GPU implemenation of motion estimation for visual saliency

{Anis.Rahman, Dominique.Houzet, Denis.Pellerin, Lionel.Agud}@gipsa-lab.grenoble-inp.fr

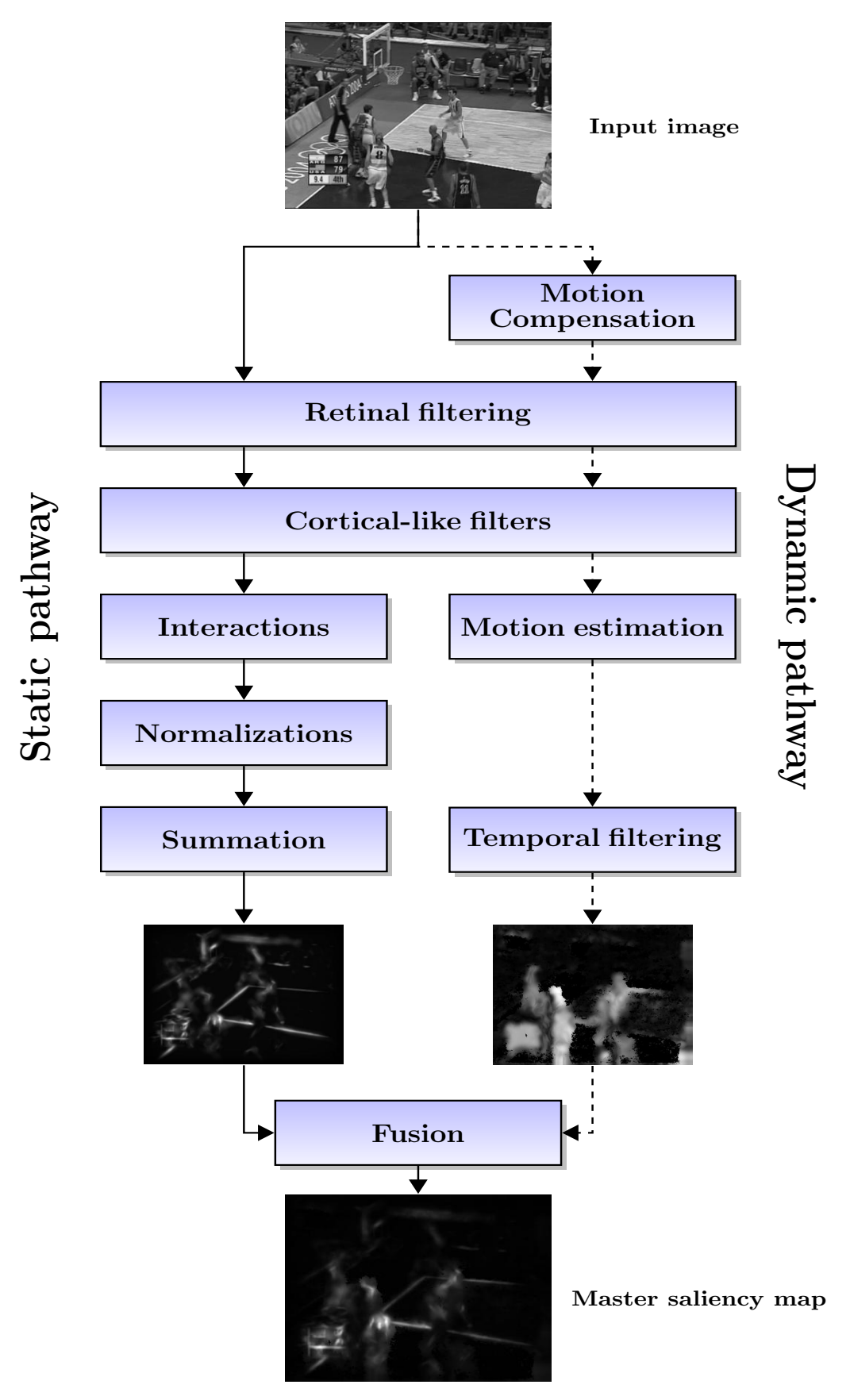
Introduction

In this work, we implement a spatio-temporal visual saliency model [3] insipired from Itti's model [1] on GPU. The model divides the visual information contained in videos into two types: static and dynamic that are processed by two separate pathways. The dynamic pathway of the model involves compute-intensive motion estimation,that when implemented on GPU resulted in a speedup of up to 40x against its sequential counterpart. The implementation involves a number of code and memory optimizations to get the performance gains, resultantly materializing real-time video analysis capability for the visual saliency model.

Visual saliency model

The human vision has been studied deeply over 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.
The bottom-up visual saliency model is used to determine where the source of attention lies and the amount of concentration used to contribute or initiate other tasks. The model is interesting because:

- is linearly modeled all the way from the retina to cortical cells
- retinal output causes separation of useful information into two distinct signals that are more efficient to process
- motion compensation extracts only the moving parts against its background
- motion estimation is used to carry out the motion contrast map



the saliency outputs from both static and dynamic pathways are fused together to get the final saliency map.

Motion estimator

The motion estimator [2] presented here employs a differential method using Gabor filters to estimate local motion. The motion vectors are calculated by averaging the movement to its spatial neighborhood. This spatial continuity within the optical flow is achieved by the convolving the spatio-temporal image sequence with a Gabor filter bank:

$$v_x \cdot \frac{\partial (I * G_i)}{\partial_x} + v_y \cdot \frac{\partial (I * G_i)}{\partial_y} + \frac{\partial (I * G_i)}{\partial_t} = 0$$

the 2D filter bank comprises of N filters Gi with the same radial frequencies. Hence, resulting in a system of N equations for each pixel:

$$\begin{pmatrix} \Omega_2^x & \Omega_1^y \\ \Omega_2^x & \Omega_2^y \\ & \dots \\ \Omega_n^x & \Omega_n^y \end{pmatrix} \cdot \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} \Omega_2^t \\ \Omega_2^t \\ \dots \\ \Omega_n^t \end{pmatrix}$$

this over-determined system is solved using least squares method(Biweight Tuckey test). The approximation starts with a sub-sampled image from the highest level of the pyramid. This multi-pattern approach allows robust estimation of motion for every pixel.

References

[1] L. Itti, C. Koch, E. Niebur. A Model of Saliency-based Visual Attention for Rapid Scene Analysis In *IEEE Trans. Pattern Analytical Machine Intelligence 1998*

[2] E. Bruno, D. Pellerin. Robust Motion Estimation using Spatial Gabor-like Filters In *Signal Processing 2002*

[3] S. Marat, T. Ho Phuoc, L. Granjon, N. Guyader, D. Pellerin, A. Guérin-Dugué. Modelling Spatio-temporal Saliency to Predict Gaze Direction for Short Videos In *International Journal of Computer Vision 2009*

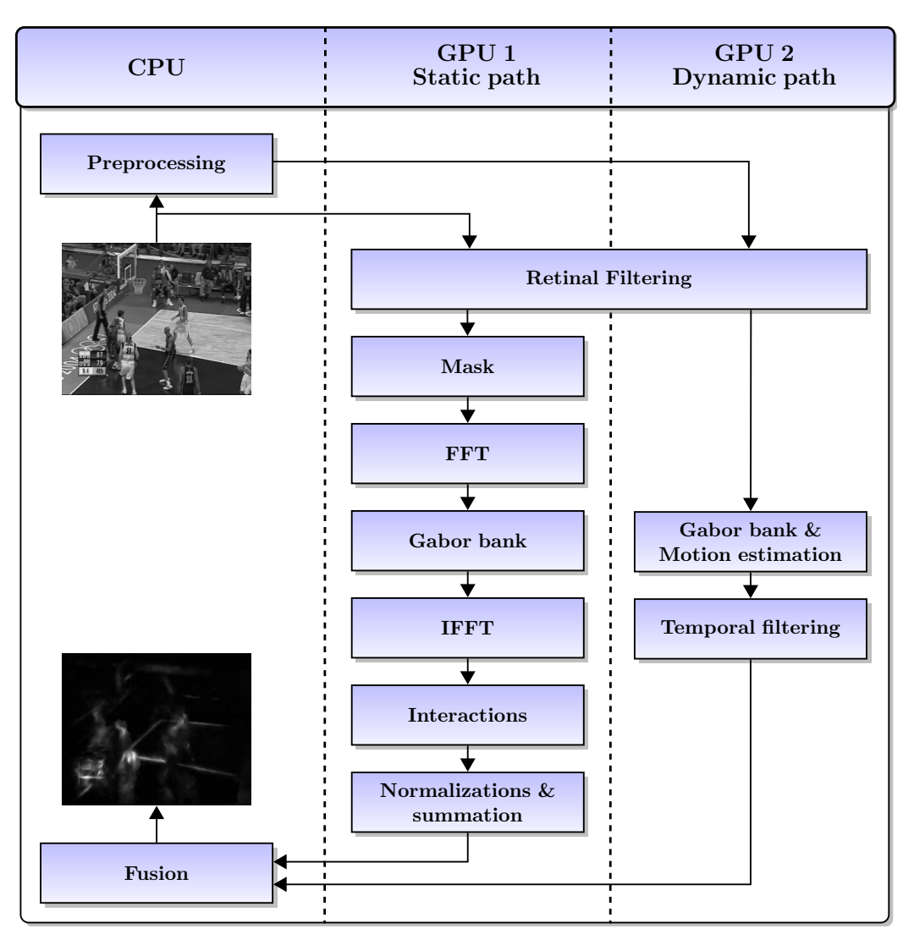
[4] A. Rahman, D. Houzet, D. Pellerin, S. Marat, N. Guyader. Parallel Implementation of a Spatio-temporal Visual Saliency Model In *Journal of Real-Time Image Processing 2010*

Algorithm

```
    Input: Pyramid of sub-sampled images
    Output: Velocity vectors $V$
 1  foreach Level $k$ of pyramid do
 2      if $k == K$ then
 3          Gabor filtering $I(x,y,t)$
 4          Gabor filtering $I(x,y,t\text{-}1)$
 5      else
 6          Projection $V$
 7          Interpolation $I(x,y,t)$
 8          Gabor filtering $I(x,y,t)$
 9          Gabor filtering $I(x,y,t-1)$
10      $(G_x, G_y, G_t) \leftarrow$ Calculate gradients
11      Vector $v \leftarrow$Resolution $I(x,y,t)$, $I(x,y,t\text{-}1)$, $G$
12      if $k == K$ then
13          $V \leftarrow v$
14      else
15          $V \leftarrow V + v$
16      Gaussian filtering $V$
```

Parallel Implementation

Over the years, computer graphics hardware has evolved into a desirable choice for general-purpose computations. These devices are also cheap, accessible to everyone, and easier to program. Our implementation executed at 25 fps on a multi-GPU platform with 3 Geforce 285GTX devices for image size 150 × 150.

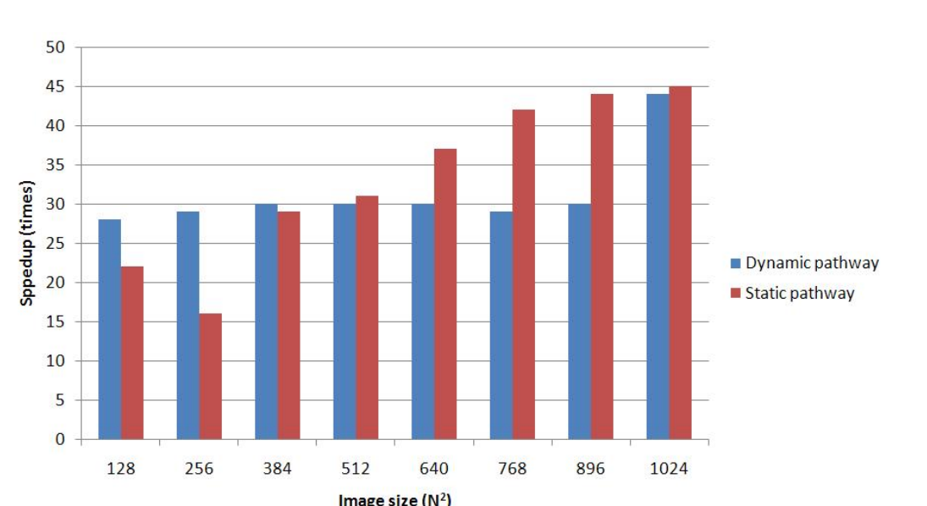


Block diagram of 2-GPU implementation

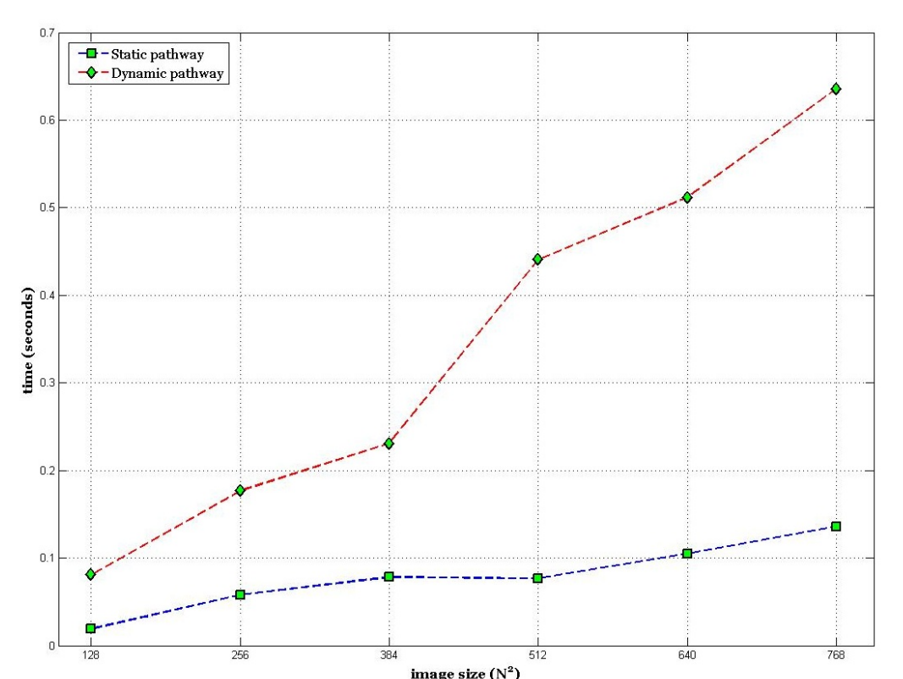
Multi-GPU platform

Speedup

Our implementation shows that the model effectively maps onto GPU, hence results in speedups of upto 40x against sequential C code.



Speedups for two pathways versus sequential C implementation



Timings for two pathways on GTX 285 for various image sizes

Results

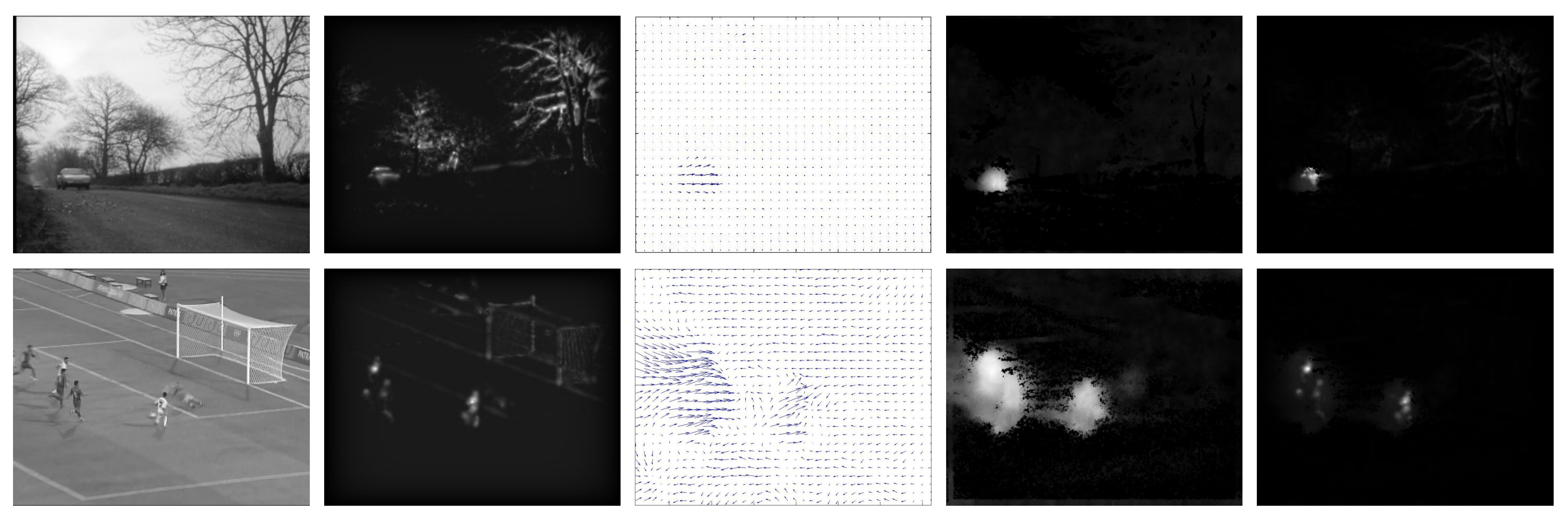


Image | Static saliency map | Velocity vectors | Dynamic saliency map | Saliency map

Conclusion

The real-time processing for our model would create an opportunity of inclusion of many other complex processes or pathways into the existing model, for example: color, face recognition, audio, and many more. Also, the performance gains on GPU will enable our model to be used for various applications such as scene recognition for mobile robots, video compression, computer graphics rendering.