SOLVING NON-TOPOGRAPHIC PROBLEMS WITH TOPOGRAPHIC AND SYNCHRONIZATION ALGORITHMS AND ARCHITECTURES

András Horváth

Theses of the Ph.D. Dissertation

Pázmány Péter Catholic University
Faculty of Information Technology

Supervisor:
Dr. Miklós Rásonyi
Scientific advisor:
Prof. Tamás Roska

Budapest, 2012

1 Introduction

It is an interesting phenomenon, how limitless the freedom of thinking is. Maybe this could be the biggest advantage and present of our life, that our imagination allows us to create anything without time, space or any physical constraints. We can not feel the difference between imagining 10 or one billion state variables.

However, when we have to implement an algorithm we have to obey physical rules, constraints, because the functionality of our device is inherently bounded in time and space. With one processing unit we only have to solve the problem of time, the space itself is not important. In contrast, on a many-core architecture, even unwillingly, we have to consider how the processing units are placed and connected. While the development of very large scale integration techniques has led us to the era of many-core architectures, a serious gap has emerged between the two representations of algorithms: between the algorithm development process and the optimization methods on specific hardwares. To avoid this problem we have to consider the well-defined place of implementation, we have to map our method onto a two dimensional wafer (not considering vertical integration) and this implies the precedence of locality and in the end – even if we have not considered this before – our algorithm will be topographic, because it is bounded in a two dimensional state space, where only cellular, local connections are allowed and efficient. Global connections are feasible as series of local connections only.

Meanwhile in the previous years the speed of Moore's law is further decreasing and the operating speed of processor has not increased significantly either, we have observed how many-core architectures and cellular processors had gain foothold. The number of transistors that can be manufactured on a silicon wafer is increasing further and further. This creates a gap between high level/abstract algorithms and the many-core architectures. We can not yield for higher processing speed but we can add additional parallel cores instead. Because of this phenomenon in parallel algorithms on many-core architectures the wire delay became the most important determining factor instead of the gate delay. The transfer of data between the cores can decrease the execution speed significantly. To avoid this we have to process all the data locally, and avoid global communication as much as possible. The only communication which is affordable in a fast, efficient way is the local topographic data exchange.

These many-core architectures have shown their usefulness in practice in numerous practical, topographic problems (mostly in molecular dynamics, solution of nonlinear partial differential equations and in image processing). These efficient, new solutions were all based on local communication and multi-parallel processing.

One of the main challenges of engineering today is however not to further enhance the current solutions and further optimize the topographic multi-parallel algorithms on even better many-core architectures, but to identify certain problems or problem classes and possibilities, which can be transformed into topographic problems and mapped on many-core architectures.

In this dissertation I would like to show, how the stochastic selection of certain elements from a given set can modify the implementation and efficiency of algorithms. Also apart from the algorithms and practical problems I would like to examine and compare two different (the localized and the global) selection mechanisms, and the 'quality' and 'diversity' of random sets generated by them.

I would like to investigate this through simplified models, where I show the usefulness of localized stochastic selection, and I will also show, that the localized method can substitute its global counterpart, also showing the advantages of the local selection, how it can be easily implemented and executed on a multi-parallel architecture. In this work I have examined local stochastic selection as a general algorithm through generally applicable models and I have also investigated stochastic selection by focusing on well-defined, specific, general problems.

In the second part of this dissertation I will show an architecture, which is inherently topographic. This network is made of Spin torque oscillators (STOs). In this case only a cellular locally connected architecture is feasible, hence the information exchange, interaction between these oscillators happens through the magnetic field. To implement a processor we can not avoid to understand the behavior of spin torque oscillators. In the second part of this thesis I will describe how we can understand and simplify the synchronization of weakly coupled oscillator networks. I have examined this architectures from a computational point of view and I have shown a few examples how they can be used to solve simple practical tasks.

I have implemented a simulator in C, Python and Matlab. With this program I managed to investigate the behavior of the STO arrays in general. I have also calculated the equilibrium of an STO with a closed formula and this way the behavior can be calculated, without solving the differential equations. Using the harmonic balance and the describing function techniques I have shown, how the behavior of the synchronized oscillation can be calculated in any arbitrary array. Apart form the transient behavior, any phase shift, frequency and spin position can be examined in any arbitrary array regardless the boundary condition, initial condition, or coupling weights between the elements in the array. I have also investigated in a more detailed way the case of two coupled oscillators. I have calculated how the phase shift between the synchronized oscillator depends on the input-current and on the coupling weight between the oscillators. These results contribute on the creation and design of O-CNN arrays built by STOs. Also the synchronization based computation can open new horizons and possibilities in the application of the CNN technology.

2 Methods used in the experiments

Unfortunately stochastic sampling is too complex to create a thorough theoretical examination and estimate bounds how an implementation would perform on a many-core architecture. Also the number of existing architectures is tremendous, this way it is almost impossible to create theories covering every architecture and problems.

To avoid this, I have introduced a general model of the stochastic optimization, which is able to mimic all the important characteristics. These models reveal different properties of stochastic optimization in different depths.

I have expanded these models to practical problems to show they are capable of simulating practical problems, too, apart from the theoretical results. Through this I have linked the general models to some practical problems and case studies. I have shown through several case studies (similarly to how it is usually done in the literature) how cellular implementations and topographic algorithms can be used in practice.

I have selected two algorithms to measure the performance of the cellular implementation. The stochastic sampling plays a crucial role in both of the methods. These two problems are the genetic algorithm and the particle filter algorithm. The test problems for the genetic algorithm were the N-Queen, Knapsack and Travelling Salesman problems. All of these are NP-hard, and general problems, and they can be extended to various practical problems.

To average out the randomness in the algorithms I have repeated all the experiments at least 1000 times, this way I can have reliable results. I have also measured the variance between the 1000 averages and it could be neglected.

Apart from this the most important part of the experiments was the architecture. Because the commercial computers are usually not many-cores and the existing many-core architectures show extreme diversity regarding the basic architecture (FPGAs, GPUs) I have implemented a virtual cellular machine, that mimics all the important characteristics. In this architecture the processing units are placed on a two-dimensional rectangular grid. This architecture and the measurements were implemented in Python.

I also wanted to measure the results on an existing chip and architecture. I chose the $Xenon_v3$ device, because I have had some experience about its capabilities and programming. The programming of the device can be done in Assembly language.

The other part of this dissertation, the investigation of the synchronization of Spin-torque oscillators contains mostly theoretical results for this reason the necessary devices, simulation and experiments was limited. But it is always a useful extension to support theoretical results by simulations and/or measurements. This was the reason why I have created a general simulator, which is capable to simulate the behavior of arbitrary STO arrays with various parameters. The implementation was done in MATLAB, C and PYTHON languages. The topography of the STO array can be arbitrary: we can simulate even globally coupled (in spite of the underlying physics is unfeasible) arrays. I have received the macro-model of the oscillators from physicists from the University in Notre Dame.

This simulator was a useful tool to test the two computational case-studies of STO array and the theoretical results in simulations (even if there were no possibilities to measure them in practice with an actual device).

3 New scientific results

1. Thesis:

1.1. Thesis: *I have investigated through general models, how the biology-motivated, localized stochastic selection affect the diversity and the quality of the generated sets. I have shown that by selecting optimal parameters (neighborhood radius, mutation factor) the quality of the generated sets (in case of the general models) is comparable to the quality of the global selection method, meanwhile the localized method can easily mapped into a many-core architecture and fits perfectly on multi-parallel, cellular devices, this way it can be executed with significant speedup. I have shown a way, how a non-topographic algorithm -stochastic selection- can be transformed into a topographic problem, and in what kind of advantages this can result.*

I have introduced, the biology-inspired, localized selection and I have compared it to its original, global counterpart through two general (and some problem dependent) models. I have done the comparison through general models, and from these models I have specified some practical problems by modifying the characteristics of these models. These general models are able to mimic the most important characteristics of stochastic selection. Through them I have shown, how the localized selection can substitute its global counterpart, and I have shown through simulations and measurement how the information propagation speed amongst elements in the selected set depends on the tuning of the neighborhood radius.

I have also shown, that the localized version is not a brand new, completely different method, it can be considered as a generalization of the global method, where we can set the speed of information propagation amongst the elements through the neighborhood radius. (If the neighborhood radius is larger than the maximal distance between any two selected elements, we will have the same algorithm as the global method; and if the neighborhood radius is set to zero we will implement the original simple Monte Carlo method, where there are no selection mechanism at all.)

I have also investigated, how the neighborhood radius can affect the diversity of a set and the distribution of the weights of the selected elements (exploitation/exploration ratio): this phenomena can be seen on Fig. 1.

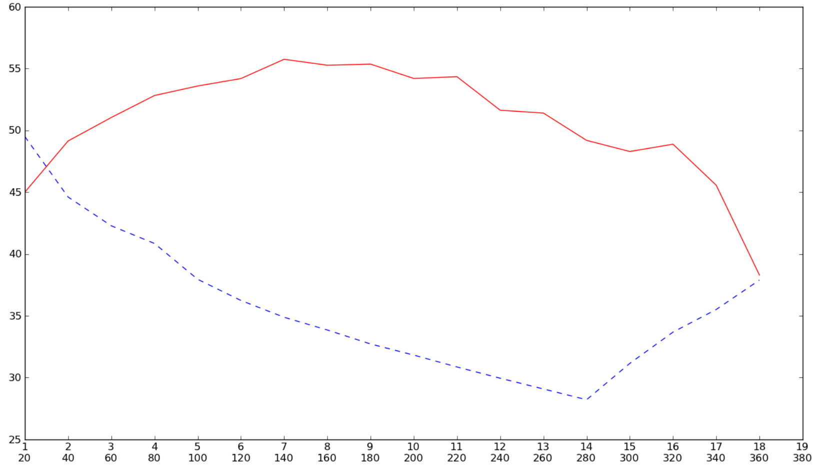

Figure 1: On this image the comparison of the local (red, continuous) and global (blue, dashed) selection methods can be seen. The numbers of the X-axis are the parameters of the methods, in every bracket the first number is the neighborhood radius and the second number is the number of the preserved elements in one set in case of the global selection. The experiments were done using 400 elements. On the Y-axis the evaluation of the fitness function can be seen for the best element in the set: higher values represent better solutions. As it can be seen from this values the characteristics of the local selection is a reversed "U" shape. From this we can derive how we can fasten the convergence of our method by tuning and selecting the optimal neighborhood radius.

1.2. Thesis: *I described general rules and guidelines about mapping the cellular genetic algorithm on a CNN architecture. I have also implemented this modified, cellular version of algorithm on a general virtual machine and an existing cellular, many-core system: on the* $Xenon_v3$*, CNN chip. I have measured the efficiency of this implementation through simulations and measurements.*

I have shown through three different case studies (the N-queen, the Knapsack and the Travelling salesman problem), how a CNN implementation of the genetic algorithm can solve difficult optimization problems in an efficient and elegant way with power consumption in the milliwatt range. The implementation gives a possibility to execute one iteration of the genetic algorithm in milliseconds (for the exact problems: $1884, 2982, 1373$ iterations per second could be executed for the N-queen, Knapsack and Travelling salesman problems) on the $Xenon_v3$ architecture. This execution times are orders of magnitudes better than other current results. I have compared these times with similar current results. In case of the Knapsack problem the fastest current result, what I have found for a problem with same complexity needed 0.92 second, which is much larger and would not be fair to be compared with the speed of the CNN implementation. In case of the Traveling Salesman problem I have compared my method with a CPU-GPU implementation, where the execution time was 0.20 seconds, meanwhile the power consumption of the GPU-card they have used (an NVIDIA GTX 280) is 310 W , not considering the additional consumption of the computer and this can not be compared to the power consumption of the CNN chip, which consumed less then 5 milliwatts and the execution time was 93 milliseconds. This shows that a two-times speed up could be reached meanwhile the power consumption remained 1500 times lower than the CPU-GPU implementation. This shows, how this implementation could be useful in tasks where complex optimization tasks have to be solved with lower power consumption within a strict time limit like in case of navigation, speech-processing, parameter optimization or in case of other problems generated by mobile devices.

Apart from the implementation I have describe a general method, how a cellular genetic algorithm can be implemented on a multi-layered CNN architecture. This description can give help and hints to implement the algorithm on other similar devices. The sketch of the general implementation divided into different steps and layers can be seen on Fig 2.

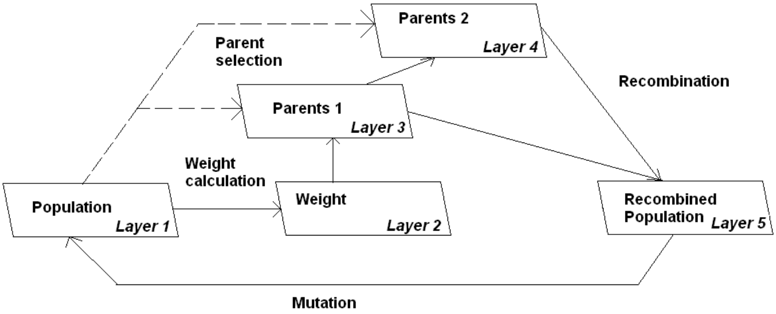

Figure 2: This image shows, how the cellular genetic algorithm can be implemented on a multi-layered CNN architecture. The easiest implementation of this method requires a $3+P$ layered CNN, where P represents the selected number of parents for each recombination. Without recombination we select only one parent ($P=1$) and the method can be implemented on a 4 layered CNN architecture. In case of regular recombination with two parents P equals two and we require 5 layers for the implementation. The continuous lines are representing operations, which can be implemented as a set of nonlinear CNN templates, the dashed lines are representing simple data transfers between the layers. The figure shows, in which layers the results can be stored, processed and how the layers are connected through the operations of the genetic algorithm. The implementation on the $Xenon\_v3$ chip was completely based on this description.

1.3. Thesis: *I have shown how the localized selection mechanism can be used in case of a dynamic state estimator: in the particle filter algorithm. I have introduced the Cellular Particle Filter algorithm. I have implemented the algorithm on a virtual cellular machine and also on an existing architecture, the $Xenon_v3$ chip.*

Also in the case of this problem I have investigated three case studies, to show how the algorithm can be implemented and utilized. Also the simulation results and the real measurements on the $Xenon_v3$ architecture were examined. I have shown in case of Hidden Markov models, how the cellular algorithm can be used for state estimation, even in problems when neither the Kalman-filter nor the Baum-equation can be applied, because our state-transition is non-linear and the stat-space is contin-

Table 1: In this table we can see the estimation errors for a commonly used benchmark model. The first column contains the number of the particles, the rest contains the mean-square errors of the global and the cellular methods. The errors are the average mean square error as the distance between the hidden and the estimated state, the lower errors mean better estimations. Every value is a result of 1000 independent measurements to avoid the noise that comes from the stochastic behaviors of our model. As it can be seen from the results the error of the cellular method is lower when a large number of particles are applied.

Number of Particles	Pfilter	Cellular PF
16	66.42	77.38
36	55.44	61.61
49	53.66	57.17
64	52.56	54.65
81	51.65	53.00
100	50.98	51.58
144	50.51	49.69
225	50.01	48.44
400	49.49	47.61
625	49.16	47.08
900	49.13	47.01

uous (or infinite). I have shown that this algorithm approximates well not only the expected value of the hidden state, but also the distribution of hidden states as well, this can be used later on to approximate probabilities and conditional expectations as well. I have shown through commonly used case studies that the cellular version can be implemented with a faster execution speed, it is easily scalable and with the proper setting of the parameters we can approximate the hidden state with a lower error rate. I have also shown through the same case studies the reason behind this lower error rate: I have measured that in the cellular, topographic method the diversity of the particles is higher than in the global method, (meanwhile the approximation is the same) because the information propagates amongst the particles in a cellular, local way.

Some example results can be seen in Table 1, for a commonly used model. As it can be seen from the results in case of certain parameters, the error of the cellular method is lower (especially, if the umber of the particles is relatively high).

2. Thesis:

2.1. Thesis: *I gave an analytic solution, a closed formula to calculate the equilibrium of the oscillation of the differential equation of spin torque oscillators. Thanks to this solution the equilibrium of the oscillation can be calculated as a function of the geometry, input current and magnetic permeability of the oscillator, without the time-consuming numerical simulation of the differential equation.*

The simple macro-model of spin-torque oscillators can be be described as:

$$\frac{d\mathbf{M}}{dt} = \gamma(\mathbf{M} \times \mathbf{H}) - \gamma\alpha\mathbf{M} \times (\mathbf{M} \times \mathbf{H}) - \gamma A\mathbf{M} \times (\mathbf{M} \times \mathbf{S}) \qquad (1)$$

Where $\mathbf{M}$ is the spin-vector, $\mathbf{H}$ is the magnetic field, $\mathbf{S}$ is the direction of the input current, A is the strength of the input current $\times$ notes the cross product between the vectors, γ and α are physical constants,the gyromagnetic-constant and the magnetic efficiency.

From this equation I have derived the fix-point of the third component of $\mathbf{M}$ (M_z), which determines the plane of the oscillation:

$$M_z^* = -\frac{A}{\alpha\, M_s(N_x - N_z)} = \frac{A}{\alpha\, M_s\, \delta} \qquad (2)$$

Where $\mathbf{N} = (N_x, N_y, N_z)^T$ is a vector determining the geometry of the oscillator, with the following constraints: $\delta = N_z - N_x = N_z - N_y$.

From this the oscillation can be reduced to a simple harmonic, circular motion, with the following parameters:

The frequency of the oscillation:

$$\omega = \frac{\gamma\, A}{\alpha} \qquad (3)$$

The amplitude of the oscillation:

$$B = \sqrt{(1 - M_z^{*2})} \qquad (4)$$

I have checked these result with the simulator, and the results match the analytical solutions.

Because from an engineering point of view we are interested in only the equilibrium and not at all in the transient states, this analytic solution can be a huge help in the investigation of spin-torque oscillators, because thanks to this solution there is no need for the time-consuming numerical simulation of the oscillations, we can derive them easily and efficiently by a closed formula.

2.2. Thesis: *I created a mapping between a current-coded input and the phase-coded output of a cellular STO array. I gave an approximation with the help of the "harmonic Balance technique" to the phase-shift of synchronized,weakly coupled STOs in a general network. I have shown that, when the coupling strength in the x and y components are the same, this method is an exact solution and not an approximation. And even when the coupling strength in the two components are different the error of the approximation is four orders smaller, than the error of the amplitude of the spin. With the help of this method the phase shifts in a general network of oscillators can be calculated, without solving a difficult and complex differential equation system. We can calculate the phase shifts by solving only an algebraic equation system. With the help of these equations I have given an example how an architecture can be built, when the processing elements are small oscillators, consisting only a few atoms (or only one), and I have also given two simple examples which kind of topographic calculations can be done by such a device.*

The connection, and this way the synchronization between weakly coupled spin torque oscillators happens through the magnetic filed:

$$\begin{aligned} \mathbf{H}_{eff} &= -\mathbf{H}_i + \mathbf{G}_j \\ &= -M_s \begin{pmatrix} N_x M_{x\,i} \\ N_y M_{y_i} \\ N_z M_{z\,i} \end{pmatrix} + M_s \begin{pmatrix} C_x^j M_{x\,j} \\ C_y^j M_{y_j} \\ C_z^j M_{z\,j} \end{pmatrix} \end{aligned} \tag{5}$$

Where the coupling strength between oscillators i and j is determined by the vector $\mathbf{C}$. And from this the effective magnetic field can be calculated: $\mathbf{H}_{eff}$.

The approximation of any general network of STOs can be done by these equations, I have obtained with the usage of spectral techniques ("Harmonic Balance" and "Describing Function") on the differential equation system. This way we can easily calculate the phase-shift between synchronized oscillators.

We can introduce the following vectors:

- $\mathbf{P} = (P_1, P_2 ... P_N)'$

- $\mathbf{K} = (K_1, K_2 ... K_N)'$
- $\mathbf{\Theta} = (0, \theta_{0-1} ... \theta_{0-N})'$
- $\mathbf{A} = (A_1, A_2 ... A_N)'$

Where $\mathbf{K}$ contains the plane of oscillations, also the fixed-point in the third component, $\mathbf{A}$ is the strength of the input current of the oscillators and ($\mathbf{P} = \sqrt{1 - \mathbf{K}^2}$) and $\mathbf{\Theta}$ is the relative phase difference between the oscillators.

$\mathbf{R_{xy}}$ is the coupling matrix in x and y components, and similarly $\mathbf{R_z}$ is the coupling weights in the third component.

The equation system obtained with this notation:

$$\begin{gathered} \eta\alpha\delta\mathbf{K} \circ \mathbf{P} - c\mathbf{A} \circ \mathbf{P} - \eta\alpha(\mathbf{R_z K}) \circ \mathbf{P} = \\ = -\eta\mathbf{R_{xy}}(\mathbf{P} \circ sin(\mathbf{\Theta})) - \eta\alpha(\mathbf{R_{xy}}(\mathbf{P} \circ (cos\Theta))) \circ \mathbf{P} \end{gathered} \tag{6}$$

$$\begin{gathered} \mathbf{P}\omega = \eta\alpha\mathbf{K} \circ \mathbf{P} + \eta(\mathbf{R_z K}) \circ \mathbf{P} \\ -\eta\mathbf{R_{xy}}(\mathbf{P} \circ cos(\mathbf{\Theta})) \circ \mathbf{P} \circ \mathbf{K} + \eta\alpha\mathbf{R_{xy}}(\mathbf{P} \circ sin(\mathbf{\Theta})) \end{gathered} \tag{7}$$

I have shown, that if the coupling strengths in the C_x and C_y components are equal, the previous equation system is an exact solution of the phase shift defined by the differential equation system of the STOs. Also when the coupling strength are not equal in the two components a minor oscillation will appear in the third, z component, but the magnitude of this oscillation is four orders smaller as the amplitude of the oscillations in the two other components.

I have also derived the simpler form of the previous equation system for only two oscillators:

$$\theta = \operatorname{asin}(\frac{A_\Delta}{2rM_s}), \tag{8}$$

where A_Δ is the different between the input current on the two oscillators, r is the coupling weight between the oscillators (which is proportional to the distance between the oscillators), M_s is a physical constant, the magnetic saturation.

From this equation in can be seen, that the phase shift between the oscillators depends linearly from the input current difference, and hyperbolically from the distance between the oscillators.

Based on these equations I have designed and simulated two examples, showing how the an cellular network of STOs can be used for computation. One example is a simple grayscale edge detection on two-dimensional input images. The other example is a more complex spatial-difference detection where not only the pixel differences but also the spatial distribution of the distances are considered during the difference calculation. This preliminary examples are showing how an STO network could be used for noise filtering and object segmentation.

4 Application of the results

The examples shown in the dissertation clearly highlight the main characteristics of the stochastic selection. We can claim that they are generally valid. Because of this, the methods and implementations are also valid, and the cellular, localized selection can be used as a substitution of the global method. Considering this, my method as an example can also help to show, how a non-topographic problem can be transformed into a topographic one, and how it can be solved by topographic algorithms on topographic architectures.

Not only the general method on the virtual machine, but also the implementation on the $Xenon_v3$ architecture can help to solve optimization problems. This way the commonly used (mostly in image processing tasks) CNN chips can also applied in tasks where parameter optimizations and/or state estimations are required. These topographic algorithm are ideal, where complex problems have to be solve with strict time limits and low power consumption. The extremely good low power consumption of the $Xenon_v3$ (under 20mW) could be ideal in mobile application such as robotics, mobile-communication or navigation.

Also not only the implementation can be applied and useful, but the similar implementation of these (or related) algorithms on other devices and architectures can be crucial e.g: implementation on FPGA-s or GPU-s.

The architectures described in the second part of the dissertation could be useful in designing computers with extremely low power consumption. These simple computational models can be extended in cases where the storing essence of the information is the spin and not the charge. The cores of these architectures are feasible even when only a few atoms are used, this makes them as a true alternative for Beyond Moore's low computation. The analytic solution of the differential equation system could help the investigation of spin-torque oscillators. Apart from the theoretical results I have shown through two simple examples how an STO array can be used for simple computational problems. Although these networks are not feasible in these days because of the physical constraints, but I hope in the future I can measure the dynamics of STOs in practice and the development of STO devices it could speed up the process since only the equation system has to be solved instead of simulating a complex differential equation system.

In the dissertation I wanted to show, how obtaining the topographic constraints can modify the algorithm development, and what we can do to avoid these limitations, also how this limits can help to solve non-topographic problems. I hope in this way topographic thinking, and considering two dimensional constraints and the precedence of locality can help in the solution of new problem classes and in their mapping on many-core architectures as well.

5 Acknowledgement

First of all I would like to thank *Miklós Rásonyi* and *Tamás Roska* for their consistent help and guidance in very many ways, for their unbroken enthusiasm and fatherly guidance during my studies.

I thank my older and younger colleagues for their advices and with whom I could discuss all my ideas: *Gábor Tornai, Mihály Radványi, Tamás Fülöp, Tamás Zsedrovits, Miklós Koller, Attila Stubendek, Vilmos Szabó, László Füredi, István Reguly, Csaba Józsa*. It is not so hard to get a Doctoral Degree if you are surrounded with talented, motivated, optimistic, wise people.

I had some really great teachers who should be mentioned here, as well. I am grateful to Professors *Barna Garay* and *László Gerencsér* for their amazing lectures during my PhD studies.

The support of *Péter Pázmány Catholic University* is gratefully acknowledged and I am also thankful to the *University of Leuven* and to the *Politecnico di Torino* where I could spend one-one semester during my studies.

My work would be less without the discussions with *Csaba Rekeczky* and Eutecus Inc who could always assist my attempts and algorithms with practical problems.

I am indebted to *Fernando Corinto*, *Giovanni Pazienza*, *Csaba György* and *Jan D'hooge* for their kind help and advice.

I am especially grateful to *Timi* who has encouraged me and believed in me all the time. I am very grateful to my mother and father and to my whole family who always believed in me and supported me in all possible ways.

6 The Author's Publications

[1] **András Horváth** and M. Rasonyi, "Implementation of cellular genetic algorithms on a cnn chip: Simulations and experimental results," *International Journal of Circuit Theory and Applications*, 2012.

[2] **András Horváth** and M. Rasonyi, "Topographic implementation of particle filters on cellular processor arrays," *Elsevier Signal Processing (Submitted)*, 2012.

[3] **András Horváth**, T. Roska, A. Stubendek, D. Voils, F. Corinto, G. Csaba, W. Porod, T. Shibata, D. Hammerstrom, and G. Bourianoff, "O-cnn vice spin torque oscillator cells and cellular spin-wave interactions in an associative memory," *IEEE Transactions on Nanotechnology*, 2012.

[4] E. An, **András Horváth**, and L. Dirk, "3d cardiac strain estimation using spatio-temporal elastic registration : in-vivo application," in *IEEE International Ultrasonics Symposium*, (Beijing,China), IEEE, November 2008.

[5] **András Horváth** and M. Rasonyi, "Fast computation of particle filters on processor arrays," in *12th IEEE international workshop on cellular nanoscale networks and applications*, (Berkeley, California), IEEE, February 2010.

[6] **András Horváth**, F. Corinto, G. Csaba, W. Porod, and T. Roska, "Synchronization in cellular spin torque oscillator arrays," in *13th IEEE international workshop on cellular nanoscale networks and applications*, (Turin, Italy), IEEE, August 2012.

[7] G. Csaba, M. Pufall, D. Nikonov, G. Bourianoff, **András Horváth**, T. Roska, and W. Porod, "Spin torque oscillator models for applications in associative memories," in *13th IEEE international workshop on cellular nanoscale networks and applications*, (Turin, Italy), IEEE, August 2012.

[8] T. Roska, **András Horváth**, A. Stubendek, F. Corinto, G. Csaba, W. Porod, T. Shibata, and G. Bourianoff, "An associative memory with oscillatory cnn arrays using spin-torque oscillator cells and spin-wave interactions – architecture and end-to-end simulator," in *13th*

IEEE international workshop on cellular nanoscale networks and applications, (Turin, Italy), IEEE, August 2012.

[9] **András Horváth** and M. Rasonyi, “Maximum likelihood estimation of quantized gaussian autoregressive processes using particle filters with resampling,” in *International Symposium on Nonlinear Theory and its Applications*, (Palma di Majorca, Spain), October 2012.