1996 International Conference on Parallel Processing

On the Scalability of 2-D Wavelet Transform Algorithms on Fine-grained Parallel Machines*

Jamshed N. Patel† Ashfaq A. Khokhar‡ Leah H. Jamieson†

†School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907
{jamshed,lhj}@ecn.purdue.edu

‡Department of Electrical Engineering
University of Delaware
Newark, DE 19716
ashfaq@ee.udel.edu

Abstract: We study the scalability of 2-D discrete wavelet transform algorithms on fine-grained parallel architectures. The principal operation in the 2-D DWT is the filtering operation used to implement the filter banks of the 2-D subband decomposition. We demonstrate that there exist combinations of the machine size, image size, and wavelet size for which the time-domain algorithms outperform the frequency domain algorithms, and vice-versa. We, therefore, demonstrate that a hybrid approach which combines time- and frequency-domain approaches can yield optimal performance for a broad range of problem and machine sizes. Furthermore, we show the effect of processor speed and the use of separable versus non-separable wavelets on the crossover points between the algorithm approaches.

Keywords: Wavelet Transform, Parallel Algorithms, Image Processing, Scalability Analysis, Fine-grained Architectures.

1 Introduction

Wavelet theory provides a unified framework for a number of techniques developed in multiresolution analysis [8, 10] and subband coding [5]. Although wavelets have been studied by mathematicians for many years, wavelet transforms have recently generated a great deal of interest as a new form of multiresolution representation for 1-D signals, e.g., for speech coding, for compression of 2-D images, and for 3-D analysis, e.g., for motion detection in video sequences [8].

The wavelet transform is an operation that transforms a function by integrating it with modified versions of some kernel function [2]. The kernel function is called the mother wavelet, and the modifications are translations and compressions (or dilations) of the mother wavelet. The DWT operates on sampled (or discrete) data in one or more dimensions.

In this paper we focus on the 2-D DWT. As with many algorithms in computer vision and image processing, the 2-D DWT is computationally intensive and operates on large data sets. A number of researchers have proposed parallel solutions for the DWT [3, 4, 6, 7, 11, 14].

This paper analyzes the effect of various machine and problem parameters on the performance of the parallel 2-D DWT algorithm. These parameters include machine size, problem size, wavelet size, processor speed, and use of separable versus non-separable wavelets. Most significantly, we also study the choice of parallel filtering algorithm.

The filtering operation in the wavelet transform can be accomplished either by convolutions with the filter kernel in the time domain, or by computing the discrete Fourier transform followed by point-by-point multiplication in the frequency domain. We show that the time domain and frequency domain approaches can be combined to achieve optimal performance for a broad range of problem and machine sizes. Our premise is that a parallel FFT algorithm is faster than a convolution operation when sizes of filter kernel and input signal are comparable. In wavelet decomposition using large filter kernels, there comes a stage due to the subsampling process when the size of the input signal is comparable to the filter size. We summarize analytical results and present experimental results on the MasPar fine-grained parallel computers.

2 2-D Discrete Wavelet Transform

This section describes the wavelet decomposition of a signal using a QMF (Quadrature Mirror Filter) filter bank. We outline the algorithms for the 2-D discrete wavelet transform (DWT). Additional details for the case of 1-D and 2-D signals can be found in [9].

In Mallat [9], it is shown that the computation of

*This work was supported in part by ARPA contract No. DABT63-92-C-0022 and by NSF Parallel Infrastructure Grant No. CDA-9015696. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

the wavelet representation can be accomplished with a pyramidal algorithm based on convolutions with quadrature mirror filters. The orthogonal wavelet representation of a discrete signal $x(n)$ can be computed by convolving with the filters $H(z)$ and $G(z)$ and retaining every other sample of the output, where $H(z)$ represents a lowpass filter and $G(z)$ represents a highpass filter. The process of decomposing the sequence into two sequences at half resolution can be iterated on either or both sequences. To achieve better resolution at lower frequencies, the scheme is commonly iterated on the output of the lower band.

In order to apply wavelet decompositions to images, 2-D extensions of wavelets are required. This can be achieved by the use of separable or non-separable wavelets as described below.

A separable filter implies that filtering can be performed in one dimension (rows), followed by filtering in another dimension (columns). A 2-D wavelet transform can be computed with a separable extension of the 1-D decomposition algorithm [9]. We first convolve the rows of $x(n,m)$ with a 1-D filter, retain every other row, convolve the columns of the resulting signals with another 1-D filter, and retain every other column. The filters used in this decomposition are the 1-D QMF filters. Further stages of the 2-D wavelet decomposition can be computed by recursively applying the procedure to the LL Band of the previous stage, where the LL band is the down-sampled signal that has passed through the $H(z)$ filter in the row and column dimensions. In general, K stages of wavelet decomposition result in a $(3K+1)$-band wavelet decomposition of the original image $x(n,m)$.

In certain cases, it is desirable to use non-separable subsampling to obtain useful 2-D wavelet representations [9, 13]. For example, non-separable wavelet orthonormal bases can be used for texture discrimination and fractal analysis [10]. In this case, we can use a non-separable 2-D subband decomposition scheme. as The input signal is convolved with $H(z_1, z_2)$ and $G(z_1, z_2)$, 2-D filters. Further iteration of the filter bank on the output of the lowpass branch leads to a 2-D DWT.

3 Scalable Parallel Implementation of 2-D DWT

In this section, we examine the scalability of the 2-D DWT on fine-grained machines. Our target architectures for fine-grained implementation are the MasPar MP-1 and MP-2. The MasPar computers are based on a 2-D mesh interconnection network. Hence we assume a 2-D mesh interconnect between processors, although the analysis can easily be generalized to other topologies.

The processors are distributed as a square grid of $\sqrt{p}\times\sqrt{p}$ processors, and each processor is connected to its eight nearest neighbors by a 2-D mesh. We assume a *block* distribution of the $\sqrt{n}\times\sqrt{n}$ image on the p processors; each processor holds a non-overlapping block of $\sqrt{n/p}\times\sqrt{n/p}$ pixels of the image, and adjacent processors in the $\sqrt{p}\times\sqrt{p}$ processor grid hold adjacent blocks of the input image. To simplify analysis, p is always assumed to be a perfect square, $\sqrt{n}$ and $\sqrt{p}$ are powers of 2, and $p \leq n$.

The separable 2-D DWT involves 1-D filtering of the rows of the input image with 1-D wavelet filters $H(z)$ and $G(z)$, followed by 1-D filtering of the columns of the image. We assume 1-D wavelet filters of size L, i.e., $h(i)$ and $g(i)$, $0 \leq i < L$. Two commonly used methods for filtering a digital signal are convolution-based filtering in the time domain, and point-by-point multiplication in the frequency domain. Due to space limitations, we refer to [12] for the detailed complexity analysis of the parallel algorithms. We have shown that the time to perform $K = \log n$ stages of convolution is bounded by the following equation.

$$T_{\text{conv}}(n,p) = O\left(L\frac{n}{p}\right). \qquad (1)$$

In the case of non-separable filter kernels, the time bound for the convolution based 2-D DWT decomposition is given as follows:

$$T_{\text{conv}}(n,p) = O\left(L^2\frac{n}{p}\right). \qquad (2)$$

To filter a signal in the frequency domain, we first compute the Discrete Fourier Transform (DFT) of the signal, perform a point-by-point multiplication with the DFT of the filter sequence, and then compute the Inverse Discrete Fourier Transform (IDFT) of the product. For this case, we have shown that the computation time for the K stages is given by:

$$T_{\text{FFT}}(n,p) = O\left(\left(\frac{n}{p}\right)\log n + \frac{n}{\sqrt{p}}\right). \qquad (3)$$

We assume that DFT coefficients of the filter are computed off-line and are stored in the local memory of each processor. The same asymptotic time bounds are achieved for the case of non-separable filters.

From equations (1) and (3), it is clear that the time taken for the convolution-based filtering increases linearly with the size of the wavelet filter kernel, while the time time taken by the FFT-based filtering operation is independent of L. Thus, for a given problem size and fixed number of processors, the frequency-domain method outperforms the time-domain method for the filtering operation when $L > K'\log n$, where K' is a constant and depends on the ratio of computation and communication power of the target machine. Comparing equations (1) and (2), the convolution-based algorithms for the non-separable case are slower by almost a factor of $L/2$. This implies that for fixed n and p, the crossover value of L at which the convolution-based algorithm becomes faster than the FFT-based algorithm is higher for the non-separable 2-D DWT.

4 Experimental Results on the MasPar MP-1 and MP-2

In this section, we discuss the implementation results on fine-grained SIMD machines. Fine-grained machines, in general, are characterized by a large number of processors with a fairly simple Arithmetic Logic Unit (ALU) in each processor. As discussed in Section 3, we use a block distribution of the $\sqrt{n}\times\sqrt{n}$ input image on the p processors. Each processor holds a non-overlapping block of $\sqrt{n/p}\times\sqrt{n/p}$ pixels of the image. Since the available memory per processor is limited, the number of image pixels per processor n/p is small ($\sqrt{n/p} \leq 32$).

For our experiments, we have used the MasPar MP-1 and MP-2 as being representative of fine-grained machines. Our algorithm implementations use the Xnet primitive for interprocessor communication. Xnet allows data to be sent in parallel in a specified direction using high-speed local routing on the mesh. We present implementation results on both the MP-1 and MP-2 to show the effects of processing power of individual processors on the performance of the algorithms. We have used an extended version of sequential ANSI C, the MasPar Programming Language (MPL), to keep our implementations free of machine-dependent software features.

To ensure consistent results using both time-domain and frequency-domain methods, we assume that the input image is periodic with period $\sqrt{n}\times\sqrt{n}$.

4.1 Separable 2-D DWT

Fig. 2 shows the performance of the time-domain and frequency-domain methods for increasing size of the 1-D wavelet filters $h(i)$ and $g(i)$, $0 \leq i < L$ on $16K$ processors of the MasPar MP-1. Fig. 2 (a) plots the time taken for an input image of size 256×256, and Fig. 2 (b) plots the time taken for an image of size 512×512 pixels. The experimental plots are consistent with the theoretical results stated in equations (1) and (3). The time for the convolution-based algorithm in the time-domain increases linearly with the filter size L, while the time for the FFT-based frequency-domain algorithm is independent of L. Similar results are obtained for larger image sizes ($\sqrt{n}\times\sqrt{n} = 1K\times1K$, $2K\times2K$, $4K\times4K$).

Fig. 2 (a) and (b) also indicate that the cross-over point – i.e. the filter-size at which the FFT-based algorithm outperforms the convolution-based algorithm – increases with image size for a fixed number of processors. We can determine the regions of the filter-size/image-size space where each algorithm is

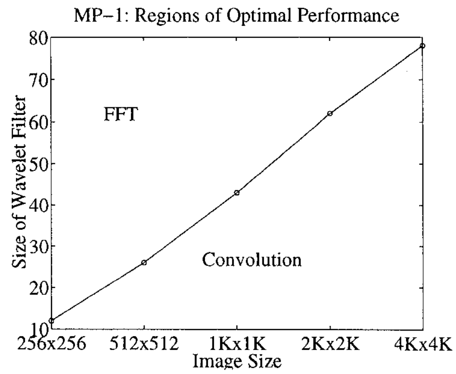

Figure 1: Experimentally determined regions in the image-size and filter-size space where each method is fastest. The plot is based on execution times for 16,384 processors of the MP-1.

fastest as shown in Fig. 1. Figures 2 and 1 clearly indicate that scalability is best achieved by changing the algorithm approach with a change in the problem parameters.

We have performed scalability experiments on the MasPar MP-1 and MP-2. Fig. 3 shows the scalability of the algorithms with respect to the problem size n. For a given wavelet filter and a fixed number of processors, the time taken by the FFT-based frequency domain method grows faster with increasing n than the time taken by the time domain method. This is consistent with the analytical results stated in equations (1) and (3).

Fig. 4 shows the scalability behavior of the algorithms with respect to increasing machine size. We notice that the execution time decreases for both algorithms with respect to p. However, beyond a certain machine size, the execution time for the FFT-based algorithm decreases faster than for the convolution-based algorithm. This can be explained by analyzing equations (1) and (3). As p increases, n/p decreases. Beyond a certain value of p, the filter size L begins to dominate the execution time of the convolution-based algorithm. Thus the frequency domain method outperforms the time domain operation for large L.

Both the MP-1 and MP-2 use the same Xnet communication network. The only difference between the architectures is the faster processors used by the MP-2. Thus, to study the impact of processor speed, we compare the performance of the algorithms using the same number of processors. It is evident from Figs. 3 and 4 that the execution characteristics of the algorithms are similar on the two machines. However, the crossover point (the ratio n/p at which the FFT-based algorithm outperforms the convolution-based algorithm) is higher on the MP-2 relative to the MP-1. This is because the ratio of computation to communication time is larger for

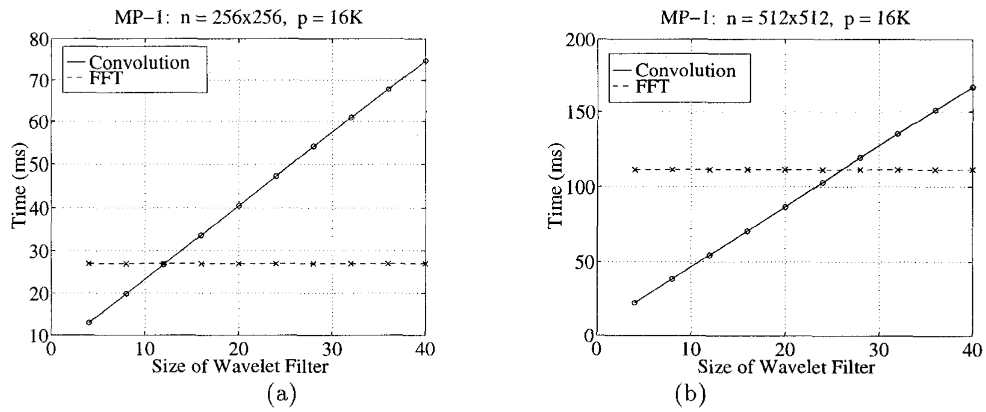

Figure 2: Performance of the algorithms for increasing size of the wavelet filter (L). The plots show the time taken by the time-domain and frequency-domain methods on 16,384 processors of the MP-1 for (a) an image of size 256×256 pixels, and (b) an image of size 512×512 pixels.

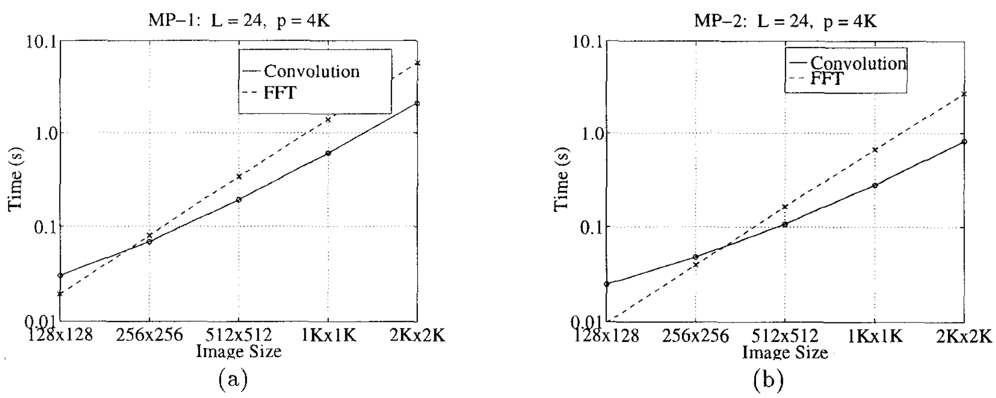

Figure 3: Comparison of performance on the MP-1 and MP-2: Scalability with respect to image size ($\sqrt{n} \times \sqrt{n}$), for a wavelet filter of size 24 on 4096 processors of (a) the MP-1, and (b) the MP-2.

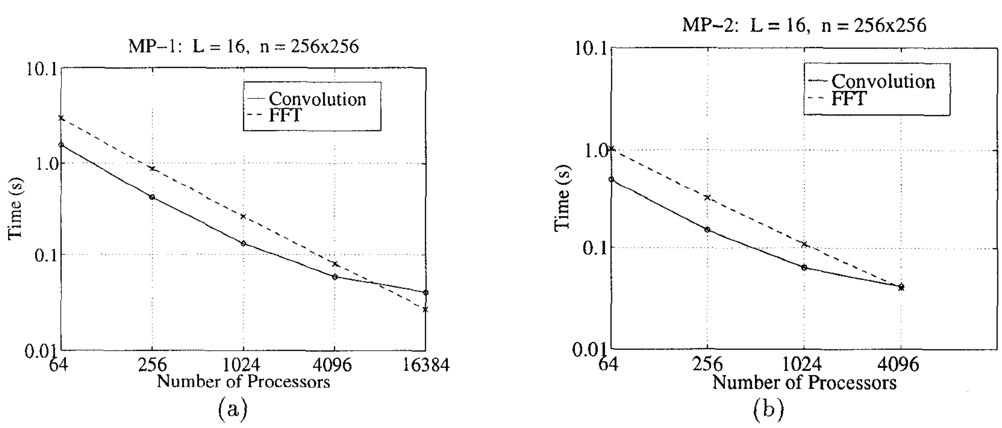

Figure 4: Comparison of performance on the MP-1 and MP-2: Scalability with respect to processor size (p), for a wavelet filter of size 16 and an input image of size 256×256 pixels, on (a) the MP-1, and (b) the MP-2.

the convolution algorithm, hence it performs better on the MP-2 due to the higher computation-speed/communication-speed ratio of the MP-2.

As described in Section 2, the filter bank implementation of the wavelet decomposition involves successive filtering and subsampling of the lowpass band of the previous stage of decomposition. The effective problem size n_k decreases at each stage of the decomposition. For a given machine size and wavelet filter, we can determine the ratio $r = n'/p$ for which the frequency domain method outperforms the time domain method on that machine. If the input image size $n > rp$, we can perform the initial stages of the wavelet decomposition using the frequency domain method. When $n_k \leq rp$, we switch to the time domain method for the remaining stages of the wavelet decomposition. Thus, a poly-algorithmic approach will provide optimal performance over a broad range of problem and machine sizes.

5 Conclusions

In this paper, we have studied the scalability of 2-D discrete wavelet transform algorithms on fine-grained parallel architectures. We have compared the analytical complexity of the time domain and frequency domain techniques used to implement the filter banks of the 2-D subband decomposition schemes on parallel machines. Experiments on the MasPar MP-1 and MP-2 have validated the analytical results.

We have shown that while one algorithm performs significantly better than the other for a certain combination of the machine size, image size, and wavelet size, the opposite may be true for a different set of problem and machine parameters. By contrasting the results on the MP-1 and MP-2, we observe the impact of architectural parameters (e.g. ratio of computation to communication speed) on the relative performance of the two algorithms. Finally, we have shown how the time- and frequency-domain approaches can be combined to achieve optimal performance for a broad range of problem and machine sizes.

Acknowledgments

The authors would like to thank Pramila N. Srinivasan for many helpful discussions.

References

[1] A. P. Calderon and A. Zygmund. On the existence of certain singular integrals. *Acta Math.*, 88:85–139, 1952.

[2] J. M. Combes, A. Grossman, and P. Tchamitchian, editors. *Wavelets: Time-Frequency Methods and Phase Space.* Springer-Verlag, New York, second edition, 1989.

[3] I. Gertner, R. Peskin, and S. Walther. Parallel computation of the continuous wavelet transform. In *Proceedings of SPIE - Adaptive Signal Processing*, volume 1565, pages 414–422, San Diego, CA, July 1991.

[4] R. W. Hall, S. Kucuk, and M. Hamdi. Wavelet transform embeddings in mesh architectures. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 596–597, New York, NY, June 1993.

[5] J. Kovacevic and M. Vetterli. Non-seperable multidimensional perfect reconstruction filter banks and wavelet bases for *R. IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):533–555, 1992. I. Daubechies, S. Mallat, and A. S. Willsky, editors.

[6] D. Krishnaswamy and M. Orchard. Parallel algorithms for the two-dimensional discrete wavelet transform. In *International Conference on Parallel Processing*, pages III–47–III–54, St. Charles, IL, August 1994.

[7] J. Lu. Parallelizing mallat algorithm for 2-d wavelet transforms. *Information Processing Letters*, 45(5):255–259, April 1993.

[8] S. G. Mallat. Multiresolution approach to wavelets in computer vision. In J. M. Combes, A. Grossman, and P. Tchamitchian, editors, *Wavelets: Time-Frequency Methods and Phase Space*, pages 313–327. Springer-Verlag, New York, second edition, 1989.

[9] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[10] Y. Meyer. Orthonormal wavelets. In J. M. Combes, A. Grossman, and P. Tchamitchian, editors, *Wavelets: Time-Frequency Methods and Phase Space*, pages 21–37. Springer-Verlag, New York, second edition, 1989.

[11] M. Misra and T. Nichols. Computation of the 2-d wavelet transforms on the connection machine-2. In *Proceedings of the IFIP WG10.3 Working Conference on Applications in Parallel and Distributed Computing*, pages 3–12, Caracas, Venez, April 1994.

[12] J. N. Patel. *Scalable High Performance Computing: A Poly-Algorithmic Approach to Computer Vision and Image Processing Libraries on Parallel Systems.* PhD thesis, School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, May 1996.

[13] O. Rioul and P. Duhamel. Fast algorithms for discrete and continuous wavelet transforms. *IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):569–586, 1992. I. Daubechies, S. Mallat, and A. S. Willsky, editors.

[14] L. R. C. Suzuki, J. R. Reid, T. J. Burns, G. B. Lamont, and S. K. Rogers. Parallel computation of 3d wavelets. In *Proceedings of the Scalable High-Performance Computing Conference*, pages 454–461, Knoxville, TN, May 1994.