# Implementation of medical imaging algorithms on kiloprocessor architectures


Gábor János Tornai

*Theses of the Ph.D dissertation*

Pázmány Péter Catholic University
Faculty of Information Technology and Bionics

Scientific advisor:
Dr. György Cserey Ph.D.
Consulent:
Dr. Tamás Roska DSc

Budapest, 2014

# 1 Introduction

Medical imaging analysis helps doctors in a wide spectrum starting from diagnosis formation to monitoring the therapy. It is possible to discover an abnormal change in a given region, to quantify the possibility of different cancer types in specific tissues. These methods work correctly thanks to the sophisticated algorithms and the development of the semiconductor devices and technology makes them realizable in feasible time. One of the most eye catching example is minimally invasive surgery, where the real trajectory of the interventional device is compared to the reference on-line.

The mentioned new devices are the many core architectures spreading and developing in the last few years. The former classical architectures with one processor core changed and more processor cores are placed on the same chip. This increase in core count is true for central processing units (CPU), graphical processing units (GPU), some application specific integrated circuits (ASIC), and field programmable gate arrays (FPGA) as well.

Each of these devices have several fundamental properties of the following like huge arithmetic performance, huge bandwidth, low power dissipation, or small chip area. However, to be able to benefit from these advantages, new algorithms, and new optimization methods shall be considered for the set of tasks to be solved. These tasks may include completely new ones or older ones considered unfeasible. In my dissertation I present two tasks and their solutions mapped on given many core devices.

The first task is the rendering of digitally reconstructed radiographs (DRR) that is a key step to several image guided therapy (IGT) applications. I am focusing on the alignment of 3D CT images taken before the intervention and 2D images taken during an intervention. This alignment procedure is called 2D to 3D registration [1]. DRRs are synthetic X-ray images. A value of a pixel of a DRR is the line integral of the CT scan along a ray emanating from a virtual source to the location of the very pixel. As it is known from the literature, the DRR rendering is the most time consuming task during registration [1, 2, 3] so it is essential the minimize the rendering time of DRRs.

The second task is to achieve as much speed-up as possible on level set (LS) based methods since they have vast applications from computational geometry through crystal growth modeling to computer vision [4, 5, 6]. The method entails that one evolves a curve, surface, mani-

fold, or image with a partial differential equation (PDE) and obtains the result at a point in the evolution. There is a subset of problems where only the steady state of the LS evolution is of practical interest like segmentation, (shape) modeling and detection. The LS method of Shi [7] gives a way to solve these tasks quickly. I have worked on determining the number of iterations and map the method of Shi on CNN-UM and GPU.

# 2 Materials and methods

DRR rendering is realized in CUDA C of Nvidia. I examined several optimization rules and parameters to be able to fit the task on a given hardware. I have made experiments on GPUs based on different architecture generations (8800 GT, 280 GTX, Tesla C2050, 570 GTX, 580 GTX). Furthermore, two different compiler and driver combinations have been used (3.2 compiler + 260.16.21 driver, and 5.5 compiler + 331.67 driver). Two different datasets have been utilized. The first is a CT scan made from a radiological torso phantom (Radiology Support Devices, Newport Beach, CA, model RS-330) with resolution $512 \times 512 \times 72$, the other is a scan taken from a pig head with resolution $512 \times 512 \times 825$ from an annotated database [8]. The phantom imitates the attenuation of human tissue like lungs, bone, arteries, etc. In the X-ray spectrum I have made measurements on complete DRRs and randomly sampled ones as well. The parameters of the rendering have been set to values that are relevant in the case of minimally invasive surgeries (region of interest, ROI and sampling rate).

During my work connected to LS methods I was required to understand the hyperbolic conservation laws as well as the concept of viscosity solutions from the field of partial differential equations (PDE). The viscosity solution is defined as the solution of the following PDE $G(u)u_x + u_t = \epsilon u_{xx}$, subject to $\epsilon$ tends to 0. The theory of LS methods and the underlying equations are essential to understand for specific tasks like segmentation and curve motion. Additionally I required the basic notions of discrete topology and convex sets to be able to construct the proofs of my theorems. Execution time measurements were done on Eye-RIS v1.3 vision system (VS) and on Nvidia 780 GTX GPU.

# 3 New scientific results

Thesis 1. *I formed a rule-set (1)-(4) allowing the rendering of DRRs to be performed efficiently on Nvidia GPUs. This step is responsible for the slowness of 2D to 3D registration. I applied the rule-set on the calculation of randomly directed line integrals for DRR rendering and systematically searched the block size parameter in the theoretically possible range. According to my findings the value of block size for efficient rendering is in the range of 8-16 threads in a block unlike the theoretical suggestions. So the 2D to 3D registration can be performed in real time for surgical need depending on the application in 0.5-10 frames per second. I showed that DRR rendering can be performed in 0.2-2.2 ms in the case of a region of interest (ROI) containing fully a lumbar vertebra (16×9 $cm^2$, 400×225 resolution).*

1. *Slow 'if else' branches shall be replaced with ternary expressions if possible that are compiled to selection 'parallel thread execution' (PTX) instructions that are faster than any kind of branching PTX instructions.*

2. *Data that is read locally and in an uncoalesced way shall be placed in texture memory provided it is not written.*

3. *Avoid division if possible and use the less precise, faster type (div.approx, dif.full instead of div.rnd).*

4. *If the denominator is used multiple times calculate inverse value and multiply with it.*

I presented measurements on randomly sampled DRRs executed on GPU first [9]. The effectiveness of the first and second optimization rules are presented in Table 1. The cumulative effect of the third and the fourth rules as a function of the block size is presented in Figure 1.

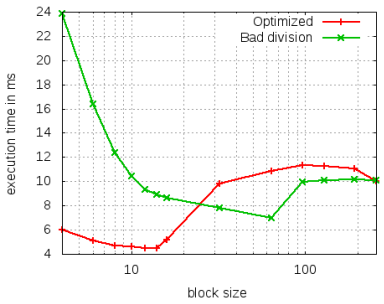
The missing branching optimization resulted in a $6 - 11\%$ performance decrease, 8% in average on Tesla C2050 GPU while $6 - 13\%$ decrease on 570 GTX GPU, if the optimized version is considered 100%. The linear memory caused a 1.75-2.4 times slowdown consequently on both GPUs.

The optimal block size in the case of the optimized kernel is always in the range of 8-16 threads in a thread block. This property was tested in a former version of compiler and driver as well as on four different top

Table 1: The effect of the lack the of first two optimization rules on the execution time. The lack of branch optimization caused 6-13% slowdown while the use of linear memory caused 2 times slowdown in average. Execution times are measured in $\mu s$.

| | Tesla c2050 | | | 570 GTX | | |
|---|---|---|---|---|---|---|
| number of threads | $t_{opt}$ | $t_{branch}$ | $t_{linear}$ | $t_{opt}$ | $t_{branch}$ | $t_{linear}$ |
| 1024 | 234 | 258 | 553 | 181 | 206 | 408 |
| 1536 | 319 | 339 | 639 | 263 | 295 | 462 |
| 2048 | 466 | 502 | 1094 | 358 | 403 | 656 |
| 3072 | 648 | 689 | 1275 | 572 | 617 | 1101 |
| 4096 | 969 | 1082 | 1935 | 693 | 742 | 1310 |
| full DRR | 2666 | 2763 | 5278 | 2259 | 2375 | 5221 |


(a) 1536 threads




(b) 20480 threads

Figure 1: The cumulative effect of third and fourth rules on the execution time as a function of block size.

GPUs (8800 GT, 280 GTX, Tesla C2050, 580 GTX). The characteristics of the optimized kernel were similar in this software environment too.

Publications connected to this thesis group: [I]. The thesis claim is specified and paraphrased in details in the second chapter of my dissertation.

Thesis group 2. I present bounds on the required number of iterations of the LS method of Shi [7] and this bound depends only on the initial condition. I propose an initial condition family that decreases the bound in a flexible and effective way. Additionally, evolutions started from this initial condition family require drastically reduced time to converge.

*Thesis 2.1 I discovered two new theorems, one for a general case and another for a convex case to determine the worst case required number of iterations of the Shi LS method to converge to the solution. These bounds depend only on the initial condition. I developed proofs for both cases and supported the bounds with experiments. The results are utilized in thesis claim 2.2.*

Let us consider a subset of $\mathbb{Z}^n$, say $D$. A point $\mathbf{x} \in D$ is characterized by its coordinates ($\mathbf{x} = (x_1, ..x_k)$). A *path* $p$ between $\mathbf{x}$ and $\mathbf{y}$ is a sequence of points $\mathbf{x}_l (l = 0, 1, ..., L) \in D$ subject to $\mathbf{x}_l \in N(\mathbf{x}_{l+1})$ and $\mathbf{x} = \mathbf{x}_0$ and $\mathbf{y} = \mathbf{x}_L$. A set of points $\mathcal{A}$ forms a *connected region* if and only if there exists a path $p$ between every $\mathbf{x}, \mathbf{y} \in \mathcal{A}$ subject to $\forall \mathbf{x}_l \in p$ is an element of $\mathcal{A}$. A *minimum path* $p_{min}$ is the shortest path meaning there are no shorter $p'$ paths between $\mathbf{x}$ and $\mathbf{y}$. Minimum path is usually not unique and can depend on the chosen discrete neighborhood. The diameter $B$ of a connected region is the longest minimum path having at least its endpoints within the connected region. A connected region is considered as convex if all minimal paths are minimum paths at the same time.

**Theorem 1** (general bound)**.** *Let the true object region be denoted by* $\Omega^*$ *and let it be composed of* $P$ *connected regions* $\Omega^*_p$ *(where* $p = 1...P$*). Similarly the true background region be denoted by* $\Gamma^*$ *and let it be composed of* $q$ *connected regions* $\Gamma^*_q$ *(where* $q = 1...Q$*). Assume that* $F > 0$ *in* $\Omega^*$ *and* $F < 0$ *in* $\Gamma^*$*. At initialization,* $C$ *is chosen such that* $\Omega = \cup_i \Omega_i$, $\Gamma = \cup_j \Gamma_j$ *and* $\Omega^*_p \cap \Omega \neq \emptyset$, $\forall p = 1...P$ *and* $(D \setminus \Omega) \cap \Gamma^*_q \neq \emptyset$, $\forall q = 1...Q$*. Then, the Shi LSM converges to* $\Omega^*$ *in* $N_{it} \leq \max(\max_i(|\Omega_i|), \max_j(|\Gamma_j|))$ *iterations, where* $|.|$ *denotes the number of elements in the region.*

**Theorem 2** (convex bound)**.** *Let the true object region* $\Omega^*$ *be composed of* $P$ *connected regions* $\Omega^*_p$ *(where* $p = 1...P$*) and the true background*

(a) concave object (b) convex object

Figure 2: Two extremities of the object types. If the initial condition section only one pixel from (a) the labyrinth, the requiring number of iterations is equal to the number of pixels within the labyrinth minus one. In the case of (b) the circle, the required number of iterations are not more than the diameter of the circle in pixels.

*region* $\Gamma^*$ *be composed of* $q$ *connected regions* $\Gamma^*_q$ *(where* $q = 1...Q$*). Assume that* $F > 0$ *in* $\Omega^*$ *and* $F < 0$ *in* $\Gamma^*$*. At initialization,* $C$ *is chosen such that* $\Omega = \cup_i \Omega_i$*,* $\Gamma = \cup_j \Gamma_j$ *and* $\Omega^*_p \cap \Omega \neq \emptyset$*,* $\forall p = 1...P$ *and* $(D \setminus \Omega) \cap \Gamma^*_q \neq \emptyset$*,* $\forall q = 1...Q$*. If either* $\Omega^*$ *or* $\Gamma^*$ *is convex than the Shi LSM converges to* $\Omega^*$ *in* $N_{it} \leq \max(\max_i(B_{\Omega_i}), \max_j(B_{\Gamma_j}))$ *iterations, where* $B$ *denotes the diameter of the given region.*

Figure 2 shows two sample objects. While Figure 2(a) shows a concave object requiring a number of iterations as its number of pixels in the worst case, Figure 2(b) shows a convex object requiring a number of iterations upper bounded by its diameter in the worst case.

Table 2 explains through an example the effect of initial condition on the bounds. The resolution of the image is $128 \times 128$ pixels, the initial condition configuration is a chessboard like pattern. The number of squares was placed in n rows and n columns according to the values of the first row of the Table. Second and third rows show the general and convex bounds corresponding to initial condition configuration. The last two rows contain the number of iterations required to converge to the objects shown in Figure 2.

*Thesis 2.2 I proved that the evolution of the Shi method can be mapped efficiently to many core architectures provided it is started from an initial condition that minimizes the bounds stated in thesis claim 2.1. I implemented it on two architectures: on CNN-UM and on GPU. The results*

Table 2: Illustrating theorems on two test images with resolution $128 \times 128$ (2(a)-(b)).

| configuration ($n \times n$) | 1 | $2^2$ | $4^2$ | $8^2$ | $16^2$ | $24^2$ | $32^2$ | $64^2$ |
|---|---|---|---|---|---|---|---|---|
| general bound | $64^2$ | $32^2$ | 256 | 64 | 16 | 9 | 4 | 1 |
| convex bound | 127 | 63 | 31 | 15 | 7 | 5 | 3 | 1 |
| $N_{it}$ for Fig. 2(a) | 145 | 68 | 18 | 7 | 6 | 3 | 3 | 1 |
| $N_{it}$ for Fig. 2(b) | 26 | 16 | 9 | 6 | 4 | 3 | 3 | 1 |

*supported the claims.*

The smaller the connected regions in the initial condition, the lesser the required number of iterations to be able to converge. This kind of initial condition is used seldom because the number of processed pixels is $O(N \times M)$ in one iteration in the case of an $N \times M$ image since the small curves fill the whole image. In the case of an evolution starting from an initial condition containing a single curve one iteration processes $O(N+M)$ pixels. It shall be noted is an initial condition is "far" from the true object region then the number of pixels to be processed increases to $O(k(N+M))$ where $k \sim \max(N, M)$ leading to complexity $O(N \times M)$. Since the initial conditions are "far" from the real object in most cases the complexity of the two different evolution is asymptotically the same.

It follows from thesis claim 2.1 that densely placed curves with small diameters keep the worst case bound on the number of iterations according to the theorems low. On the Eye-RIS VS the execution time of one iteration is independent from the type of initial condition while in the case of GPU a mild deviation is experienced together with the drastic decrease of the number of iterations.

The algorithm mapped to CNN-UM is implemented on the Eye-RIS 1.3 VS. The realization uses only simple templates, one step of the algorithm is performed in $400 - 440\mu s$ on a QCIF image. It must be noted that the actual computing is finished within $60 - 70\mu s$ and the remaining time $(340 - 370\mu s)$ is required for the data movement from the main memory of the Eye-RIS (on the Altea NIOS-II microprocessor) to the Q-Eye chip memory.

The execution times of the algorithm mapped to GPU are summarized in Table 3. It is clear that evolutions started from the proposed initial condition family perform much better in all cases than the ones started from conventional initial conditions. In an extreme case it caused 24 times speedup ($2,048 \times 2,048$ image resolution, $210 \cdot 560$ vs. $7 \cdot 684$).

Table 3: Time measurements on NVIDIA GTX 780 GPU compared to Intel core i7 CPU, $t_{GPU}$ and $t_{CPU}$ shows mean of one iteration in $\mu s$.

| Data size | Initial condition | $t_{\mathbf{GPU}}$ | $t_{\mathbf{CPU}}$ | $N_{\mathbf{it}}$ | Speedup |
|---|---|---|---|---|---|
| $256 \times 256$ | $1 \times 1$ | 129 | 1,610 | 32 | 12.5 |
| $256 \times 256$ | $2 \times 2$ | 126 | 2,242 | 59 | 17 |
| $256 \times 256$ | $8 \times 8$ | 140 | 3,164 | 20 | 22 |
| $256 \times 256$ | $32 \times 32$ | 143 | 8,874 | 8 | 62 |
| $512 \times 512$ | $1 \times 1$ | 317 | 3,190 | 64 | 10 |
| $512 \times 512$ | $4 \times 4$ | 167 | 8,724 | 40 | 52 |
| $512 \times 512$ | $16 \times 16$ | 157 | 12,897 | 25 | 82 |
| $512 \times 512$ | $64 \times 64$ | 123 | 16,246 | 18 | 132 |
| $1,024 \times 1,024$ | $1 \times 1$ | 534 | 6,431 | 129 | 12 |
| $1,024 \times 1,024$ | $8 \times 8$ | 548 | 27,461 | 55 | 50 |
| $1,024 \times 1,024$ | $32 \times 32$ | 590 | 43,739 | 32 | 74 |
| $1,024 \times 1,024$ | $128 \times 128$ | 490 | 84,078 | 12 | 171 |
| $2,048 \times 2,048$ | $1 \times 1$ | 560 | 14,972 | 210 | 26 |
| $2,048 \times 2,048$ | $16 \times 16$ | 703 | 79,920 | 79 | 113 |
| $2,048 \times 2,048$ | $64 \times 64$ | 830 | 198,980 | 28 | 239 |
| $2,048 \times 2,048$ | $256 \times 256$ | 684 | 327,541 | 7 | 478 |

Presented results are the mean value of 100 runs.

It can be seen that both on CNN-UM and GPU a significant speedup can be achieved in the case of the LS evolution of Shi if the proposed initial condition family is used.

Publications connected to this thesis group: [II, III, IV]. The thesis claim is specified and paraphrased in details in the third chapter of my dissertation.

# 4 Application fields

I demonstrated that it is possible to perform 2D to 3D registration during image guided therapy applications at the speed of (0.5-10 fps). This is essential and has great impact on the following applications. Furthermore, the problem was solved with the constant consulting with field experts from GE Healthcare and the technical knowledge and codebase were forwarded to the French research and development team.

The claims of the second thesis group can be utilized for faster segmentation or detection. The application fields of these methods are known. Naturally I emphasize the analysis of medical images. It is straightforward that I managed to utilize an initial condition that was considered unfeasible until now. Additionally the results from thesis claim 2.1 give guarantee which is essential in time critical applications.

# 5 Acknowledgements

*„Mi mindnyájan az Ő teljességéből kaptunk kegyelmet kegyelemre halmozva." (Jn1,16)*

I deeply thank *Tamás Roska* and *Péter Szolgay* the former and the present head of the doctoral school for their advice and encouragement giving me the strength to go on. It is hard to precisely articulate how I thank *György Cserey* my advisor the fascinating work lasting more than six years. Without his excellent ideas and inspiration I would not have been successful in my doctoral studies. We managed to find a way to discuss the real life problems and joys as well.

I thank *Ion Pappas* his support, encouragement, advices, and inspiration. During our collaboration I learned how industry and academy can benefit from each other. The Öveges scholarship of General Electric Healthcare and the datasets provided by the company are gratefully acknowledged.

To the other doctoral students: *András Horváth, Attila Stubendek, Domonkos Gergely, Miska Radványi, Tamás Fülöp, Ádám Rák, Tam Zsedrovits, Miki Koller, Csaba Józsa, Csaba Nemes, Bence Borbély, Zoli Tuza, Jani Rudan, Dóri Bihari, Anna Horváth, András and Zsolt Gelencsér*. Thanks for the chat near the lunch, the help, the seminars on Fridays, the football.

I thank *Vida Tivadarné Katinka* her eternal service and patience to make the administrative side of life much easier, the work of the dean's office, the administrative and financial personnel.

Thank you *Father, Mother, Piri, Tamás, Ildi* and the whole family. Everything.

Niki, Magdi. You worth to fight for, and joy to return back home to you.

# The author's publications

[I] **G. J. Tornai**, G. Cserey, and I. Pappas, "Fast DRR generation for 2D to 3D registration on GPUs," *Medical Physics*, vol. 39, no. 8, pp. 4795–4799, 2012.

[II] **G. J. Tornai** and G. Cserey, "Initial condition for efficient mapping of level set algorithms on many-core architectures," *EURASIP Journal*

*on Advances in Signal Processing*, 2014:30.

[III] **G. J. Tornai**, G. Cserey, and A. Rák, "Spatial-Temporal level set algorithms on CNN-UM," in *International Symposium on Nonlinear Theory and its Application, (NOLTA 2008)*, pp. 696–699, 2008.

[IV] **G. J. Tornai** and G. Cserey, "2D and 3D level-set algorithms on GPU," in *Cellular Nanoscale Networks and Their Applications (CNNA), 2010 12th International Workshop on*, p. 1–5, 2010.

A. Horváth, **G. J. Tornai**, A. Horváth and G. Cserey, "Fast, parallel implementation of particle filter on GPU," *EURASIP Journal on Advances in Signal Processing*, 2013:148.

# References


[1] P. Markelj, D. Tomazevic, B. Likar, and F. Pernus, "A review of 3D/2D registration methods for image-guided interventions," *Medical Image Analysis*, Mar. 2010.

[2] G. P. Penney, P. G. Batchelor, D. L. Hill, D. J. Hawkes, and J. Weese, "Validation of a two-to three-dimensional registration algorithm for aligning preoperative CT images and intraoperative fluoroscopy images," *Medical physics*, vol. 28, p. 1024, 2001.

[3] J. Wu, M. Kim, J. Peters, H. Chung, and S. S. Samant, "Evaluation of similarity measures for use in the intensity-based rigid 2D-3D registration for patient positioning in radiotherapy," *Medical Physics*, vol. 36, no. 12, p. 5391, 2009.

[4] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.* No. 3 in Cambridge monographs on applied and computational mathematics, New York: Cambridge Univ Pr, 2nd edition ed., 1999.

[5] G. Sapiro, *Geometric partial differential equations and image analysis.* New York: Cambridge Univ Pr, 1st edition ed., 2001.

[6] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, pp. 61–79, Feb. 1997.

[7] Y. Shi and W. Karl, "A Real-Time algorithm for the approximation of Level-Set-Based curve evolution," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 645–656, 2008.

[8] 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.

[9] G. J. Tornai, G. Cserey, and I. Pappas, "Fast DRR generation for 2D to 3D registration on GPUs," *Medical Physics*, vol. 39, no. 8, pp. 4795–4799, 2012.