

Reprinted from the PROCEEDINGS OF THE IEEE
 VOL. 60, NO. 7, JULY, 1972
 pp. 887-891

COPYRIGHT © 1972—THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.
 PRINTED IN THE U.S.A.

A Comparison of Computations for Spatial Frequency Filtering

ERNEST L. HALL, MEMBER, IEEE

Abstract—Three computational algorithms for performing spatial frequency filtering are compared and tradeoffs developed. Although each method is defined by a convolution relation, the convolution computations are different. Equal filter point-spread functions are assumed to effect the comparison. If the filter point-spread function is nonzero only over a small area, then the computation tradeoff is simply the well-known comparison between direct convolution and the fast Fourier transform (FFT). If the filter point-spread function is nonzero over a large area, then a recursive filter is competitive with the FFT. Core memory requirements for this case are smallest with the recursive filter. Experimental examples are given to illustrate the subjective evaluation problem.

INTRODUCTION

IN AN IMAGE PROCESSING system, the need to design a filter with given spatial frequency domain characteristics often arises. Some examples are: a spatial frequency equalizing filter might be needed to compensate

for imaging-system aberrations; a spatial notch or bandpass filter might be desired to remove or enhance a systematic line structure; a smoothing or low-pass spatial filter might be needed to reduce speckled or "snow" type noise; a high-pass spatial filter might be needed to remove the contrast information and retain the edge information; a high-frequency emphasis spatial filter might be desired to enhance the image for human visualization; a spatial matched filter might be desired to detect a certain feature in an image.

Most of the previous spatial frequency filter design methods have used either direct spatial convolution or the two-dimensional fast Fourier transform (FFT) to implement the filter [1], [2]. Since recursive difference equations can often lead to an order of magnitude speed advantage over convolution or the FFT in one-dimensional digital filtering [3], it is natural to inquire about this possibility in two-dimensional filtering. This question is of particular interest in real time or small computer applications for which computation time and core memory storage are limited.

Manuscript received December 13, 1971; revised February 18, 1972.
 The author is with the Departments of Electrical Engineering and Radiology, University of Missouri, Columbia, Mo. 65201.

In this paper, three computational algorithms for digital spatial frequency filtering will be compared. The comparison shows that each method is advantageous under certain conditions.

COMPUTATIONAL ALGORITHMS

Since the defining relationship of the linear position-invariant spatial filter is a convolution, a direct method of filtering is via convolution. If the filter impulse response is nonzero only over a small area or truncated to meet this requirement, then convolution is competitive in speed to the other methods. The filter design in this case consists of specifying the filter point-spread function $h(nX, mY)$.

The filter is characterized by its point-spread function, and the filtering relation is given by

$$g_1(nX, mY) = \sum_{i=0}^{M_1-1} \sum_{j=0}^{M_2-1} h_1(iX, jY) f(nX - iX, mY - jY) \quad (1)$$

where $g_1(nX, mY)$ is the filter output at point (nX, mY) , $f(nX, mY)$ is the filter input, and both functions are defined for

$$\begin{aligned} n &= 0, 1, \dots, N_1 - 1, \dots, N_1 + M_1 - 1, \dots \\ m &= 0, 1, \dots, N_2 - 1, \dots, N_2 + M_2 - 1, \dots \end{aligned} \quad (2)$$

and $h_1(iX, jY)$ is the filter point-spread function.

In the general case, the index space of both f and g is the two-dimensional array of positive integers. An important special case consists of considering the input image f defined on an N_1 by N_2 array; then since convolution is a broadening operation, the output image g is defined on an $(N_1 + M_1)$ by $(N_2 + M_2)$ array. Another important special case restricts consideration of both the input f and output g images to the finite frame of N_1 by N_2 points. Only this case will be considered in the following to simplify the comparison of algorithms and also because it is the most commonly occurring practical situation.

The filter point-spread function is usually nonzero only for an area much smaller than the frame area, i.e., $h(iX, jY)$ is nonzero only for

$$i = 0, 1, \dots, M_1 - 1 \quad (3)$$

$$j = 0, 1, \dots, M_2 - 1 \quad (4)$$

and

$$M_1 \ll N_1$$

$$M_2 \ll N_2.$$

The convolution filter defined in the above manner is often called a nonrecursive filter.

Filtering may be accomplished via the FFT by computing the transform of the image function, multiplying by the filter transfer function, and inverse transforming the result. One advantage of filtering via the FFT is the ease and versatility of the frequency domain design techniques. That is, the filter transfer function may be directly specified in the frequency domain. Also an isotropic (rotationally invariant) filtering may be easily performed if the images are scanned accordingly. Since any of the four types—low-pass, high-pass, band-

pass, or band-reject—may be easily obtained from a low-pass prototype, only the low-pass prototype need be designed.

A class of isotropic spatial digital filters will be used for comparison with the other filters. However, it is first useful to review one requirement of an image scanning system.

It is often desirable to obtain an isotropic (rotationally invariant) frequency space since the two-dimensional filter design problems can be reduced to a one-dimensional problem and because many naturally occurring images do not have nonisotropic frequency spaces. The discrete Fourier transform of f is given by

$$F(u, v) = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} f(iX, jY) e^{-j2\pi} \left(\frac{iX}{N_1} + \frac{jY}{N_2} \right) \quad (5)$$

with

$$\begin{aligned} u &= 0, 1, \dots, N_1 - 1 \\ v &= 0, 1, \dots, N_2 - 1. \end{aligned} \quad (6)$$

If an isotropic Fourier transform filtering is to be performed, then the filter transfer function H must satisfy

$$H(f_r) = H(f_u, f_v) \quad (7)$$

where f_u and f_v are the rectangular spatial frequencies and f_r is the radial spatial frequency. If $H(f_u, f_v)$ is a continuous function defined on a region R of two-dimensional space, then even if $H(f_u, f_v)$ is sampled on a Cartesian grid, the isotropic requirement may be satisfied. For example if

$$f_r = \sqrt{f_u^2 + f_v^2} \quad (8)$$

then the isotropic condition follows. Some difficulties may be encountered in the discrete representation of H ; for example, if the sampling increments X and Y are not equal, or if a continuous filter function is not available; however, these difficulties may be minimized by scanning with equal sampling increments. This is usually not a severe restriction.

The third method for realizing a digital spatial filter is by recursive partial difference equations. Using the two-dimensional z transform, spatial and frequency domain design methods may be developed.

The general form of the digital imaging computation is given by

$$\begin{aligned} g(nX, mY) &= \sum_{k=0}^{K_1-1} \sum_{l=0}^{K_2-1} a_{kl} f(nX - kX, mY - lY) \\ &+ \sum_{k=0}^{L_1-1} \sum_{l=0}^{L_2-1} b_{kl} g(nX - kX, mY - lY). \end{aligned} \quad (9)$$

Assuming that the system is spatially causal, that is, $f(nX, mY) = 0$ and $g(nX, mY) = 0$ for $n, m < 0$, one may use the shifting theorems of the ZW transform to develop the transfer function for the system. Therefore, assuming $b_{00} = 1$,

$$H(z, w) = \frac{G(z, w)}{F(z, w)} = \frac{\sum_{k=0}^{K_1-1} \sum_{l=0}^{K_2-1} a_{kl} z^{-k} w^{-l}}{\sum_{k=0}^{L_1-1} \sum_{l=0}^{L_2-1} b_{kl} z^{-k} w^{-l}} \quad (10)$$

The equation for $H(z, w)$ is the defining equation of the two-dimensional digital filter. Clearly, given the transfer function of the filter $H(z, w)$ in the form shown in (10) one can reverse the above steps to obtain the difference equation describing the computation. Also, given the filter transfer function $H(z, w)$, the ZW transform of the output may be found by simply performing the multiplication

$$G(z, w) = H(z, w) F(z, w).$$

Note that the filter is a feedback type. That is, previous outputs may be used to compute the present output. This also provides a virtual infinite memory for the filter computation, although only a small finite memory is actually required.

COMPUTATIONAL TRADEOFFS

In this section, the three computational algorithms will be considered and certain tradeoffs given. Although each method is defined by a convolution relation, the actual convolution computation for each method is different. Since the normal aperiodic convolution computation requires the same number of computations for each output point, it may be called a constant number convolution. The FFT method produces a periodic convolution and for each output point the computation is over the full range of input points. The equivalent convolution for the recursive equation is also aperiodic, but the number of points used in the computation of each output point is variable. Thus, a comparison of the three methods is somewhat complicated.

The first computational method, aperiodic convolution, is described by (1), (3), and (4). The number of computations required for filtering an N_1 by N_2 point image is approximately

$$NC_1 = M_1 M_2 N_1 N_2 \quad (11)$$

real operations where a real operation is defined as a real multiplication and addition.

For the FFT computation, the periodic convolution which is equivalent to FFT transform multiplication and inverse transformation is given by

$$g_2(nX, mY) = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} h_2(iX, jY) f\{[n-i]X, [m-j]Y\} \quad (12)$$

where

$$\begin{aligned} [n-i] &= (n-i) \bmod N_1 \\ [m-j] &= (m-j) \bmod N_2. \end{aligned} \quad (13)$$

The modulus expressions indicate the periodic or "wrap-around" nature of FFT equivalent convolution. Also note that the computation is over the entire frame. The number of computations using an FFT algorithm for computing a transform is proportional to $2N_1 N_2 \log_2 N_1 N_2$ complex operations, where a complex operation is defined as a complex multiplication and addition. Therefore, if the filter operation consists of computing the transform of the input function, point-by-point multiplication of the input by a filter transform, and inverse transforming the result, the number of computations required is approximately

$$NC_2 = 4N_1 N_2 \log_2 N_1 N_2 + N_1 N_2. \quad (14)$$

The final computational algorithm produces an equivalent convolution given by

$$g_3(nX, mY) = \sum_{i=0}^n \sum_{j=0}^m h_3(iX, jY) f(nX - iX, mY - jY). \quad (15)$$

If the output image frame was computed by the above equation, the number of computations required would be

$$\sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} (i+1)(j+1) = \left(\frac{N_1^2 + N_1 + 2}{2}\right) \left(\frac{N_2^2 + N_2 + 2}{2}\right). \quad (16)$$

However, the recursive filter computation requires only

$$NC_3 = (K_1 K_2 + L_1 L_2) N_1 N_2 \quad (17)$$

real operations.

In order to compare the three algorithms, one must assume equal impulse responses. It is useful to consider two cases corresponding to small and large nonzero convolution areas. For simplicity, one may assume square areas, i.e.,

$$K_1 = K_2, \quad L_1 = L_2, \quad M_1 = M_2, \quad N_1 = N_2. \quad (18)$$

Case 1. Small Convolution Area

Suppose the filter responses are equal, i.e.,

$$h_1 = h_2 = h_3 = h \quad (19)$$

and the convolution area is small, i.e.,

$$h(nX, mY) = 0, \quad n, m > M. \quad (20)$$

The number of computations by direct convolution is

$$NC_1 = M^2 N^2 \quad (21)$$

real operations while the number of computations for FFT filtering is

$$NC_2 = N^2 \{8 \log_2 N + 1\} \quad (22)$$

complex operations. Finally, the number of computations for the recursive equation is

$$NC_3 = (K^2 + L^2) N^2. \quad (23)$$

However, since the impulse response has been constrained to be finite, the filter has been essentially constrained to be non-recursive. That is, the primary computational comparison is between the direct FFT and finite convolution computations. The more detailed comparison between direct and sectioning or block mode FFT filtering is described in [5]. Also, given a finite impulse response it may be possible to design a recursive filter which approximates the response and requires fewer computations [6]. Therefore,

$$\begin{aligned} K &= M \\ L &= 0. \end{aligned} \quad (24)$$

Thus, for this case, the direct convolution and nonrecursive filter are identical, and the computational tradeoff is simply the well-known comparison between direct convolution and the FFT. If this comparison is made and is favorable for direct convolution then a more detailed comparison with the recursive filter should be made.

Case 2. Large Convolution Area

Again assume the filter responses are equal, but suppose the nonzero response is large, i.e.,

$$h(nX, mY) \neq 0, \quad n, m \leq N. \quad (25)$$

For this case, the number of computations required by direct convolution is

$$NC_1 = N^4 \quad (26)$$

while the FFT still requires NC_2 computations as given in (22), the number of computations required for the recursive filter is still NC_3 as given in (23). For large values of N , clearly NC_2 and NC_3 will be much smaller than NC_1 . Thus, the main comparison is between the FFT and recursive filter. Using the conservative estimate that one complex operation is equivalent to two real operations permits one to develop the situation in which recursive filters are advantageous as

$$\frac{K^2 + L^2 - 2}{16} < \log_2 N. \quad (27)$$

If $K=L$ and $N=256$, the above equation requires $K^2 < 65$.

Thus, a recursive filter with 32 feed-forward and 32 feed-back coefficients would still be advantageous. Therefore, for the case in which the convolution area is large, both the FFT and the recursive filter are much faster than direct convolution. Furthermore, since K^2 and L^2 may not increase as fast as $\log_2 N$, the recursive filter is at least competitive and may have a computation advantage over the FFT.

It is useful to note that both the FFT and the recursive filter compute a different convolution than the aperiodic convolution. However, the "variable" type response is usually not as bothersome in image processing as a periodic "wrap-around" or aliasing.

The amount of core memory required to perform a computation is also an important consideration, especially in real time digital image processing. Both aperiodic convolution and recursive filters require only a small core memory when they are computationally advantageous. That is, for the small filter response case, M^2 is small. For the large filter response case, K^2+L^2 is also small. Thus both methods are suitable for small computers and real time computations. For the FFT using a single image array, N^2 core locations are required. Large array transforms can only be computed in real time with a large computer or special purpose FFT processor.

DISCUSSION

To effect the previous comparison, equal point-spread functions were assumed. Clearly, this assumption cannot always be satisfied directly: for example, a nonrecursive filter has a finite area impulse response, while the recursive filter response may be nonzero over an infinite area. However, techniques such as sectioning [4], [5] may be used to compute



Fig. 1. Original image scanned at 256 by 256 point resolution.



Fig. 2. Enhanced image computed via isotropic FFT filter.



Fig. 3. Enhanced image computed via nonisotropic recursive filter.

aperiodic convolutions via the FFT. Also, it is usually possible to construct approximately equal point-spread functions.

If equal point-spread functions cannot be attained, the filtering methods may be compared subjectively. For example, the original Lincoln image shown in Fig. 1 was high-frequency emphasis filtered with a Gaussian isotropic filter via the FFT with the result shown in Fig. 2, and with a nonisotropic Butterworth [9] filter with the result shown in Fig. 3. One may subjectively compare these results to select the desired filter. The filters used have approximately equal gains and cutoff frequencies but have different shapes. Note that both the high-frequency emphasis images show more edge detail than the original, but there is a difference in contrast levels

between the images in Figs. 2 and 3. The computation times for the 256 by 256 images on an IBM 360/50 were 3 min for Fig. 2 and 1 min for Fig. 3.

Finally, the design simplicity should be noted. It is very easy to design FFT filters and many programs are available. The stability of recursive filters has been considered [6], [7] and some design methods developed [8]–[10]; however, the design theory for two-dimensional recursive filters is not yet complete. Therefore a judgment on filter use based on design simplicity may be premature.

ACKNOWLEDGMENT

The author wishes to thank several members of the University of Missouri, Columbia, Image Analysis Laboratory, particularly Professors S. J. Dwyer, III, and G. S. Lodwick, for their encouragement. The image in Fig. 2 was computed by Mr. Adnan Kahveci. Finally, the Mathew Brady photo of Lincoln was obtained from T. Siegel of IBM.

REFERENCES

- [1] H. C. Andrews, *Computer Techniques in Picture Processing* New York: Academic Press, 1970.
- [2] O. J. Tretiak and T. S. Huang, "Notes of image processing workshop," Mass. Inst. Technol., Cambridge, Mass., June 1970.
- [3] C. M. Rader and B. Gold, "Digital filter design techniques in the frequency domain," *Proc. IEEE*, vol. 55, pp. 149–171, Feb. 1967.
- [4] T. B. Stockham, Jr., "High speed convolution and correlation with applications to digital filtering," in *Digital Processing of Signals*. New York: McGraw-Hill, 1969, pp. 203–232.
- [5] B. R. Hunt, "Minimizing the computation time for using the technique of sectioning for digital filtering of pictures," IEEE Comp. Soc. Rep.
- [6] J. L. Shanks, "Two dimensional recursive filters," in *SWIEEEO Rec.* (Dallas, Tex., Apr. 1969), pp. 19E1–19E8.
- [7] C. H. Farmer, "Stability of spatial digital filters," private corresp., Apr. 1971.
- [8] J. L. Shanks, "The design of stable two dimensional recursive filters," in *Proc. Univ. Missouri, Rolla—Mervin J. Kelly Commun. Conf.*, Rolla, Mo., Oct. 1970, pp. 15-2-1, 15-2-6.
- [9] E. L. Hall, "Digital filtering of images," Ph.D. dissertation, Univ. Missouri, Columbia, Mo., 1971.
- [10] C. H. Farmer, "Rotation and stability of a recursive digital filter," in *Proc. Univ. Missouri, Columbia, Two Dimensional Digital Signal Processing Conf.* (Columbia, Mo., Oct. 1971), pp. 1-2-1, 1-2-12.