

Moving Object Detection Strategy for Augmented-Reality Applications in a GPGPU by Using CUDA

Daniel Berjón, Carlos Cuevas, Francisco Morán, and Narciso García

Abstract— A spatial-color-based non-parametric background-foreground modeling strategy in a GPGPU by using CUDA is proposed. This strategy is suitable for augmented-reality applications, providing real-time high-quality results in a great variety of scenarios.

I. INTRODUCTION

The last generation of consumer electronic devices, such as smart-phones, tablets or home and portable video game consoles, are endowed with Augmented-Reality (AR) tools [1]. These tools require new, fast and efficient (i.e. lightweight) computer vision applications [2] where moving object detection is a key step for several high level analysis tasks such as segmentation, tracking, classification, or event detection.

Recently, non-parametric based moving object detection strategies, able to provide satisfactory results (low number of misdetections and false detections) in complex scenarios with dynamic backgrounds, non-static cameras and illumination changes, have been proposed [3]. These strategies, modeling the background and the foreground from spatial-color sets of reference samples, are able to improve the quality provided by other strategies [4]. Nevertheless, they require the evaluation of very large amounts of multidimensional computations and, thus, they are not suitable for AR applications.

However, these strategies can greatly benefit from moving them to a modern programmable general-purpose graphical processing unit (GPGPU). These devices are becoming commonplace in regular computers and are even advancing into mobile computing scenarios [5]. They provide very good numeric performance and a high degree of parallelism, being capable of executing simultaneously hundreds or even thousands of threads concurrently [6].

In this paper we propose a novel non-parametric based moving object detection strategy in a GPGPU by using CUDA (Compute Unified Device Architecture) [7], which provides real-time high-quality results in a great variety of scenarios and, accordingly, can be used in AR applications.

II. NON-PARAMETRIC MODELING

Let us consider a pixel x^n , in the current image, defined by its RGB color components (R^n, G^n, B^n) and by its coordinates (r^n, c^n) . The probability of x^n belonging to the background, β , or to the foreground, ϕ , can be evaluated using Bayes'

theorem:

$$\Pr(\phi | x^n) = \frac{\Pr(\phi) p(x^n | \phi)}{\Pr(\phi) p(x^n | \phi) + \Pr(\beta) p(x^n | \beta)}$$

where $\Pr(\phi)$ and $\Pr(\beta)$ are the prior probabilities for foreground and background, $p(x^n | \phi)$ is the probability density function (pdf) estimated for the foreground and $p(x^n | \beta)$ is the correspond pdf estimated for the background.

Applying 5-dimensional Gaussian kernels with diagonal bandwidth matrices, $H = \text{diag}(h_R, h_G, h_B, h_r, h_c)$, the background pdf for x^n can be estimated, from N_β reference samples, $x_{\beta,i} = (R_{\beta,i}, G_{\beta,i}, B_{\beta,i}, r_{\beta,i}, c_{\beta,i})$, from the T_β previous images, as:

$$p(x^n | \beta) = \frac{1}{N_\beta (2\pi)^{5/2} |H|^{1/2}} \sum_{i=1}^{N_\beta} \prod_{j=1}^5 \exp\left(-\frac{(x^n(j) - x_{\beta,i}(j))^2}{2h(j,j)}\right)$$

and the foreground pdf can be estimated from N_ϕ reference samples, $x_{\phi,i} = (R_{\phi,i}, G_{\phi,i}, B_{\phi,i}, r_{\phi,i}, c_{\phi,i})$, from the foreground pixels detected along the previous T_ϕ images as a mixture of a uniform function, γ , and a kernel density estimation:

$$p(x^n | \phi) = \alpha\gamma + \frac{(1-\alpha)}{N_\phi (2\pi)^{5/2} |H|^{1/2}} \sum_{i=1}^{N_\phi} \prod_{j=1}^5 \exp\left(-\frac{(x^n(j) - x_{\phi,i}(j))^2}{2h(j,j)}\right)$$

where α is a mixture factor.

To decrease the computational cost of this non-parametric based strategy, we propose to fix the bandwidth matrices and to use lookup tables (LUT). Nevertheless, for each pixel at each new image, a very large amount of memory accesses are required (typically, more than 10^7 samples) and, as a result, background and foreground modeling cannot be efficiently computed in typical Central Processing Units (CPUs).

III. GPU-CUDA IMPLEMENTATION

Our algorithm, described in Fig.1, very naturally lends itself to parallelization because belonging to foreground or background is evaluated in a per-pixel basis, which allowed us to devote a thread for each pixel in every stage of the process. New frames are transferred to the global memory of the GPU as textures, which allows for faster read operations. Then each of the steps of the algorithms is implemented in parallel stages as CUDA kernels, which we describe in the following paragraphs.

Background model: The input image is tessellated into windows sized 16x16 pixels to be processed by one CUDA thread block each. To evaluate the sum of Gaussians on each pixel, we need to retrieve all the samples from past images belonging to the window and its neighborhood. However, most pixels (those inside the window) will be input to several

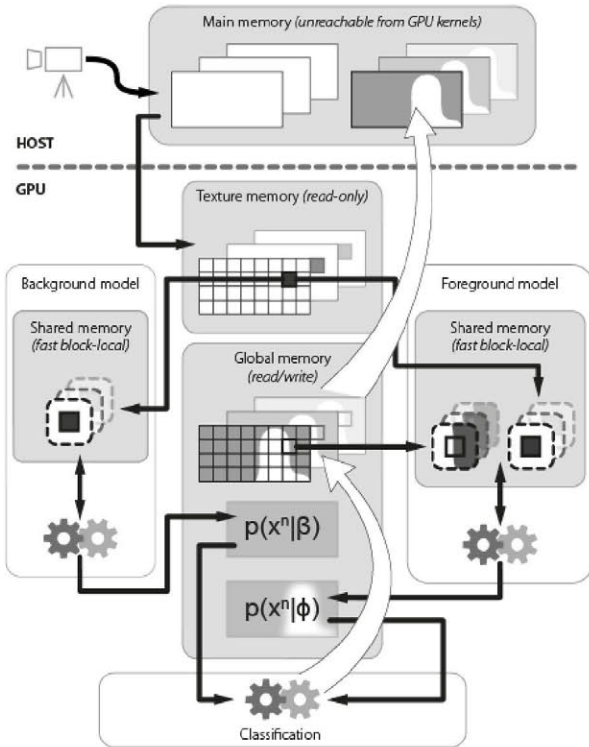


Fig. 1. Memory flows for the proposed algorithm.

threads, so we avoid reading them several times from the relatively slow texture memory. Pixels to be read are distributed evenly among the threads in the block, which concurrently copy them into the shared memory region, which is much faster. Finally, each thread computes the value of the sum of Gaussians for its own pixel. High arithmetic intensity and operands stored in a fast memory region make this operation very efficient.

Foreground model: The process for this model is largely the same than that for the background model. However, in addition to the current image, this model requires the past foreground images. Since holding in memory both the whole past input images and the past foreground images does not make sense, the latter being a subset of the former, foreground images are stored as binary masks packed as 16-bit integers. This coding method allows the threads to disregard in a single comparison a whole row of the 16x16 window, greatly improving the speed of this phase despite the fact that the input spatial window is usually larger than in the background model.

Classification: Only one read operation per pixel is needed, so the shared memory strategy is unnecessary. However, the image is tessellated in 256x16 windows. Thus, each thread is responsible for a 16x1 sub-window, yielding a whole 16-bit mask as described earlier, thereby eliminating the need for synchronization among threads.

IV. RESULTS

Fig. 2 depicts the quality of the obtained results in different scenarios containing multimodal backgrounds. These results

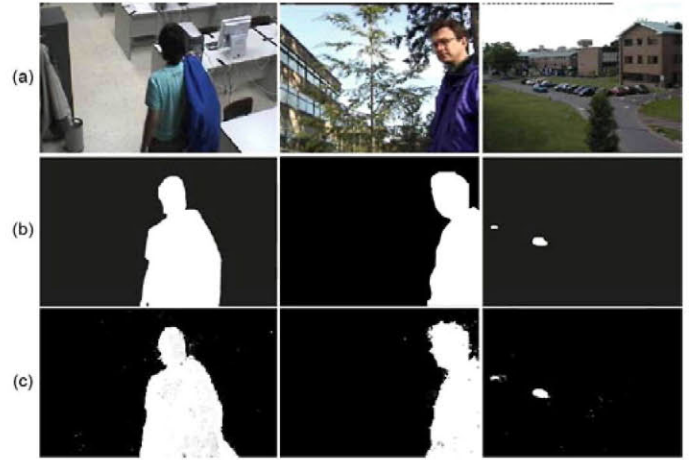


Fig. 2. Quality detection analysis in different scenarios. (a) Original images. (b) Ground truth. (c) Detections obtained with the proposed strategy.

show that the proposed strategy provides high-quality detections where moving objects are correctly identified and well defined. Finally, Table I resumes with the computational reduction resulting from the use of the parallel programming with CUDA.

TABLE I
MEAN PROCESSING TIMES (SECONDS)

| | CPU | GPGPU | Speedup |
|---------------------|-----|-------|---------|
| Background modeling | 24 | 0.15 | 160x |
| Foreground modeling | 8 | 0.04 | 200x |
| Overall strategy | 32 | 0.19 | 168x |

V. CONCLUSIONS

Here we have described a novel background-foreground non-parametric based moving object detection strategy in a GPGPU by using CUDA.

Obtained results have demonstrated that the proposed strategy is able to provide high-quality real-time results in a large variety of multimodal and complex scenarios and, therefore, it is suitable for AR tools required by the last generation of consumer electronic devices endowed with integrated smart cameras.

REFERENCES

- [1] D. W. F. van Krevelen, and R. Poelman, "A Survey of Augmented Reality Technologies, Applications and Limitations", *The International Journal of Virtual Reality*, vol. 9, no. 2, pp. 1-20, 2010.
- [2] A. Mulloni, and T. Drummond, "Real-Time Detection and Tracking for Augmented Reality on Mobile Phones", *IEEE Trans. Visualization and Computer Graphics*, vol. 16, no. 3, pp. 355-368, 2010.
- [3] Y. Sheikh, O. Javed, and T. Kanade, "Background Subtraction for Freely Moving Cameras", *IEEE Int. Conf. Computer Vision*, pp. 1219-1225, 2009.
- [4] Y. Sheikh, and M. Shah, "Bayesian modeling of dynamic scenes for object detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1778-1792, 2005.
- [5] S. P. Mohanty, "GPU-CPU Multi-Core For Real-Time Signal Processing", *Int. Conf. Consumer Electronics*, 2009.
- [6] S. F. Tsai, C. C. Cheng, C. T. Li, and L. G. Chen, "A Real-Time 1080p 2D-to-3D Video Conversion System", *IEEE Trans. Consumer Electronics*, vol. 57, no. 2, pp. 915-922, 2011.
- [7] NVIDIA CUDA development team, "NVIDIA CUDA Compute Unified Device Architecture Programming Guide", NVIDIA corporation, 2007.