Image and Video Processing on CUDA: State of the Art and Future Directions

Roberto Di Salvo, Carmelo Pino
Department of Electrical, Electronics and Computer Engineering - University of Catania
V.le Andrea Doria, 6 – 95125 Catania - Italy
E-mail: {roberto.disalvo, carmelo.pino}@dieei.unict.it

Abstract: - In the last few years a myriad of computer graphic applications have been developed using standard programming techniques, which are mainly based on multicore general-purpose processors (CPUs) architectures. Due to the rapid turning towards high definition multimedia, more and more researches have been done that need both computational resources and memory space to achieve high performance. To this end, more recently the general-purpose computing on graphic processing units (GPGPUs) architectures, which are becoming increasingly programmable and scalable, have been adopted. Since the GPUs provide a vast number of simple, data-parallel, deeply multithreaded cores and high memory bandwidths, they have been used to provide a valid support in several fields such as image and video processing research area, medical and human applications. In this paper the most recent image and video processing techniques and relevant applications are reviewed, highlighting the advantages of using CUDA in terms of efficiency.

Key-Words: - Graphics processors, Parallel programming, Multicore, Image processing, CUDA platform, Medical and Human applications

1 Introduction

Nowadays, the advent of general purpose computing on GPUs have drawn a lot of researchers to prefer this new parallel processing platform to the multicore CPUs architectures. Since the massive parallelism, high memory bandwidth, and general purpose instruction sets (including single/double precision IEEE floating point arithmetic) of Computed Unified Device Architecture (CUDA) systems, it becomes important to revisit parallel programming model in order to find the best trade-off between computational convenience and hardware implementation costs [1]. Due to the power of the graphic processing units designed to exploit the multi-threaded capabilities of the multi-core structure [2], several approach have been proposed for computing intensive application fields.

In particular, the GPUs have recently used in the field of computer vision [3-7] especially for all those operations characterized by multiple iterations, making this new technology increasingly competitive as regards price, speed and programmability. Thus, CUDA systems were taken into account as the reference hardware architecture for parallel processing created by NVIDIA.

The CUDA programming model (e.g. CUDA-C) allows the developers to exploit parallelism by writing simple C code that run in thousands of parallel invocations, or threads. These threads are managed by representing parallel tasks as kernels (the sequence of works to be executed in each thread) mapped over a domain (the set of threads to be invoked). The data to be processed by the GPU are copied to the graphics board's memory. Data transfer is done by DMA (Direct Memory Access) and are persistent unless they are de-allocated, thus remaining available for next kernels. Fig.1 shows a reference schema for CUDA threads architecture.

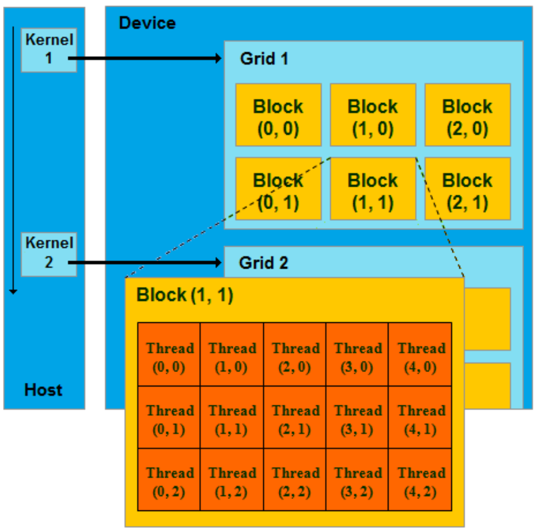

Figure 1 – CUDA Thread model: each kernel is assigned to a grid, each grid contains a number of blocks, each block contains threads (512 maximum per block)

ISBN: 978-1-61804-046-6

To appreciate the impact of CUDA in the modern programming scenario and potentialities, the paper first discusses the advantage of CUDA programming on GPUs starting from the limits of CPUs multicore technologies, then in Sect. 3 a review of most recent works for GPU video and image processing in different domains will be illustrated. Concluding remarks and future directions are given in the last section.

2 Problems and Motivations

Today, one of the most interesting programming topic concerns the advantages of using the new GPU paradigm rather than CPU standard programming. In the past, when an algorithm working on standard single or multi-core processor was too slow, we waited for a future CPU able to execute the same code in a more efficiently way. Despite in the last few years the CPUs architecture have been increasingly improved with "many-core" processors, providing high parallelism, a lot of operations could be executed in a tiny time, especially in research fields such as i) video-surveillance for vehicles [8-10], animals [11-13] and humans [14-16], ii) medical image analysis [17-21], and iii) image restoration [22-24].

For this reason, parallel implementations of such applications have been developed more and more using GPU systems with the aim of achieving high speedups ranging from 10× to 100× and above. In the earliest versions, these GPU based applications were implemented via a graphics language. This limited its flexibility. With the advent of NVIDIA's CUDA platform a hugely multithreaded general purpose architecture is provided, with up to 512 processor cores and thousands of threads executed at the same time. CUDA is programmable in C and capable of millions of floating-point operations per second [25]. This allowed the CUDA technology to be adopted across several domains ranging from image and video processing applications [26-30], computational chemistry [31], astrophysics [32], biomedical purposes including for example gene sequencing [33], CT reconstruction [34], and medical image segmentation [35-36].

In the next section a survey of such approaches, organized per domain, is given by pointing out how their higher performance opens new frontiers for CUDA based systems.

3 Review

Among the huge quantity of approaches that exploits CUDA for investigating improved efficiency, three main domains are reviewed: 1) general video and image processing applications, 2) biomedical applications and 3) video-surveillance applications.

3.1 CUDA for Image and Video Processing

Since the C-CUDA technologies have been made available, many image and video processing applications, such as image segmentation, edge detection for video segmentation and video coding, have been ported to CUDA.

3.1.1 Image Processing Approaches

Several approaches for image processing that exploit CUDA have been proposed in the last few years. They usually deal with edge detection and image segmentation algorithms that are commonly time-consuming.

For example, Yang *et al.* in [30] analyze the distinct features of GPU for developing histogram equalization (40× speedup), removing clouds (79× speedup), edge detecting (8× speedup) and DCT encoding and decoding (200× speedup and above). The architecture of the adopted CUDA implementation for histogram equalization is shown in fig. 2 where the input image data is split into N threads that compute the probability density function (needed for the image equalization) for a subset of the input data and then a master collect the results of each thread and provide the output data.

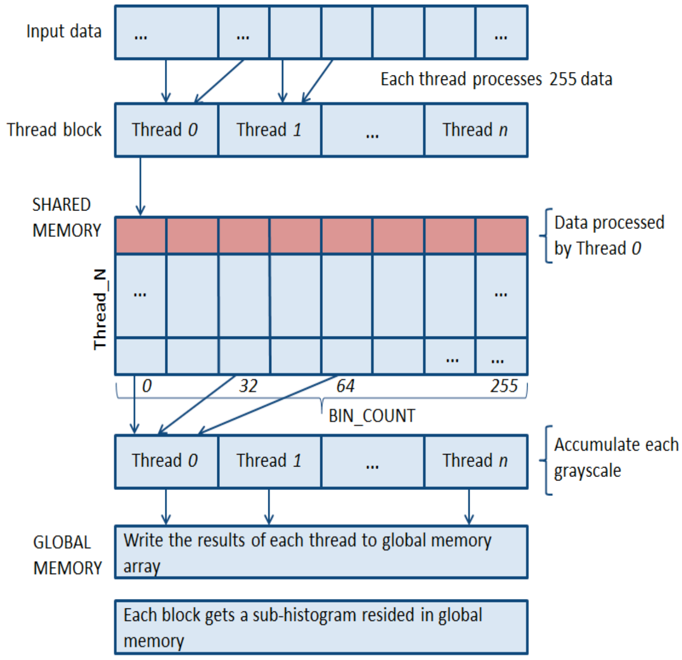

Figure 2 – CUDA implementation of a histogram equalization algorithm

Most of the image and video processing approaches existing in the literature share a similar philosophy (data-based-parallelization) to port the sequential code into a CUDA implementation and it relies mainly on splitting the input into N threads and then a master node reassembles the results of each thread and provides the global result.
According to this approach, Catanzaro *et al.* in [28] examine efficient parallel algorithms for performing image contour detection, with particular attention paid to local image analysis as well as the generalized eigensolver used in Normalized Cuts. Yang *et al.* in [30] analyze the distinct features of GPU for developing histogram equalization (40× speedup), removing clouds (79× speedup), edge detecting (8× speedup) and DCT encoding and decoding (200× speedup and above).

Luo and Duraiswami in [37] and Park *et al.* in [38] propose two different approaches for Canny's edge detection algorithm using CUDA. In the first case the authors present a comparison with CPU implementations using Matlab toolbox, wheras in the second case the authors demonstrate the efficiency of their approach by a parallelization and optimization of the Canny algorithm for data-intensive video motion tracking.

As concerns image segmentation, stereo vision and image restoration in [39] Vineet and Narayanan suggest a new implementation of graph-cuts algorithm to find the optimal MAP estimation of various Computer Vision problems defined over an MRF (Markov Random Field).

Finally in [40] Zhang and Nevatia describe an efficient design for scan-window based object detectors using a GPGPU framework. They use a generic methodology that can be applied to other objects, using different features and classifiers (like SVM) they achieve a more-than-ten-times speed up with no loss on detection rates.

3.1.2 Video Processing Approaches

Differently from image processing, few approaches that use CUDA have been proposed for video processing and the existing ones are mainly related to video segmentation and video coding. Gomez-Luna *et al.* in [41] present the parallelization of a video segmentation application on GPU hardware, which implements an algorithm for abrupt and gradual transitions detection. Colic *et al.* in [29] explore the use of parallelization opportunities in graphics processors to accelerate video encoding. They evaluate the capabilities of NVIDIA/CUDA and develop a process for implementing video/multimedia applications discovering that the difference in performance when CUDA is not used properly can be over 100×.

Lastly, Yu and Medioni in [26] propose a GPU-based implementation for building the background model and detecting motion regions (18 fps on 320×240 videos), in order to achieve significant acceleration over standard CPU.

3.2 Medical Applications

Due to the massive GPUs computational power for scientific computing, several medical applications have been rewritten by using CUDA technology following a data-based parallelization analogue to the one shown in fig. 2.

To this end in [42] Boyer *et al.* demonstrate how a biology application for the detection and tracking of white blood cells in video microscopy can be accelerated by 200× using a CUDA-capable GPU. In [43] Hartley *et al.* illustrate a CPU-GPU cooperative parallelization approach by implementing a large-scale biomedical image analysis application that contains a number of different kernels including streaming operators, co-occurrence matrices, convolutions, and histograms. The highest speed-up is achieved when all in

ISBN: 978-1-61804-046-6

coexisting methods in CPUs and GPUs are combined.

Image segmentation is considered a fundamental task also in medical domain. Pan *et al.* in [35] and Ruiz *et al.* in [36] provided different implementation of several medical image segmentation algorithms using CUDA and CUDA-enabled GPUs, finding high improved performance in terms of speedup with respect to the sequential version of the considered algorithms.

Another aspect to be tackled when dealing with medical images is the registration phase. In [44] Modat *et al.* present a parallel-friendly formulation of the “free-form” deformation algorithm to perform non-rigid registration in medical image analysis, suitable for GPUs execution. In this approach they perform registration of T1-weighted MR images in less than 1 min providing the same level of accuracy of the classical sequential implementation. Similarly, Scherl *et al.* in [34] propose a C-CUDA implementation of the most time-consuming parts of FDK method (i. e. filtering and back-projection), to solve the three-dimensional reconstruction task in cone-beam CT. The authors explain the required transformations to parallelize the algorithm for the CUDA architecture providing an “on-fly” reconstruction, which happens right after the end of data acquisition.

Another interesting topic that have been addressed using CUDA is the data mining for discovering associations between biological entities [45]. On this field, to best of our knowledge, only one approach has been faced with CUDA [46], where the authors propose an evolutionary computing technique named Artificial Immune System (AIS) to refine explanatory models and successfully identify highly complex associations between genotypes and diseases.

3.3 Video-surveillance Applications

Recently, even in video-surveillance applications the CUDA technology has been successfully adopted giving an important contribution in the detection and tracking tasks. For example, Bilgic *et al.* in [47] propose a fast pedestrian localization framework that integrates the cascade-of-rejectors approach with the Histograms of Oriented Gradients (HoG) features, selecting the most informative histogram blocks with the AdaBoost algorithm, in a parallel architecture. By using the NVIDIA-CUDA framework on a GPU, they report a speed up of a factor 13 over the corresponding CPU implementation. This approach is also able to run at 8 frames per second for a 1.2 Megapixel image achieving near real time detection performance.

Similarly in [30] Tang and Goto proposed a GPU-based implementation to accelerate the computation of a multi scale block histogram of a template feature (MB-HOT) for pedestrian tracking which integrates gray value and gradient value information, and reflects relationship of three blocks. Like the previous approach this method provides very good performance and it is suitable enough to be adopted in several real time applications. Sugano and Miyamoto in [48] also tackles the problem of pedestrian recognition for automotive and security applications, focusing their attention on the tracking task. In particular to construct an accurate tracking scheme suitable for pedestrian recognition from distant observation points, they use multiple cues: HSV histogram and a HOG feature. This approach was implemented on NVIDIA Tesla C1060 processor (one of the latest GPGPU on the market) and achieved real-time processing performance. Lillywhite *et al.* in [49] propose a similar real-time human detection approach using histograms of oriented gradients method through the NVIDIA CUDA architecture on a GPU. The implementation significantly speeds up computation, achieving approximately 38 frames per second on a standard VGA video, whereas the accuracy in human detection remains comparable to the equivalent CPU implementation.

Another domain on which CUDA has been widely used is face detection and recognition. Nasse *et al.* in [50] consider the problem of face detection under pose variations by using an efficient implementation of convolutional neural networks, exploiting the computational power of modern graphics cards. Likewise, Devrari and Kumar in [14] propose a new and fast approach to perform real time face detection, which includes the enhanced Haar-like features and uses SVM (Support Vector Machine) for training and classification. Experimental result shown that the proposed architecture works well for high-resolution images.

4 Conclusions and Future Directions

The field of GPU computing is nearly to be mature as demonstrated by the massive growing of applications in different domains.

In this paper a survey of CUDA based approaches for image and video processing in different domains has been given. We have highlighted how many applications in these areas have been parallelized using CUDA, to achieve very high-performance in time processing keeping the same performance in terms of accuracy.

The main part of the available CUDA based approaches dealt with the parallelization of generic image processing operations, whereas much more work should be done in the biomedical and video-surveillance fields. In image and video processing fields, we expect that CUDA based clustering techniques should allow to improve the real time control of metropolitan systems that highly influence life quality, e.g., the traffic flow management and the understanding of people behaviors to prevent dangerous situations. Often these systems rely on video processing approaches but they are very time-consuming making their application on real life environment a challenge.

In the next years, we would expect new work than simply "porting" existing algorithms to the GPU, in order to demonstrate principles, techniques and real advantages of using the available hardware. Fortunately, the accumulated body of developed approaches represents an interesting background for developers that want to use GPUs for general-purpose computing. Moreover, research groups are working on performing large-scale GPU computing exploiting large clusters of GPU-equipped computers [51].

In conclusion, the computational capacity and the rapid growth of GPU systems has allowed computer vision applications to reach performance in terms of processing times unimaginable even with the best performing multicore processors. However, in the near future we look forward to see improved the programmability and generality of future GPU architectures.

References:

[1] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, K. Skadron, "A performance study of general-purpose applications on graphics processors using CUDA", *J. Parallel Distrib. Comput. 68 (2008) 1370–1380*

[2] J. Pan, D. Manocha, "GPU-based Parallel Collision Detection for Real-Time Motion Planning", *Springer Tracts in Advanced Robotics*, 2011, Volume 68/2011, 211-228, DOI: 10.1007/978-3-642-17452-0_13

[3] K.S Kyong, K. Jung, "GPU Implementation of Neural Network", *Pattern Recognition*, Vol. 37, Issue 6, pp. 1311-1314, 2004.

[4] K. Moreland, E. Angel, "The FFT on a GPU", *Proceedings of SIGGRAPH Conference on Graphics Hardware*, pp. 112-119, 2003.

[5] J. Mairal, R. Keriven, A. Chariot, "Fast and Efficient Dense Variational Stereo on GPU", *Proceedings of International Symposium on 3D Data Processing, Visualization and Transmission*, pp. 97-704, 2006.

[6] R. Yang, G. Welch, "Fast Image Segmentation and Smoothing using Commodity Graphics hardware", *Journal of Graphics Tools*, Vol. 17, issue 4, pp. 91-100, 2002.

[7] J. Fung, S. Man, "OpenVIDIA: Parallel GPU Computer Vision", *Proceedings of ACM International Conference on Multimedia*, pp. 849-852, 2005.

[8] A. Faro, D. Giordano, C. Spampinato, Integrating location tracking, traffic monitoring and semantics in a layered ITS architecture, IET Intelligent Transport Systems 5, 197 (2011).

[9] A. Faro, D. Giordano, C. Spampinato, Fusing Visual Perceptions And CNN Processing Of Webcam Images To Evaluate The Traffic Parameters In A Metropolitan Area, IEEE Trans- actions on Neural Networks, Volume 19, Number 6, 2008.

[10] A. Faro, D. Giordano, C. Spampinato, Soft-computing agents processing webcam images to optimize metropolitan traffic systems, Springer Book Series on Computational Imaging and Vision, Computer Vision and Graphics, Volume 32 pp. 968-974. 2006

[11] C. Spampinato, D. Giordano, R. Di Salvo, J. Chen Burgher, R. B. Fisher, G. Nadarajan, Automatic Fish Classification for Underwater Species Behavior Understanding, Proc. of the ACM Int. Workshop Anal. and Retrieval of Tracked Events and Motion in Imagery Streams (ARTEMIS), October 29th, 2010, Florence, Italy.

[12] C. Spampinato, Adaptive Objects Tracking by using Statistical Features Shape Modeling and Histogram Analysis, Proceedings of the 7th IEEE International Conference on Advances in Pattern Recognition (ICAPR09), February 4th – 6th, 2009, Kolkata, India.

[13] C. Spampinato, J. Chen Burger, G. Nadarajan, R. B. Fisher, Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos, Proceedings of VISAPP 08, January, 22nd – 25th, 2008, Madeira, Portugal.

[14] K. Devrari, K.Vinay Kumar, "Fast Face Detection Using Graphics Processor", *International Journal of Computer Science and Information Technologies*, Vol. 2 (3), 2011,1082-1086.

[15] S. Tang, S. Goto, “Multi Scale Block Histogram of Template Feature for Pedestrian Detection”, *Proceedings of 2010 IEEE 17th International Conference on Image Processing*, September 26-29, 2010, Hong Kong.

[16] A. Faro, D. Giordano, C. Spampinato, An Automated Tool for Face Recognition Using Visual Attention and Active Shape Models Analysis, Proceedings of the 28th IEEE EMBS Annual International Conference, Aug 30th - Sept 3rd, 2006, New York City, USA.

[17] . Giordano, R.Leonardi, F. Maiorana, C. Spampinato, Cellular Neural Networks and Dynamic Enhancement for Cephalometric Landmarks Detection, Lecture Notes in Computer Science, Artificial Intelligence and Soft Computing - ICAISC 2006, Volume 4029, pp. 768-777. Springer Berlin / Heidelberg.2006

[18] A. Faro, D. Giordano, C. Spampinato, S. Ullo, A. Distefano, Basal ganglia activity measurement by automatic 3D Striatum segmentation in SPECT images, Appearing on IEEE Transactions on Instrumentation and Measurement (2011) Special Issue on Medical Measurements, Article in Production.

[19] A. Faro, C. Spampinato, G. Scarciofalo, R. Leonardi, An Automatic System for Skeletal Bone Age Measurement by Robust Processing of Carpal and Epiphysial/Metaphysial Bones, IEEE Transactions on Instrumentation and Measurement , Vol. 59, pp. 2539-2553, 2010.

[20] R.Leonardi, D. Giordano, F. Maiorana, C. Spampinato, Automatic Cephalometric Analysis: A Systematic Review, The Angle Orthodontist, Vol. 78, No.1, pp.145-151, February 2007.

[21] A. Faro, D. Giordano, F. Maiorana, G. Scarciofalo, C. Spampinato, EMROI Extraction and Classification by Adaptive Thresholding and DoG Filtering for Automated Skeletal Bone Age Analysis, Proceedings of the 29th Conference of IEEE Engineering in Medicine and Biology Society, pp.22-26, August 23rd – 26th, 2007, Lyon, France.

[22] Faro, D. Giordano and C. Spampinato Neural Network Combined with Fuzzy Logic to Remove Salt and Pepper Noise in Digital Images, in Tiwari et al (Eds.), Applications of Soft Computing: Recent Trends, pp 23-33, Springer Berlin Heidelberg New York. 2006

[23] A. Faro, D. Giordano, G. Scarciofalo and C. Spampinato, Bayesian Networks for Edge Preserv- ing Salt and Pepper Image Denoising, Proceedings of the first IEEE International Workshops on Image Processing Theory, Tools and Applications (IPTA08), November 24th – 26th, 2008, Sousse, Tunisia.

[24] F. Cannavo’, D. Giordano, G. Nunnari, C. Spampinato, Variational Method for Image Denoising by Distributed Genetic Algorithms on GRID Environment, Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enter- prises (WETICE-2006), pp. 272-232, June 26th - 28th, 2006, Manchester, UK.

[25] NVIDIA. 2007. CUDA Programming Guide 1.1;*http://developer.download.nvidia.com/compute/cuda/1_1/NVIDIA_CUDA_Programming_Guide_1.1.pdf.*

[26] Q. Yu, G. Medioni, “A GPU-based implementation of Motion Detection from a Moving Platform”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1-6, Doi: 10.1109/CVPRW.2008.4563096.

[27] B. Bilgic, B. K. P. Horn, I. Masaki, “Efficient Integral Image Computation on the GPU”, *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 21-24 2010, pp.528-533, DOI: 10.1109/IVS.2010.5548142.

[28] B. Catanzaro, B-Y. Su, N. Sundaram, Y. Lee, M. Murphy, K. Keutzer, “Efficient, high-quality image contour detection”, *IEEE 12th International Conference on Computer Vision, 2009*, Sept. 29 2009-Oct. 2 2009, pp. 2381-2388, DOI: 10.1109/ICCV.2009.5459410.

[29] A. Colic, H. Calva, B. Furht, “Exploring NVIDIA-CUDA for Video Coding”, *MMSys '10 Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, New York, NY, USA ©2010.

[30] Z. Yang, Y.Zhu, Y. Pu, “Parallel Image Processing Based on CUDA”, *2008 International Conference on Computer Science and Software Engineering*, 2008 Dec. 12-14, pp. 198-201.

[31] J. Stone, J. Phillips, D. Hardy, P. Freddolino, L. Trabuco, K. Schulten., “Accelerating molecular modeling applications with graphics processors,” *Journal of Computational Chemistry*, Vol. 28, No. 16 , pp 2618 – 2640, 25 Sep 2007.

[32] R. G. Belleman, J. Bédorf and S. F. Portegies, “High Performance Direct Gravitational N-body Simulations on Graphics Processing Units II: An implementation in CUDA”, *arXiv:0707. 0438v2 [astro-ph], Jul 2007 (Accepted to New Astronomy)*.

[33] M. C Schatz, C. Trapnell, A. L Delcher, and A. Varshney, “MUMmerGPU: High-throughput

sequence alignment using Graphics Processing Units”, *BMC Bioinformatics 2007*, 8:474, Dec 2007.

[34] H. Scherl, B. Keck, M. Kowarschik, J. Hornegger, “Fast GPU-Based CT Reconstruction using the Common Unified Device Architecture (CUDA)”, *2007 IEEE Nuclear Science Symposium Conference Record*, Oct. 26 2007-Nov. 3 2007, pp. 4464 – 4466.

[35] L. Pan, L. Gu, J. Xu, “Implementation of medical image segmentation in CUDA”, *International Conference on Information Technology and Applications in Biomedicine*, 30-31 May 2008, pp. 82-85.

[36] A. Ruiz, K. Jun, M. Ujaldon, K. Boyer, J. Saltz, M. Gurcan, “Pathological image segmentation for neuroblastoma using the GPU”, *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 14-17 May 2008, pp. 296-299.

[37] Y. Luo, R. Duraiswami, “Canny edge detection on NVIDIA CUDA”, *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 23-28 2008, Anchorage, AK, USA.

[38] S. I. Park, S. P. Ponce, J. Huang, Y. Cao, F. Quek, “Low-cost, high-speed computer vision using NVIDIA's CUDA architecture”, *2008 37th IEEE Applied Imagery Pattern Recognition Workshop*, Oct. 15-17 2008, Washington, DC, USA.

[39] V. Vineet, P. J. Narayanan, “CUDA cuts: Fast graph cuts on the GPU”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 23-28 June 2008, pp. 1-8.

[40] L. Zhang, R. Nevatia, “Efficient scan-window based object detection using GPGPU”, *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 23-38 2008, Anchorage, AK, USA.

[41] J. Gomez-Luna, J. M. González-Linares, J. I. Benavides, N. Guil, “Parallelization of a Video Segmentation Algorithm on CUDA–Enabled Graphics Processing Units”, *Lecture Notes in Computer Science*, 2009, Vol. 5704/2009, pp. 924-935, DOI: 10.1007/978-3-642-03869-3_85.

[42] M. Boyer, D. Tarjan, S. T. Acton, K. Skadron, “Accelerating leukocyte tracking using CUDA: A case study in leveraging manycore coprocessors”, *2009 IEEE International Symposium on Parallel&Distributed Processing*, May 23-29 2009, Rome, Italy.

[43] T. D. R. Hartley, U. Catalyurek, A. Ruiz, F. Igual, R. Mayo, M. Ujaldon, “Biomedical image analysis on a cooperative cluster of GPUs and multicores”, *ICS '08 Proceedings of the 22nd annual international conference on Supercomputing*, New York, NY, USA ©2008, DOI: 10.1145/1375527.1375533.

[44] M. Modat, G. R. Ridgway, Z. A. Taylor, M. Lehmann, J. Barnes, D. J. Hawkes, N. C. Fox, S. Ourselin, “Fast free-form deformation using graphics processing units”, *Computer Methods and Programs in Biomedicine*, Volume 98, Issue 3, HP-MICCAI 2008, June 2010, pp. 278-284, DOI: 10.1016/j.cmpb.2009.09.002.

[45] A. Faro, D. Giordano, F. Maiorana, C. Spampinato, Discovering Genes and Diseases Associations from Specialized Literature using GRID, IEEE Transactions on Information Technology in Biomedicine, Volume 13, Number 4, pp. 554-560, 2009

[46] N. A. Sinnott-Armstrong, D. Granizo-Mackenzie, J. H. Moore, “High Performance Parallel Disease Detection: an Artificial Immune System for Graphics Processing Units”, *Computational Genetics Laboratory Dartmouth Medical SchoolLebanon, NH 03756*.

[47] B. Bilgic, B. K. P. Horn, I. Masaki, “Fast human detection with cascaded ensembles on the GPU”, *2010 IEEE Intelligent Vehicles Symposium (IV)*, June 21-24 2010, pp. 325-332, DOI: 10.1109/IVS.2010.5548145.

[48] H. Sugano, R. Miyamoto, “Parallel implementation of pedestrian tracking using multiple cues on GPGPU”, *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 900-906, DOI: 10.1109/ICCVW.2009.5457606, Sept. 27-Oct. 4 2009, Kyoto, Japan.

[49] K. Lillywhite, L. Dah-Jye, Z. Dong, “Real-time human detection using histograms of oriented gradients on a GPU”, *2009 Workshop on Applications of Computer Vision (WACV)*, Dec. 7-8 2009, pp. 1-6, Snowbird, UT, USA, DOI: 10.1109/WACV.2009.5403100.

[50] F. Nasse, C. Thurau, G. A. Fink, “Face Detection Using GPU-Based Convolutional Neural Networks”, *Lecture Notes in Computer Science*, 2009, Vol. 5702/2009, pp. 83-90, DOI: 10.1007/978-3-642-03767-2_10

[51] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krüger, Aaron E. Lefohn, and Tim Purcell. "A Survey of General-Purpose Computation on Graphics Hardware". Computer Graphics Forum, volume 26, number 1, 2007, pp. 80-113.