
# Proceedings of the First PhD Symposium on Sustainable Ultrascale Computing Systems (NESUS PhD 2016)
Timisoara, Romania

Jesus Carretero, Javier Garcia Blas
Dana Petcu
(Editors)

February 8-11, 2016

# Towards a Smart Selection of Hybrid Platforms for Multimedia Processing

Sidi Ahmed Mahmoudi and Pierre Manneback

University of Mons, Belgium
sidi.mahmoudi@umons.ac.be

**Abstract**

*Nowadays, images and videos have been present everywhere, they can come directly from camera, mobile devices or from other peoples that share their images and videos. The latter are used to illustrate different objects in a large number of situations. This makes from image and video processing algorithms a very important tool used for various domains related to computer vision such as video surveillance, medical imaging and database (images and videos) indexation methods. The performance of these algorithms have been so reduced due the the high intensive computation required when using new image and video standards. In this paper, we propose a new framework that allows users to select in a smart and efficient way the processing units (GPU or/and CPU) within heterogeneous systems, when treating different kinds of multimedia objects : single image, multiple images, multiple videos and video in real time. The framework disposes of different image and video primitive functions that are implemented on GPU, such as shape (silhouette) detection, motion tracking using optical flow estimation, edges and corners detection. We have exploited these functions for several situations such as indexing videos, segmenting vertebrae in in X-ray and MR images, detecting and localizing event in multi-user scenarios. Experimentation showed interesting accelerations ranging from 6 to 118, by comparison with sequential implementations. Moreover, the parallel and heterogeneous implementations offered lower power consumption as a result for the fast treatment.*

***Keywords*** GPU, Heterogeneous architectures, Image and video processing, Medical imaging, Motion tracking

## I. Introduction

Recently, the architecture of CPUs has so changed and evolved that the number of integrated computing units has been multiplied. This evolution is reflected in both general (CPU) and graphic (GPU) processors which present a large number of computing units, their power has far exceeded the CPUs ones. In this context, image and video processing algorithms are well adapted for acceleration on the GPU by exploiting its processing units in parallel, since they are mainly based on applying the same computation over many points or pixels. Many GPU and parallel computing approaches have been developed recently. Although they present a great power of GPU architecture, any is able to process high definition image and video efficiently and accordingly to the type of Medias (single image, multiple image, multiple videos and video in real time). Thus, there was a need to develop a framework capable of addressing the outlined problem. In literature, we can categorize two types of related works based on the exploitation of parallel and heterogeneous platforms for multimedia processing: one related to image processing on GPU such as presented in [1], [2] which proposed GPU implementations that use CUDA [1] for basic image processing and medical imaging algorithms. A performance evaluation of GPU-based image processing algorithms is presented in [3]. These implementations offered high improvement of performance thanks to the exploitation of the GPU's computing units in parallel. However, these accelerations are so reduced when processing image databases with different resolutions. Thus, an efficient exploitation of CPU, GPU and

[1] CUDA. https://developer.nvidia.com/cuda-zone

hybrid (Multi-CPU/Multi-GPU) platforms is needed with an effective management of the related memories. Notice also that the processing of images with low resolutions cannot benefit from the high power of GPUs since few computations will be launched. This implies an analysis of algorithms complexities before their parallelization. On the other hand, video processing algorithms require generally a real-time treatment. We may find several methods in this category, such as understanding human behavior, event detection, camera motion estimation, etc. These methods apply mainly motion tracking algorithms that can exploit several techniques such as optical flow estimation [4], block matching technique [5], and scale-invariant feature transform (SIFT) [6] descriptors. In this case also, several GPU implementations have been proposed for sparse [7] and dense [8] optical flow computation.

## II. Research idea

Despite the high speedups presented in the previous section, none of the above-mentioned implementations can provide real-time processing of high definition videos. Therefore, we propose a new framework that allows a smart, effective and adapted processing of different type of Medias exploiting parallel and heterogeneous platforms. This framework enables to select the units (GPU or/and CPU) for processing, and also the related implementations to be applied. The latter are selected after checking the type of media to treat and the algorithm complexity. The framework offers several scheduling strategies that allow an equivalent distribution of tasks over the available processors. The data transfer times are also reduced as a result of the efficient management of GPU memories and to the overlapping (CUDA streaming) of data copies by kernels executions. Otherwise, the framework disposes of several GPU-based image and video primitive functions, such as shape detection, motion tracking using optical flow estimation, edges and corners extraction. We have exploited these functions for several situations such as indexing videos, segmenting vertebrae in X-ray and MR images, detecting and localizing event in multi-user scenarios. The primitive functions are presented in detail in our previous publication [9]. Figure 1 illustrate the proposed framework, presenting different applications that can exploit in an adapted way the heterogeneous systems, which offers a low energy consumption as a result for the fast and accelerated treatment. The main contributions of our framework can be summarized within five points :

1. Smart selection of resources (CPU or/and GPU) based on the estimated complexity and the type of media. Additional computing units are exploited only in case of intensive and tasks;

2. Several image and video GPU primitive functions;

3. Efficient scheduling of tasks and management of memories in case of heterogeneous computation;

4. Acceleration of real-time image and video processing applications;

5. low energy consumption.

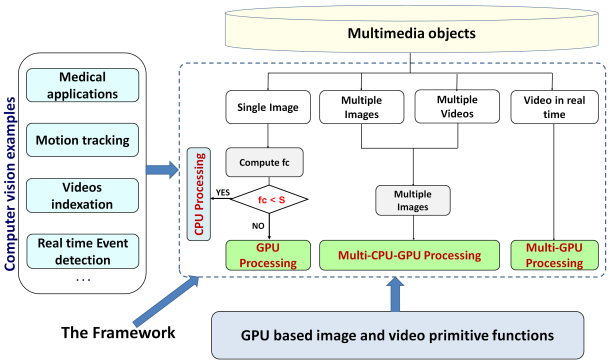


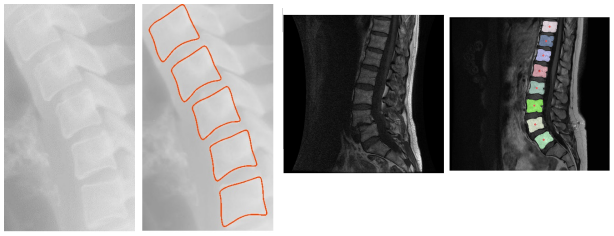

*Figure 1: Multi-CPU/Multi-GPU based Framework for Multimedia Processing*

## III. Experimental results

The proposed framework has been exploited in several high intensive applications related to image and video processing such as vertebra segmentation, videos indexation, event detection and localization, etc.

### III.1 Heterogeneous vertebra segmentation

The main objective of this method is the cervical vertebra mobility analysis on X-Ray or MR images. The aim is to detect vertebra automatically. The computation time presents one of the most important requirements for this application. Based on our framework, we propose a hybrid implementation of the most intensive steps, which have been defined within our estimation complexity equation [9]. Our solution for vertebra detection on Multi-CPU/Multi-GPU platforms is detailed in [10] for X-Ray images, and in [11] for MR images. Fig. 2(a) presents the results of vertebra detection in X-ray images, while Fig. 2(b) is related to present the detected vertebra in MR images. Notice that the use of heterogeneous platforms allowed to improve performance with a speedup of 30 $\times$ for vertebra detection within 200 high resolution (1472$\times$1760) X-ray images, and a speedup of 118 $\times$ when detecting vertebra in a set of 200 MR images (1024$\times$1024).



(a) X-ray images (b) MR images

*Figure 2: Vertebra detection in X-ray and MR images*

### III.2 Multi-CPU/Multi-GPU based videos indexation

The context of this application is to develop a new browsing environment for images and videos databases. This method consists on calculating similarities between videos sequences (composed of consecutive images), based on detecting the feature of images (frames) that compose videos [12]. The main drawback of this application is the high computing time that increases considerably when enlarging videos databases and definitions. Using our framework, we developed a hybrid version of the most intensive step of the features extraction process. This step, detected within our complexity estimation equation defined in [9], consists of contours extraction algorithm that provides relevant information for localizing motion's areas. This implementation is detailed in [13] showing a total gain of 80%, compared to the total time of the sequential version, when treating 800 frames of a video sequence (1080x720).

### III.3 Multi-GPU based Event detection and localization in real time

The aim of this method is to detect and localize events in video sequences in real time. The method is based on modeling normal behaviors, and then estimating the difference between the normal behavior model and the observed events of behaviors. The detected variations are labeled as emergency events, and the deviations from examples of normal behavior are used to characterize abnormality. After the detection of each event, we localize the related areas in video frames where motion behavior is surprising compared to the rest of motion. Using our framework, we developed a Multi-GPU version of the most intensive steps of the application. The latter are detected within our complexity estimation equation defined in [9]. This implementation is detailed in [14]. Notice that performed tests show that our application can turn in multi-user scenarios, and in real time even when processing high definition videos such as Full HD or 4K standards. The scalability of our results is also achieved thanks to the effective exploitation of multiple GPUs. A demonstration of GPU based features detection, features tracking, and event detection in crowd video is shown in this video sequence: `https://www.youtube.com/watch?v=PwJRUTdQWg8`.

## IV. Conclusion and future work

We proposed in this paper a new framework that allows a smart and efficient exploitation of Multi-CPU/Multi-GPU platforms accordingly to the type of multimedia (single image, multiple images, multiple videos, video in real time) objects. This framework

enables to select the units (GPU or/and CPU) for processing, and also the related implementations to be applied. The latter are selected after checking the type of media to treat and the algorithm complexity. Experimental results showed different use case applications that have been improved thanks to our framework. Each application has been integrated in an adapted way for exploiting resources in order to reduce both computing time and energy consumption. As future work, we plan to port our algorithms on GPU Tegra Mobile Processors [2] that allow to reduce significantly the power consumption, with maintaining high performance of computation.

## Acknowledgment

Authors would like to thank the support of European COST NESUS action IC1305 " Network for Sustainable Ultra-scale Computing"

## References

[1] Yang. Z and Zhu. Y and pu. Y, " Parallel Image Processing Based on CUDA " *HPCCE Workshop, IEEE International Conference on Cluster Computing*, pp. 198-201, 2008.

[2] Mahmoudi. Sidi Ahmed and Lecron. F and Manneback. P and Benjelloun. M and Mahmoudi. S, " GPU-Based Segmentation of Cervical Vertebra in X-Ray Images " *HPCCE Workshop, IEEE International Conference on Cluster Computing*, pp. 1-8, 2010.

[3] Park. Kyu and Nitin. Singhal and Man. Hee Lee, " Design and Performance Evaluation of Image Processing Algorithms on GPUs " *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, pp. 1-14, 2011.

[4] Horn. B. K and Schunk. B. G, " Determining Optical Flow " *Artificial Intelligence*, vol. 2, pp. 185-203, 1981.

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

[6] Lowe. D. G, " Distinctive image features from scale-invariant keypoints " *International Journal of Computer Vision (IJCV)*, vol. 60(2), pp. 91-110, 2004.

[7] Mahmoudi. Sidi Ahmed and Kierzynka. Michal and Manneback. Pierre and Kurowski. K, " Real-time motion tracking using optical flow on multiple GPUs " *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 62, pp. 139-150, 2014.

[8] Marzat. J and Dumortier. Y and Ducrot. A, " Real-time dense and accurate parallel optical flow using CUDA " *In Proceedings of WSCG*, pp. 105-111, 2009.

[9] Mahmoudi. Sidi Ahmed and Manneback. Pierre, " Multi-CPU/Multi-GPU Based Framework for Multimedia Processing " *Computer Science and Its Applications*, vol. 456, pp. 54-65, 2015.

[10] Lecron. Fabian et al., " Heterogeneous Computing for Vertebra Detection and Segmentation in X-Ray Images " *International Journal of Biomedical Imaging: Parallel Computation in Medical Imaging Applications*, vol. 2011, pp. 1-12, 2011.

[11] Larhmam. Mohammed Amine et al., " A Portable Multi-CPU/Multi-GPU Based Vertebra Localization in Sagittal MR Images ", *International Conference on Image Analysis and Recognition, ICIAR 2014*, pp. 209-218, 2014.

[12] Damien Tardieu et al., " Video Navigation Tool: Application to browsing a database of dancers' performances " , *QPSR of the numediart research program*, vol. 2, number. 3, pp. 85-90, 2009.

[13] Mahmoudi Sidi Ahmed and Manneback Pierre, " Efficient exploitation of heterogeneous platforms for images features extraction " *3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 91-96, 2012.

[14] Mahmoudi Sidi Ahmed and Manneback Pierre, " Multi-GPU based event detection and localization using high definition videos " *International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 81-86, 2014.

[2] Tegra Mobile Processors :http://www.nvidia.com/object/tegra.html