Neural, Parallel & Scientific Computations 15 (2007) 489-500

An Efficient Partitioning-Based Scheme for 2-D Convolution and Signal Processing Applications

Kostas Berberidis

*Department of Computer Engineering & Informatics / RACTI-R&D

University of Patras

26500, Rio-Patras, Greece

Email: berberid@ceid.upatras.gr

February 26, 2007

Abstract

A new scheme for efficiently computing the 2-dimensional (2-D) linear convolution between a fixed filter and an input sequence is presented. The scheme is particularly suitable for a special type of linear convolutions encountered in several 2-D applications, such as image registration and 2-D block adaptive filtering. In this type of convolution the involved filter is assumed fixed for a number of outputs which is less (or much less) than the number of filter taps. By properly partitioning both the filter and the input sequences the whole problem is divided into a number of smaller convolution problems which in turn are solved efficiently in the frequency-domain using Fast Fourier Transform (FFT). The scheme exhibits considerably reduced computational complexity as compared to other well-known FFT-based techniques. Moreover it lends itself for parallel implementation as well as for efficient VLSI implementation since it employs FFTs of relatively small size.

Key Words:

Image Processing, Image Registration, Fast Convolution, Block Adaptive Filtering

*This work was partially supported by the General Secretariat for Research and Technology, under grant ΠΕΝΕΔ-No.03ΕΔ838.

1061-5369 $15.00 © Dynamic Publishers, Inc.

1 Introduction

The task of computing efficiently the 2-D linear convolution between a fixed Finite Impulse Response (FIR) filter and an input sequence is encountered in many 2-D applications. Among several techniques suggested in literature, those implemented in the frequency domain, using Fast Fourier Transform (FFT), are particularly attractive due to their computational efficiency. Such frequency domain techniques, performing true linear convolution, are the well-known overlap-save and overlap-add methods [1]. The main trait of these methods is that the input sequence is divided into a number of blocks of equal area which is related to the region of support of the filter. It should be noted however that in both methods the region of support of the input sequence is considered to be much larger than the region of support of the 2-D filter.

In several applications, however, the required number of convolution outputs is quite small as compared to the region of support of the involved filter. In other cases the filter is spatially (or time) varying and can be considered fixed only for a number of outputs which is less (or much less) than the number of filter taps. A typical application of the former type is image registration [2] while of the latter type is 2-D block adaptive filtering [3]. Note also that, in these cases it is meaningless to apply the above mentioned FFT-based techniques since the input sequence has a region of support which is comparable to that of the filter. Of course, one could still compute the required convolution (or correlation) in the frequency domain using FFT after padding appropriately with zeros the involved sequences to avoid wraparound errors, [2]-[5]. However this widely used technique turns out to be inefficient in the applications of interest due to the fact that large FFTs need to be computed for relatively few outputs, and thus the cost per output sample is quite high.

In the new method proposed here, a proper partitioning is applied to both the filter and the input sequences and the overall convolution problem is divided into a number of smaller convolution sub-problems. These small sub-problems are subsequently computed efficiently in the frequency domain using FFT. Due to this partitioning, the scheme exhibits considerable computational savings as compared to the above mentioned FFT-based methods. In addition, it offers implementation advantages due to the use of relatively small FFTs and the parallelizable structure of the method.

Although the new scheme is developed for the general 2-D linear convolution problem, it can be easily extended to a form suitable for cross-correlation based image registration. We have focused on this application since in many cases in practice the image registration task turns out to be a fundamental prerequisite for any further processing and analysis. Recall that the image registration task consists in finding the underlying correspondence between two or more images. The different images are either images of the same object taken from different sensors or images of the same object taken at different times. Thus, between these images there may be translational shifts, rotational differences, scale and perspective view differences [2]. In some applications the shifts and differences are detected and corrected off-line, while in others the acquired sequence of images has to be aligned in real time. In the latter applications the computational complexity issue is even more critical. Since the new scheme can be applied for computing efficiently the spatial cross-correlation between two images, it can be incorporated to any correlation-based image registration technique. It should be emphasized that the scheme provides an exact computation of the spatial cross-correlation sequence (i.e., as if it were computed directly by its definition).

Another typical application in which the new scheme could be adopted is 2-D block adaptive filtering (BAF) [3]. By definition, in block adaptive filtering the involved filter taps are updated on a block-by-block basis (every, say, $L \times L$ input data samples) while the filtering operation is performed at every step. BAF schemes are very popular in one-dimensional signal processing and communications applications as a means to dramatically reduce complexity, [6], [7]. The computational saving is achieved due to the use of fast convolutional techniques and/or sampling rate reduction. BAF is, likewise, a promising means to reduce complexity in 2-D applications in which adaptive filters are employed such as, restoration, denoising, blind deconvolution, etc, [3].

The following notation is used throughout the paper. Spatial domain and frequency domain 2-D arrays are denoted by upper case and bold upper case letters, respectively. $X(m,n)$ denotes a sample of array X at (m,n) and $X_{i,j}$ is the (i,j) subarray of X. The new 2-D convolution scheme is presented in Section 2 while in Sections 3 and 4 its application to image registration and block adaptive filtering is discussed, respectively. Finally, Section 5 concludes the work.

2 Efficient 2-D Convolution Scheme

Let us assume that an $M \times M$ filter H has to be convolved with the 2-D sequence X and that $L \times L$ outputs of this convolution are required all belonging to a cohesive square area of dimensions $L \times L$. For the sake of simplicity, we assume that both H and X are real valued. The extension to the complex case is straightforward. Also, for simplicity again, we assume that M and L are powers of two. Let us now define a 2-D mirror filter H^e as

$$H^e = JHJ \tag{1}$$

where matrix J is the so-called exchange matrix of dimensions $M \times M$ having ones along the antidiagonal and zeros elsewhere. Then the required convolution of H with X can be equivalently written as

$$Y(m,n) = \sum_{k=1}^{M} \sum_{l=1}^{M} H^e(k,l) X(m - M + k, n - M + l) \tag{2}$$

for $m, n = M+1, \ldots, M+L$. If we compute the 2-D convolution directly from its definition then the required complexity will be $O(M^2L^2)$. If instead we choose to carry out the convolution in the frequency domain via FFT, then a straightforward way would be to pad with zeros the filter H^e up to an area equal to $(M+L) \times (M+L)$, then compute its cyclic convolution with the respective area of X using FFT and finally keep the $L \times L$ outputs corresponding to the required linear convolution. The complexity of this approach is $O((M+L)^2 log(M+L))$.

To reduce the above complexity and at the same time avoid using large FFTs we suggest partitioning appropriately both X and H^e and solve a number of smaller convolution problems whose outputs can be finally combined in the frequency domain leading to the desired convolution outputs. The new scheme is summarized below.

The new partitioning-based scheme

1. Define the mirror filter H^e in terms of H as in (1). The $M \times M$ samples of filter H^e are denoted as $H^e(k,l)$, $k, l = 1, 2, \ldots, M$.

2. Let $X(k,l)$, $k, l = 1, 2, \ldots, M+L$ be the samples of the $(M+L) \times (M+L)$ area of X that will take part in the required convolution.

3. Divide this $(M+L) \times (M+L)$ area of X into $(M/L)^2$ overlapped blocks of dimensions $2L \times 2L$. The (i,j) block of X, i.e. $X_{i,j}$, contains the samples $X((i-1)L+p, (j-1)L+q)$, $p, q = 1, 2, \ldots, 2L$.

4. Divide the $M \times M$ filter H^e into $(M/L)^2$ non-overlapped and consecutive blocks of dimensions $L \times L$. Thus the block $H^e_{i,j}$ contains the samples $H^e((i-1)L+k, (j-1)L+l)$, $k, l = 1, 2, \ldots, L$. For each $H^e_{i,j}$ form a respective zero padded $2L \times 2L$ array $\tilde{H}^e_{i,j}$ as follows:
$$\tilde{H}^e_{i,j} = \begin{pmatrix} JH^e_{i,j}J & 0 \\ 0 & 0 \end{pmatrix}$$

5. For each (i,j), $i, j = 1, 2, \ldots, (M/L)^2$, compute the 2-D FFT of the respective $X_{i,j}$.

6. For each (i,j), $i,j = 1,2,\ldots,(M/L)^2$, compute the 2-D FFT of the respective $\tilde{H}^e_{i,j}$.
7. For each (i,j), $i,j = 1,2,\ldots,(M/L)^2$, compute the point-wise product $\boldsymbol{Y}_{i,j} = \boldsymbol{X}_{i,j} \odot \tilde{\boldsymbol{H}}^e_{i,j}$.
8. Compute the summation:
$$\boldsymbol{Y}_{tot} = \sum_{i=1}^{M/L} \sum_{j=1}^{M/L} \boldsymbol{Y}_{i,j}$$
9. Compute the inverse 2-D FFT of $\boldsymbol{Y}_{tot}$, and, from the resulting $2L \times 2L$ array, keep only the bottom right quarter, which is equal to the required convolution.

Note that the mirror filter H^e is used in the algorithm instead of H. In this manner, as it will be shown in next section, the scheme is more easily applicable for correlation purposes as well. Of course, with trivial modifications, the algorithm can be written directly in terms of H.

The proposed partitioning-based method should not be confused with the well-known overlap-save (or overlap-add) method. In the latter method, the input signal (i.e., the image) is again partitioned into blocks but the 2-D filter is very small compared to the input and remains as is. On the contrary, in the proposed method the required convolution output has a small area compared to the filter area and both the input image and the filter are properly partitioned.

The computational complexity of the new 2-D convolution scheme is $O(M^2 logL)$. Thus, not only small FFTs are employed but, also, depending on M and L, a noticeable computational saving may be achieved. More specifically, the total number of real multiplications required for computing $L \times L$ convolution outputs equals:

$$M_{total} = 32M^2 log_2 L + 16L^2 log_2 L + 16L^2 + 48 \qquad (3)$$

Table I illustrates the substantial computational savings which can be achieved by the new algorithm for some indicative and practical values of M and L. The proposed scheme is compared to the direct 2-D convolution and the conventional FFT-based 2-D convolution. As it can be seen, the new scheme exhibits considerably reduced complexity and offers computational savings as high as 70%. Furthermore, it is suitable for parallel as well as efficient VLSI implementation since, instead of large FFTs, it employs FFTs of relatively small size.

3 Application Example 1: Image Registration

As mentioned in the introduction, the new fast convolution scheme (and more generally the idea of partitioning) is almost directly applicable to image registration. Let us denote as A

the so-called reference image and as B the second image which is a translated and rotated version of the first one. Recall that, in most cases, the aim in image registration is to correct translation and rotation displacements between A and B. Spatial cross-correlation is among the most widely used and studied criteria for image registration. Apart from its simplicity, the main advantage of this criterion is its robustness against additive noise.

Recently a new method, based on spatial cross-correlation, has been suggested for estimating both translational and rotational differences, [8]. This is achieved in an iterative manner and after pre-whitening the images. The method offers considerable improvement in performance as compared to other methods which tackle the same problem. In any case, the most consuming part is the one concerned with translations finding since it constitutes a 2-D search as opposed to the rotation finding part which can be viewed as an 1-D search. Thus, we apply the new efficient scheme to the former part.

If we assume that the maximum translational shift at each dimension is equal to L pixels, then the $M \times M$ area of image B has to be correlated with a corresponding $(M+L) \times (M+L)$ area of image A. Moreover, the cross-correlation output has to be computed for approximately L^2 possible shifts. Commonly this procedure is done efficiently in the frequency domain using FFT and the required complexity is $O((M+L)^2 log(M+L))$, [5].

It should be noticed however, that in many cases in practice, it can be reasonably assumed that $L << M$, meaning that the translational shifts are much less than the image size. Indeed, in many applications the images are acquired at such a rate that during the interframe period there are no significant shifts and in any case the search area and the searching window have dimensions much larger than the possible shifts. In such cases, the spatial correlation is of a special type, since only a small number of correlation outputs relative to the window size is required. This observation was the motivation to apply the new scheme of Section 2 to image registration.

The scheme can be applied after taking into account the following remarks. In place of X we have now the search area of image A and in place of H^e we have the window of image B. Note that step 1 of the scheme is skipped here since correlation and not convolution is to be computed. The overall complexity for performing the spatial cross-correlation is now $O(M^2 logL)$.

In Figure 1 the whole procedure is shown for a simplified example with $L = M/2$. As it can be seen the search area of A is partitioned into four overlapped blocks of dimensions $2L \times 2L$. Such a block is shown at the left bottom of the figure. The searching window is in turn partitioned into four non-overlapped blocks. Each block is zero padded appropriately as shown at the right bottom of the figure.

Recall that the new efficient scheme provides exact solution to the problem at hand, i.e., a solution which is equal to the one provided directly by the cross-correlation definition. Therefore, it is rather meaningless to conduct experiments for evaluating the new scheme since the performance of spatial cross-correlation has been extensively studied in literature. However, in order to test the validity of the new scheme we conducted some typical registration experiments. More specifically, two classical test images of sizes 256×256 were used, the so-called "forest" and "peppers" images shown in Figures 2 and 3, respectively. For each image case, a 128×128 window and a 160×160 search area were taken. The latter image contained a translated version of the window image. The translation shifts, at each axis, were in the range [0,32]. For each image case, the task was to register (align) a number of 1089 pairs of images (i.e., for all 33×33 possible translations). The same number of experiments were repeated for the noisy case, in which the translated versions of the original images were contaminated with additive Gaussian noise resulting at SNR=10dB. The performance criterion used in all cases was the number of completely successful registration (hits).

As shown in Table II, in the case of the "forest" image all pairs were successfully aligned even in the noisy case, whereas for the "peppers" image the percentage of hits was 96.5% for the noiseless and about 94.5% for the noisy case, respectively. This difference in performance between the two image cases is due to the fact that the "forest" image is a more "detailed" image with a sharper autocorrelation peak as compared to the "peppers" image. As expected, the same scores achieved by applying the mathematically equivalent spatial cross-correlation criterion directly by its definition.

4 Application Example 2: Block Adaptive Filtering

Adaptive filtering algorithms has been an area of active research over the last two decades due to their wide applicability in many signal processing and communications applications. Several important factors have to be taken into account in designing an adaptive filtering algorithm, such as, accuracy of steady-state solution, convergence speed, tracking abilities, computational complexity, numerical robustness, ease of implementation etc., [6], [7]. In many real-time applications the issue of complexity plays a critical role and many research efforts have been directed towards deriving adaptive filtering algorithms with relatively low computational cost. A significant reduction in complexity may be achieved with block adaptive algorithms especially when implemented in the frequency domain.

Let us first describe the key concept for efficient filtering operation in the one dimensional BAF case. As already mentioned, in block adaptive filtering the involved filter taps are updated on a block-by-block basis (every, say, L input data samples) while the filtering operation is performed for every step. We focus here on the filtering part since this is the most computationally demanding. The computations of the updating part can be treated in a similar fashion. Let us form vector $\boldsymbol{y}_L(n+L-1)$ containing all L filter outputs within current block, i.e.,

$$\boldsymbol{y}_L(n+L-1) = X_{L,M}\boldsymbol{w}_M(n) \tag{4}$$

where vector $\boldsymbol{w}_M(n)$ is the M-length filter, which is kept fixed for the current block (containing the time steps from n to $n+L-1$), and matrix $X_{L,M}$ contains the input samples as

$$X_{L,M} = \begin{bmatrix} x(n) & \dots & x(n-M+1) \\ \vdots & \ddots & \vdots \\ x(n+L-1) & \dots & x(n+L-M) \end{bmatrix} \tag{5}$$

For simplicity, and in order to make easier to understand the key concepts, we further assume that $L = M$, i.e., the block length is equal to the filter length. The extension to $L < M$ is straightforward. Since matrix $X_{L,M}$ is Toeplitz, we may implement efficiently the product $X_{L,M}\boldsymbol{w}_M(n)$ in the frequency-domain by embedding $X_{L,M}$ into the $2M \times 2M$ circulant matrix C as follows,

$$C = \begin{bmatrix} X_{L,M} & \tilde{X}_{L,M} \\ \tilde{X}_{L,M} & X_{L,M} \end{bmatrix} \tag{6}$$

The first column of C is defined as

$$\boldsymbol{c} = [\boldsymbol{x}^H \;\; 0 \;\; \tilde{\boldsymbol{x}}^H J]^H$$

where vector $\boldsymbol{x}$ is the first row of Toeplitz matrix $X_{L,M}$. Vector $\tilde{\boldsymbol{x}}$ consists of the last $M-1$ elements of $\boldsymbol{x}$, and matrix J is the so-called exchange matrix defined in Section 2.

Let us now define as F the $2M \times 2M$ Discrete Fourier Transform (DFT) matrix. That is, if we left multiply a vector $\boldsymbol{v}$ by F then we obtain the DFT of $\boldsymbol{v}$. The operation FCF^{-1}, with C defined as in (6), results in a diagonal matrix D, whose diagonal is equal to the DFT of vector $\boldsymbol{c}$, i.e.,

$$FCF^{-1} = diag(F\boldsymbol{c}) \equiv D \tag{7}$$

Thus, the matrix-by-vector product $X_{L,M}\boldsymbol{w}_M(n)$ can be computed via

$$\begin{bmatrix} X_{L,M}\boldsymbol{w}_M(n) \\ \boldsymbol{0}_M \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_M & \boldsymbol{O}_M \\ \boldsymbol{O}_M & \boldsymbol{O}_M \end{bmatrix} \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{F} \begin{bmatrix} \boldsymbol{w}_M(n) \\ \boldsymbol{0}_M \end{bmatrix} \tag{8}$$

In other words, three DFTs and a point-wise vector-by-vector product suffice to carry out the above matrix-vector multiplication. The computational complexity of the scheme is $O(logL)$ operations per sample, when $L = M$.

Unfortunately, the FFT-based efficient scheme of eq.(8) cannot be extended to the 2-dimensional case. Let us recall that, in the 2-D case, an $M \times M$ filter H has to be convolved with the 2-D sequence X and that $L \times L$ outputs of this convolution need to be computed. By taking row-wise, the elements of H and X we can still express this convolution operation as a matrix-by-vector similar to the one in (4). However, now, the involved data matrix is not a Toeplitz one and hence the scheme of eq.(8), derived for the one dimensional case, cannot be directly applied. If the block is restricted to be one dimensional (i.e., $1 \times L$) the resulting data matrix has a block Toeplitz structure which can be exploited for deriving schemes with lower complexity as compared to the direct computation. Even in that case, however, the complexity would be much higher than $O(logL)$ operations per sample attained in the 1-D case above.

On the contrary, the efficient scheme developed in Section 2 is directly applicable to the 2-D BAF problem and results in significant reduction of the computational load of the filtering part. It can be easily seen by eq.(3) that the required complexity would be $(M/L)^2O(logL)$ operations per sample, thus for $L = M$ the scheme attains the complexity of the 1-D case. To the author's knowledge, the suggested partitioning based scheme is the most efficient among the existing ones when the block size is less than or equal to the filter size.

5 Conclusion

A new fast scheme has been developed for computing a special type of 2-D linear convolution which is encountered in several applications. The scheme is directly applicable to the computation of the spatial cross-correlation sequence between images thus reducing the complexity of the image registration task. Also, the new scheme lends itself for efficient block adaptive filtering with significantly reduced complexity as compared to existing schemes.

Acknowledgements: The author would like to thank Dr. Irene Karybali for useful discussions on the subject of image registration.

References

[1] J. Lim, "2-D Signal and Image Processing", Prentic-Hall, 1990.

[2] W. P. Pratt, "Digital Image Processing", John Willey and Sons, 2001.

[3] W.B. Mikhael, S.M. Ghosh, "Two-Dimensional Block Adaptive Filtering Algorithms with Optimum Convergence Factors", *IEEE Transactions on Circuits and Systems II*, vol. 42, No. 8, Aug. 1995, pp. 505-515.

[4] L. G. Brown, "A Survey of Image Registration Techniques", ACM Computing Surveys, vol. 24, no. 1, pp. 326-376, Dec. 1992.

[5] R. J. Althof, M. G. J. Wind, and T. Dobbins, III, "A Rabit and Automatic Image Registration Algorithm with Subpixel Accuracy", *IEEE Transactions on Medical Imaging*, vol. 16, No. 3, June 1997, pp. 308-315.

[6] S. Haykin, "Adaptive Filter Theory", Prentice-Hall, 1996.

[7] G. Glentis, K. Berberidis, S. Theodoridis, "Efficient Least Squares Adaptive Algorithms for FIR Transversal Filtering: A Unified View", *IEEE Signal Processing Magazine*, vol.16, no.4, pp. 13-41, July 1999.

[8] I. Karybali, K. Berberidis, C. J. Balas, "A High-Performance Image Registration Technique and Application to Multispectral Imaging for Medical Diagnosis", Proc. of BIOSIGNAL-2002 (EURASIP), June 2002.

TABLE I
COMPUTATIONAL COMPLEXITIES (REAL MULTIPLICATIONS/BLOCK)

Filter dimensions $M \times M$	Required Area $L \times L$	Direct 2-D Convolution	Conventional FFT-based Scheme	Partitioning-based Scheme
128×128	16×16	4194304	6291456	2117680
128×128	32×32	16177216	6291456	2719792
128×128	64×64	67108864	6291456	3604528
256×256	16×16	16777216	28311552	8409136
256×256	32×32	67108864	28311552	10584112
256×256	64×64	268435456	28311552	13041712
512×512	16×16	67108864	125829120	33574960
512×512	32×32	268435456	125829120	42041392
512×512	64×64	1073741824	125829120	50790448

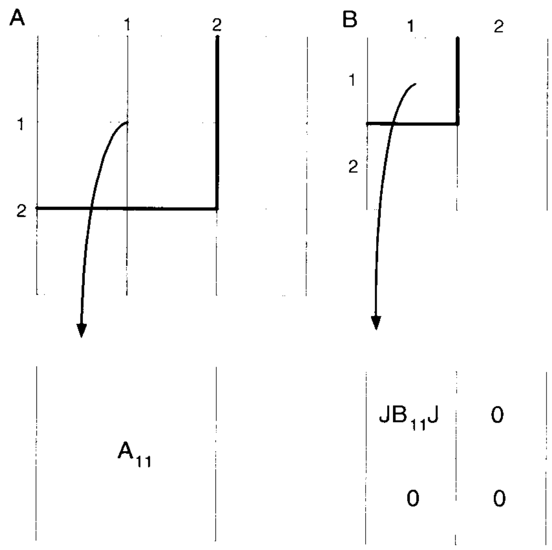

Figure 1: Schematic representation of the new scheme

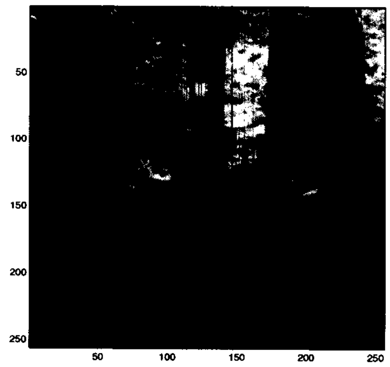

Figure 2: Image "forest"

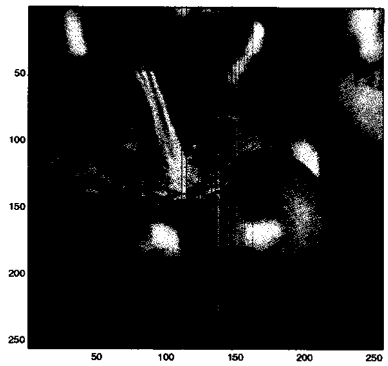

Figure 3: Image "peppers"

TABLE II

Number of successful registrations (hits)

IMAGES	no. of Hits
"forest"	1089/1089
"forest" with noise	1089/1089
"peppers"	1052/1089
"peppers" with noise	1027/1089