E S2 camera, which operates with $R = 64$ vertically aligned beams. The *Courtyard* sequence contains 2500 frames with four people walking in a $25m^2$ area in 1-5m distances from the Lidar, with crossing trajectories. The rotation speed was set to 20Hz. In the background, heavy motion of the vegetations make the accurate classification challenging. The *Traffic* sequence was recorded with 5Hz from the top of a car waiting at a traffic light in a crowded crossroad. The adaptive background model was automatically built up within a few seconds, then 160 time frames were available for traffic flow analysis. We have compared our DMRF model to three reference solutions:

1) *Basic MoG*, introduced in Sec. III, which is based on [4] with using on-line K-means parameter update [7].
2) *uniMRF*, introduced in Sec. IV, which partially adopts the uniform foreground model of [2] for range image segmentation in the DMRF framework.
3) *3D-MRF*, which implements a MRF model in 3D. We define here point neighborhoods in the original $\mathcal{L}^t$ clouds based on Euclidean distance, and use the background fitness values of (2) in the data model. The graph-cut algorithm [8] is adopted again for MRF energy optimization.

Qualitative results on two sample frames are shown in Fig. 2. For Ground Truth (GT) generation, we have developed a 3D point cloud annotation tool, which enables labeling the scene regions manually as foreground or background. Next, we manually annotated 700 relevant frames of the *Courtyard* and 50 frames of the *Traffic* sequence. For quantitative evaluation metric, we have chosen the point level F-rate of foreground detection [3], which can be calculated as the harmonic mean of precision and recall. We have also measured the processing speed in frames per seconds (fps). The numerical performance analysis is given in Table I. The results confirm that the proposed model surpasses the *Basic MoG* and *uniMRF* techniques in F-rate for both scenes, and the differences are especially notable at the *Courtyard*. Compared to the *3D-MRF* method, our model provides similar detection accuracy, but the *proposed DMRF* method is significantly quicker. Observe that differently from 3D-MRF, our range image based technique is less influenced by the size of the point cloud. In the *Traffic* sequence, which contains around 260000 points within a time

Aspect	Sequence	Seq. property	Basic MoG	uniMRF	3D-MRF	**DMRF**
Detection rate (F-rate in %)	*Courtyard*	4 obj/frame	55.7	81.0	88.1	95.1
	Traffic	20 obj/frame	70.4	68.3	76.2	74.0
Processing speed (frames per sec)	*Courtyard*	65K pts/frame	120 fps	18 fps	7 fps	16 fps
	Traffic	260K pts/frame	120 fps	18 fps	2 fps	16 fps

TABLE I
NUMERICAL EVALUATION ON THE *Courtyard* AND *Traffic* SEQUENCES: DETECTION ACCURACY (F-RATE IN %) AND PROCESSING SPEED (FPS, MEASURED IN A DESKTOP COMPUTER)



(a) Basic MoG, *Courtyard* sequence

(b) Proposed DMRF, *Courtyard* sequence

(c) Basic MoG, *Traffic* sequence

(d) Proposed DMRF, *Traffic* sequence

Fig. 2. Point cloud classification result on sample frames with the *Basic MoG* and the proposed DMRF model: foreground points are displayed in blue (dark in gray print).

frame, we measured 2fps processing speed with 3D-MRF and 16fps with the proposed DMRF model.

VI. CONCLUSIONS

I have introduced a Dynamic MRF model for foreground segmentation in point clouds obtained by a rotating multi-beam Lidar system. We have proposed an efficient spatial foreground filter to decrease artifacts of angle quantitization and background motion. The model has been quantitatively validated based on Ground Truth data, and the advantages of the proposed solution versus three reference methods have been demonstrated. The author thanks his supervisor, Tamás Szirányi for guidance and help, and Csaba Benedek for his contributions in the concerning work.

REFERENCES

[1] I. Schiller and R. Koch, "Improved video segmentation by adaptive combination of depth keying and Mixture-of-Gaussians," in *Proc. Scandinavian Conference on Image Analysis, Ystad, Sweden*, ser. LNCS, vol. 6688, 2011, pp. 59–68. [Online]. Available: http://dl.acm.org/citation.cfm?id=2009594.2009602

[2] Y. Wang, K.-F. Loe, and J.-K. Wu, "A dynamic conditional random field model for foreground and shadow segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 279 –289, 2006.

[3] C. Benedek and T. Szirányi, "Bayesian foreground and shadow detection in uncertain frame rate surveillance videos," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 608 – 621, 2008.

[4] R. Kaestner, N. Engelhard, R. Triebel, and R.Siegwart, "A Bayesian approach to learning 3D representations of dynamic environments," in *Proc. International Symposium on Experimental Robotics (ISER)*. Berlin: Springer Press, 2010.

[5] B. Kalyan, K. W. Lee, W. S. Wijesoma, D. Moratuwage, and N. M. Patrikalakis, "A random finite set based detection and tracking using 3D LIDAR in dynamic environments." in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Istanbul, Turkey: IEEE, 2010, pp. 2288–2292. [Online]. Available: http://dblp.uni-trier.de/db/conf/smc/smc2010.html

[6] N. Muhammad and S. Lacroix, "Calibration of a rotating multi-beam Lidar." in *International Conference on Intelligent Robots and Systems (IROS)*. Taipei, Taiwan: IEEE, 2010, pp. 5648–5653. [Online]. Available: http://dblp.uni-trier.de/db/conf/iros/iros2010.html

[7] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 747–757, 2000.

[8] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2004.60

The Fermi GPU architecture as a CNN simulator

Endre László
(Supervisor: Ph.D. Péter Szolgay)
lasen@digitus.itk.ppke.hu

***Abstract*—During the years different solutions have been created to substitute the CNN (Cellular Neural Network) hardware if it is not available. Simulators have been created to help the implementation of algorithms based on CNN processor. Commercially available graphics cards with high computing capabilities make this simulator feasible. The aim of this work is to present the use of nVidia's Fermi architecture for speeding up such simulations. Different implementation approaches are considered and compared to a multi-core, multi-threaded CPU and some earlier GPU implementations. A detailed analysis of the introduced GPU implementation is presented.**

***Keywords*-CNN, simulator, Fermi, CUDA, GPGPU.**

I. Introduction

The use of GPU platforms is getting more and more general in the field of HPC (High Perfomance Computing) and in real-time systems. This lead to the evolution of GPGPU (General Purpose GPU) computing. The bare computing power of these devices is tremendous compared to the conventional CPUs. At this point it is worth noting that the memory wall in this computing environment is the most problematic bottleneck. Usually the data is stored in the main memory of the system. A CPU can access this memory through a cache hierarchy (with a maximum 21 GB/s bandwidth), which is quite fast compared to the a GPU that can access this data only through a PCIe (PCI Express) bus (with 4 GB/s aggregated speed through a 16 lane PCIe connector). On the other hand when the data is on the device (GPU) memory it can be accessed by the GPU with 128 GB/s through a 256 bit width GDDR5 memory interface (according to the specification of nVidia GeForce GTX 560 - Asus DCII Top graphics card). The computing power of these devices is given by the large amount of computing units, the cores. Different GPU manufacturers construct their cores differently. But it is common that each core contains pipelined ALU (Arithmetic Logic Unit) that implements the most essential arithmetic functions. These units have smaller instruction set implemented than CPUs have. The GeForce GTX 560 GPU has 336 CUDA (Compute Unified Device Architecture) cores in it. An Intel i5 660 CPU has two computing units (without the integrated GPU unit). These two units are used as 4 cores by the help of a hardware implemented Hyper-Threading Technology developed by Intel.

After a brief introduction to CNN state equation, CUDA programming architecture and Fermi hardware architecture the discussion of the optimized GPU and CPU implementation follows. The simulation performance results are then compared to prior implementations published earlier.

II. The CNN model

During the analisys the conventional CNN model introduced in [2], [1] is used. The original model is an analogue one defined by the following differential equation:

$$\dot{x}_{i,j}(t) = -x_{i,j}(t) + \sum_{C(k,l)\in S_r(i,j)} A(i,j,k,l)y_{k,l}(t) + \sum_{C(k,l)\in S_r(i,j)} B(i,j,k,l)u_{k,l}(t) + z_{i,j}$$

where $x(t)$ is the state variable, $C(k,l)$ is an element in the $S_r(i,j)$ neighbourhood, $A(i,j,k,l)$ and $B(i,j,k,l)$ are the feed-back and feed-forward templates, $u_{k,l}(t)$ is the input, $z_{i,j}$ is the offset and $y_{k,l}(t)$ is the output that is calculated according to the following formula:

$$y_{k,l}(t) = f(x_{k,l}) = 0.5(|x_{k,l}+1| - |x_{k,l}-1|)$$

The solution of this state equation is approximated using the forward-Euler method.

$$x_{i,j}(k+1) = x_{i,j}(k) + h\left(- x_{i,j}(k) + \sum_{C(k,l)\in S_r(i,j)} A(i,j,k,l)y_{k,l}(k) + \sum_{C(k,l)\in S_r(i,j)} B(i,j,k,l)u_{k,l}(k) + z_{i,j}\right)$$

If one assumes that the input image is permanent during the solution (i.e. $u_{k,l}(k) = u_{k,l}$) of the state equation, the calculation can be divided into two parts: the feed-forward and the feed-back part. The feed-forward part contains all the calculations that has to be done only once, at the beginning of the calculation:

$$g_{i,j} = \sum_{C(k,l)\in S_r(i,j)} B(i,j,k,l)u_{k,l} + z_{i,j}$$

The feed-back part uses the results of the feed-forward part to perform the given number of iterations:

$$x_{i,j}(k+1) = x_{i,j}(k) + h \left(-x_{i,j}(k) + \sum_{C(k,l)\in S_r(i,j)} A(i,j,k,l) y_{k,l}(k) + g_{i,j} \right)$$

This way unnecessary calculations can be avoided.

In this paper the diffusion template with $h = 0.2$ step size has been used to measure the performance of the presented implementations. The result of the template on a grey-scale image is shown in Fig. 1.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0.1 & 0.15 & 0.1 \\ 0.15 & 0 & 0.15 \\ 0.1 & 0.15 & 0.1 \end{bmatrix} z = 0$$

Fig. 1. Example of diffuson template. Original image on the left, diffused image on the right.

III. The CUDA architecture

CUDA is a combination of hardware and software technology [4] to provide programmers the capabilities to create well optimized code for a specific GPU architecture that is general within all CUDA enabled graphics cards. A thread in the CUDA architecture is the smallest computing unit. Threads (1024 in Fermi) are grouped into thread blocks. Thousands of thread blocks are organized into a block grid and many block grids form a CUDA application. This thread hierarchy with thread blocks and block grids make the thread organisation simpler. Thread blocks in a grid are executed sequentially. At a time n Streaming Multiprocessors (SM) are executing the n blocks (a portion of a block grid). The execution and scheduling unit within an SM is a warp. A warp consists of 32 threads. Thus the scheduler issues an instruction for a warp and that same instruction is executed on each thread. This procedure is similar to SIMD (Single Instruction Multiple Data) instruction execution and is called SIMT (Single Instruction Multiple Threads).

The memory hierarchy of the CUDA programming model can be seen in Fig. 2. Within a block each thread has its own register, can access the shared memory that is common to every thread within a block, has access to the global (graphics card RAM) memory and has a local memory that is allocated in the global memory but private for each thread.

A. Fermi hardware architecture

The CUDA programming architecture is used to implement algorithms above nVidia's hardware architectures such as the Fermi. Capabilities of the Fermi architecture are summarized in [7]. The base Fermi architecture is implemented using 3 billion transistors, features up to 512 CUDA cores that are organized in 16 SMs each encapsulates 32 cores. Compared to the earlier G80 architecture along with others a 10 times faster context switching circuitry, a configurable L1 cache (16 KB or 48 KB) and unified L2 cache (768 KB) is implemented. The L1 cache and the shared memory is allocated within the same 64 KB memory segment. The ratio of these two memories can be chosen as 16/48 KB or as 48/16 KB.

Just like the previous nVidia architectures, a read-only texture cache is implemented in the Fermi architecture. This cache is embedded between the L2 cache and ALU, see Fig. 2. This type of cache has many advantages in image processing. During a cache fetch, based on the given coordinates, the (dedicated) cache circuitry calculates the memory location of the data in the global memory, if a miss is detected the geometric neighbourhood of that data point is fetched.

Beside the texture cache the Fermi architecture has a constant memory cache (prior architectures had it as well). This memory is globally accessible and its size is 64 KB. Constant data are stored in the global memory and cached through a dedicated circuitry. It is designed for use when the same data is accessed by many threads in the GPU. If many threads want to access the same data but a cache miss happens, only one global read operation is issued. This type of cache is useful if the same, small amount of data is accessed repeatedly by many threads. If one would like to store this data in the shared memory and every thread in the same block wanted to access the same memory location, that would lead to a bank conflict. This conflict is resolved by serializing the memory access, which leads to a serious performance fall.

In Fermi nVidia introduced the FMA (Fused Multiply and Add) [7] operation using subnormal floating point numbers. FMA performs the multiplication, and all the extra digits of the intermediate result are retained. Then the addition is performed on the denormalised floating number. Truncation is performed in the last step. This highly improves the precision of iterative algorithms.

IV. Fermi based CNN implementation

Earlier different implementations of CNN using GPU were reported [10], [9], [3]. In this paper a different approach is presented.

The memory access pattern of the CNN simulation makes this problem a memory-bandwidth bounded problem. Fortunately, the introduced memory caching structure perfectly fits the problem and hides most of the latency and bandwidth limitations.

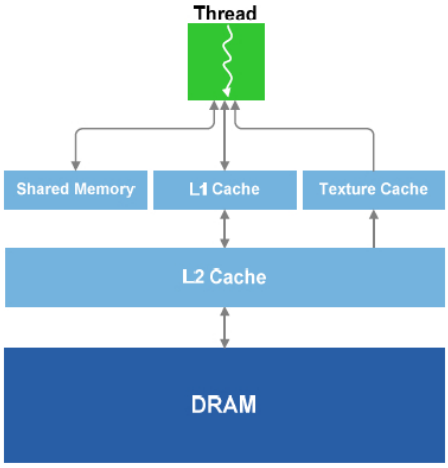


Fig. 2. Fermi memory hierarchy according to [8]

The following discussion details the use of texture and constant caching methods. A texture array reference is defined in the code as global variable. The reference is bounded to the float array using the *cudaBindTexture2D()* function. In this way the array can be written using global memory access (through L2/L1 cache) and read as read-only memory through the texture cache. After the texture cache is defined it can be read using the *tex2D()* function. Similarly to the previous texture definition, the constant memory definition works in the following way. The array containing the template values is stored in a float array. A global constant memory reference is defined. This reference is bounded to the float array using the *cudaMemcpyToSymbol()* function. After the constant cache is defined it can be used as a conventional array using square brackets.

Using these techniques the CNN state equation can be solved easily. If the input image considered static, the feed-forward part of the equation has to be calculated only once. This result can be reused in every iteration of the feed-back part of the equation. The equation is solved with the conventional forward (explicit) Euler method discussed earlier.

Every thread that realizes the kernel calculates one pixel of the state equation. Threads are organized in 32x32 sized blocks, and the grid size is calculated according to the size of the image. The block size has been obtained by trial and error method.

One iteration of the state equation is calculated in each kernel invocation. Thus 50+1 (feed-backs+feed-forward) kernel invocations are necessary to perform the calculations. This is an important fact, because kernel invocations are OS dependent. Under Windows each kernel invocation takes $36\mu s$. This value is significantly lower under Linux. The overall time saved by running the entire simulation under Linux is nearly $0.9ms$.

The available computing power is not utilised perfectly. The memory wall problem can not be avoided even with these techniques. Using texture and constant cache the code becomes clear and efficient.

V. CPU BASED CNN IMPLEMENTATION

For the sake of comparison an optimized CPU code has been produced. During the tests an Intel i5 600 is used with 2 cores at 3.33 GHz clock rate, 4 MB L3, 256 KB/core L2 and 64 KB/core L1 cache. The latter is subdivided to data and instruction cache, in 32 KB and 32 KB partitions. The processor has hardware support for multi threaded applications, called Hyper-Threading Technology (HTT), which is Intel's implementation of SMT (Simulatanious MultiThreading). This processor is based on the Westmere architecture.

The optimized CNN simulator code exploits the benefits of the SSE4.2 (Streaming SIMD Extension 4, subset 2) instruction set. These instructions are operating on the 128 bit wide specialized registers and 4 single precision floating point operations can be performed in each clock cycle.

For the implementation the Intel Integrated Performance Primitives (IPP) library was used. This library contains highly optimized image and signal processing functions for Intel CPUs. It takes advantage of the SSE4.2 instruction set and the HTT, thus providing an easy to use yet efficient tool for solving the state equation of CNN.

VI. COMPARISON OF IMPLEMENTATIONS

Exhaustive comparison of the presented implementations together with the previous FPGA and GPU implementations is detailed in this section. First, the presented GPU and CPU implementation is discussed. Then these results are compared to prior implementations created by different authors.

In order to compare the two devices the theoretical FLOPS (single precision) rates provide a good basis. The test system was a desktop machine with an Intel Core i5 660 processor, 4GB of RAMs, an Intel motherboard and Asus graphics card with nVidia GeForce GTX 560 DCII Top (over-clocked by Asus to 925MHz engine and 1850 MHz shader frequency). The operating sytem was a 64bit Ubuntu 11.10 with kernel version 3, nVidia driver version 285.05.33 and CUDA 4.1 toolkit was used. According to Intel [5] the performance of the processor is 26.664 GFLOPS (at 3.33 GHz) and 29 GFLOPS (at 3.6 GHz Single Core Max Turbo). According to [6] the dual warp scheduler in the Fermi architecture is capable of issuing 2 instructions per clock cycle, thus the total number of instructions issued per second is the product of clock cycles per second and the n number of CUDA cores times the 2 instruction issued per second: $f \times n \times 2 = 1850Hz \times 336cores \times 2 = 1243.2GFLOPS$. However there are only n number of cores, the graphics processor is capable of performing only $f \times n = 1850Hz \times 336cores = 621.6GFLOPS$, i.e. each core can initiate one floating point operation (Addition, Multiplication or FMA - Fused Multiply and Add) per clock cycle [7]. FMA instructions are counted as 2 instructions, therfore theoretical peak performance is $1243.2GFLOPS$. Performance metrics of the different architectures are compared on Table I.

TABLE I
COMPARISON TABLE

	i5 660	**GTX560**	CELL	8800GT	XC5VSX240T
Frequency [MHz]	**3330**	**1850**	3200	1350/1500	550
Cores [pcs]	**2(4threads)**	**336**	8	120/112	117
Peak performance [GFLOPS]	**26.664**	**1243.2(FMA)**	25.6	504	-
Power consuption [W]	**73**	**150 (base version)**	86	160	25
Cell iteration / s [10^6]	**397**	**4397**	3627	590	64350
Communication time [ms]	**0**	**1.56**	-	-	4.48
Speedup	**1**	**11.07**	9.13	1.49	162.1

Results of the presented GPU and CPU implementations on 512x512 sized image compared to the previous results [10], [11], [12] are summarized on Table I.

Using the presented GPU and CPU implementations 4397 and 397 cell iteration per second was achieved respectively. The results show that the Xilinx Virtex-5 FPGA implementation is the fastest during the solution of the CNN state equation. Taking into account that the first CELL processor was introduced in 2005, this solution has a remarkable performance even nowadays. The presented solution using the GTX560 GPU outperformed the previous GPU implementation reported by Soos et al. [10]. The current implementation is 7.45 times faster than the 8800GT implementation, if the memory transfer time is not considered. This factor is reduced to 5 if the data transfer overhead is taken into account. Partially this is explained by the facts that the number of cores, the frequency, the context switching, the FMA instructions and other minor developments have been introduced in the Fermi architecture. The main difference between the previous and the current implementations is the way state image is accessed and stored. In the previous work [10] the shared memory is used to store parts of the state image, whereas in the present case texture cache has been used for this purpose.

VII. CONCLUSION

In the present article a novel Fermi based GPU and an IPP library based CPU implementation of CNN simulator has been introduced. A comparison of different solutions with the presented results has been discussed. The presented GPU implementation is significantly 11.07 times faster than the near optimal CPU implementation and 7.45 times an earlier 8800GT GPU implementation. It is even faster than the CELL processor implementation. The FPGA implementation is the fastest but developing proper circuitry for an FPGA takes much more effort than to prepare a GPU code. The presented GPU solution is simple and effective, thus it is a good choice for simulation of the CNN dynamics.

VIII. ACKNOWLEDGEMENTS

The support of the grants TÁMOP-4.2.1.B-11/2/KMR-2011-0002 and TÁMOP-4.2.2/B-10/1-2010-0014 is gratefully acknowledged.

REFERENCES

[1] Roska, T. and Chua, L.O., *"The CNN universal machine: an analogic array computer,"* in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 40, no. 3, 1993, pp.163-173.

[2] L.O. Chua and L. Yang, *"Cellular Neural Network: Theory,"* in IEEE Transactions on Cicuits and Systems, vol. 35, no. 10, 1988, pp. 1257-1272.

[3] R. Dolan and G. DeSouza, *"GPU-Based Simulation of Cellular Neural Networks for Image Processing,"* in Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, 2009, pp. 730-735.

[4] NVIDIA. (2012, Apr 12). *NVIDIA CUDA C Programming Guide - Version 4.2* [Online]. Available: http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf

[5] Intel. (2011, Sept 6). *Intel microprocessor export compliance metrics* [Online]. Available: http://www.intel.com/support/processors/sb/CS-032815.htm

[6] T. R. Halfhill. (2009, Sept). *Looking Beyond Graphics, White Paper* [Online]. Available: http://www.nvidia.com/content/PDF/fermi_white_papers/T.Halfhill_LookingBeyondGraphics.pdf

[7] NVIDIA. *Whitepaper, NVIDIA's Next Generation CUDA Compute Architecture: Fermi v1.1* [Online]. Available: http://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIAFermiComputeArchitectureWhitepaper.pdf

[8] D. Triolet. (2010, Jan 26), *Nvidia GeForce GF100: the geometry revolution?* [Online]. Available: http://www.behardware.com/articles/782-1/nvidia-geforce-gf100-the-geometry-revolution.html

[9] A. Fernandez, R. San Martin, E.Farguell and G.Egidio Pazienza, *"Cellular Neural Networks Implementation on a Parallel Graphics Processor Unit,"* in Proceedings of the 11th International Workshop on Cellular Neural Networks and their Applications, 2008.

[10] B. G. Soós, Á. Rák, J. Veres and Gy. Cserey, *"GPU boosted CNN simulator library for graphical flow-based programmability,"* in EURASIP J. Adv. Signal Process, Hindawi Publishing Corp., New York, NY, United States, 2009, no. 8, pp. 1-11.

[11] Zs. Voroshazi, A. Kiss, Z. Nagy and P. Szolgay, *"Implementation of embedded emulated-digital CNN-UM global analogic programming unit on FPGA and its application,"* in Int. J. Circ. Theor. Appl., Wiley, 2008, vol. 36, pp. 589–603.

[12] L. Füredi, Z. Nagy, A. Kiss and P. Szolgay, *"An improved emulated digital CNN architecture for high performance FPGAs,"* in International symposium on nonlinear theory and its applications, Krakow, 2010, pp. 103-106.

Finite Element Algorithms and Data Structures on Graphical Processing Units

István Z. Reguly
(Supervisors: András Oláh, Tamás Roska)
reguly.istvan@itk.ppke.hu

***Abstract*—Solution of partial differential equations on unstructured meshes is one of the most important problems in engineering. The finite element method can provide an approximate solution with a higher accuracy than other methods at the expense of increased computational requirements. Because the evolution of computing architectures shows an increasing gap between memory bandwidth and computational power, the FEM is better suited for long-term scaling.**

We discuss both the assembly and the solution part of the finite element method with a special attention to the balance of computations and data movement. We present a GPU assembly algorithm that scales to arbitrary degree polynomials used as basis functions without requiring additional on-chip resources at the expense of redundant computations. We show how the storage of the stiffness matrix affects the performance of both the assembly and the solution. We investigate two approaches: global assembly into the CSR and ELLPACK matrix formats and matrix-free algorithms, and show the trade-off between the amount of indexing data and stiffness data.

We discuss the performance of different approaches in light of the caching behaviour on Fermi GPUs and show a speedup over the CPU of up to a 100 times in the assembly and up to 40 times in the solution phase. We present our sparse matrix-vector multiplication algorithms that are part of the conjugate gradient iteration in the solution phase and show that a matrix-free approach may be up to two times faster than global assembly approaches and up to 5 times faster than NVIDIA's cuSPARSE library. Finally, we compare the performance of the GPU to a 12 core CPU using OpenMP.

***Keywords*-Graphical Processing Unit, Performance Analysis, Finite Element Method, Sparse Matrix-Vector Multiplication, Conjugate Gradient Method**

I. Finite Element Problem

The finite element method (*FEM*) is a powerful numerical method for approximating the solution of partial differential equations (*PDEs*) [1]. The method is based on the polygonal discretisation of the domain Ω over which the PDE is to be solved. Equation (1) describes a simple elliptic problem with a Dirichlet boundary condition:

$$-\nabla \cdot (\kappa \nabla u) = f \text{ in } \Omega, \tag{1}$$

$$u = 0 \text{ on } \partial\Omega. \tag{2}$$

The solution is sought in the form of $u : \Omega \to \Re$, with $u = 0$ on $\partial\Omega$. The standard finite element method constructs a finite dimensional space V_h of functions over Ω, and searches for an approximate solution $u_h \in V_h$. If the number of vertices in the discretisation of Ω is N_e, then let $\{\phi_{1\dots N_v}\}$ be a basis for V_h, then:

$$u_h = \sum_i \bar{u}_i \phi_i \tag{3}$$

To find the best approximation to u, it is necessary to solve the system:

$$K\bar{u} = \bar{l}, \tag{4}$$

where K is the $N_v \times N_v$ matrix, usually called the *stiffness matrix*, defined by:

$$K_{ij} = \int_\Omega \kappa \nabla\phi_i \cdot \nabla\phi_j \, dV, \quad \forall i,j = 1,2,\dots,N_v, \tag{5}$$

and $\bar{l} \in \Re^n$, usually called the *load vector*, is defined by:

$$\bar{l}_i = \int_\Omega f \cdot \phi_i \, dV, \quad \forall i,j = 1,2,\dots,N_v. \tag{6}$$

If the underlying discretisation mesh has nodes $\bar{x}_i$, it is possible to choose a finite element space V_h with basis functions such that $\phi_i(\bar{x}_j) = \delta_{i,j}$. In this case u_h is determined by its values at $\bar{x}_i$, $i = 1,2,\dots N_v$. The mesh is a polygonal partitioning of the domain Ω into a set of disjoint elements $e_i \in E$, $i = 1 \dots N_e$. The basis functions are constructed so that ϕ_i is nonzero only over those elements e which have $\bar{x}_i$ as a vertex. This means that finite element basis functions ϕ_i have their support restricted to neighbouring elements, thus the integral in equation (5) is non-zero only if the two vertices belong to the same element.

II. Storage formats

In our GPU implementation the stiffness matrix described by equation (5) is stored in three formats:

1) CSR uses three vectors, one for pointers to the first element of each row, one storing the values of the non-zeros and one storing their column indices.
2) ELLPACK stores non-zeros and column indices in a `dimRow` by `K` matrix, where `K` is the maximum row length. This matrix is transposed in GPU memory so that the n^{th} non-zero of each row is in one contiguous block of memory, enabling coalesced memory transfers.
3) LMA stands for Local Matrix Approach which exploits the fact that during the iterative solution of the linear system $K\bar{u} = \bar{l}$ the stiffness matrix K is not required explicitly. During matrix assembly an $m*m$ local matrix

is calculated for each element. In the global matrix approach these values are scattered according to the vertex indices, but the local matrix approach stores them as they are, and calculates the matrix-vector product in the following way [2], [3]:

$$\bar{y} = \mathcal{A}^T(K_e(\mathcal{A}\bar{x})), \tag{7}$$

where K_e is the matrix containing the local matrices in its diagonal and $\mathcal{A}$ is the local-to-global mapping from the local matrix indices to the global matrix indices. In a similar way to ELLPACK the values of the local matrices are stored in a way that the n^{th} value of each one is in one contiguous block of memory. This approach stores stiffness data redundantly, however it does not have to store row and column indices explicitly as it is available in the structural description of the underlying mesh.

III. The Finite Element Algorithm on GPUs

The algorithms in this paper are based on quadrilateral elements and can work for elements of any order. The calculation of the coefficients of the basis functions, the local quadrature points and the gradients are based on a transformation from the reference square. This bilinear transformation is calculated for every element and applied to the data of the reference square stored in constant memory. The pseudocode for each element is described by algorithm 1.

When increasing the degree of polynomials used as basis functions, both the number of degrees of freedom and the number of quadrature points increases as a square function of the degree. For 2D quadrilateral elements it is equal to: $(degree + 1)^2$. To perform the minimal amount of computations, the local quadrature points and the inverse of the jacobian evaluated at each one of them can be precomputed and reused in the innermost loop of algorithm 1. Alternatively, any of these can be recalculated every time they are needed thereby saving local storage. This enables us to trade computations for local storage space. In the CPU versions, where register pressure does not pose a problem, all these values are precomputed. However, the register pressure makes GPU implementation infeasible even at $degree = 2$. Thus, our GPU kernel recalculates the positions of the local quadrature points and jacobians every time they are needed in the innermost loop, which enables the kernel not to store anything in local arrays that would grow in size with the increasing degree of polynomials used - our kernel uses the same number of registers for any degree.

Another approach to the assembly is to exchange the loops over the quadrature points and the pairs of degrees of freedom, in which case the jacobian does not have to be recalculated. However, in this case the values of the local stiffness matrix are updated repeatedly. This is a viable option on the CPU, but on the GPU these local matrices cannot fit in either the local memory or the cache which results in dramatically increased memory traffic.

For testing purposes meshes were generated with different numbers of elements and with different degree elements in a way that a mesh with first degree elements would have corresponding higher degree meshes with the same number of degrees of freedom. The default numbering of the degrees of freedom is based on iterating through elements and their degrees of freedom and assigning a number in an increasing order. In several scenarios a colouring of these elements [4] is required to avoid write conflicts.

Colouring is done on two levels: block and element. Since there is no explicit synchronisation between blocks in CUDA, blocks of elements with different colour are processed by different kernel calls, thereby making sure no two blocks access the same data at the same time. Within these blocks different elements are assigned to each thread along with their colour. When these threads access memory in a potentially conflicting way, these accesses are executed colour-by-colour with an explicit synchronisation between each one. The colouring of elements was done by iterating through them and assigning the first colour available that was not used by any of the elements neighbours.

Algorithm 1 Local stiffness matrix and load vector assembly

```
I_local ← M_e(e)
generate bilinear mapping based on the four vertices
calculate local quadrature points quadPoints_local
Calculate jacobians J to map to each quadrature point
for i = 1 to length(I_local) do
  if i is not constrained then
    for j = i to length(I_local) do
      if k is not constrained then
        for each quadrature point p in quadPoints_local
        do
          K_local(i,j) += weights(p) * ∇φ_i(p) * ∇φ_j(p)
        end for
        K_local(i,j) = K_local(j,i)
      end if
    end for
    for each quadrature point p in quadPoints_local do
      l̄_local(i) += weights(p) * φ_i(p) * f(p)
    end for
  end if
end for
```

Most GPU algorithms differ only in the way the local stiffness matrix and load vector are written to global memory.

IV. Performance Evaluation

The performance measurements were obtained on a workstation with two Intel Xeon X5650 6-core processors, 24GBytes of system memory running Linux kernel 2.6.35. The system has 2 NVIDIA Tesla C2070 GPUs, both with 6GB global memory clocked at 1.5GHz and 384-bit bus width, 448 CUDA cores in 14 streaming multiprocessors (SMs) clocked at 1.15GHz. The GPU codes were compiled with NVIDIA's nvcc compiler with the CUDA 4.0 framework with the

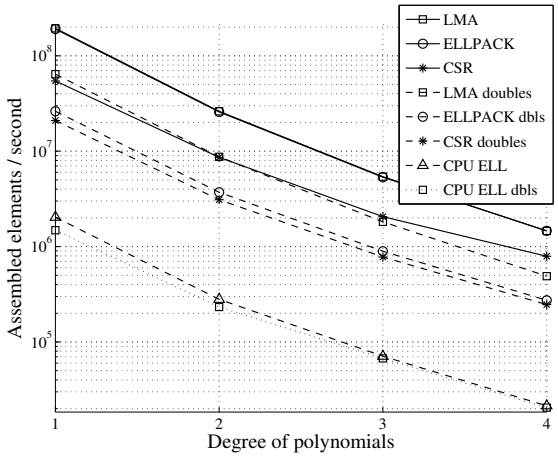


Fig. 1. Number of elements assembled and written to global memory when using different storage formats and storage precision.

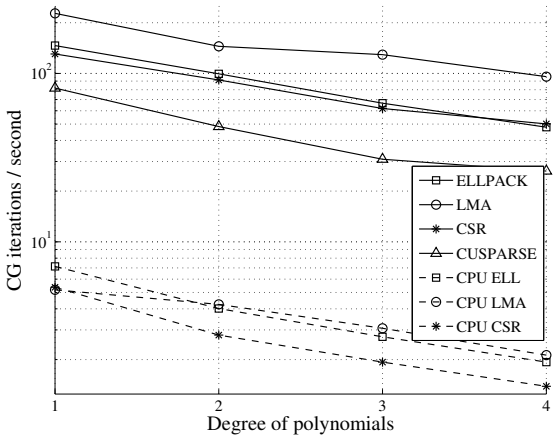


Fig. 2. Number of CG iterations per second with different storage formats at single precision.

$--use\_fast\_math$ flag. The L1 cache size was set to 48kB, turning it off or reducing its size to 16kB decreased performance in all cases.

A. Test problem

Since our goal was to investigate the relative performance of the finite element method using different approaches on different hardware we chose a simple Poisson-problem with a known solution:

$$-\Delta u(\bar{x}) = sin(\pi x_1) \cdot sin(\pi x_2), \tag{8}$$

$$u(\bar{x}) = 0 \; on \; \partial\Omega. \tag{9}$$

The underlying two dimensional grid consists of quadrilateral elements with number of nodes up to 16 million and the order of elements ranges from 1 to 4. The performance figures are constructed in a way that the tests run on grids that have the same number of degrees of freedom; so the actual number of elements in a fourth degree test case is one sixteenth of the number of elements in a first degree test case. The conjugate gradient iterative method was used to approximate the solution of the linear system $K\bar{u} = \bar{l}$.

V. Performance Analysis

A. The assembly phase

Both the assembly and the spMV phase have to move the entire stiffness matrix to or from memory, and this transfer makes up the bulk of all memory traffic in both phases. Looking at the performance degradation when the order of elements increase, it is apparent that the assembly becomes much slower than the spMV. This means that while increasing the order of the elements results in having to move more data to and from memory, it also requires more computation in the assembly phase because the number of quadrature points also increases as a square function (in 2D) of the degree of polynomials used. These factors indicate that the assembly phase is compute limited. CSR throughput figures on the other hand show an increasing tendency, while its bandwidth utilisation is the same as that of the other two approaches. The reason for this is the high percentage of cache misses; while threads writing to LMA and ELLPACK data layouts work on a small set of cache lines at the same time, thanks to their transposed data layout, threads writing to the CSR layout do not.

Also shown in figure 1 is that even though the LMA approach has coalesced memory accesses and transfers significantly less data than the ELLPACK format, assembly performance is the same. Based on these observations, it can be stated that the assembly kernel is increasingly compute limited with higher order elements. According to the Visual Profiler's output and our own calculations the instruction throughput of the LMA approach is around $540 * 10^9$ instructions per second, which is a good proportion of theoretical 1TFLOPS throughput considering the amount of integer arithmetic and control overhead.

When moving to double precision, it can be seen in figure 1 that the LMA assembly performance performance is only a third of its single precision version. Double precision ELLPACK and CSR have to use colouring due to the lack of native support for atomic operations with doubles - the performance ratio between single and double precision assembly using colouring in both cases is only 2:1. In the case of the CPU, the compute limits are again apparent, even though the CPU versions have to perform less computations at the expense of having to store local quadrature points and their jacobians. Regardless of the memory access pattern, all versions perform similarly.

The GPU's speedup over a single CPU thread is between a factor of 70 and 100 in single precision using atomics, and between 25 and 40 in double precision. The speed difference in the actual calculations and in the theoretical performance of the GPU versus a single core of the CPU are very close.

B. The spMV phase

The sparse matrix-vector product is commonly known to be a bandwidth limited operation [5]. In fact it has to move just a little less data than the assembly phase, but the number of operations is significantly less: approximately one fused multiply-add for each nonzero element in the matrix. Using the LMA approach incurs an overhead of having to avoid race conditions using atomics or colouring. The LMA approach offers fully coalesced access to the elements of the stiffness matrix, and both ELLPACK and CSR use optimizations to improve bandwidth efficiency. However, the access to the elements of the multiplicand vector is not coalesced, but caching can improve performance. Global matrix approaches like CSR and ELLPACK store the least amount of stiffness data, but they have to read a column index for every nonzero of the sparse matrix. The LMA approach uses the mapping M_e to get row and column indices. Figure 2 shows that global matrix approaches have very similar performance, but the less data-intensive local matrix approach provides the best performance. As expected, the difference increases with the increasing degree of elements. This also supports the conclusion that the sparse matrix-vector product is bandwidth limited. For first order elements the LMA approach utilises close to 100% of theoretical bandwidth; it is important to note however that these figures include the effects of caching, which in this case greatly improves the performance of LMA by serving global memory requests from L1 and L2 caches. For higher order elements the ELLPACK layout shows the best bandwidth utilisation, but since it has to move more data it is still up to 50% slower than the LMA layout. Although using either the CSR or the ELLPACK layout results in having to move the same amount of useful data, the transposed layout of ELLPACK provides up to 10% higher effective bandwidth. The zeros padding the rows of ELLPACK are not factored into these figures.

On average the GPU's speedup over the CPU using global matrix approaches is around a factor of 25 in single precision and 15 in double precision. Local matrix approaches outperform the CPU by up to 40 times in single and 15 in double precision.

VI. Conclusion

A study of the finite element method has been presented that analyses two main approaches to the construction and solution of the finite element problem: the local matrix approach and the global matrix approach using either the CSR or the ELLPACK storage format. We present an assembly algorithm that scales well with the increasing degree of polynomial used as basis functions. The implications of the choice of storage approach are analysed in detail: performance bottlenecks resulting from patterns of memory access, amount of computation, handling of race conditions, use of resources and occupancy. The local matrix assembly is demonstrated to be viable alternative to global assembly providing the best performance because of the reduced data transfer required. It is shown that at higher order elements the advantage of the local matrix approach in the sparse matrix-vector multiplication increases. A variation of the local matrix approach, the matrix-free method which avoids the storage of stiffness data by recomputing them on-the-fly is shown to become heavily compute-limited at higher order elements.

The most widely used sparse matrix storage format CSR (compressed sparse row) turns out to be a slightly worse choice than LMA or ELLPACK, being on average two times slower in the assembly and 10% slower in the spMV phase because of its complicated indexing scheme and non-coalesced access. NVIDIA's CUSPARSE library is also used for the spMV phase, its performance being on average just 50% lower than our hand-coded kernel. The ELLPACK storage format proves to be the best choice for sparse matrix-vector multiplication, because it makes coalesced access possible when transposed. In the assembly phase it performs close to the LMA approach. The number of assembled quadrilateral elements per second outperforms triangular assembly shown in [6], [3] by a factor of up to 10.

We also show the viability of extension to multiple GPUs and multi-threaded CPUs. Using multiple threads on a 12 core hyper-threaded CPU increases performance by up to 10 times in the assembly phase and 5 times in the solution phase. Considering that the assembly is compute-limited, the factor of 10 speed difference between the fully utilised GPU and CPU matches the difference in their theoretical performance.

Acknowledgment

The author would like to thank Prof. Mike Giles at Oxford University for his help developing these algorithms and preparing a paper on the subject.

References

[1] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, 1987.

[2] C. Cantwell, S. Sherwin, R. Kirby, and P. Kelly, "From h to p efficiently: Strategy selection for operator evaluation on hexahedral and tetrahedral elements," *Computers & Fluids*, vol. 43, no. 1, pp. 23 – 28, 2011, symposium on High Accuracy Flow Simulations. Special Issue Dedicated to Prof. Michel Deville. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045793010002057

[3] G. R. Markall, D. A. Ham, and P. H. Kelly, "Towards generating optimised finite element solvers for GPUs from high-level specifications," *Procedia Computer Science*, vol. 1, no. 1, pp. 1815 – 1823, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050910002048

[4] E. L. Poole and J. M. Ortega, "Multicolor ICCG Methods for Vector Computers," *SIAM Journal on Numerical Analysis*, vol. 24, no. 6, pp. 1394–1418, 1987.

[5] N. Bell and M. Garland, "Efficient sparse matrix-vector multiplication on CUDA," NVIDIA Corporation, NVIDIA Technical Report NVR-2008-004, Dec. 2008.

[6] C. Cecka, A. J. Lew, and E. Darve, "Assembly of finite element methods on graphics processors," *International Journal for Numerical Methods in Engineering*, vol. 85, no. 5, pp. 640–669, 2011. [Online]. Available: http://dx.doi.org/10.1002/nme.2989

Bandwidth-Limited Mesh Partitioning

Antal Hiba
(Supervisors: Péter Szolgay and Miklós Ruszinkó)
hiban@digitus.itk.ppke.hu

***Abstract*—In case of many computational problems an unstructured mesh is given (computation on sensor data, simulations of physical systems - PDEs), where the vertices represent computations with dependencies represented by the edges. Utilization of processing elements (PEs) during these computations is mainly depends on the node indexing of the mesh. If the adjacent nodes are stored close to each other in main memory, the reloading of node data can be ignored by using the on-chip memory. The mesh and an ordering of its nodes, define the graph bandwidth, which determines the minimum size of on-chip memory, which can provides zero reloading. If the required on-chip size is higher than the available resources, the mesh must be divided into parts. In this paper two methods are presented, which constructs reordered parts from a given unstructured mesh, where each part meets the constraint on graph bandwidth.**

I. Introduction

Nowadays many-core architectures like GPUs and FPGAs have hundreds of processing elements (PEs), which leads to high theoretical computational power. Unfortunately, the PEs of these architectures are often waiting for input data, and the utilization drops below 100%. The optimization of memory-accesses is a possibility to increase utilization of processing elements.

In case of FPGA the memory accesses can be fully determined by the designer. Nearly the theoretical bandwidth of the off-chip DRAM can be utilized by moving data in long sequential bursts between the off-chip memory and the PEs in the FPGA. However, optimized input data is necessary, where all dependent data are inside an index-range (in main memory), which can be stored on-chip. If the dependencies are described by a mesh, the result of the optimization is an ordering of nodes, where the index-range is minimized.

This task is similar to a well-studied optimization problem, called Matrix Bandwidth Minimization. One of the most practical heuristic solutions is GPS(Gibbs, Pole and Stockmeyer)[1], which is fast enough to handle graphs with many million nodes effectively. If the reordered input has grater on-chip memory requirement than the available resources, the input mesh must be divided into parts.

Famous partitioning methods, for instance METIS[2], minimize the edge-cut between the parts, and balance the size of the generated parts. The size-balance is important because each part is given to a multi-processor, and the overall runtime is determined by the processor which get the largest part. The edge-cut is proportional to the communication required between the processors. Bandwidth of the resulting parts is smaller than the bandwidth of the whole mesh, but the methods do not deal with the bandwidth directly.

One possible direction of handling the graph bandwidth constraint, is an extended ordering method, which estimates the graph bandwidth, and starts new part, when the estimated value reaches the bound.

In many cases the covering surface (set of extremal nodes) of the mesh is also known, which gives information about the geometry, but not used by traditional partitioners. In this paper a novel partitioning method is shown, which creates parts with minimized bandwidth, using geometrical information derived from the cover. The proposed method is an example, which presents new possibilities in mesh partitioning.

A. Objective

The main goal is to reduce the bandwidth of the resulting parts. However, the objective of bandwidth minimization alone is meaningless, because the bandwidth of the resulting parts can be decreased optionally by increasing the edgecut. The edgecut is also important, because communication between the processors is proportional to the edgecut.

Reducing the bandwidth of the parts with acceptable communication requirement is the objective of the proposed methods. The acceptable communication requirement (edgecut) is an application-specific parameter. In novel FPGA array the cost of reading data from the off-chip memory of an adjacent FPGA is usually 10 times slower than reading from its own off-chip memory. In case of using Alpha-Data ADM-XRC-6T1 cards, the theoretical memory bandwidth is 12.8 Gbyte/s inside, and 1.25 Gbyte/s between the cards[4][1]. When only 10% of the whole memory accesses are external reads and their occurrences are balanced, the whole memory bandwidth can be utilized to feed PEs.

II. Extended ordering method (AM1)

Let G(V,E) a graph with vertex set V, $|V| = n$ and edge set E. Labeling is a function f(v) which assigns integers[1..n] to vertices. f(v)=f(u) if and only if u=v $u, v \in V$. N(v) is the set of vertices which adjacent to v. The bandwidth of a vertex v is $B_f(v) = Max\{|f(v) - f(u)| : u \in N(v)\}$, and the graph bandwidth corresponds to G with labeling f() is $B_f(G) = Max\{B_f(v) : v \in G\}$.

The minimal size of the Local (on-chip) Memory can be given as:

$$Minimal\ Size = BW * sizeof(node)$$

$$BW = (B_f(G) * 2 + 1)$$

[1] the number of adjacent parts is limited

A. AM1 Bounded BW method

The task of AM1 is to reorder the nodes into a different sequence such a way to minimize BW, therefore size of the LM is also minimized.
The structure of a solution part **P** with n elements is shown in Figure 1. All nodes in **P** have been indexed, therefore **P** is the ordered part of the mesh-data.

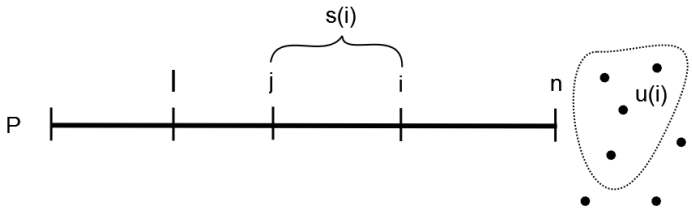


Fig. 1. Structure of solution part P.

s(i): is the distance between **i** and its lowest indexed neighbor in the part: $s(i) = MAX\{i.index - j.index : j \in N(i)\}$.
u(i): is the set of nodes which uncovered by **P**, but must be added in later steps because of node **i**: $u(i) = \{v : v \in N(i)\ AND\ v \notin P\}$.
I: is the index of the first elemet which has not empty u() set, so for every node **i** where $i.index < I$ all neighbors covered by P.
imp(i): Importance of node **i**, used for decisions: $imp(i) = (n - i.index) + |u(i)| + s(i)$
The starting node is chosen by the same way in GPS[1]. In each step a node added to the part from u(node(I)). AM1 selects the node which is adjacent to a node in P with highest importance.
Given an AM1 Part, the task is to estimate its BW value. Every time when index I is changed, the following equation have to be checked:

$$S = s(node(I))$$

$$E = (n - I) + |u(node(I))|$$

$$BWBound \geq MAX\{S, E\} * 2 + 1 \qquad (1)$$

New nodes are added to the part P when Eq. 1 holds, otherwise the part is finalized and a new instance of AM1 is started on the rest of nodes, when Eq. 1 is not hold.

III. Depth Level Structure (DLS) Based Bisection

DLS is a hidden structure in every unstructured mesh for which the covering surface is defined. Depth is the distance from the cover. Nodes of the mesh with same depth belongs to a level. Nodes in the deepest levels represent the critical areas of the mesh in case of bandwidth minimization.

A. Basic Entities and Operations

DLS-Based partitioning uses some subroutines which are general tools for manipulating node sets. The most important node set is the covering surface, furthermore the separators are also node sets in our nomenclature. These operations are based on waves (breadth-first search - BFS) which are starting from a set of nodes and spreading through the mesh.

a) ***Cover****:* Set of nodes belonging to the covering surface.

b) ***Deepest****:* Set of nodes in the deepest levels of DLS. The Deepest set contains the three deepest levels of the DLS structure.

c) ***Level_Structure(in: in_set, out: LS)****:* Generates a level-structure from in_set. LS is a series of sets (levels), where the elements of in_set form the zero level, and the rest nodes associated to the level according to their minimal distance from in_set.

d) ***Level(in: node, LS)****:* A function which returns the level index of node in LS.

e) ***Pseudo_Diameter(in: in_set, out: (u,v))****:* Gives the two endpoints of a pseudo diameter on in_set. The method is similar to the first step of GPS method, returns two points which have maximal distance from each other.

f) ***Grow(in: start_node, border_set, out: out_set)****:* Grows a set from start_node, by adding the neighbors of included elements into the set. An element is added if it has no node from border_set as its neighbor.

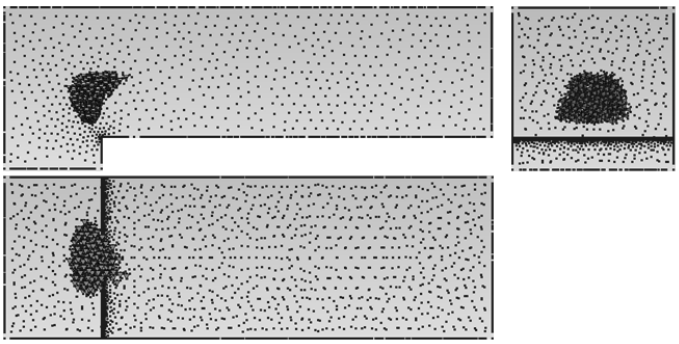


Fig. 2. Deepest set of tunnel202, xyz projections are shown. Points are the vertices of covering surface, nodes of Deepest set represented by tetrahedrons.

B. DLS Bisection

The base concept of DLS bisection is the division of the mesh along the deepest set of nodes. GPS method creates level-structures from an extremal node and indexes nodes level by level. Bandwidth of the solution is proportional to the size of the largest level. In geometrical view, the ordering starts from an extremal surface and creates onionskins through the mesh, and the bandwidth is proportional to the largest cutting surface. In case of a structured grid of a rectangle, the bandwidth of GPS solution is proportional to the smaller side, which is often optimal. The elements of Deepest set take place on a line which is perpendicular to the smaller side, furthermore the line separates the rectangle into two equal sized parts.
DLS can be obtained by the Level_Structure() routine, starting the BFS from the Cover set, the resulting level structure will be the DLS structure. The Deepest set is the union of the three deepest levels in DLS. With pseudo diameter routine, the method gets two endpoints of Deepest set, which have

maximal distance from each other in the whole mesh(Deepest set is not necessarily connected). DLS-Based bisecting method generates the separating surface in two steps. In the first stage a set of nodes are obtained which have the same distance from the two endpoints of the Deepest set's pseudo-diameter. The resulting set is used during the second stage.The final set separates the mesh into two parts.

// Get Separator
1 $Level\_Structure(Cover, DLS)$
2 $Pseudo\_Diameter(Deepest, (u, v))$
3 $Level\_Structure(\{u\}, LU)$
4 $Level\_Structure(\{v\}, LV)$
5 $sep1 = \{x : Level(x, LU) - Level(x, LV) \in \{0, 1\}\}$
6 $Pseudo\_Diameter(sep1, (u, v))$
7 $Level\_Structure(\{u\}, LU)$
8 $Level\_Structure(\{v\}, LV)$
9 $sep2 = \{x : Level(x, LU) - Level(x, LV) \in \{0, 1\}\}$

The resulting separator is parallel to the pseudo diameter of the deepest set, but the separator not necessarily intersects the Deepest set. The correction step is responsible for placing the separator to the middle of the Deepest set.
The partition method is not completed by determining the separator surface, because our separator is a set of nodes, therefore the partition is ambiguous. Two parts are obtained by using the Grow subroutine, but the separator and its nearest neighborhood still remains unpartitioned. Unpartitioned nodes are added to the smaller part. The method can be finished at this point, but the size balance is not guaranteed. A simple solution is to grow the smaller part till balance reached.

// Get parts from separator
1 $s \notin Separator$
2 $Grow(s, Separator, part1)$
3 $s \notin Separator \cup part1$
4 $Grow(s, Separator, part2)$
5 $rest = \{x : x \notin part1 \cup part2\}$
6 $Add\ rest\ to\ the\ smaller\ part$
7 $Grow\ smaller\ part\ till\ balance\ reached$

IV. Results

A. Extended Ordering Method

It is obvious that the proposed algorithm generates access patterns which has lower BW than a given bound. The solution quality can be measured by the node reload factor **k** which is defined by the ratio of the length of the access pattern and the number of nodes.

Measurements on three meshes with different BW bounds can be found on Table I. The results shows that the BW value of large meshes can be reduced more effectively(with better k factors), with BW_Bound=0.5*AM1_BW, we get k={1.65 1.31 1.08}. BW_Bound is determined by the on the on-chip memory capabilities of the FPGA, which is increasing with every new generation of the technology, furthermore the ratio becomes better for larger problems.

TABLE I
Results of AM1 bounded bandwidth optimization

Case	AM1_BW	BW Bound	num. of parts	N	overall length	k	time(s)
3d_075	411	412	1	3562	3562	1	0,264
3d_075	411	400	3	3562	4489	1,26	0,485
3d_075	411	300	8	3562	5241	1,471	0,932
3d_075	411	200	15	3562	5899	1,656	1,247
3d_035	1893	1894	1	33730	33730	1	2,367
3d_035	1893	1800	3	33730	37952	1,125	6,979
3d_035	1893	1500	5	33730	39987	1,185	7,14
3d_035	1893	1000	15	33730	44439	1,317	9,523
3d_015	14985	14986	1	417573	417573	1	76,88
3d_015	14985	14000	2	417573	430693	1,031	79,869
3d_015	14985	10000	3	417573	439136	1,052	154,14
3d_015	14985	7500	7	417573	452510	1,084	68,43
3d_015	14985	5000	21	417573	483391	1,158	59,01
3d_015	14985	2500	97	417573	577474	1,383	1170,8

AM1_BW: the bandwidth provided by AM1 for the whole mesh
overall length: length of the generated access pattern
N: number of vertices

B. DLS-Based Bisection

Our test environment is based on GMSH[3], which is a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. In GMSH simple 3D models can be defined, meshed and partitioned. The covering surfaces are given as physical surfaces, which makes possible for our algorithm to get the cover of the mesh.
The test models are shown in Figure 3. Meshes generated from sgrid are structured grids of hexahedrons, from snake and weight unstructured tetrahedron-based meshes are generated with uniform density, in case of tunnel the density of the mesh is increased around the step.

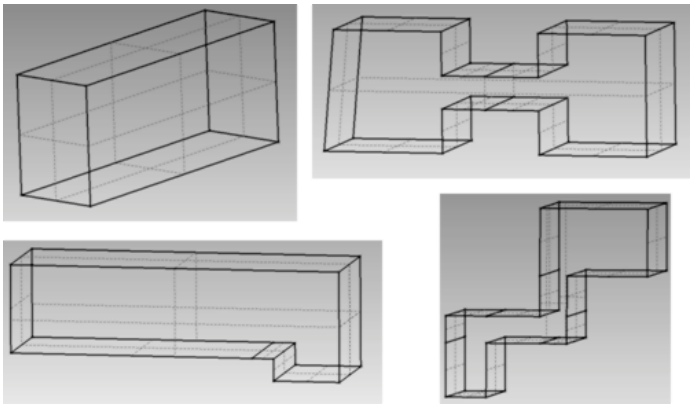


Fig. 3. Test shapes. sgrid, weight (up), tunnel, snake(down)

A comparison is shown in Table II between the DLS-Based and the METIS-recursive partitioning. The objective of the two partitioner is different, but there is no other known method, which minimizes the bandwidth of the resulting partitions, and METIS is one of the most popular solver.
For the structured brick-shaped problems, the DLS method provides average 28% better BW partition, with acceptable communication ratio (external reads / inner). The COMM ratio is better for larger problems, in case of sgrid4, which has 1M vertices, the COMM ratio is only 1,6%.

TABLE II
RESULTS OF DLS BASED PARTITION

Problem	N	Orig BW	MET BW	MET COMM	DLS BW	DLS COMM
sgrid1	2200	221	181	0,0388	141	0,1216
sgrid2	16800	841	641	0,0192	479	0,0628
sgrid3	131200	3281	2517	0,0087	1719	0,0323
sgrid4	1036800	12961	9967	0,0045	6599	0,0164
snake100	7821	777	689	0,0254	531	0,079
snake038	158544	5701	5371	0,0074	4941	0,0095
tunnel202	18210	2353	1385	0,0292	1303	0,0262
tunnel100	191592	12525	5675	0,017	5949	0,027
weight045	4899	641	581	0,0169	311	0,1764
weight022	35922	2363	2131	0,0078	1411	0,0559
weight012	230891	8087	8785	0,0037	8785	0,0075

N: number of vertices. Orig BW: GPS bandwidth for the whole mesh.
MET/DLS BW: bandwidth of partitions.
MET/DLS COMM: number of outgoing edges / number of internal edges

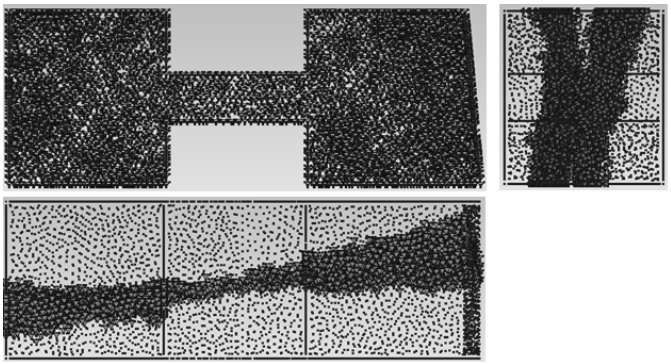


Fig. 4. Separator of weight022.

The communication ratio (edge-cut ratio) is getting better when the mesh density is increased for all problem instances. This is obvious because the cutting surface has N-1 dimension in case of an N dimensional mesh. This feature is important, because DLS computes a kind of N-1 dimensional surface, which separates the mesh into two parts. DLS-Based solutions can have unacceptable communication need for small meshes, for example weight045, where the COMM ratio is 17,5%.
Using DLS-Based bisection the resulting partitions have 40% reduced bandwidth compared to the whole mesh, and create 20% better solutions than METIS. METIS minimizes the edge-cut with providing size-balance, the DLS-Based solutions have same size balance quality, however the edge-cut is several times higher. There is a tradeoff between bandwidth and communication need, and DLS creates partitions with higher COMM ratio to provide reduced bandwidth.
DLS reduces the bandwidth by 40-50% by separating the mesh along the Deepest set, in case of tunnel geometry the METIS solution has the same reduction ratio, for tunnel202 the partition is nearly similar and DLS has better COMM ratio, for tunnel100 the METIS solution has better bandwidth. The difference between the traditional methods and DLS-Based partitioning can be observed on Figure 4, where the separator of weight022 is shown. DLS-Based bisection cuts the weight shape in longitudinal direction, instead of choosing the small edgecut between the two weights. This partition leads to 34% better bandwidth reduction with 5,6% COMM ratio.

If the DLS separator do not provide size balance, the proposed method grows the smaller part until the balance is reached. This strategy can harm the bandwidth of the solution. For weight012 the bandwidth of the resulting parts are 6741 without size balance, and 8785 after, because the balance leads to a quasi similar partition to METIS. The bandwidth of the parts can be higher than the original mesh, because GPS is a heuristic algorithm, and the bandwidth mainly depends on the largest cutting surface.
The proposed method is created for providing partitions which meet constraints on the bandwidth. If the original mesh has larger GPS bandwidth than the given bound, the method bisects the mesh. Parts can be bisected if the constraint on the bound is still unsatisfied. The constraints on bandwidth can be reached in less bisection steps with DLS-Based method, than with other traditional partitioners.

V. CONCLUSIONS

In this paper a novel partitioning problem is presented, which is a combination of the classical partitioning problem, and the bandwidth reduced node ordering on the resulting parts. The motivation of this partitioning problem is based on FPGA designs, where the graph-bandwidth of the input parts has to be below a bound. Two solvers are shown: an extended ordering method, which can be applied to every unstructured mesh, and the DLS-Based partitioner, which needs additional information from the covering node set.
The covering surface is often given (physical simulations), but is not used by traditional partitioners. The proposed method is an example of using covering surface in partitioning techniques. The components of the DLS-Based method are operations on node sets, which operations use spatial waves to compute abstract surfaces (node sets) inside the mesh.
The results show that the proposed partitioning algorithm creates partitions with better graph-bandwidth quality than METIS, with acceptable edgecut. The size-balancing steps of the method should be improved, because the graph-bandwidth quality of the solution can be damaged.

REFERENCES

[1] N.E. Gibbs, W.G. Poole, P.K. Stockmeyer, An algorithm for reducing the bandwidth and profile of sparse matrix, SIAM Journal on Numerical Analysis 13 (2) 236-250 (1976)
[2] G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing Vol. 20, No. 1, pp. 359-392 (1998)
[3] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, Volume 79, Issue 11, pages 1309-1331, (2009)
[4] www.alpha-data.com

Appendix

PhD-studies started in 2009:
- Tamás Fülöp
- Domonkos Gergelyi
- András Horváth
- Miklós Koller
- László János Laki
- Csaba Nemes
- Mihály Radványi
- Ádám Rák
- Attila Stubendek
- Gábor János Tornai
- Tamás Zsedrovits

PhD-studies started in 2010:
- Dóra Bihary
- Bence József Borbély
- Zsolt Gelencsér
- Petra Hermann
- Antal Hiba
- Csaba Máté Józsa
- Bálint Péter Kerekes
- Márton Zsolt Kiss
- András József Laki
- György Orosz
- István Zoltán Reguly
- János Rudan
- Norbert Sárkány
- Emília Tóth
- Zoltán András Tuza

PhD-studies started in 2011:
- István Endrédy
- Dániel Györffy
- Anna Horváth
- Balázs Knakker
- Péter Lakatos
- Endre László
- Dömötör Molnár
- Attila Novák
- Balázs Oláh
- Borbála Siklósi
- Zóra Solymár