A Hardware Implementation of a Levelset Algorithm for Carotid Lumen Segmentation in CTA

André van der Avoird[a], Ning Lin[a], Bram van Ginneken[b] and Rashindra Manniesing[b]

[b] BIC Design, Eindhoven, the Netherlands;
[a] Diagnostic Image Analysis Group, Department of Radiology, Radboud University Nijmegen Medical Centre, Nijmegen, the Netherlands

ABSTRACT

This work presents a novel hardware implementation of a levelset algorithm for carotid lumen segmentation in computed tomography. We propose to use a field programmable gate array (FPGA) to iteratively solve the underlying finite difference scheme. A FPGA processor can be programmed to have a dedicated hardware architecture including specific data path and processor core design with different types of parallelizations which is fully tailored and optimized toward its application. The method has been applied to ten carotid bifurcation of six stroke patients and the results have been compared to the results obtained from the same method implemented in C++. Visual inspections revealed similar segmentation results. The average computation time in software was 1663 ± 86 seconds, the computation time on the FPGA processor was 28 seconds yielding approximately a 60-fold speed-up which to our knowledge has been unmmatched before for this class of algorithms.

Keywords: hardware, field programmable gate array (FPGA), levelset, segmentation, carotid bifurcation, CT Angiography

1. INTRODUCTION

An important class of algorithms in medical image analysis are partial differential equations (PDEs) driven methods for object segmentation. Because PDE driven segmentation methods can easily handle complex varying shapes, as opposed to for example active shape models, they have become the cornerstone in many vessel analysis applications and in methods aimed at segmentation of anatomical structures from brain Magnetic Resonance images. A general and now commonly used framework has been presented by Caselles et al.[1] In their seminal paper a connection was established between explicit contours and implicit geodesic active contours (GAC) thereby extending the classical PDE for levelset evolution suggested by Osher and Sethian[2] with one additional term to track boundaries with high gradient variations, called the advection term. However, the main disadvantage of these methods is their large computation time, especially if second order terms for curvature calculations are included, and even if a narrow-band approach, only updating the values at the boundaries of the evolving object, has been adopted.

The purpose of this work is to demonstrate the potential of dedicated hardware based solutions for the implementation of computational intensive algorithms in medical image analysis.

To meet these computational demands we propose to use a *Field Programmable Gate Array* (FPGA). The FPGA is essentially a processor chip available up to very high-performances and which can be programmed at a very low level (called the Boolean gate level) to have a specific functionality. They have been introduced in the mid-eighties and ever since their use has increased tremendously, but mainly in industrial, communication and networking applications.[3] The use of FPGA solutions in *medical imaging* applications (and those solutions that have been published in literature) has actually been limited to a few only.[4,5,6,7,8] If we consider biomedical applications in general, hardware based solutions are more commonly found. A simple case-sensitive query on the number of publications that have appeared on Pubmed (www.pubmed.com) with in its title or abstract the word 'FPGA' or 'GPU' (Graphical Processor Unit, a type of processor typical found on a video card), and after removal of possible medical meanings of these acronyms, gives the results that are shown in Figure 1. The debut

Medical Imaging 2013: Image Processing, edited by Sebastien Ourselin, David R. Haynor,
Proc. of SPIE Vol. 8669, 86691V · © 2013 SPIE · CCC code: 1605-7422/13/$18
doi: 10.1117/12.2007231

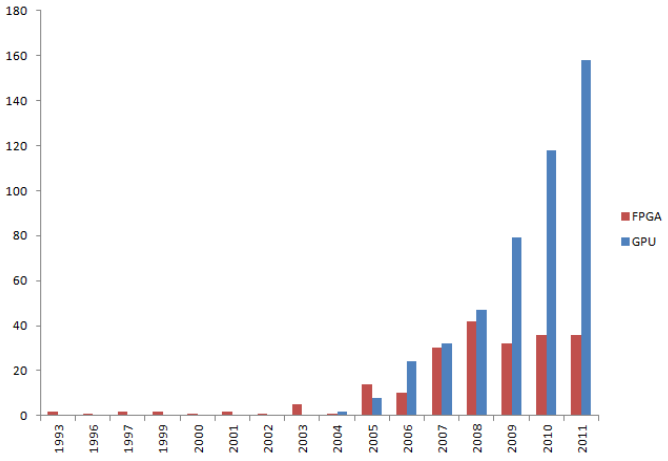

Figure 1. Total number of publications per year that have appeared on Pubmed with the word 'GPU' or 'FPGA' in the title or the abstract.

appearing of FPGA on Pubmed was in 1993 and GPU in 2004, and only after 2004 the number of publications steadily started to rise to reach a peak in 2008 for FGPA based solutions.

The reasons why we think FPGA based approaches should be considered now, are the following.

High performance-cost ratio The highest performances are achieved by designing a dedicated processor, which is a costly procedure and gives the least flexibility in changing its functionality later on. On the other hand, there are general purpose processors (typical found in a desktop or a laptop) which are cheap and flexible (any algorithm can be implemented) but may not give the required performance. Inbetween these extremes, there are the FPGA and GPU based approaches. There is an important difference between these two.

Highly efficient parallelization The program that is executed on the FPGA is not like a traditional software program, instead, the program is actually a very low level hardware definition of functions and connections of the circuitry. This gives the speed of an architecture and a physical circuit fully dedicated to the algorithm, but the flexibility of a general purpose processor. The speed-up is expected because the dedicated data-path and memory architecture will do useful calculations *each* clock cycle. There is no overhead caused by instruction and data manipulations like in a general purpose processor (found in a desktop computer) or in a GPU (found on a video card). Furthermore, specific data-path design enables very low level task parallelization, e.g. duplication of bottleneck adders and/or multipliers, the re-use of single memory words, gate level pipelining, retiming, and data-skewing. As a consequence, a highly efficient parallelization can be achieved on a completely different level than can be done on parallel architectures like the GPU.

Acknowledgments This work has been supported by two grants from the Dutch organization for scientific research (NWO-NCF, grant numbers NRG-2009.06 and NRG-2011.04).

2. METHODS

A basic levelset equation is considered without curvature or advection terms: $\phi_t + F|\nabla\phi| = 0$, with ϕ the levelset function and F the image based speed function. This equation is iteratively solved using a standard explicit numerical discretization scheme.[9] This scheme has been implemented on a FPGA board (Figure 2).

The top level architecture of the hardware system is shown in Figure 3. The color coding indicates the design type of the specific block. The initial level set and speed function are sent from the host computer to the board via the physical layer interface (PHY) chip and Media Access Control (MAC) block to the on-board DDR3 memory. Then the levelset process starts and after N iterations the final levelset is sent back to the host computer. In the block diagram 'IF' stands for interface, 'Proc' is the processing block and 'MIG' is the controller block for the memory. At the input of the system, the initial levelset and the speed function images with fixed sizes are

Send correspondence to r.manniesing@rad.umcn.nl

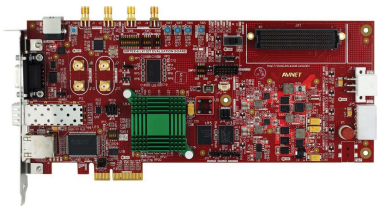

Figure 2. The Xilinx Virtex 6 FPGA board that is used to implement the system. Several key components can be distinguished, including the FPGA processor (the green square), several connection ports including ethernet, serial and USB, various switches and LEDs. The ethernet port is used to transfer image data to and from the host computer at 1 Gbps. The user LEDs and switches can be used for development and debugging purposes. The board can be connected using one of the communication ports or can directly be plugged into a free PCI slot of a desktop computer.

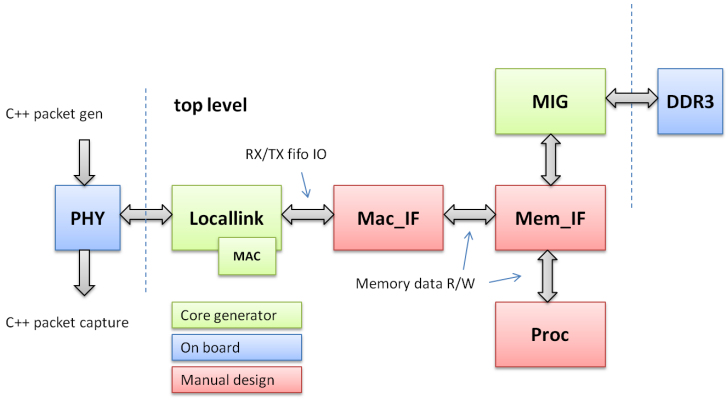

Figure 3. The top level architecture of the system.

expected. The data is sent from the host computer over the ethernet connection to the onboard memory, then the processing block iterates over the memory data until the final result is available and can be sent back to the host computer. The functional blocks are either actual hardware that already is placed on the board (e.g. the DD3 memory), or generated by a core generator (Xilinx) or manually designed in a Hardware Description Language (HDL).

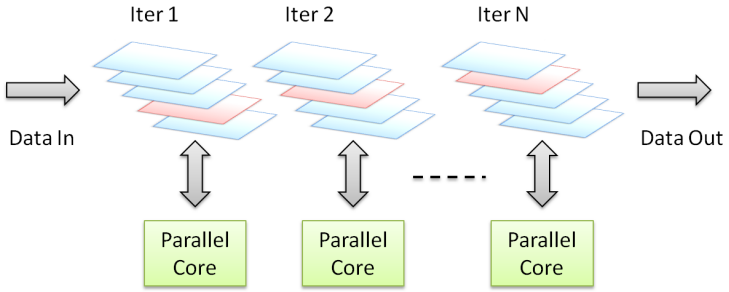

Figure 4. The parallel processing dedicated cores. Each available core needs several slices of the 3D levelset image to work on in different stages of the iterative process.

The speedup is achieved by mainly two types of optimization. First is the design of a dedicated datapath for the core calculations. The datapath contains the arithmetic operations at the required precision. It calculates the result for a single pixel, in a single iteration based on its surroundings and the speed function. The datapath is pipelined, which gives some latency but allows us to stream in new pixel data and stream out results every clock cycle. Second is dividing the processing over parallel instances of the arithmetic core. Optimizing this parallelization means trading off and making optimal use of the off-chip bandwidth to the dynamic random access memory (DRAM), the on-chip available static random access memory (SRAM) size and the number of parallel cores. This distributed processing over the parallel cores is illustrated in Figure 4. At the data-in, a number of

Figure 5. Volume renderings of the final segmentations obtained by the software implementation

Figure 6. The results from the FPGA implementation.

slices of the 3D images are brought to the on-chip memory. Each core needs several slices to work on in different stages of the iterative process. The memory architecture is also manually optimized to guarantee data integrity, meaning each core works on the correct pixels in the correct state of iteration. The resulting processing and memory architectures have a very generic setup and can be easily scaled when hardware bottlenecks are resolved in the future.

3. EXPERIMENTS AND RESULTS

The levelset algorithm is used in a framework for carotid bifurcation segmentation in CTA. The levelset algorithm has also been implemented in C++ as reference for comparison with the FPGA implementation. Ten atherosclerotic carotid bifurcations from six stroke patients were used as input data for both implementations. A detailed description of the CTA patient data is given in.[10] The user defined seed points have been used to crop the image data to obtain a fixed size ROI of 160x160x640 pixels. Having fixed size imaging data is a requirement imposed by the hardware design. The number of levelset iterations has been set at 500. The software program ran on a standard desktop computer with a dual-core Intel Celeron processor at 2.20 GHz and 4G of memory. The average computation times are given in Table 1, 3D visualizations of the final levelset images are shown in Figures 5 and 6. The average speed-up that is achieved in this setting is 60-fold.

software C++	1663 ± 86 (27.7 ± 1.4 min)
hardware FPGA	28
speed-up	59.4

Table 1. Average processing times in seconds for segmenting ten carotid bifurcations using the C++ software implementation and the processing time of the FPGA implementation running for 500 levelset iterations.

4. DISCUSSION AND CONCLUSION

To our knowledge, this is the first work describing the implementation of a levelset algorithm on a FPGA processor. The results show the feasibility of a FPGA based approach for the implementation of levelset algorithms, which is an important class of segmentation algorithms in medical imaging, to achieve ten-folds of speed-up in computation time compared to a similar implementation in software.

This work has limitations. Currently, the curvature and advection terms are lacking in the FPGA implementation. Implementing these terms is part of immediate future work which recently has been supported by a follow-up parallelization grant.

In the prototype system, the limiting factors for hardware system performance are the off-chip bandwidth to the DRAM, the size of the on-chip available SRAM and the number of parallel arithmetic cores. In further work all of these bottlenecks will be addressed. Given the scalability of the current implementation a new target of 600-1000 times acceleration is realistically expected when these bottlenecks have been resolved.

Automation of the parallelization process was one of the targets for this work but turned out not feasible due to hardware as well as design tool limitations. This target remains open for further research.

REFERENCES

[1] Caselles, V., Kimmel, R., and Sapiro, G., "Geodesic active contours," *International Journal of Computer Vision* **22**, 61–79 (1997).

[2] Osher, S. and Sethian, J., "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics* **79**, 12–49 (1988).

[3] Worchel, J., "The field-programmable gate array (FPGA): Expanding its boundaries," NPD In-Stat (2006).

[4] Neuenhahn, M., Blume, H., and Noll, T., "Pareto optimal design of an FPGA-based real-time watershed image segmentation," in [*Proceedings ProRISC*], (2004).

[5] Leeser, M., Coric, S., Miller, E., and Yu, H., "Parallel-beam backprojection: An FPGA implementation optimized for medical imaging," *Journal of VLSI Signal Processing Systems* **39**, 295–311 (2005).

[6] Dandekar, O. and Shekhar, R., "FPGA-accelerated deformable image registration for improved target-delineation during CT-guided interventions," *IEEE Transactions on Biomedical Circuits and Systems* **1**, 116–127 (2007).

[7] Hasan, S., Yakovlev, A., and Boussakta, S., "Performance efficient FPGA implementation of parallel 2D MRI image filtering algorithms using Xilinx system generator," in [*7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP)*], 765 –769 (2010).

[8] Lin, S.-J., Hwang, W.-J., and Lee, W.-H., "FPGA implementation of generalized Hebbian algorithm for texture classification," *Sensors* **12**, 6244–6268 (2012).

[9] Sethian, J. A., [*Level set methods and fast marching methods*], Cambridge University Press, 2 ed. (1999).

[10] Manniesing, R., Schaap, M., Rozie, S., Hameeteman, R., Vukadinovic, D., van der Lugt, A., and Niessen, W., "Robust CTA lumen segmentation of the atherosclerotic carotid artery bifurcation in a large patient population," *Medical Image Analysis* **14**, 759–769 (2010).