Ph.D Annual Progress Report

# Visual Saliency model on Multi-GPU

Anis ur RAHMAN

Supervised by: Dominique HOUZET & Denis PELLERIN

GIPSA-lab, 961 rue de la Houille Blanche
38402 St Martin d'Heres, France

June 28, 2010

**Abstract**

The human vision has been studied deeply in the past years, and several different models have been proposed to simulate it on computer. Some of these models concerns visual saliency which is potentially very interesting in a lot of applications like robotics, image analysis, compression, video indexing. Unfortunately they are compute intensive with tight real-time requirements. Among all the existing models, we have chosen a spatio-temporal one combining static and dynamic information. We present in this report a very efficient implementation of this model with multi-GPU reaching real-time. We present the algorithms of the model as well as several parallel optimizations on GPU with higher precision and execution time results. The real-time execution of this multi-path model on multi-GPU makes it a powerful tool to facilitate many vision related applications.

## 1 Introduction

Visual attention models translate the capability of human vision to concentrate only on smaller regions of the visual scene. More precisely, such regions are the spotlight of focus, which either may be an object or a portion of the scene. A number of modalities are used to locate regions of attention like intensity, color, orientation, motion, and many others. The attention model acts as an information processing bottleneck to reduce the overall information into a region of useful information. This model when guided by salient stimuli falls into a category of bottom-up approach, which is fast and primitive. On the other hand, models driven by cognition using variable selection criteria are the basis for top-down approaches, and are slower and more complex. The human visual system uses either saliency-based or top-down approach, or the combination of both these approaches to find the spotlight of focus.

## 2 Goals of the thesis

- to improve the existing visual attention model by adding different paths i.e. face recognition, sound, color, characters etc
- to optimize the parallel algorithms using specific data structures to split the computation among multi-GPUs
- to virtualize memory management for 2D/3D data processing, and implement a tuneable library of data structures and access will be built, adapted to image processing applications
- to validate the model on a real application (bio-inspired robot, video indexing)
- to improve the model, and compare the performance gains

The scientific aim of this study is the proposition of variants for algorithms dedicated to visual attention, allowing better results quality as well as higher execution performances.

## 3 Spatio-temporal model

The bottom-up visual saliency model [4] implemented on GPU mimics the human vision system all the way from the retina to the visual cortex. The model uses a saliency map to determine where the source of attention lies within the input scene, which may further be used to initiate other tasks. Also, it is linearly modeled and based on the human visual system. The forking of the entire pathway into different sub-paths using various modalities is more efficient to compute. In the end, the output of both pathways is combined into a final saliency map using several adaptive coefficients like mean, maximum, and skewness. The model is validated against large datasets of images, and the results are compared against that of a human visual system using an eye tracker. The model is efficient, and results in a stable prediction of eye movements.
This model to predict the areas of concentration finds its worth in applications like for video content analysis to perform structural decomposition to build indexes, for video reframing to deliver comforting viewing experience on mobile devices, for video compression to reduce the bandwidth required to transmit, for realistic video synthesis. The notion of sketching a biologically-inspired model is to build robust and all-purpose vision systems adaptable to various environmental conditions, users, and tasks.

### 3.1 Merit to qualify for parallelization

Often visual saliency models incorporate a number of complex tasks that make a real-time solution quite difficult to achieve. This objective is only achievable by simplification of the overall model as done by Itti [3] and Nabil [5]. Resultantly, making impossible the inclusion of other processes into the existing model. Over the years, GPUs have evolved from fixed function architecture into completely programmable shader architecture. All together with a mature programming model like CUDA [1] makes the GPU platform a preferable choice for acquiring high performance gains. Generally, vision algorithms are a sequence of filters that are relatively easier to implement on GPU's massively data-parallel architecture. Also, the graphics device is cheaper, accessible to everyone, and simple to program than its counterparts.

### 3.2 Which, and why

The static pathway includes the retina filter with low-pass filters using 2D convolutions, the normalizations with reduction operations, shifts operations, and Fourier transforms. On the other hand, the dynamic pathway involves recursive Gaussian filters, projection, modulation/demodulation, and kernels for calculation of spatial and temporal gradients are simple and classically implemented. Notably, the kernels that are compute-intensive and interesting to be

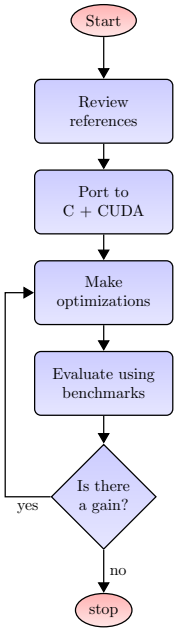


Figure 1: Proces flow for porting onto GPU

implemented on the GPU are: interactions kernel and gabor filter bank from the first pass, while the motion estimator kernel from the second pass. Hence, we present these kernels because they can be potentially improved by employing various optimizations, and making several adjustments to the kernel launch configurations.

## 3.3 Importance of optimizations

Naturally, due to immense computational power of the graphics device, the memory management becomes more critical. There are several strategies to achieve the desired performance like memory coalescing to execute burst memory accesses, and to use the cache memories, internal shared memories and registers. Most importantly, shared memory is an on-chip high bandwidth memory shared among all the threads on a single SM. It has several uses: as a register extension, to avoid global memory accesses, to give fast communication among threads, and as a data cache.

## 3.4 General guidelines

In short, the porting process of saliency model can be summarized into a number of key points:

- Identify parts in the source code for data initialization, compute intensive task and data retrieval.
- Aligning a structure type containing 16 bytes of data to 16-byte boundary provide coalesced memory transaction for the global memory thus increasing global memory bandwidth efficiency.
- Page-locked memory buffer can be requested for higher data transfer rate however multiple requests over small amount of data degrade the overall performance significantly.
- Examine computation demanding task for data dependencies as well as sequence. This is important to recognize whether the computation can be done in parallel. Usually an arrays or matrices operation that is done in a loop without any dependencies can be directly ported.

- Although the global memory has the highest latency among the other types of memory in the device, the latency can be hidden by compute intensive instructions.
- Memory management is crucial in getting most of the performance from CUDA. By using shared memory as a temporary memory for operations could increase the performance.
- The amount of size for memory transfer must be kept at minimum as possible for efficiency. Generally, a big ratio between computations to memory transfer must be maintained.
- For multi-dimensional array computation, it is more convenient to flatten the array for memory allocation and memory transfer but the data will still be able to be computed in multi-dimensional computational model by managing the kernels blocks and threads.

### 3.5 Multi-GPU implementation

Multi-GPU implementation is quite interesting to increase the computational efficiency of the entire visual saliency model. We have employed a shared-system GPU model, where multiple GPUs are installed on a single CPU. If the devices need to communicate, they do it through the CPU with no inter-GPU communication. A CPU thread is created to invoke kernel execution on a GPU, accordingly, we will have a CPU thread for each GPU. To successfully execute our single GPU solution on multi-GPUs, the parallel version must be deterministic. Our first implementation, the two pathways of the visual saliency model; static and dynamic, are completely separate with no inter-GPU communication required. The resulting saliency maps are simply copied-back to the host, where they can be fused together into the final saliency map.

- valuating aspects and affects of using dynamic modalities in the saliency model
- integration of the dynamic pathway into the existing platform
- implementing the model on a shared-system multi-gpu system, to prepare the platform to be used on cluster of GPUs

## 4 Obtained results

All implementations are tested on a 2.80GHz quad core system with 12GB of main memory, and Windows 7 running on it. On the other hand, the parallel version is implemented using latest CUDA v3.0 programming environment on NVIDIA GTX285 series graphics cards. This graphics device consists of 24 SMs with 240 scalar cores in total, 16KB shared memory, and 8192 registers per SM.

### 4.1 Speedup for static pathway

The implementation of the static pathway on GPU resulted in performance gains, which is illustrated in 2 for various image sizes. The original code for the pathway is in MATLAB, which is redone using C for the sequential and multithreaded solutions resulted in performance gains of 2x and 3.5x respectively. On the other hand, the CUDA implementation resulted in speedup of more than 400x to the MATLAB solution.

### 4.2 Speedup for Dynamic pathway

The CUDA solution resulted in an immense speedup against its sequential MATLAB and C counterparts as illustrated in table 1. The speedup are for three input image datasets ”Treetran”, ”Treediv” and ”Yosemite”, where the first two are of resolutions $150\times150$ pixels, while the last one is of $316\times252$ pixels.

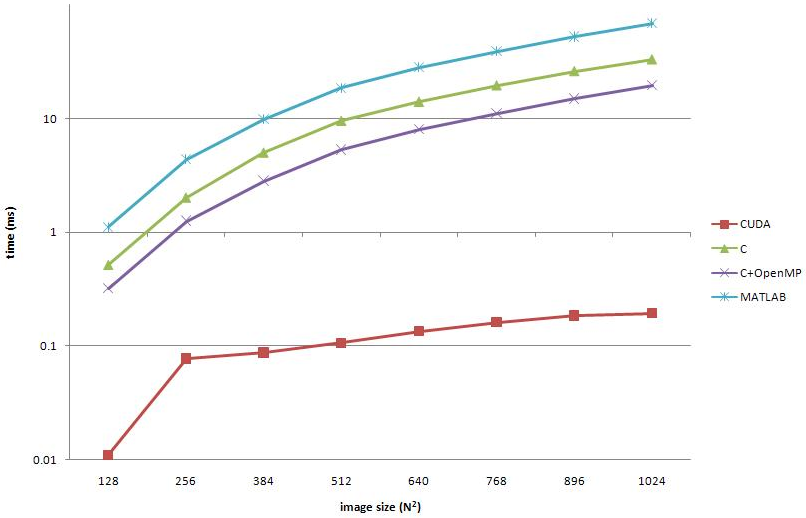


Figure 2: Timings of different versions of static pathway

Table 1: Timings for the dynamic pathway after Optimizations

| | treetran | treediv | yosemite |
|---|---|---|---|
| MATLAB | 13.30s | 12.86s | 46.61s |
| C | 1.75s | 1.76s | 6.28s |
| CUDA | 0.12s | 0.12s | 0.30s |

## 4.3 Precision

The original code is developed using double precision in MATLAB, hence it is important to verify the effects of lower precision support on graphics devices. It mainly depends on the algorithm implemented, whether it can produce usable results or not. We have checked the accuracy of our results to be 99.66% using universal image quality index [6]. Also, we found that with each progressing stage the useful information increases that lead to more overall accuracy of the pathway.

## 4.4 Real-time streaming solution

After the parallel implementation of the visual saliency algorithm, we used OpenCV to demonstrate the real-time processing. The demonstration is done on a quad core machine with three GPUs installed, and the library is used to interface with the webcam. This resulted in execution of the static pathway at 28 fps on the platform shown in the figure 3. Finally, the performance gains on GPU will enable our model to be used for various applications such as automatic video reframing process [2]. This application extracts a cropping window using the regions of interest from the model. These windows are then smooth to increase the viewing experience.

## 4.5 Activity report

Drawing on these results, several publications have been written, as well as other activities during the year are listed below:

Figure 3: Platform for real-time solution

**International Journal:**

- Rahman, A.; Houzet, D.; Pellerin, D.; Guyader, N. & Marat, S. Parallel Implementation of a Spatio-temporal Visual Saliency Model. Journal of Real-Time Image Processing, JRTIP. Elsevier (2010) {Accepted}

**Conference:**

- Houzet, D.; Huet, S. & Rahman, A. SysCellC: a data-flow programming model on multi-GPU. International Conference on Computational Science, ICCS '10 (2010) {Accepted}
- Rahman, A.; Houzet, D. & Pellerin, D. GPU implementation of motion estimation for visual saliency. Conference on Design and Architectures for Signal and Image Processing, DASIP (2010) {Submitted}

**Book chapter:**

- Rahman, A.; Houzet, D. & Pellerin, D. Visual Saliency model on Multi-GPU. mei Hwu, W. (ed.) GPU Computing Gems. Elsevier (2010) {Accepted abstract}

**Presentations:**

- Rahman, A. Parallel implementation of visual saliency model on GPU. Presentation during CUDA programming course, Nov, GIPSA-lab (2009)
- Rahman, A. Bio-inspired visual attention model on GPU. Invited speaker for the meeting L'utilisation des GPU pour les applications de traitement du signal et des images, GDR-ISIS, 06th May, Telecom Paristech (2010)

**Poster:**

- Rahman, A.; Houzet, D.; Pellerin, D.; Guyader, N. & Marat, S. Parallel Implementation of a Visual Saliency Model, NVIDIA Research Summit, GTC '09, Oct, San Jose, USA (2009)

- Rahman, A.; Houzet, D.; Pellerin, D.; Guyader, N. & Marat, S. Visual saliency on multi-GPU. CNRS - EEFTIG '10 Ecole d'été francophone de traitement d'image sur GPU, 29 June - 02 July, Grenoble (2010)

**Courses taken:**

- **Introduction du reparti calcul:** The aim of this course is to provide an introduction to supercomputing, parallelism and their concepts and tools. It is based on numerous practical examples illustrated in the context of TP. **(36h)**
- **Recherche operationelle:** The course give insight of graph theory, shortest paths algorithms, scheduling, maximum flow problems, and linear Programming. **(16h)**
- **Sensibilisation á la propriété industrielle:** Intellectual property and industrial property in particular are essential tools for the legal protection of innovations from the company and research laboratories. **(10h)**
- **Gestion de projets:** The aim of this course is to understand the organization of an industrial project or research in order to position themselves effectively. After an overview, we descend into the level of detail to offer behaviors of individual agents that promote the collective productivity. The study will be of practical scenarios with use of software to promote productivity. **(20h)**
- **Initiation á l'éthique:** The course is an introduction to ethics, its importance in our daily life, its relation to religion, philosophy and science. In the end, several scenario from the area of research are presented to create an ethical sense in a researcher. **(12h)**
- **Industrial marketing:** To present main strategies of the companies on the markets, and the knowledges and the know hows to allow the researchers to place their work in the middle of innovation processes. **(18h)**

**Courses assisted:**

- Ordinateurs et microprocesseurs **(24h TD)**
- CNRS - EEFTIG '10 Ecole d'été francophone de traitement d'image sur GPU, 29 June - 02 July, Grenoble (2010) **(8h TP)**

# 5 Emerging point of view

Beyond their appeal as cost-effective HPC accelerators, GPUs also have the potential to significantly reduce space, power, and cooling demands, and reduce the number of operating system images that must be managed relative to traditional CPU-only clusters of similar aggregate computational capability. In support of this trend, NVIDIA has begun producing commercially available Tesla GPU accelerators tailored for use in HPC clusters.
Although successful use of GPUs as accelerators in large HPC clusters can confer the advantages outlined above, they present a number of new challenges:

- Resource sharing
- Heath monitoring and data security
- Node allocation
- code development tools

# 6 Future work plan

## 6.1 Porting to GPU cluster

To build a cluster of CPU-GPU nodes with the following characteristics:

- GPU computation enabled with supplementary CPU-GPU communication overlapping capabilities
- fast interconnection network in between nodes
- enough power supply on each node to support several devices.

After setting up the cluster with the characteristics desired, we can evaluate various parameters and tools for potential of GPU for high performance computing in the context of image processing.

- **Power consumption:** To evaluate the GPU power consumption correlated with different kernel workloads, we can evaluate measures for power consumption for a memory-access intensive kernel, and another for a compute-intensive kernel.
- **Host-device bandwidth and latency:** The measurements of sending data between the host and device memories i.e when the data is sent from the memory attached to the same CPU.
- **Benchmark:** To study the peak CPU-GPU single node performance against peak node performance on multiple nodes. It will be interesting to use different precision, and also test other benchmarks.
- **Development tools:** To evaluate different GPU programming languages and toolkits. It is evident that many of the HPC applications have been implemented using MPI for parallelizing the application. The simplest way to start building an MPI application that uses GPU-accelerated kernels using NVIDIAs nvcc compiler.

## 6.2 Developing a multicore solution

To prepare a multicore solution for comparison purpose against using GPUs. With the continuing increase in the number of cores with each CPU generation, there will be a significant need for efficient mechanisms for sharing GPUs among multiple cores, particularly for legacy MPI applications that do not use a hybrid of shared-memory and message passing techniques within a node. Resultantly, This implementation will provide us a testbed to evaluate:

- usage of OpenMP to take advantage of the multicore CPUs
- usage of different MPI implementation which support different interconnection networks (OpenMPI)

## 6.3 Inclusion of pathway

The main advantage of the performance gain accomplished will allow the inclusion of face recognition, stereo, audio, and other complex processes. The three pathway model with face recognition is already evaluated to get more accurate salient regions. Whereas, a colleague is also investigating and modeling sound as a stimuli for attention.

## 6.4 Creation of a library or tool

It is our main objective to prepare an application or tool to investigate the correctness of the model for decision making. This will be possible due to the faster platform for investigating different videos, and also it will be used by other collaborators (e.g DPC Dept., GIPSA-lab) interested to evaluate our model. Finally, the performance gains on GPU will enable our model to be used for various applications such as automatic video reframing process. This application will extract a cropping window using the regions of interest from the model. These windows can be smoothened to increase the viewing experience.

# 7 Conclusion

In the report, we presented the multi-GPU implementation of a visual saliency model to identify the areas of attention. The main advantage of the performance gain accomplished will allow the inclusion of face recognition, stereo, audio, and other complex processes. Consequently, this real-time solution finds a wide application for several research and industrial problems i.e. video compression, video reframing, frame quality assessment, visual telepresence and surveillance, automatic target detection, robotics control, super-resolution, computer graphics rendering, and many more.

# References


[1] *NVIDIA CUDA Compute Unified Device Architecture - Programming Guide*, 2007.

[2] C. Chamaret and O. Le Meur. Attention-based video reframing: Validation using eye-tracking. In *ICPR08*. 2008.

[3] L. Itti. Real-time high-performance attention focusing in outdoors color video streams. In B. Rogowitz and T. N. Pappas, eds., *Proc. SPIE Human Vision and Electronic Imaging VII (HVEI'02), San Jose, CA*, pp. 235–243. SPIE Press, 2002.

[4] S. Marat, T. Ho Phuoc, et al. Modelling spatio-temporal saliency to predict gaze direction for short videos. *Int. J. Comput. Vision*, 82:231–243, 2009.

[5] N. Ouerhani and H. Hgli. Real-time visual attention on a massively parallel simd architecture. *Real-Time Imaging*, 9:189–196, 2003.

[6] Z. Wang and A. C. Bovik. A universal image quality index. *Signal Processing Letters, IEEE*, 9:81–84, 2002.