2-D Wavelet Transform using Fixed-Point Number Representation*

A. Ruiz†, J.R. Arnau†, J. M. Orduña†, V. Arnau†, F. Silla‡, and J. Duato‡
†Universidad de Valencia. Departamento de Informática.
Av. Vicente Andrés Estellés, s/n.
46100 - Burjassot (Valencia). SPAIN
‡Universidad Politécnica de Valencia.
DISCA.
Camino de Vera, s/n. 46022 - Valencia (SPAIN)

Abstract

The application of Discrete Wavelet Transform to real-time digital signal processing requires the use of dedicated hardware to provide the necessary speed. The use of integer arithmetic in this dedicated hardware may be crucial to speed-up the signal processing.

In this paper, we study the implementation of 2-D DWT using fixed-point number representation. The results show that this implementation can be performed without significantly losing quality.

Keywords: Real time vision, wavelet transform, signal coding.

1 Introduction

The Discrete Wavelet Transform (DWT) has received considerable attention in the field of image processing due to its flexibility in representing non-stationary image signals and its ability in adapting to human visual characteristics [1].

This transform needs enormous computational power, and in order to satisfy these requirements in many real-time applications, dedicated hardware implementation is required [4]. In order to develop a DWT hardware implementation, the use of fixed-point number representation provides two main advantages with respect to floating-point number representation: faster computation of DWT and easier design of electronic circuitry.

In this paper, we study the implementation of 2-D DWT using fixed-point number representation. The purpose is to find the minimum number of bits required

*Supported by the Spanish CICYT under Grant TIC2000–1151–C07–04

to ensure that both image processing and the intermediate representation of the coefficients are performed without significantly losing quality.

The rest of the paper is organized as follows: Section 2 briefly describes how to apply 2-D DWT to an image. Section 3 studies the required number of bits to be used in a 2-D DWT when using fixed-point number representation. Finally, Section 4 presents some concluding remarks.

2 Applying 2-D DWT to a Picture

A DWT is an orthogonal function that can be applied to a finite group of data. Functionally, it is very much like the Discrete Fourier Transform, where the transforming function is orthogonal, a signal passed twice through the transformation is unchanged, and the input signal is assumed to be a set of discrete-time samples. Both transforms are convolutions. Whereas the basis function of the Fourier transform is a sinusoid, the wavelet basic is a set of functions, which are defined by a recursive difference equation [3].

Generally, the wavelet transform is made by a Quadrature Mirror Filters (QMF) that provides a Low-Pass filter L and a High-Pass filter H. 1-D Wavelet transform is a lineal operation that transforms an input vector X with size 2^K, in two vectors with size 2^{K-1} [2]. The first vector is the result of applying the Low-Pass filter and the second vector is the result of applying the High-Pass filter. These filters can be repeatedly applied to the low-pass output of each previous iteration. Each step of re-transforming the low-pass output is called a *dilation*, and if the number of input samples is $N = 2^K$ and the filter is formed by two coefficients then a maximum of K dilations can be performed. The last dilation results in a single low-pass value and single high-pass value. Figure 1 shows this process.

The 2-D Wavelet Transform is a generalization of the 1-D transform. First, 1-D transform is applied to every line of the image, and then the 1-D transform is applied to every column of the result image. That is, the filters L and H are applied first in horizontal direction and then in vertical direction.

The result obtained after the application of the 2-D Wavelet Transform is an image divided in 4 quadrants, LL, LH, HL, and HH. The LL area contains the softened original image. The HL area shows horizontal details, the LH area shows vertical details and finally the HH area shows oblique details. Similarly to the 1-D transform, the process can be repeated on the quadrant LL of the previous transformation, so that the output decreases by n/4 in each step. Figure 2 shows three dilations of 2-D DWT applied to an image.

After the application of 2-D Wavelet Transform on an image, there will be a

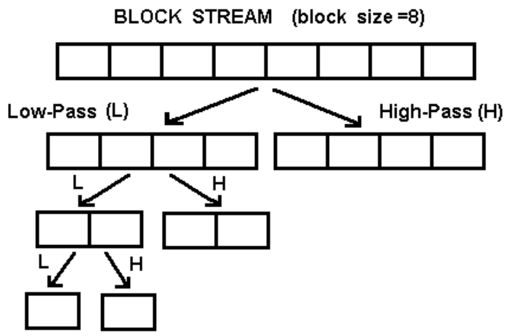

Figure 1: Dilations of a eight-sample block of data.

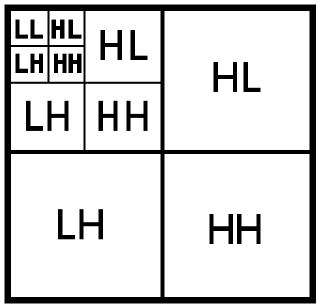

Figure 2: Three dilations applied to an image.

great quantity of coefficients with values very close to zero, providing a threshold that will allow, in a later compression process, to achieve bigger compression rates.

3 Fixed-point Data Representation

The coefficients of the Daubechies wavelet transform are real coefficients. Therefore, the use of integer arithmetic to apply this wavelet transform to an image introduces errors. Thus, it is necessary to study the required number of bits to be used in order to represent, using integer arithmetic, both the transform coefficients and also the image transformed values. The selected number of bits must guarantee a good quality of the final image.

First, we will study the required number of bits to represent the transformed image values. Since the sum of the filter coefficients is equal to 2, each time we compute a wavelet transform of a given image we are multiplying by a factor of two the pixel values of the image. Therefore, the transformed coefficients will require an additional bit for each new dilation. On the other hand, the transformed coefficients may be negative, requiring a new additional bit for the sign.

Figure 3 shows the required number of bits to represent the resulting image of applying three dilations of the Daubechies4 filter to a 512x512 image, with 8-bit pixel values. It can be seen that for this number of dilations the required number of bits is 12.

Figure 4 shows the required number of bits to represent only the high-pass area of the resulting image. Since the resulting values are obtained as subtractions of pixel values, the average value for this area does not need more bits to be properly

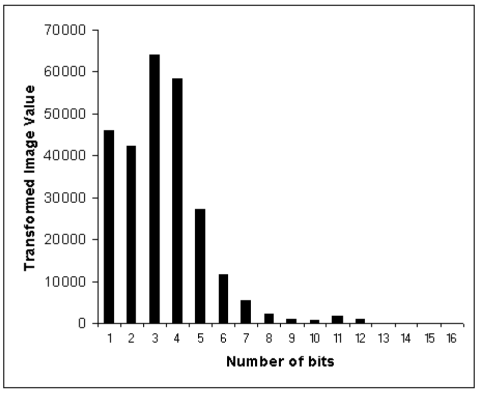

Figure 3: Histogram of the entire image.

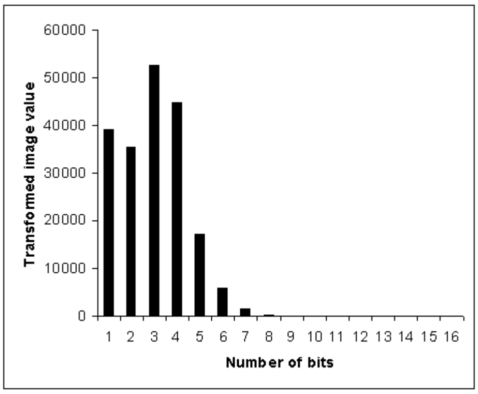

Figure 4: Histogram of high-pass area.

represented. Even more, the required number of bits is often less than 8 bits.

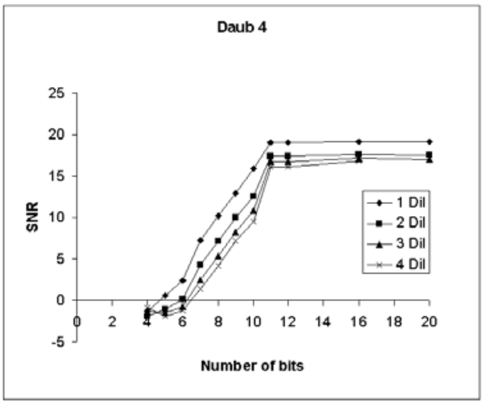

Figure 5: SNR for different number of bits used in Daub4 filter.

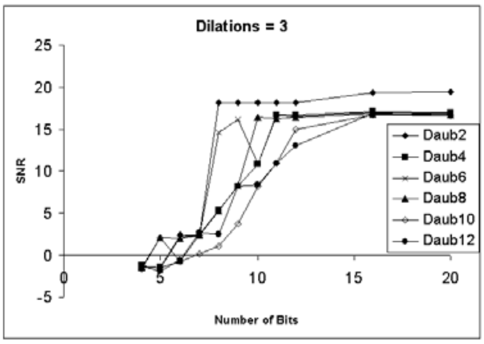

Figure 6: SNR for different number of bits and filter sizes.

Figure 5 shows the signal-to-noise ratio (SNR) for a reconstructed image after applying some different number of dilations of the Daub4 filter. This Figure shows that, since each dilation introduces errors due to the use of integer arithmetic, these errors are being accumulated as more dilations are applied, degrading the SNR obtained for the reconstructed image. It can also be seen that in order to represent the Daub4 transform coefficients and obtaining an acceptable SNR, at least 10 bits are required. Additionally, the use of more than 11 bits does not improve the SNR.

Figure 6 shows the SNR for different number of bits and for different filter sizes. It can be seen that using 10 bits for representing the filter coefficients results in a good SNR, regardless of the filter size. Although it is not shown here due to space limitations, the resulting images show that when using a small number of bits for the coefficients then the reconstructed image has a low luminance (the average pixel values decreases).

Additionally, we have studied the possible effects of image and size in the required number of bits to be used. Figures 7 and 8 show that the SNR of the reconstructed image is not significantly affected by the filter size nor by the image size.

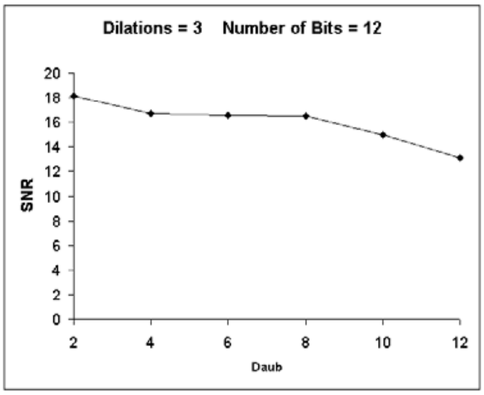

Figure 7: SNR obtained for different filter sizes.

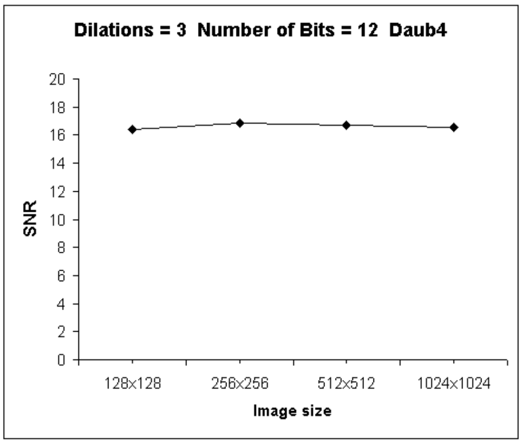

Figure 8: SNR obtained for different image sizes.

Figure 9 shows the obtained SNR for the reconstructed image when fixing the number of used bits and varying the number of applied dilations of different Daubechies filters. This figure clearly shows that the bigger the filter size is, the worse SNR is obtained for the reconstructed image. It is due to the errors introduced by the use of integer number representation. Although they are not shown here due to space limitations, the reconstructed images show that for more than 7 dilations there exists some areas in the reconstructed images with worse quality than in the original images.

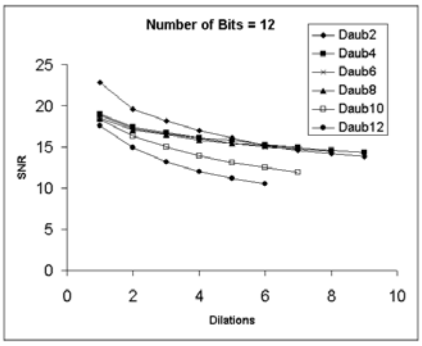

Figure 9: SNR obtained for different number of dilations.

4 Conclusions

The purpose of this paper has been to implement the wavelet transform in a dedicated hardware, obtaining simple circuits capable of providing the wavelet transform in real time. We have studied parameters such as the required number of bits to represent filter coefficients and pixel values, filter sizes and number of dilations.

The results show that in order to obtain a reconstructed image with good quality, the minimum required number of bits for representing the filter coefficients is 10. Additionally, an additional bit for each applied dilation must be used to represent the pixel values of the low-pass area of the resulting image, due to the average value of the filter coefficients. On the other hand, the required number of bits to represent the pixel values of the reconstructed image is 12 bits. Finally, for image sizes of 512x512 pixels, the application of three dilations provides a good quality of the reconstructed image.

References

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform", in *IEEE Trans. on Image Processing*, vol. 5, pp. 205-220, Apr. 1992.

[2] A. S. Lewis and G. Knowles, "Video compression using 3-D wavelet transforms", in *Electron. Lett.*, vol. 26, pp. 6, pp. 396-398, Mar. 1990.

[3] G. Strang, T. Nguyen, "Wavelets and Filter Banks", Wellesley-Cambridge Press Ed., 1996.

[4] C. Yu, S. J. Chen, "VLSI implementation of 2-D discrete wavelet transform for real-time video signal processing", in *IEEE Trans. on Consumer Electronics*, vol. 43, No. 4, pp. 1270-1279, Nov. 1997.