USING APPROXIMATION AND RANDOMNESS TO SPEED-UP INTENSIVE LINEAR FILTERING

J. Inglada

CNES - DCT/SI/AP - BPI 1219
18 avenue Edouard Belin
31401 TOULOUSE CEDEX 9 - France

J. Michel

C-S - IGI
Parc de la Plaine
Rue Brindejonc des moulinais, BP 5872
31506 TOULOUSE CEDEX 5 - France

1. INTRODUCTION

The convolution product is an intensive operator whose complexity depends on the number of pixels in the image as well as the radius of the neighborhood to account for. On the other hand, it is required in numerous applications like any linear filter, texture detectors using Gabor filters or digital elevation modeling and displacement estimation using fine grain correlation. To apply such operators to images from the market often takes hours, especially with a high spatial resolution, while operational applications like rapid mapping requires the tightest possible schedule. One possible way to overcome this shortening is to use up-to-date performing hardware : multi-cores CPU, or more recently GPU can considerably speed-up the process. Yet, one may wonder if increasing the processing power is the only answer to computational intensity. In this paper, we try to sketch alternative approximate solutions to lower the complexity while keeping the error rate under control. The remaining of this paper is organized as follows. The first section briefly recalls definitions of the operators we are working on, and presents the related work. The second section details our use of approximation to speed-up these operators, whereas the fourth section demonstrates the use of randomness in this context.

2. DEFINITIONS AND RELATED WORK

This section briefly recalls the main definitions for convolution, correlation and sub-sampling operators, as well as related work on trying to speed-up their computation.

2.1. Definitions

Given a convolution kernel $S(i,j)$ of size (m,n) and an image $I(i,j)$ of size (M,N), the convolution of I by S is defined in equation 1. The complexity of the convolution operator is $O(MNmn)$.

$$(I * S)(i,j) = \sum_{k=0}^{k=m} \sum_{l=0}^{l=n} I(i-k+m/2, j-l+n/2) \cdot S(k,l) \tag{1}$$

Given two images $I_1(i,j)$ and $I_2(i,j)$, and a neighborhood radius r, and a search radius R , the correlation map D of the two images is given in equation 2. The complexity of this operator is $O(MNR^2r^2)$.

$$D(I_1,I_2)(i,j) = Max_{(t_x,t_y)\in[-R/2,R/2]^2} \left(\sum_{k=-r/2}^{k=r/2} \sum_{l=-r/2}^{l=r/2} I_1(i+t_x+k, j+t_y+l) \cdot I_2(i+t_x+k, j+t_y+l) \right) \tag{2}$$

As one can see, the correlation map is computed from a series of convolutions for each pixel.

2.2. Related work

One classical way to accelerate convolution computation is by performing products in the Fourier domain, however, this approach is only efficient when convolution kernels are large. Often, when local linear filtering operators are applied, the Fourier domain approach is not useful.

The idea we wanted to explore in this work was inspired by recent advances in the Compressive Sensing field [1], where the underlying idea is to concentrate on the essential information in the signals. Compressive Sensing, in some of its implementations, uses random projections in order to perform the signal acquisition. This is related to older approaches which were designed to overcome the Shannon/Nyquist limits [2, 3, 4]. Many other approaches have been developed for different applications [5].

One approach that will interest us comes from computer graphics and its goal is to reduce aliasing in under-sampled images [6], since one of our applications will be generating small quick-looks for large images with small error. This approach is again related to the compressive sensing field as shown by a recent work [7].

3. APPROXIMATION

This section describes the use of approximation to speed-up processes like convolution or correlation on huge images with large neighborhood radiuses. Although not directly related to the approaches cited in section 2.2, since the convolution kernel is known, it seems interesting to use a deterministic and simple way to reduce the number of computations performed.

3.1. Principle

Operators like convolution or correlation implicitly consider all pixels equal. They do not account for the fact that pixels with high value in the kernel are more likely to contribute to the final result than pixels whose value is almost null. Meanwhile, when looking at oscillant convolution kernels such as Gabor filters, on can observe that lots of pixels lay in that second low-valued group of pixels. Therefore, we shall be able to ignore those low-valued pixels without losing to much of the final information. This induces a preliminary step before actually processing data :

1. Order the pixels from the kernel decreasing L_p norm,
2. Select the most significant pixels from the kernel according to some criterion,
3. At each location of the image, compute the operators using only the set of selected pixels.

The criterion to select the set of significant pixels can be given either :

- in term of number of pixels to keep, which allows to get the best accuracy under the constraint of fixed complexity,
- or in term of the amount of kernel information to keep, which allows to get the lowest complexity under the constraint of fixed accuracy.

Please note that this only applies to highly perturbed filters, or kernels from real data as in the correlation map case. Handling kernels with pixels of equal significance will be dealt with in section 4.

4. RANDOMNESS

The idea here is to find a strategy to apply when the deterministic selection strategies presented in section do not perform well. Some experiments, which will be included in the final paper, show that when the convolution kernels are smooth and with a single mode (mainly low pass filters), ranking the coefficients by value introduces a spatial bias in the computation (all coefficients are located around the same location).

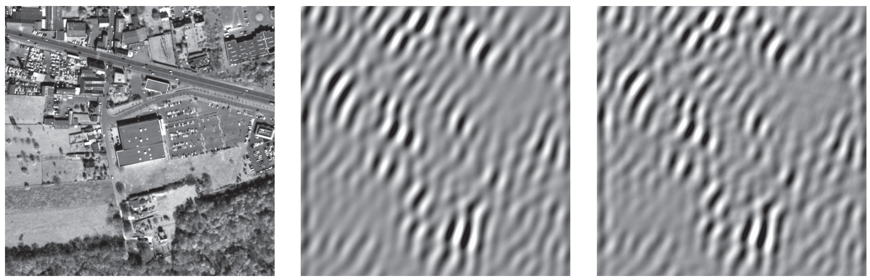
In those cases, inspired by Compresive Sensing and also by techniques introduced in the computer graphics fields, we found that random selection of coefficients was more efficient.

Since random number generation may also be costly in terms of computation, pre-computed coefficient locations can be used without loss of quality in the results.

5. APPLICATIONS

5.1. The deterministic convolution case: computing textures

To apply the simple scheme of section 3.1 to the convolution operator, one only needs to perform the preliminary steps 1 and 2 once. This implies of course an overhead with respect to the classical convolution process, but at the expense of this light overhead, one can benefit from a fast approximative result with either complexity or accuracy under control. Figure 5.1 shows comparative results for a given Gabor filter between the standard convolution operator and our method with only 25% of kernel pixels kept. We achieve a PSNR of 15.5 dB and is 3 times faster than the standard convolution.



(a) Input image (b) Standard convolution (c) Approximated convolution

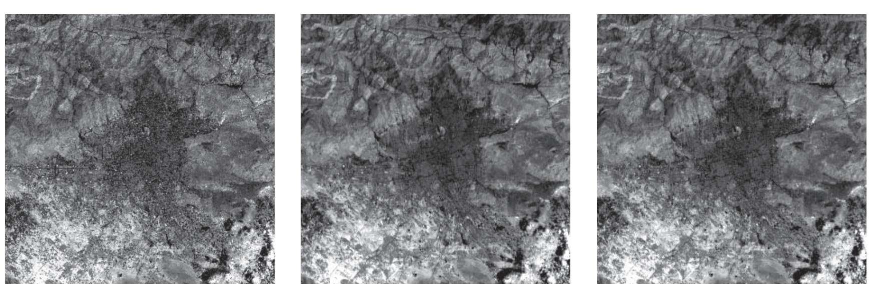
Fig. 1. Comparative result of standard and approximate convolution with only 25% of kernel pixels kept.

5.2. The random convolution case: quick-looks

As stated above, when the convolution kernels perform low-pass filtering, the deterministic selection does not work efficiently. One application where low pass filtering is used intensively is image quick-look generation. In this case the 2 classical approaches are:

- quick and dirty down-sampling by taking 1 pixel over N;
- exact computation by low-pass filtering before down-sampling.

The first approach yields highly aliased images and the second one can be very time consuming. In this case, the random selection of coefficients in the convolution kernel yields results of better quality than the raw down-sampling with a computation cost which can be highly reduced. Figure 5.2 shows such results on a SPOT5 full scene : the random version is almost as fast as the simple sub-sampling and 3 times faster than the clean low-pass filtered one. It achieves a PSNR of 18,9 dB wrt the low-pass filtering, while the simple sub-sampling only achieves a PSNR of 13.6 dB.

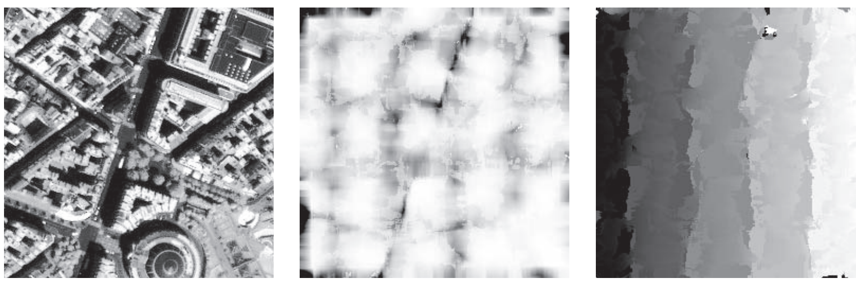


(a) Simple sub-sampling (b) Low-pass filtered sub-sampling (c) Random sub-sampling

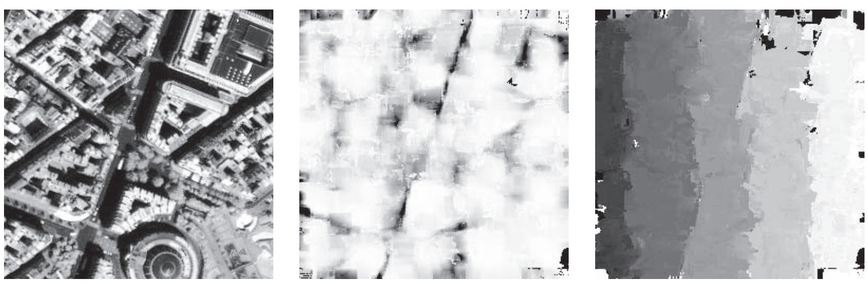
Fig. 2. Comparative result of simple sub-sampling, ideal sub-sampling with low-pass filtering and random sub-sampling using only 2.5% of the samples.

5.3. The correlation case: deformation field estimation

Disparity map estimation is the task of estimating local displacement between two images of the same scene. This displacement can be produced by changing view point or moving objects from one image to another. It is used to compute digital elevation models or estimate speed of moving objects in the scene. It implies computing the correlation map as described in equation 2, and keeping the relative position of the maxima associated to each location. Here the case is slightly different from the deterministic convolution case presented section 5.1, in that it can not be done only once for all the processing. At each location, the kernel corresponding to the accounting neighborhood in the reference image is extracted and ordered. It is then used to approximate the correlation with the other image at each location in the search windows. The two options on controlling either the accuracy or the complexity presented section 5.1 remain valid. Figure 5.3 shows the comparative results of the classical correlation between two images and the approximate correlation derived in this paper. The approximate correlation is 2 times faster than the standard one, and PSNR value are 16.9 dB on the correlation map, and 12 dB for the deformation field.



(a) Reference image (b) Standard correlation field (c) Standard deformation field



(d) Warped image (e) Approximated correlation field (f) Approximated deformation field

Fig. 3. Comparative result of standard and approximate correlation with only 50% of kernel pixels kept.

6. REFERENCES

[1] R.G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, 2007.

[2] F. Papenfuss, Y. Artyukh, E. Boole, and D. Timmermann, "Optimal sampling functions in nonuniform sampling driver designs to overcome the Nyquist limit," in *Proceedings of ICASSP*, 2003.

[3] Mark A. Z. Dippé and Erling Henry Wold, "Antialiasing through stochastic sampling," in *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1985, pp. 69–78, ACM.

[4] Robert L. Cook, "Stochastic sampling in computer graphics," *ACM Trans. Graph.*, vol. 5, no. 1, pp. 51–72, 1986.

[5] Abdul J. Jerri, "The shannon sampling theorem-its various extensions and applications: A tutorial review," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565–1596, Nov. 1977.

[6] Don P. Mitchell, "Generating antialiased images at low sampling densities," in *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1987, pp. 65–72, ACM.

[7] J. Romberg, "Compressive sensing by random convolution," *SIAM J. Imaging Sci. to appear*, 2009.