FAST PATTERN MATCHING IN 3D IMAGES ON GPUS

Patrick Eibl[♯], Dennis Healy[♮], Nikos P. Pitsianis[†‡] and Xiaobai Sun[†]

[♯]Duke University, Department of Electrical and Computer Engineering, Durham, NC, USA

[†]Duke University, Department of Computer Science, Durham NC, USA

[‡]Aristotle University, Department of Electrical and Computer Engineering, Thessaloniki, Greece

[♮]University of Maryland, Department of Mathematics, College Park MD, USA

Introduction. We introduce first a fast computation method for similarity comparison of a 3D object template against a 3D image scene, as needed in object recognition or detection in cluttered 3D scenes [1, 2] or automated image registration and segmentation.Three dimensional images include also spatio-temporal images as in detection and estimation of motion of other changes in a temporal sequence of 2D spatial images, and spatio-spectral images. We describe next an effective scheme for mapping the algorithm to the existing graphics processing units (GPUs) to exploit the modern computer architecture for further acceleration of the computation. Preliminary experiments and results are presented.

Local correlation coefficients. Many signal and image processing applications require fast location of close matches, if any, in a 2D or 3D scene image S to a particular pattern or template T, according to certain criteria. We describe a particular matching criterion that accounts for non-rigid and nonlinear local changes in the scene image. We then describe a fast method with full respect to the criterion. The description is for the 2D case, with straightforward extension to higher-dimensional cases.

Suppose there are N_S and N_T pixels or voxels in scene S and template T, respectively. We may assume that both S and T are real valued. Let P be a typical panel of S that the template T overlaps with. Denote the mean value and the standard deviation of a pixel/voxel collection Q by $\mu(Q)$ and $\sigma(Q)$, respectively. When neither T nor P is constant-valued, $\sigma(T)\sigma(P)$ is nonzero. The correlation coefficient between the template T and the panel P for the 2D case can be described as follows,

$$c(P,T) = \frac{\sum_q [P(q) - \mu(P)] \cdot [T(q) - \mu(T)]}{N_{\mathrm{T}} \cdot \sigma(T) \cdot \sigma(P)}. \qquad (1)$$

The correlation coefficient is a particular measurement of the similarity between T and P. By the Cauchy-Schwartz inequality, $c(P,T) \in [-1,1]$. When $c(P,T) = 1$, P is a perfect match to T, pixel by pixel, or voxel by voxel. When $c(P,T) = 0$, the panel P is orthogonal to the template T.

This work is supported in part by DARPA/MTO/DESA program.

All the panels of S with their relative locations in S can be described in the 2D case as follows, $P_{uv}(x,y) = S(x,y)B_T(x-u,y-v)$, where B_T is the binary-valued characteristic function of the spatial support of the template. To avoid redundant computation, we translate and normalize the template T once for all so that $\mu(T) = 0$ and $\sigma(T) = 1$. Then, the local correlation coefficients (LCCs) of T with all candidate panels can be described as follows,

$$c(u,v) = \frac{1}{\bar{\sigma}(u,v)} \sum_{x,y} [P_{uv}(x,y) - \mu_{u,v}]\, T(x-u, y-v), \quad (2)$$

where $\mu_{u,v} = \mu(P_{uv})$ and $\bar{\sigma}(u,v) = \sigma(P_{uv})\sqrt{N_T}$. These coefficients represent the collective and basic information for subsequent location and analysis of best template matching. It is nonlinear in the local means and standard deviations.

There is a connection from the LCC computation to certain other important signal and image processing operations. For example, without the translation in the mean value and the normalization in the standard deviation, the LCC computation becomes the convolution of S with an impulse function that has the elements of T in reverse order.

Fast LCC calculation via the FFT. When the size of T reaches to certain point, the calculation of the LCCs can be made faster via the use of the FFT, in comparison to the direct evaluation by (2). The computation complexity and hence efficiency had been a divisive issue in the choice of similarity criteria between the so-called local correlation methods and the Fourier domain methods employed for template matching [3, 4]. In the latter, the local normalization is omitted. In the method introduced below, without compromising the local normalization, we use only two and a half 2D FFTs for computing the LCCs for each scene image S, with a common template.

We write the right-hand side of (2) into three cross-correlation components,

$$\sum_{x,y} [P(x,y|u,v) - \mu(u,v)]\, T(x-u, y-v) = \{S * T\}(u,v) - \frac{1}{N_T}\{S * B_T\}(u,v) \cdot \{B_S * T\}(u,v),$$

with $\{S * T\}(u,v) = \sum_{x,y} S(x,y)\, T(x-u, y-v)$, where B_S is the binary-valued characteristic function of the support of S. We re-write the expression of the standard deviation distribution $\sigma^2(u,v)$ on the left side of (2) into cross-correlation terms as well,

$$N_T\sigma^2(u,v) = \{S^2 * B_T\}(u,v) - N_T\mu^2(u,v)$$
$$= \sum_{x,y} S^2(x,y) B_T(x-u, y-v) - N_T\mu^2(u,v).$$

The reformulation above leads to efficient array operations and enables the use of multi-dimensional FFTs.

We elaborate on how to use only 2.5 FFTs in computing the LLC distribution per scene. By the cross-correlation theorem, the four cross-correlation components can be computed via the following five quantities in the Fourier domain $F(B_T)$, $F(T)$, $F(B_S)$, $F(S)$ and $F(S^2)$ and the inverse FFT of the point-wise products, $F(T) \cdot F(B_S)$, $F(T) \cdot F(S)$, $F(B_T) \cdot F(S)$ and $F(B_T) \cdot F(S^2)$. For a sequence of scene images, the quantities associated with B_T, B_S and T only shall be computed once for all. Thus, for each scene image, we can use one FFT with $S + i \cdot S^2$ to obtain both $F(S)$ and $F(S^2)$ at the same time, then use one inverse FFT from the scaled quantities $F(B_T) \cdot F(S)$ and $F(B_T) \cdot F(S^2)$ to obtain two cross-correlation components $S * B_T + iS^2 * B_T$, and use another inverse FFT to obtain $S * T$ from the scaled quantity $F(S) \cdot F(T)$ in the Fourier domain. Exploiting the real-valued property of $S * T$, the operations in the last inverse FFT can be reduced to a half.

The total number of real arithmetic operations in computing the LCC distribution per scene is about $12.5N \log(N) + \alpha N$ with N the number of computed coefficients. For equilateral image S and T with sides N_S and N_T respectively per dimension, $N = (N_T + N_S - 1)^d$, where d is the dimension. The constant α in the linear term is a modest number, accounting for the element-wise products in the Fourier domain and the packing and unpacking operations before and after each FFT.

Mapping to GPUs via CUDA. There are certain distinct advantages in using GPUs for pattern matching in 3D images. First, there are intimate connections and similarities in processing and rendering between graphics and image applications. Secondly, GPUs can be easily integrated as a special accelerator into a host computer such as a desktop or an existing embedded imaging system. Thirdly, GPU programming environments, such as CUDA by NVIDIA, leverage the increasing degree of parallelism in many-cores hardware for computing tasks beyond graphics processing and rendering.

The parallelization of the LCC computation centers on the partition of the output data space. To utilize the hierarchical GPU memory structure, we partition the LCC output array into hierarchical subarrays accordingly. Every element of the LCC output is assigned to a distinct CUDA thread. In CUDA, threads are grouped into blocks. The threads in each block share a memory local to the block, which contains a subarray of the input scene and a copy of the template. There is no overlapping in the LCC output array among the threads within a block or across blocks. In other words, the mutual exclusion in writing is guaranteed by the algorithmic arrangement. The subarrays of the input scene image on neighbor blocks overlap at their boundary areas, and the element-wise threads on each block read from the same input subarray. The size of input and output subarrays is determined to maximally utilize the shared memory of each thread block.

The experiments with GPU-accelerated calculation were carried out with the following setups. The host machine is a Dell T5400 workstation with a quad-core Intel Xeon at 2.50GHz. One core is used for the host control. Two different NVIDIA Tesla cards are used as accelerators in separate experiments. One is the C870 GPU with 128 streaming processor cores running at 1.35GHz. It has 1.5GB of dedicated memory and a 384-bit memory interface with a bandwidth of 76.8GB/sec. The other is the C1060 GPU with 240 streaming processor cores running at 1.35GHz. It has 4GB of dedicated memory and a 512-bit memory interface with a bandwidth of 102GB/sec. The experiments without GPU acceleration were carried out on the host machine with only one core activated for the calculation.

Both the direct calculation and the calculation via the FFT are implemented. In the latter, the CUFFT from the CUDA 2.1 SDK is used. Preliminary results showed substantial GPU acceleration, with each of the GPU cards. The LCC calculation with GPUs is 32 to 43 times faster than that without, with the size of scene images up to 513^3 and the size of templates up to 8^3. With larger template size, the approach via the FFT is indeed faster. The cross-over point in the template size between the direct approach and the FFT approach, by the present implementations, is about 8×8 for the 2D case and $6 \times 6 \times 6$ for the 3D case.

1. REFERENCES

[1] A. E. Johnson and M. Hebert, "Efficient multiple model recognition in cluttered 3-D scenes," *Proc. IEEE Conf. Comp. Vision and Pat. Rec.*, pp. 671–677, 1998.

[2] S. Allney and C. Morandi, "Digital image registration using projections," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 222–233, Mar 1986.

[3] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, 1992.

[4] B. S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *IEEE Trans. Image Proc.*, vol. 5, no. 8, pp. 1266–1271, 1996.