MULTI-FOVEA ARCHITECTURE AND ALGORITHMS BASED ON CELLULAR MANY-CORE PROCESSOR ARRAYS

Thesis of the Ph.D. Dissertation

Balázs Gergely Soós

Scientific adviser:
Csaba Rekeczky, Ph.D.

Péter Pázmány Catholic University
Faculty of Information Technology
Multidisciplinary Technical Sciences Doctoral School

Budapest 2010

1. Introduction and Aims

Intel produced the 8080 processor chip in 1974, which consisted of approximately 4500 transistors. The revolutionary development of silicon-based manufacturing technology led to the possibility to produce integrated circuits (IC) using several billion transistors in a single chip [8]. Such a large amount of basic elements opens the way to commercialize chips with thousands of connected – but independently controlled – parallel computation components, called processor cores. From the aspect of computer science it raises a new challenge to formulate the theory of many-core computing: the structure of the communication network to support maximal data transfer rate between computing nodes to keep them busy with new inputs to process and to transfer outputs to their destination. New algorithms are needed to maximally exploit these novel hardware capabilities, and new methods are required to split problems into parts that can be evaluated in parallel. The optimization of data-communication time enforces the precedence of locality and the utilization of cellular structures [9].

The rapid development in material science and semiconductor technology and processor enhancements lead to the greatly increased performance, falling prices and widespread applications. Digital cameras with 10 Mega Pixels have become commodity. High-end video cameras are capable of capturing sequences of up to many thousand frames per second and sensors operating in the infrared domain can form images in darkness. Medical diagnostics deeply rely on imaging, for example x-ray or ultrasound, whereas physics boosted by computer science offers further methods like magnetic resonance imaging (MRI) or positron emission tomography (PET). The acquired data set has implicit data parallelism in 2D or 3D topology.

Most common everyday computer programs like word processors or internet applications do not take advantage of multi-core processing since their tasks cannot be easily split into parallel segments. Processor vendors have so far been competing in increasing the operation frequency of their serial-execution systems. Lately, however, the spread of digital multimedia (pictures, videos and music) introduced new kind of data and new kind of processing task to personal computers. The remarkable degree of structure within this type of data opened up the necessity to split up the data for parallel processing using multiple arithmetic cores and later produce the output after joining the partial results. This commercial motivation prompted large investments in multi- and many-core technology.

The need for scientific modeling of parallel processing is obvious concerning the huge variable space dimension and data size. Numerous important phenomena can be modeled via joint dynamic systems [10], which in turn raises the need to extend the classical algorithm definition originally formulated for integer numbers. The underlying cellular structure of broad classes of problems calls for the design of topologically connected many-core processor arrays and the algorithmic approaches could rely on the results of the cellular wave computing theory [11][12].

Sensory systems of animals collect an extremely large amount of information from the surrounding environment. One of the key tasks is to extract interesting portion of the wide data-flow to support decisions for modulating actual action or triggering new ones. The visual pathway of mammals shows strong convergence from sensory to cortical levels. An extremely parallel processing takes place in this network of neural cells to deal with the input signal generated by millions of sensory cells concurrently. Spatial representation of the scene is not homogenous: the so-called fovea – a densely represented part of the retina – can be focused to interesting regions of the scene by fast movements of the eye-ball.

The obvious efficacy of biological vision systems may motivate engineers to mimic two major characteristics (parallelism and foveal processing) during the design of artificial solutions that are embedded into real world-environments. When many salient regions arise after preprocessing, all of these should be better explored. This is the multi-fovea model. Successful works (for example [13]) highlighted the importance and efficiency of this design concept that called for a unified algorithm description.

Surveillance based on moving cameras is an important application field of image processing results. For instance Mini Unmanned Aerial Vehicles (m-UAVs) support the inspection of large open fields for security reasons, traffic flow design. Furthermore, overview images can help in rescue missions in case of natural disasters like fire or flood. Tele-operation requires all-time human control that can be tedious thus ineffective. An alternative possibility is to create autonomous planes to acquire images from the most significant locations. In order to do so, an onboard computational system is required to alter the predefined path before passing over critical events on the ground. This visual navigation system must be designed considering the hard time and power consumption constraints.

2. Research Goals, Results

In my work I primarily focused on data-parallel interpretation and processing of topological problems, in particular image processing tasks, by using cellular processing structures within the Multi-Fovea Architecture and Algorithmic Framework. The research question from the theoretical point of view was to create a unified software model that supports the selection of parallel architecture and also covers hardware specific details for a given device. The particular practical application inspiring the theoretical work was to analyze algorithms for visual navigation systems applicable in mini / micro unmanned air vehicles (UAVs).

The elaborated virtual platform consists of separate processor arrays specialized for parallel execution of preprocessing and foveal computation. The proposed heterogeneous structure can fit the special characteristic of various operators processing the highly parallel data input. I gave a design guideline for Multi-Fovea

Architecture and presented it by the comparison of 2D registration methods for ground object motion detection from mini unmanned aerial vehicles. After giving an analytic comparison of registration methods, I proposed a novel method exploring the proposed architecture by running a larger percent of the task in parallel and in cellular structures.

3. Methods Used in the Experiments

My research was motivated by the most recent neurobiological results in retina-modeling and other neuromorphic engineering solutions [14][15] along with psychophysical experiments.

Backgrounds for the proposed architecture are collected from projective geometry, image processing, topologic cellular operators and algorithms [16], parallel computing and graph theory.

The preference of local communication (cellular structures) both at the virtual and physical level is a fundamental part of the model [12].

The notation used for describing the algorithms of the thesis within the mathematical framework is a directed acyclic graph (DAG), which is widely used for scheduling problems. This description may be considered as a generalization of the Universal Machine on Flows (UMF) diagram [17] that was specially designed for cellular neural networks (CNN) algorithms executed on SIMD/MIMD type many-core processors built in highly regular topology.

After setting up the theoretical framework for complexity evaluation, I modeled relevant state of the art algorithms. To test quantitative quality of the algorithms I made a software framework in PC environment. Furthermore, I made measurements using many-core hardware configurations such as the ACE16K chip [18], the EyeRis chip [19] and the Nvidia GeForce 8800 platform [20], using their specific development tools and programming languages.

Within the framework of the ALFA project [9] I participated in the field experimental series, using a small UAV that was flying above the airport of Gödöllő (small town near Budapest), thus I could also use real video sequences in the algorithm development and testing beside the ones rendered via 3D simulation.

The comparative analysis was performed in the Matlab / Simulink programming environments [9][21]. In addition, some modules were implemented in the C/C++ language, and some reference implementations were also used from third party sources.

4. New Scientific Results

I. Algorithms dealing with direct topographic sensory inputs may contain large number of steps suitable for data-parallel execution due to the natural structure of the data. Based on this observation, I have worked out a novel virtual hardware architecture model (Multi-Fovea Architecture) enabling communication-effective decomposition of those family of image processing algorithms that are convergent, starting from direct sensory input and can be described by acyclic dataflow graphs. The proposed structure effectively evaluates

algorithms consisting of operators with different radii of coupling and topology, and nonhomogoneous spatial coverage by using three specific processor arrays. This heterogeneous structure fits the family of algorithms better than the general homogenous parallel structures without losing general programmability.

Published in [1][5]

Processors built to deal with integers or floating point numbers are calculating operators on a small set of inputs (two in most cases). They incorporate a small amount of dedicated memory – the so-called registers – for storing only a few numbers that can be used as operands. The great majority of the data is stored in an external memory chip. The programs consist of calculation and data transfer instructions. The evaluation of the code may branch depending on some results or it can run in iterations (loops).

The computation deals with neighboring data elements frequently; this property is called locality. Locality is also characteristic for instructions: commands of a subroutine can be aligned in memory resulting sequences and a minimal number of branching instructions. The off chip communication to load new data has massive latency – typically two or three order of magnitude higher compared to numerical calculation – thus caching into on-chip memory is essential to exploit locality.

The principal fraction of computer problems is formulated in serial form, although the intermediate results are not used immediately and the operand of a binary operator can be evaluated in an arbitrary order (instruction level parallelism). To permit the optimizing, the pre-fetching of instructions is necessary. Processors in modern PCs contain more execution lines; moreover, some independent instructions may be reordered and processed while waiting for memory access (out-of-order execution). In the case of handling branches in the program, all paths should be analyzed or some heuristic method is needed to make a decision. The sweep of vector instruction sets and wide registers for handling structured data have further increased the complexity of the already extremely complex processor structure.

Handling the extreme workload of numerous problems on a single processor is not fast enough to be beneficial at all. Nobody would wait hours for the answer to a simple train-schedule query. For these cases it is important to identify tasks of the problem that can be executed in parallel without altering the final result, and distribute them to multiple processors.

All devices should be supplied with task and data and in addition, a communication network should be built to connect them for enabling exchange of partial results and synchronization. Parallel computing raises new challenges: multiple access to common resources (most important ones are the memory and the communication network) should be granted while maintaining (cache) coherency. The further increase of the operation frequency is limited by physical constraints. On the other hand, the placement of many cores on a chip is feasible, thus exploiting parallelism efficiently is essential.

Imperative programming languages handle large data stacks by processing elements in loops. The multimedia inputs offer a great level of data parallelism. Each element should be handled the same way; in most cases, calculations depend only on neighboring structures. These functions are called cellular operators. Loops implementing such operators may be evaluated using many execution contexts. Due to the known locality of the given operators, the large amount of automatically handled cache memory can be replaced with a small amount of local memory circuits dedicated to a given arithmetic unit loaded with direct instructions. The most extensively used imperative language is C/C++. OpenMP [22] is an extension for shared memory multi-threading that can create parallel code for loops to be executed on multi-core processors using the explicit locality constraints given by the programmer.

Parallel data container structures without constraints for traversal but having explicit notion of locality offer great potential for compile-time optimization (RapidMind [23], Intel Threading Building Blocks [24]). In the case of serial description of a program, the dependencies between operators should be scouted by the compiler to enable parallelization. It is worth to ask the programmer to explicitly express producer-consumer locality by drawing dataflow diagrams.

Custom application-specific integrated circuits (ASIC) that were directly designed to boost performance of specific tasks can exploit all parallelism in a given algorithm, in exchange for completely losing software programmability. For example, encoder chips for multimedia compression are common in handheld devices. Homogenous structures of general purpose processors offer a programmable and more scalable solution. However, it is worthwhile to design hybrid hardware structure to fit specific algorithmic classes.

I have designed a virtual heterogeneous many-core architecture (Fig. 1.) for image processing algorithms that are convergent, starting from direct sensory input and can be described by acyclic dataflow graphs. Convergence is referring to the extraction of compact information from inputs represented with topological maps with smaller resolution, image parts, or scalar values. This property calls for heterogeneous processor structures. In applications, where the high frame-rate is important (e.g. 10.000 fps), sensor pixels can be built in the processing topology to eliminate the need of wide and/or ultrafast cross chip communication circuits. If the program can be transformed into a representation containing iterations and recursions only at operator level an acyclic dataflow graph can be created. In this case, program execution can be mapped to many cooperative processors requiring clean and pre-calculated synchronizations. The virtual architecture hides the details of the physical hardware from the software engineering point of view, but models the scheduling and communication latencies.

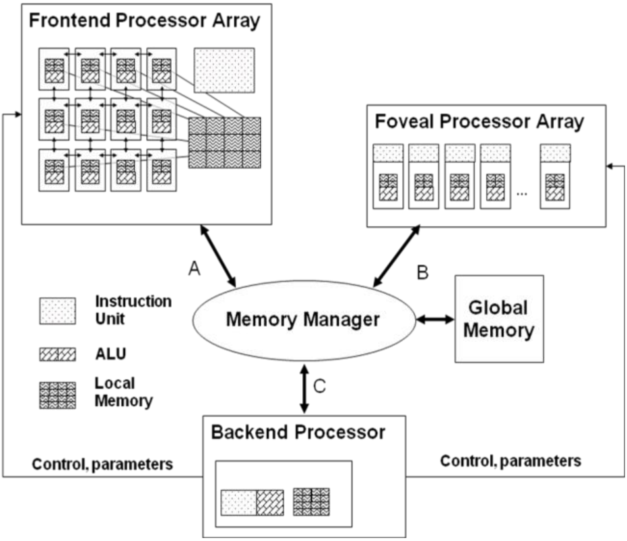

Figure 1. **Functional components of the Multi-Fovea Architecture**
the cellular topologic processor array for full frames: Frontend Processor Array
the window-based Foveal Processor Array
the serial processor: Backend Processor.
The processor arrays are composed of many computing cores (arithmetic and logic units, ALU) controlled by Instruction Units. A small amount of fast access storage – the Local Memory – is placed inside the arrays, while the system contains a massive storage – the Global Memory – with substantial capacity. Window extraction from frames for foveal processing and global memory access is handled by the intelligent Memory Manager Unit via the A, B and C communication lines. Coordination and syncronization of the arrays is done by the serial Backend Processor. The Backend Processor is also responsible for merging the partial results supported by the parallel array of foveal processors and to give final classification results.

The proposed virtual architecture contains a dedicated cellular processor array to take advantage of cellular locality and data parallelism which is characteristic of the family of cellular image processing operators [25]. This is designed to handle full images through space invariant operators without data-dependent branching, thus it may be controlled by a single instruction unit in SIMD fashion.

For the majority of image processing algorithms the final goal is to highlight specific regions (segmentation and classification), or to detect and possibly identify objects and/or events. This means that after some topological steps they focus on selected portions of the input image flow. The thorough analysis of chosen windows is supported by another dedicated processor array, the Foveal Processor Array. The arithmetic units (or small groups) in this array are controlled by independent instruction units to support data dependent branching. The communication and synchronization between the foveas is indirect. The generated results are either window sized topological data or descriptor vectors.

The non-parallelizable tasks and the evaluation of descriptor vectors are executed in a serial processor, the Backend Processor. It is also responsible for synchronizing the other arrays. The data transfer (sensor sized or scaled images, windows, and scalars) is supported by an intelligent multi-port Memory Management unit.

The Multi-Fovea Architecture is designed to hide a specific physical implementation. Execution time can be assigned to data transfers and global image processing operators, thus different algorithms for the same problem can be compared. The most important issues from the hardware implementation aspect are (i) used global operators (with a minimal instruction set for the core operators, explicit locality and topology description), (ii) the separation of program code segments with and without branching (iii) continuous data block definitions for data transfer.

One of the most powerful universal parallel technologies in PC environment is the General Purpose Graphic Processing Unit (GP-GPU) based Compute Unified Device Architecture (CUDA) from Nvidia [22]. AMD-ATI also offers GP-GPU computing, although their software support is more focused on graphical applications at the present stage.

The fovea-based parallel array and the serial processor are common elements in both, thus the uniqueness of the latter is the dedicated cellular unit for evaluating 2D topologic operators that are working on image parts in a small connection radius. In the preprocessing stage most of the image processing algorithms need such operators (for example, convolution and image morphology) transforming the whole input image with each of them working on a neighborhood of a few pixels with full-grain connectivity, thus the parallel evolution requires local communication on a massive scale. I have shown that the GP-GPU array may implement the functions of the Frontend Preprocessor Array, thus this architecture may also covered by the virtual architecture.

II. I have shown that the proposed virtual architecture can uniformly cover important problems in sensory image processing. As a proof of concept, 2D registration based ground motion detection methods for mini Unmanned Aerial Vehicles (UAV) were presented. I have given models for the state-of-the-art solutions by creating the data-flow diagrams, mapped the operators to processing structures and compared the algorithms within a unified simulation environment. I have given performance analysis concerning computational complexity, registration quality, detection robustness, and parameter sensitivity of the algorithms. Based on the detailed analysis of all methods I have proposed specific modifications leading to significant improvements, which will be summarized in the corresponding two sub-theses.

Published in: [4]

2D image registration techniques can be applied to data from Mini UAVs operating at the altitude of 80-100m flying over a flat inspection area to detect ground motion [26]. Consecutive projections of a flat screen captured from different locations can be aligned to a common coordinate frame. Changes may occur due to moving objects on the ground.

Salient points in the given image respecting a robust metric – for example *corner* at the intersection of edges – are called Feature points (or Point features). Point feature based alignment is one of the mainstream solutions for 2D image registration.

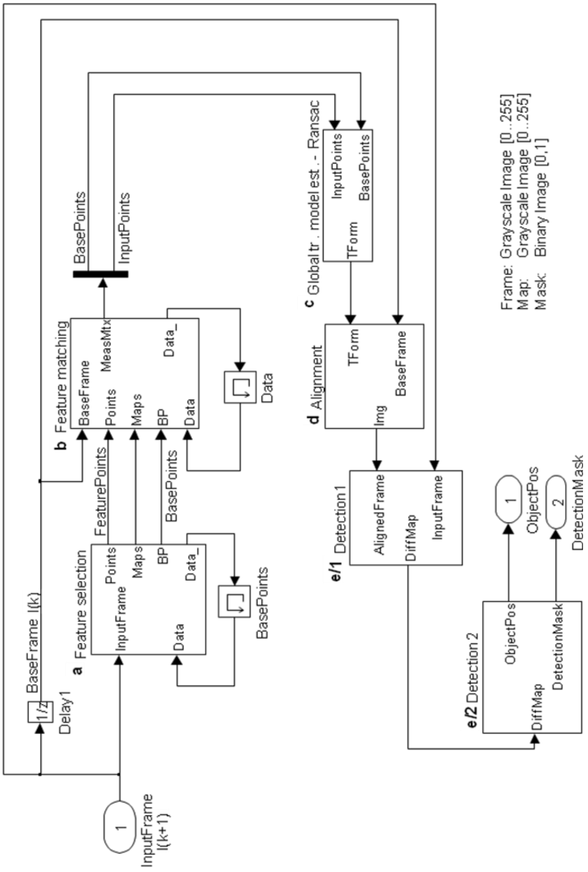

Figure 2. **General Flowchart of Global Registration Based Ground –Motion Detection Algorithms**

Feature points (Keypoints) are extracted from the InputFrame (a) and stored for matching. Using the next input image, pairs are assigned to all feature points (b), and based on the point correspondances a global transformation is estimated between the two consecutive images (c). All pixels from the previous image are then aligned to the new coordinate frame (d), and a grayscale detection mask is calculated describing the changes on the screen (e/1). The final detection blobs are derived from the detection map after binarization (e/2).

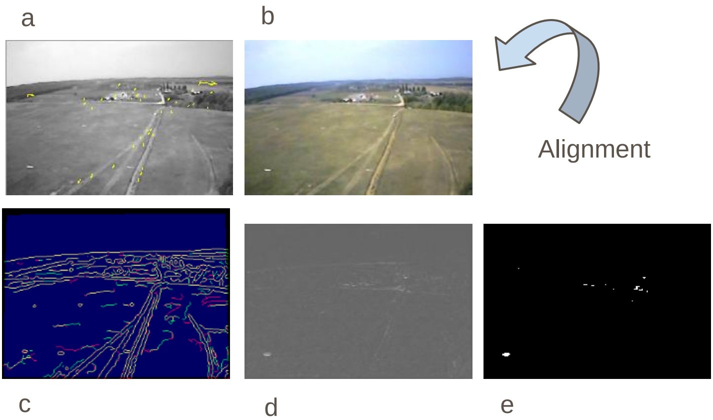

Figure 3. **Intermediate Results for the Global Registration Based Ground – Motion Detection Algorithms:**
matched feature point pairs diplayed over the previous frame (a); the InputFrame (b); graycale detection result (d); binary result with white blobs for moving objects (e). The overlayed edge-images (c) demonstrate the quality of the alignment.

Besides the thoroughly discussed Harris Corner [27] (used by the Corner Pairing Algorithm, CPA) and Kanade-Lucas Tracker – KLT [28][29] the most cited point-feature extractor is the Scale Invariant Feature Transform – SIFT [30]. Video compression pushes for the improvement of block matching techniques [31] (BMA) as well.

Robust registration [32] and motion detection methods based on these representative groups of algorithms were covered in my assessment. Many surveys give comparative results for registration quality [33] [34], although none of them include hardware complexity factors in the evaluation metrics. The general flowchart of the diagram is given in Figure 2. and the intermediate results of the corresponding processing are presented in Figure 3.

I have evaluated the algorithms on synthetic and many real-world video sequences for ground object motion detection and presented the design trade offs. In case of large-field inspection the common flight path follows a rectangular shape with long straight edges and short rotating maneuvers. The sequences were taken from the straight portions of flight videos. The simulated sequence consists of sharp still images rendered from different calculated locations; on the contrary, the real shots are slightly distorted by the motion-blur effect and video compression artifacts. All sequences use 320x240 pixel resolution and were taken at 20 frames per second. The maximal measured displacement was 12 pixels in any direction. Higher sampling rate with high-sensitivity sensors would be desired to keep the average displacement in the 1-2 pixel range.

All algorithms used conceptually identical registration and detection steps.

The complexity of the CPA, BMA and KLT algorithms may be parameterized using two factors: (i) the number of foveal regions and (ii) the size of template images used for matching. The preprocessing phase is non-tunable. On the other hand, the SIFT algorithms can be described via parameters of preprocessing.

The detection robustness of the algorithms was compared after selecting the optimal parameters. Table 1 shows that their capabilities are similar even though they consume computational complexity in a quite different range. If the mission of the UAV is not only to detect the presence of moving object, but also to localize and identify them, the considerably high complexity of KLT and SIFT are justified since the results may be efficiently reused in further processing.

The registration capabilities for frame-to-frame alignment were compared in the high spatial frequency domain via the overlapping ratio of the binary edge images. In the case of high precision estimation, the consecutive transformation matrices may be accumulated and longer series of images can be transformed to a common coordinate frame. This advanced capability can be used to detect objects at relatively low motion speeds compared to the image sampling frequency.

The different algorithms are optimal for different hardware setups. SIFT is outstanding in quality, but it requires a very complex preprocessor and foveal arrays to be effective. On the other hand, the calculated description vectors may also be used for object identification. KLT involves fovea intensive calculations using branches and fractional number representation; although the preprocessing computation is less intensive compared to the one in the case of SIFT. Both methods are characterized with high registration quality.

Block Matching methods do not need a preprocessing array (although their performance could be enhanced by preprocessing). The different strategies of Diamond Search and Full Search can balance complexity and registration quality. The latter incorporates high number of searching steps (n) and finds global optima. It does not need branching, thus it can be implemented in less complex hardware; on the other hand, the first method converges to a local optimum rapidly ($\sqrt{n}$).

The Corner Pairing Algorithm offers lower registration quality; its small computation complexity, however, is remarkable for detection purposes.

Based on the detailed analysis of the algorithms, I have proposed specific modifications leading to significant improvements which will be summarized in the corresponding three sub-theses.

II.1. I have shown that up-scaling is unnecessary for scale-space based point feature detection in the sequences captured by a mini UAV flying at medium altitude. This means that processing does not need to be more fine-grained than the original sensor resolution. Furthermore, I have experimentally validated that the visible zoom factor due to altitude changes between consecutive frames is small (less than one percent) and as a result, evaluating more than two inter-octave scales does not give any benefit in alignment capability.

The SIFT algorithm applies a series of Gaussian filters (low-pass characteristic in space, smoothing the image) and calculates differences between filtered images. These intermediate maps can be used to robustly localize blob-like features with different sizes in the original image. The Gaussian filter with $\sigma=2$ parameter gives an output with an effective resolution of half of the original, meaning an octave in scale-space. Lowe in [30] proposed to start scale-space generation from an interpolated double resolution image (up-scaling) and to calculate three intermediate scales for each octave.

Gaussian filtering can be effectively calculated in a dedicated full-grain cellular processor array [18]. Creating a double resolution array increases complexity at least by a factor of four, and using more intermediate scales implies additional components.

During the evaluation of registration quality I have experimentally shown that adequate feature pairing can be calculated in case of two intermediate scales, and that the number of feature point pairs is large enough to robustly estimate alignment transformation.

The simplified algorithm does not differ from the original regarding its average detection robustness.

II.2. I have shown that the KLT and SIFT algorithms can be merged effectively, with the assumption that direct hardware support for diffusion and extrema localizing operators working in a 3x3 neighborhood exist on the cellular processor array.

The extrema localizing operator working in a 3x3x3 pixel neighborhood is needed for feature point localization in the case of the scale-space approach. This can be calculated using a subroutine in current hardware configurations while direct realization is also possible. Using the simplifications that were described in the previous section, the running time of feature point localization is reduced to the millisecond range.

The most important additional results for average flight paths are the following: (i) it is not important to maintain rotation invariance in local motion estimation level to achieve good registration results; (ii) the fine sub-pixel based matching part of the KLT algorithm is reliable even in case of a small number of point pairs. Therefore, the combination of scale-space based feature extraction and KLT like feature matching offers a good solution.

III. I have proposed a novel independent motion detection algorithm (Elastic Grid Multi-Fovea Detector) that exploits the parallel and cellular capabilities and communication model of the proposed architecture. I have experimentally shown that the model based connected multiple displacement method is adequate for the detection of ground objects moving on an open field from a mini Unmanned Air Vehicle (UAV). I have proven by using a multiple criteria (quality-execution-complexity) metrics that the proposed algorithm offers a better trade-offs for important classes of problems than previous solutions using the same scene assumptions.

Published in [2] [3]

The mainstream methods estimate a single global image transformation (projection) for 2D registration, directly fitting the flat-world assumption. Their estimation process needs complex floating point operations and the transformation itself requires a non-continuous memory access pattern. However, the proposed model-driven multiple displacement estimation can deal with moderate relief variation and operates only by using the continuous coalescing windowing memory access mechanisms.

The scheme of the method is presented in Figure 4. The computation is focused on the foveal processor array. The local partial results converge through iterations, using the results from the neighboring foveas.

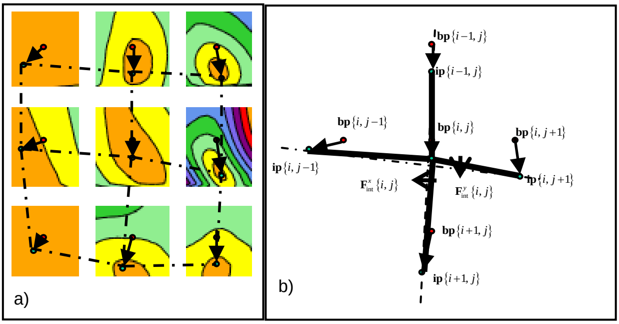

Figure 4. **Elastic Grid Multi-Fovea Detector algorithm**

The extraction of point pairs is done by foveal search in a 2D topology. Small regions – called templates – are extracted around vertices of a regular grid. These templates are matched against corresponding image parts of the new InputImage with a given displacement inside the search radius. Matching values are displayed for all foveas in the same image (a). As a first step the best value is selected from each fovea. The endpoints of the displacement vectors span the elastic grid used for iterative optimization. The grid is deformed by two effects: the matching surface drives all vertices towards local optima, while the joined edges of the grid tries to keep parallel. The transient is controlled by the Backend Processor.

The grayscale and binary detection maps are also calculated inside the foveas using the estimated displacement values.

	Artificial 135/130	Godollo_1 120/79	Godollo_2 300/230	Godollo_3 35/31	Complexity (operation per pixel)
SIFT	130	52	200	29	~1100
KLT	130	52	217	29	~450
BMA	128	61	208	29	~100
CPA	125	55	183	27	~75
ELG	92	52	194	28	~50

Table 1. **True-positive detection results and computational complexity of the algorithms**

Total number of frames in a sequence / the frame number on which target ‘1’ is visible is given for each sequence in the header. Rows contain the number of true-positive detection results based on hand-made references together with the average computational steps normalized with the pixel count of the frames.

I have proven the efficiency of the proposed new algorithm using a simulated sequence and multiple real-life sequences. In case of momentous plane maneuvers, on average the Elastic Grid Multi-Fovea Detector gives similar detection results to the more complex algorithms for usual surveillance flight paths. However, it requires far less computational effort (Table 1).

5. Application of the Results

The Multi-Fovea Architecture is an adequate computational model for small and compact embedded detection-classification systems. In the frame of the VISCUBE project, a multi-foveal chip is under design and prototype production. My algorithmic research fundamentally influenced the decision on what hardware-implemented instruction set is to be used in the first generation of the VISCUBE chip to be manufactured in 3D silicon technology.

The Multi-Fovea Architecture can be implemented using other many-core devices, such as FPGAs. The related design considerations that were discussed in the dissertation are applicable to the parallel implementation of a wide range of video processing algorithms. Numerous integrated circuit and system manufacturers (e.g. IBM, Intel, Nokia and Apple) support the upcoming standard called OpenCL that is extremely similar to CUDA in its concept. The predictable general spread of those platforms will grant application potential to my results.

6. Acknowledgements

First of all I would like to thank my supervisor Csaba Rekeczky for introducing me to scientific work and for his constant support and belief in my research activity. Working with him gave me the opportunity to join several international projects and also to work at Berkeley, USA.

I thank Tamás Roska, head of the Cellular Sensory Wave Computing Laboratory and head of the doctoral school, for the opportunity to be a Ph.D. student and for directing my work with great empathy. Thanks to the Faculty of Information Technology (ITK) at Pázmány Péter Catholic University, and the Computer and Automation Research Institute (SZTAKI) of the Hungarian Academy of Sciences for providing the intellectual and material background for my research.

Special thanks to my host professors at the University Notre Dame (Indiana, USA) and at Johann Wolfgang Goethe Universität (Frankfurt, Germany): Wolfgang Porod and Ronald Tetzlaff.

I would like to express particular thanks to my colleagues at SZTAKI who influenced this work with their attitude to scientific research and with the atmosphere of a cutting edge research institute: Ákos Zarándy, Péter Földesy, László Kék, László Orzó, István Szatmári, András Radványi, and Péter Szolgay.

The cooperation of my closest colleagues György Cserey, Ádám Rák, József Veres, Zsolt Szálka, Dániel Hillier, Gunter Geis and

Alpár Sándor is greatly acknowledged. The work with them greatly contributed to my scientific results.

By working with Tamás Roska and Csaba Rekeczky, I was given the possibility to take part in the education at ITK. Thanks for the help related to my classes to Zsuzsa Vágó and Ágnes Bércesné Novák. I appreciate the work of Vilmos Szabó and András Gelencsér former M.Sc. students, who helped my with many ideas and arguments.

The colleagues and students of the Robotics Laboratory at ITK provided constant technical help during this research. Thanks to György Cserey for the overwhelming support to bring our ideas into reality.

I am thankful to my colleagues at SearchLab Ltd., particularly Zoltán Hornák, and Balázs Kiss. Their comments helped me to augment the quality of the dissertation.

I am thankful to my colleagues at SearchLab Ltd., particularly Zoltán Hornák, and Balázs Kiss. Their comments helped me to augment the quality of the dissertation.

I am especially graceful to my class-mates and other graduate students for their professional and non professional help here and abroad: Éva Bankó, Csaba Benedek, Barnabás Hegyi, Ferenc Lombai, Tamás Harczos, Zsolt Szálka, Gergely Gyimesi, Béla Weiss, Tamás Zeffer, Mária Ercsey-Ravasz, Giovanni Pazienza, Norbert Bérczi, Gergely Feldhoffer, Ákos Tar, Gábor Vásárhelyi, Kristóf Iván, Kristóf Karacs, András Kiss, Frank Gollas and Martin Eichler.

I am grateful to Ildikó Csurgay and Árpád Csurgay who supported me at Notre Dame and to the other Hungarian students: Gergő Szakmány, Edit Varga and Kristóf Tahy.

I appreciate the help of the colleagues at Analogic Ltd. and Eutecus Inc. who guided me in hardware related topics, particularly István Horváth, Tibor Gyimesi and Gábor Erdősi.

I acknowledge the kind help of Anna Csókási, Livia Adorján, Judit Tihanyi, Katinka Vida and the rest of administrative and financial personnel.

I acknowledge the kind help of Anna Csókási, Livia Adorján, Judit Tihanyi, Katinka Vida and the rest of administrative and financial personnel.

The finacial support of Hungarian and USA R&D founds is acknowledged: Hungarian Scientific Research Fund (OTKA) NI61101; National R&D Program GVOP 3.1.1-2004-05-0388/3.0; National R&D Program NKFP ALFA-2/046/04 and TELESENSE-035/02/2001; Multidisciplinary University Research Initiative (MURI) of the Office of Naval Research (ONR, USA) and the VISCUBE 07-09 project, Eutecus Inc. (ONR).

Thanks to all my colleagues at ITK and at SZTAKI.

The loving support of all my family and my girlfriend Zsófi helped me through the hardest moments of this period.

7. Publications

Journal Publication, Book Chapter

[1] Soós, B. G., A. Rák, J. Veres, and G. Cserey, "GPU boosted CNN simulator library for graphical flow based programmability," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2009, p. 11.

[2] Soós, B. G., V. Szabó, and C. Rekeczky, "Elastic Grid Based Multi-Fovea Algorithm for Real-Time Object-Motion Detection in Airborne Surveillance," *C. Bhaatar, W. Porod and T. Roska : Cellular Nanoscale Sensory Wave Computing*, Springer, 2009.

Conference papers, Technical report

[3] B.G. Soós and C. Rekeczky, "Elastic Grid Based Analysis of Motion Field for Object-Motion Detection in Airborne Video Flows," *Circuits and Systems, ISCAS 2007. IEEE International Symposium on*, 2007, pp. 617-620.

[4] B.G. Soós, V. Szabó, and C. Rekeczky, *Multi-Fovea Architecture and Algorithms for Real-Time Object-Motion Detection in Airborne Surveillance*, Budapest, Hungary: Pazmány Peter Catholic University, 2009.

[5] B.G. Soós, A. Rák, J. Veres, and G. Cserey, "GPU powered CNN simulator (SIMCNN) with graphical flow based programmability," *Cellular Neural Networks and Their Applications, CNNA 2008. 11th International Workshop on*, 2008, pp. 163-168.

[6] A. Rák, F. Gergely, B.G. Soós, and G. Cserey, "CPU-GPU hybrid compiling for general purpose: Case studies," *Cellular Neural Networks and Their Applications, CNNA 2010. 12th International Workshop on*, 2010.

Co-authored journal publication

[7] A. Rák, B.G. Soós, and G. Cserey, "Stochastic Bitstream Based CNN and its Implementation of FPGA," *International Journal Circuit Theory and Applications*, vol. published online, Nov. 2008.

8. Related publications

[8] B. Stackhouse, S. Bhimji, C. Bostak, D. Bradley, B. Cherkauer, J. Desai, E. Francom, M. Gowan, P. Gronowski, D. Krueger, C. Morganti, and S. Troyer, "A 65 nm 2-Billion Transistor Quad-Core Itanium Processor," *Solid-State Circuits, IEEE Journal of*, vol. 44, 2009, pp. 18-31.

[9] ITRS, "International Technology Roadmap for Semiconductors," *http://www.itrs.net*, 2009.

[10] A. Turing, "The chemical basis of morphogenesis," *Philosophical Transactions of the Royal Society (part B)*, vol. 237, 1953, pp. 37-72.

[11] L.O. Chua and T. Roska, *Cellular neural networks and visual computing: foundation and applications*, Cambridge University Press, 2002.

[12] T. Roska, "Cellular Wave Computing in Nanoscale via Million Processors," *C. Bhaatar, W. Porod and T. Roska : Cellular Nanoscale Sensory Wave Computing*, Springer, 2009.

[13] C. Rekeczky, I. Szatmari, D. Balya, G. Timar, and A. Zarandy, "Cellular multiadaptive analogic architecture: a computational framework for UAV applications," *Circuits and Systems I: Regular Papers, IEEE Transactions on [Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on]*, vol. 51, 2004, pp. 864-884.

[14] B. Roska and F. Werblin, "Vertical interactions across ten parallel, stacked representations in the mammalian retina," *Nature*, vol. 410, 2001, pp. 583-587.

[15] D. Bálya, B. Roska, T. Roska, and F.S. Werblin, "A CNN framework for modeling parallel processing in a mammalian retina," *International Journal of Circuit Theory and Applications*, vol. 30, 2002, pp. 363-393.

[16] L. Kek, K. Karacs, and T. Roska, *Cellular Wave Computing Library V2.1*, Budapest: Cellular Sensory Wave Computers Laboratory, Computer and Automation Research Institute, Hungarian Academy of Science, 2007.

[17] T. Roska, "Computational and computer complexity of analogic cellular wave computers," *Cellular Neural Networks and Their Applications, 2002. (CNNA 2002). Proceedings of the 2002 7th IEEE International Workshop on*, 2002, pp. 323-338.

[18] A. Rodriguez-Vazquez, G. Linan-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona-Galan, F. Jimenez-Garrido, R. Dominguez-Castro, and S. Meana, "ACE16k the third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, 2004, pp. 851-863.

[19] AnaFocus, "eye-RIS 1.1 Leaflet of AnaFocus Ltd," 2007.

[20] NVIDIA, *GeForce 8800 GPU Architecture Overview*, NVIDIA Corp., 2006.

[21] MathWorks, "Simulink - Simulation and Model-Based Design," *http://www.mathworks.com/products/simulink/*, 2008.

[22] R. Chandra, R. Menon, L. Dagum, D. Kohr, D. Maydan, and J. McDonald, *Parallel Programming in OpenMP*, Morgan Kaufman, 2000.

[23] Rapidmind, "RAPIDMIND," *http://www.rapidmind.net/*.

[24] "Intel Performance Libraries," 2008.

[25] A. Zarándy and C. Rekeczky, "Many-core Processing Array Architecture Selection Guide for Topologic Problems," *C. Bhaatar, W. Porod and T. Roska : Cellular Nanoscale Sensory Wave Computing*, Springer, 2009.

[26] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, Hai Tao, Yanlin Guo, K. Hanna, A. Pope, R. Wildes, D.A.-.H. Hirvonen, M.A.-.H. Hansen, and P.A.-.B. Burt, "Aerial video surveillance and exploitation," *Proceedings of the IEEE*, vol. 89, 2001, pp. 1518-1539.

[27] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings Fourth Alvey Vision Conference*, Manchester, UK: 1988, pp. 147--151.

[28] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *International Joint Conference on Artificial Intelligence*, Vancouver: 1981, pp. 674-679.

[29] J. Shi and C. Tomasi, "Good features to track," *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR*

'94., 1994 IEEE Computer Society Conference on, 1994, pp. 593-600.

[30] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, Nov. 2004, pp. 91-110.

[31] S. Zhu and K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *Image Processing, IEEE Transactions on*, vol. 9, 2000, pp. 287-290.

[32] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2004.

[33] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, 2005, pp. 1615-1630.

[34] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, Oct. 2003, pp. 977-1000.