Rapid Digital Filtering

Tom R. Miller, Kondapuram S. Sampathkumaran, and Michael A. King

The Edward Mallinckrodt Institute of Radiology, Washington University School of Medicine, St. Louis, Missouri, and Department of Nuclear Medicine, University of Massachusetts Medical School, Worcester, Massachusetts

Image filtering with the larger, and potentially most valuable, digital filters is very time-consuming, thus precluding use of these filters in routine clinical applications. A recently developed algorithm for spatial-domain filtering is described, and its speed is compared with those of conventional methods with and without an array processor. Using the new Chebyshev method, a 64 by 64 pixel image can be filtered on a standard 16-bit minicomputer with filters of size 3 by 3 to 23 by 23 in 1.4–9.2 sec. The conventional spatial-domain algorithm requires 3.8–71 sec. With an array processor, filtering is accomplished in 0.19–0.54 sec. Filtering in the frequency domain requires 34 sec without an array processor and 0.12 sec with one. Thus with this new Chebyshev algorithm, clinically practical digital filtering can be performed with large filters even without an array processor.

J Nucl Med 24: 625–628, 1983

Digital filtering of images is used in several areas of nuclear medicine, including gated cardiac studies (*1*,*2*) and the processing of static images (*3–6*). Unfortunately, application of the larger, and potentially most valuable, filters can be quite time-consuming without an array processor, thus precluding their routine use in many clinical applications. In this paper, rapid methods of computation of digital filters are evaluated. The performance of a recently developed algorithm for spatial-domain filtering will be compared with conventional methods with and without an array processor. It will be shown that this new method makes routine clinical use of large digital filters practical for those users who do not have an array processor.

MATERIALS AND METHODS

Basic theory. The theory and application of digital filters in nuclear medicine are described in detail elsewhere (*3*,*4*). Briefly, filtering can be performed either in the "spatial domain," where the image and filter are both described in X-Y coordinates, or in the "frequency domain," where a Fourier transform is performed leading to representation of the image and filter as Fourier series of differing spatial frequencies. In spatial domain, or finite impulse response (FIR), filtering a two-dimensional convolution operation is performed in which a square "mask" filter is passed across the image. The widely used "nine-point smooth" is an example of a simple 3×3 FIR filter. In frequency-domain filtering, a two-dimensional Fourier transform is performed, the resulting transformed image is multiplied by the desired filter function, and the inverse Fourier transform is computed to yield the filtered image.

The performance of different filtering methods can be characterized by the number of multiplications required to effect the filtering operation, since on most minicomputers multiplication is much slower than addition and is, therefore, the principal determinant of speed of computation.

Filter algorithms. The simplest algorithm for computation of FIR filters is direct convolution (*3*,*4*). For a filter mask of size $(2N + 1) \times (2N + 1)$, this method requires $(2N + 1)^2$ multiplications for each point in the image, or $M^2(2N + 1)^2$ multiplications for an image of size $M \times M$. By exploiting the symmetry properties of circularly symmetric, zero-phase filters (*4*), the number of multiplications can be significantly reduced by first adding all the image elements with symmetric filter values, and then multiplying. With this symmetric algorithm, the number of multiplications is approximately $0.6M^2(N + 1)^2$.

A popular method of implementing FIR filters—especially for filters of approximately size 11×11 or greater—uses the Fourier transform (*7*,*8*). First, the two-dimensional filter mask is Fourier transformed to yield the desired filter function in the frequency domain. The calculation then proceeds as described above for conventional frequency-domain filtering. The Fourier-transform and spatial-domain methods yield mathematically identical results. The Fourier-transform method requires approximately $M^2(4 \log_2 M + 1)$ multiplications using the fast Fourier transform (*7*).

A completely new class of filtering algorithms has recently been proposed (*9*); they yield mathematically exact computation of FIR filters with many fewer multiplications than the conventional convolution methods. These algorithms use a small filter mask that is passed repeatedly across the image. One such filtering scheme,

Received Dec. 6, 1982; revision accepted Feb. 28, 1983.

For reprints contact: Tom R. Miller, PhD, MD, Mallinckrodt Institute of Radiology, 510 S. Kingshighway Blvd., St. Louis, MO 63110.

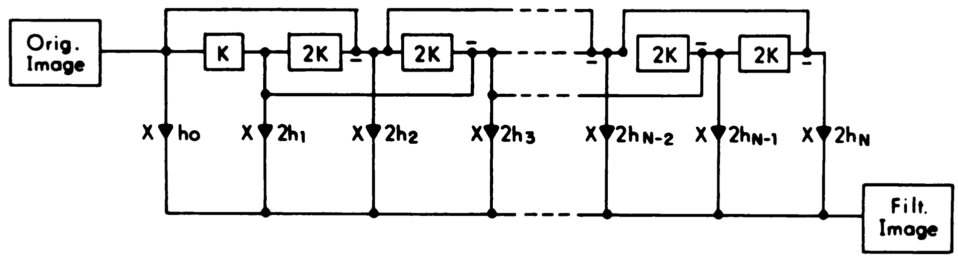

FIG. 1. Network diagram showing Chebyshev method of spatial domain filtering with filter of length 2N + 1. K is a 3 × 3 mask, or kernel, that is passed over original unfiltered image. 2K is a mask with coefficient values twice those of K. $h_0, h_1, \ldots, h_N$ are coefficients of the one-dimensional filter with same frequency response as desired for the two-dimensional filter.

using the recursion relation between the Chebyshev polynomials, is shown in Fig. 1 (*10*). The h_i are the one-dimensional filter coefficients with the same frequency-response characteristics as those of the desired two-dimensional filter. The 3 × 3 mask, or kernel, K is

$$\frac{1}{8}\begin{pmatrix} 1 & 2 & 1 \\ 2 & -4 & 2 \\ 1 & 2 & 1 \end{pmatrix}. \quad (1)$$

This kernel (or twice its value after the first pass) is repeatedly passed over the image, as shown, with the resulting image after each pass multiplied by the successive filter coefficients $h_0, h_1, \ldots$ and added together to give the filtered image, as shown in Fig. 1 and described in more detail in the Appendix. The number of multiplications for this Chebyshev algorithm is only $M^2(4N + 1)$ (*9*). Note the striking resemblance of the kernel K to the popular nine-point smooth:

$$\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}. \quad (2)$$

This filter is available on many nuclear medicine computers as a fast, assembly-language program that uses bit shifting to effect the multiplications and division. This nine-point smooth program can readily be modified to yield K, thus further enhancing the speed of the Chebyshev method. The number of multiplications is then reduced to $M^2(N + 1)$ assuming the bit-shift operations are essentially instantaneous relative to multiplication (*9*).

When digital filtering is performed entirely in the frequency domain, $M^2(4 \log_2 M + 1)$ multiplications are required, as in the Fourier-transform implementation of the FIR method described above.

Computer hardware and measurements. The filter algorithms were implemented in FORTRAN on a 16-bit minicomputer* and in AP assembly language on an array processor†. All programs were written to fit in 32K of host memory without requiring storage of intermediate results on disk or extended main memory. The programs for the array processor did not use the symmetry properties of the filters. A 64 by 64 pixel image size was used, since that matrix size is widely used in nuclear medicine, especially in cardiac studies. Larger arrays would require storage of intermediate results on disk, resulting in loss of computational speed. FIR filters from size 3 × 3 to 23 × 23 were evaluated. The computation time was measured for each method, excluding the time required to read the image in or out of main memory, but including the time to pass the images between the host and array-processor memories. To avoid the wrap-around error in FIR filtering, a border of width N was deleted from the filtered image as the last step in the processing (*3*).

RESULTS

Figure 2 shows the timing results for FIR and frequency-domain filtering both with and without use of the array processor. The FIR values are shown for filters of size 3 × 3 to 23 × 23 using the symmetric algorithm, the new Chebyshev method, and the array processor. The timing is the same for all FIR-filter sizes using the Fourier transform implementation. Note the marked reduction in computation time for the Chebyshev method compared with the symmetric algorithm. Without the array processor the Fourier-transform implementation is faster than the symmetric algorithm for filters larger than 13 × 13, while the Chebyshev computation is superior to the Fourier-transform method for filters of all sizes shown here. Computation time is identical for pure frequency-domain filtering and for the Fourier-transform implementation of FIR filtering, since the mathematical operations are the same. Computation time for FIR filtering with the array processor is 10–20 times faster than the Chebyshev method, the fastest conventional technique. Fourier-transform filtering on the array processor is 10–80 times faster than the Chebyshev method.

The relative number of multiplications required by each method was computed from the formulas given above. A very close correlation was observed between the measured timing values and the computed number of multiplications. Such a comparison was not performed for the array processor, since the pipeline architecture of the add and multiply units on those machines invalidates a direct comparison based on the number of multiplications.

DISCUSSION

If digital filters are to be applied in routine clinical use, the computation time must be rapid. The popular "nine-point smooth" is an example of a widely used small filter that can be applied very quickly to an image, especially if an assembly-language implementation is used. There are, however, reasons to use larger, and hence slower, filters. These more elaborate filters, computed in either the spatial- or frequency-domain, can be designed with special properties tailored to match the characteristics of the imaging equipment or to enhance certain features of the image (*2–6*). A larger filter will also more closely approximate the desired frequency response (*11,4*).

As shown in Fig. 2, the new Chebyshev method yields remarkably fast computation of even large FIR filters without the use of an array processor. This approach is faster by a factor of ~4–10 than the conventional convolution (symmetric algorithm) and the fast Fourier-transform methods. The most rapid computation continues to be with the array processor, although the gap between the conventional computer and array processor is significantly narrowed with the new algorithm.

Use of the Chebyshev algorithm may improve the speed of computation with the array processor. Unfortunately, storage of the intermediate images would require an array processor with

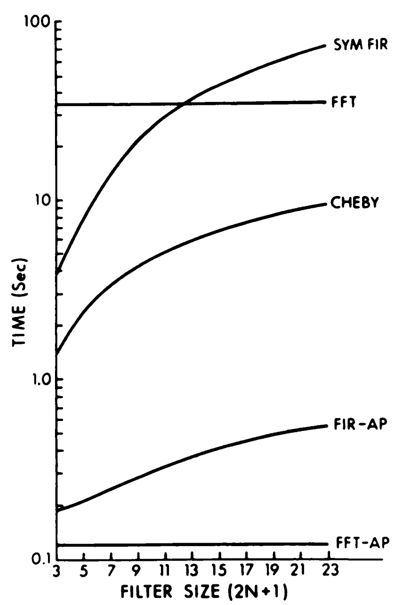

FIG. 2. Time required to filter a 64- by 64-pixel image is shown using Fourier transform (FFT) method and symmetric FIR and Chebyshev spatial-domain filters of size 3 × 3 to 23 × 23. AP denotes results obtained with an array processor.

memory larger than is now commonly available, or time-consuming transfers from array processor to host memory.

To illustrate these results in a practical setting, a 32-frame gated cardiac blood-pool study can be analyzed in less than 3 min (5.2 sec/frame) with an 11 × 11 filter using the Chebyshev algorithm, excluding a small amount of disk I/O time, whereas the computation would require almost 15 min (27 sec/frame) using the symmetric convolution. With the array processor, the calculation requires 10 sec (0.32 sec/frame) using the FIR method or only 4 sec (0.12 sec/frame) with the Fourier-transform implementation.

The timing results may differ substantially when using computer hardware, programming languages, or image sizes other than those applied in this work. While different computers and array processors will, of course, have different inherent computational speed, the relative performance of the different algorithms should parallel the results in Fig. 2 since the timing results reported here closely correlated with the number of multiplications, the rate-limiting step in computational speed in most computers. Images larger than 64 × 64 pixels may exceed the capacity of the host computer and array-processor memories, thus requiring time-consuming transfer of intermediate results to disk. The data reported here for the Chebyshev method were obtained with the bit-shifting implementation of the kernel, Eq. (1). If integer multiplication is used instead, the performance will be somewhat slower.

In conclusion, rapid—and thus clinically practical—digital filtering can be performed on a conventional minicomputer using a new, efficient algorithm.

FOOTNOTES

* Digital Equipment Corporation PDP-11/34A equipped with a hardware floating-point processor and cache memory.

† Analogic AP-400.

‡ A FORTRAN-IV program is available from the authors to perform the Chebyshev calculation.

ACKNOWLEDGMENTS

We thank David M. Epstein, M.D. for his valuable suggestions. We are grateful to Lorraine DiPlacido and Linda Stanfield for their assistance in the preparation of the manuscript.

The study was supported in part by National Institutes of Health Grant No. HL13851 and American Cancer Society Grant No. IN-129.

APPENDIX‡

The Chebyshev algorithm, shown in Fig. 1, may be programmed as follows using the arrays M_1, M_2, M_3, and R, and the filter values h_0 to h_N:

1) Place the original image in M_1.
2) Apply kernel K to M_1 and place the result in M_2 (without changing M_1).
3) Multiply M_1 by h_0; multiply M_2 by $2 \times h_1$; place the sum of the results in R (without changing M_1 or M_2).
4) Do the following loop: (each pass of the loop applies 2K twice.)
   5) Start with i = 2.
   6) Apply 2K to M_2, and place the result in M_3.
   7) Subtract M_1 from M_3, and place the result back in M_1;.
   8) Multiply M_1 by $2 \times h_1$. Add the result to the current pixel values in R, and place the result back in R.
   9) If i = N, skip to step 14. Otherwise, increment i by 1 and continue with step 10.
   10) Apply 2K to M_1, and place the result in M_3.
   11) Subtract M_2 and M_3, and place the result back in M_2.
   12) Multiply M_2 by $2 \times h_i$. Add the result to R, and place the result back in R.
   13) If i = N, skip to step 14. Otherwise, again increment i by 1, and go back to step 6.
   14) The filtered image is now in R.

REFERENCES

1. MILLER TR, GOLDMAN KJ, SAMPATHKUMARAN KS, et al: Analysis of cardiac diastolic function: Application in coronary artery disease. *J Nucl Med* 24:2-7, 1983
2. VERBA J, BORNSTEIN I, ALMASI J, et al: The application of three dimensional fourier filtering techniques to nuclear cardiac studies. *J Nucl Med* 20:658, 1979 (abst)
3. MILLER TR, SAMPATHKUMARAN KS: Digital filtering in nuclear medicine. *J Nucl Med* 23:66-72, 1982
4. MILLER TR, SAMPATHKUMARAN KS: Design and application of finite impulse response digital filters. *Eur J Nucl Med* 7:22-27, 1982
5. KING MA, MILLER TR, JACOBS DA, et al: Nonstationary image processing in the frequency domain. In *Digital Imaging, Clinical Advances in Nuclear Medicine.* New York, The Society of Nuclear Medicine, 1983
6. TODD-POKROPEK A: Image processing in nuclear medicine. *IEEE Trans Nucl Sci* NS 27:1080-1094, 1980
7. MERSEREAU RM, DUDGEON DE: Two-dimensional digital filtering. *IEEE Proceedings* 63:610-623, 1975
8. CAPPELLINI V, CONSTANTINIDES AG, EMILIANI P: Digital Filters and Their Applications. New York, Academic Press, 1978, pp 221-227
9. MECKLENBRAUKER WFG, MERSEREAU RM: McClellan transformations for two-dimensional digital filtering: II—Implementation. *IEEE Trans Circuits Syst* CAS-23:414-422, 1976

10. McCLELLAN JH, CHAN DSK: A 2-D filter structure derived from the Chebyshev recursion. *IEEE Trans Circuits Syst* CAS-24:372–378, 1977
11. RABINER LR, KAISER JF, SCHAFER RW: Some considerations in the design of multiband finite-impulse-response digital filters. *IEEE Trans Acoust, Speech, Signal Proc* ASSP-22:462–472, 1974