Implementation of Wavelet Decomposition and Reconstruction for an Image using TMS320C6701

Chung S. Leung, Sunil Kulkarni

Electrical Engineering Department
Texas A&M University-Kingsville
Kingsville, Texas 78363

Abstract

The discrete wavelet transform provides sufficient information both for analysis and synthesis of the original image with a significant reduction in the computation time. There are two approaches for working on the above algorithm, one being by using two dimensional filters and the other one by using separable transforms that can be implemented using one-dimensional filter on the rows first and then on the columns. In this research, we have implemented wavelet decomposition and reconstruction using one-dimensional transform applied on the rows first and then the columns. For an $N \times M$ image size, we filter each row and then columns with the analysis pair of low-pass and high-pass filters and down sample successively to obtain four bands after decomposition. Later during image reconstruction, we up sample and filter each column and then rows with pair of synthesis low pass and high pass filters to obtain the original image size. This algorithm has been implemented in real-time by using the floating-point processor TMS320C6701 chip manufactured by Texas Instruments (TI) that is widely used for image processing applications. The wavelet transform has become the most powerful tool for still image analysis. Yet there are many parameters within a wavelet analysis and synthesis that govern the quality of the image. In this paper, we discuss the wavelet decomposition and reconstruction strategies for a two-dimensional signal and their implications on the reconstruction of the image. A pool of grey scale image has been wavelet transformed using a set of bi-orthogonal filters (wavelet filter bank) that undergoes the decomposition and reconstruction process.

Introduction

The dyadic and wavelet transform is originally defined only for one-dimensional signals. The application of wavelet transform to still image offers different possibilities to decompose a signal. In this paper, we investigate the effects of a non-standard method used for the decomposition and reconstruction of the image. An empirical evaluation of each coding performance on grey-scale images focuses on the visual quality. Yet there are many parameters

as mentioned earlier, that govern the quality of a decoded image: choice of the wavelet filter bank, image boundary policy, decomposition and reconstruction strategies used. In our algorithm, we have implemented it by choosing a bi-orthogonal 4.4 filter as its features help in getting better reconstruction of the image. Also we have performed symmetric periodic extension of the signal around the boundaries both during the signal decomposition and reconstruction.

With an N×M image size, each row is filtered with the analysis pair of low pass and high pass filters to obtain two $N \times M/2$ image. We then filter each column and sub-sample the filter output to obtain four $N/2 \times M/2$ images. Therefore, after performing one level two-dimensional wavelet transform with an image, we obtain the following four sub-images:

LL: obtained by low pass filtering the rows and columns.
LH: obtained by low pass filtering the rows and high pass filtering the columns.
HL: obtained by high pass filtering the rows and low pass filtering the columns.
HH: obtained by high pass filtering the rows and high pass filtering the columns.

Just as forward wavelet decomposition, the inverse wavelet transform or reconstruction is done exactly the opposite way to that of the decomposition. Of the four obtained sub-band images, we do the up sampling first followed by filtering applied to the columns and then to the rows to get the reconstructed image. To obtain better reconstruction of the original image, we used periodic symmetric extension of the signals both during the decomposition and reconstruction process.

LL	*HL*
LH	*HH*

Fig. 1: One level 2-D wavelet transform

Wavelet Transform

We have introduced the concepts of wavelets and multi-resolution analysis and how efficiently wavelets are used for decomposition and reconstruction of signal. The wavelet is an ideally compact function, i.e., outside a certain time interval it vanishes. Implementations are based on the fast wavelet transform, where a given wavelet, i.e., mother wavelet is shifted and dilated so as to provide a base in the function space. In other words, a one-dimensional function is transformed into a two-dimensional space, where it is approximated by coefficients that depend on time (determined by the translation parameter) and on scale, i.e., frequency (determined by the dilation parameter). The zoom phenomenon of the wavelet transform offers high temporal localization for high frequencies while offering good frequency resolution for low frequencies. Consequently, the wavelet transform is especially well suited to analyze local variations such as those in still images.

By introducing multi-resolution, Mallat [1] made an important contribution to the application of wavelet theory to multi-media; the transition from mathematical theory to filters. Multi-resolution analysis is implemented via high-pass filters, respectively band-pass filters (wavelets) and low-pass filters (scaling functions). In this context, the wavelet transform of a signal can be realized by means of a filter bank via successive application of a 2-channel filter bank consisting of low-pass and high-pass filters: the detail coefficients resulting from the application of high-pass, respectively band-pass filters) of every iteration step are kept apart, and the iteration starts again with the remaining approximation coefficients (from application of the low-pass filter) of the transform. The multi-resolution theory is 'per se' defined only for one-dimensional signals. Since still images are two-dimensional discrete signals, our current research is restricted to separable filters. The successive convolution of filter and signal in both dimensions opens to two potential iterations: standard and non-standard decomposition. Since we are dealing with non-standard decomposition strategies, it iterates only the purely low-pass filtered approximations while leaving the mixed terms unchanged.

Bi-orthogonal Filters

With orthogonal filters, the wavelet transform can be viewed as projecting the input signal onto a set of orthogonal basis function. However, the standard orthogonal transform has some shortcomings that make it less than ideal for use in a coding system. One shortcoming is that total number of input coefficients, N does not equal to the total number of coefficients L using the maximally decimated wavelet transform. In general L is greater than N and the wavelet transform results in co-efficient expansion. This expansion is illustrated with a small example. Consider a length N (even input) and length M (even) wavelet filters. The outputs of the filters h and g will be $N+M-1$, and the output of the decimators would be $(N+M)/2$. Thus, the N original input samples result in a total of $N+M$ wavelet coefficient after one level of transform. More levels of wavelet analysis only make the problem worse, since more levels result in more $N+M$ samples. Coefficient expansion is a problem for coding where the aim is to reduce and not increase the coefficients. These extra coefficients are undesirable because:

- They require more bits to code so that the reconstructed signal accurately represents the input.
- They do not represent any information present in the original signal, but rather are an artifact of the method used to perform the transform.

The lack of linear phase filters in orthogonal wavelets led to research in extending wavelet analysis to more general forms, which would allow for linear phase filters and more general form of wavelets known as bi-orthogonal wavelet. Thus, because of these drawbacks of the orthogonal wavelets we have implemented this algorithm using bi-orthogonal wavelets with periodic symmetric extension for achieving better reconstruction. In summary, the bi-orthogonal wavelet transform has the advantage that it can use linear phase filters, but the disadvantage that it is not energy preserving. Therefore it is not a big problem that bi-orthogonal wavelets are not energy preserving due to the fact that there are linear phase bi-orthogonal filter, which coefficients are close to being orthogonal.

Periodic extension of signals

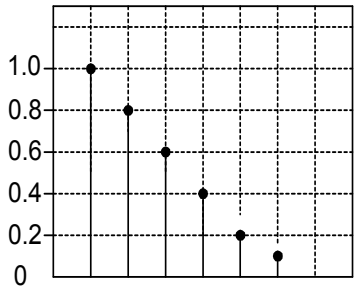

Fig. 2: Input signal

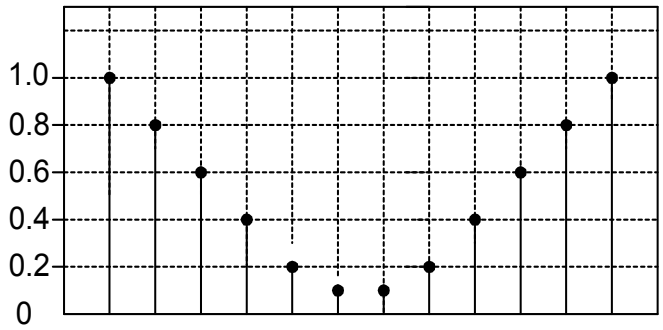

Fig. 3: Symmetric extension of signal

Performing a symmetric periodic extension of the input signal instead of periodic extension can eliminate the border or edge artifacts introduced with the use of wavelets [2]. This symmetric extension guarantees continuity across replicas of the input and eliminates the large wavelet coefficients caused by border discontinuities. However, the symmetric extension doubles the number of input samples. This is not a problem as half of the samples are redundant due to symmetry. When the input signal is filtered and decimated, the outputs are not necessarily symmetric and periodic. As a result, half the coefficients cannot be eliminated due to symmetry, and there is a doubling of the number of coefficients required to represent the input. Fortunately, symmetry can be preserved across the scales of wavelet transform by imposing an additional constraint on the wavelet filters must be either symmetric or anti-symmetric (also known as linear phase in signal processing terminology). For this special case, with a periodic and symmetric input, the output is also periodic as well as symmetric which thus resulted in no co-efficient expansion. The use of symmetric extensions and linear phase wavelet filters does solve the problem of border effects in the wavelet transform. The symmetric extensions are of quite a few types, but the three basic periodic symmetric extensions used in this algorithm are as follows:

- The symmetric extension of the boundary is of type (1,1), i.e., the first and last samples appear only once and are whole samples.
- The symmetric extension of the boundary is of type (2,1), i.e., the first sample appears twice where as the last samples appears once.
- The symmetric extension of the boundary is of type (1,2), i.e., the first samples appear only once and the last sample appears twice.

Detailed explanation of the wavelet decomposition for 2D image

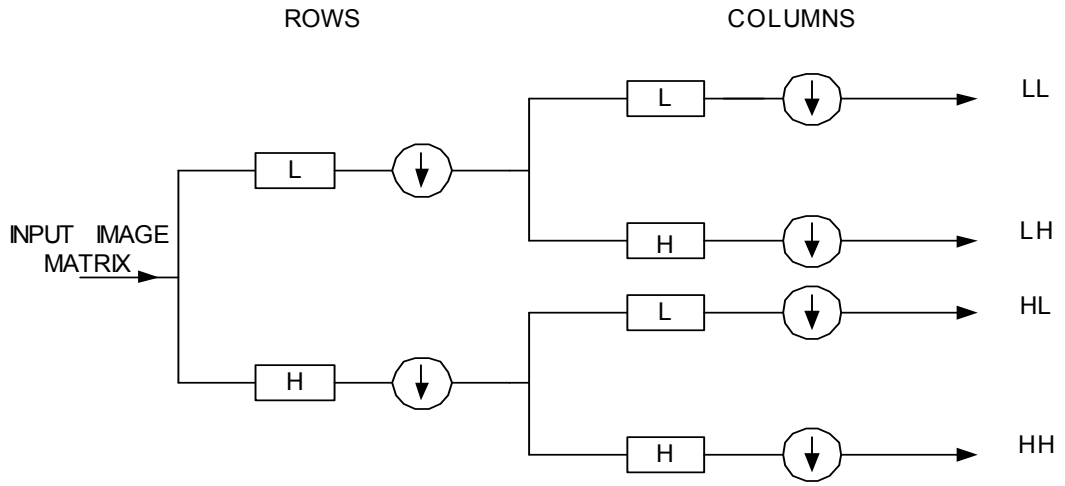

Fig. 4: Wavelet decomposition for 2D Image

In this algorithm, we are basically concentrating only on two-dimensional images. Each 2-D image can be classified as a grey scale image and that the corresponding grey scale value of any image is determined by using MATLAB. Therefore, the input two-dimensional array for this entire program would be the grey-scale value obtained. Once the coefficients are stored in the two-dimensional array, we first do the symmetric periodic extensions along the rows. This algorithm has been implemented using a bi-orthogonal filter due to its unique property of getting better reconstruction with the images. The input array obtained after symmetric periodic extension is then low-passed as well as high-passed along each rows of the array. After completing the filtering operation along each of the rows, each of the row is then down sampled by a factor two which means every odd sample is eliminated from the array obtained after convolution Then we undergo the same process of filtering and convolution but on the columns this time [3]. Functions were written for all the important process during this algorithm like the convolution, down sampling as well as up sampling and for the corresponding transforms applied along the rows. All the resultant arrays are then low-pass filtered as well as high-pass filtered. Later, all these are down sampled by the same factor 2 to obtain four sub-bands, i.e., *LL, LH, HL* and *HH* after the first stage of decomposition.

Detailed explanation of the wavelet reconstruction for 2D image

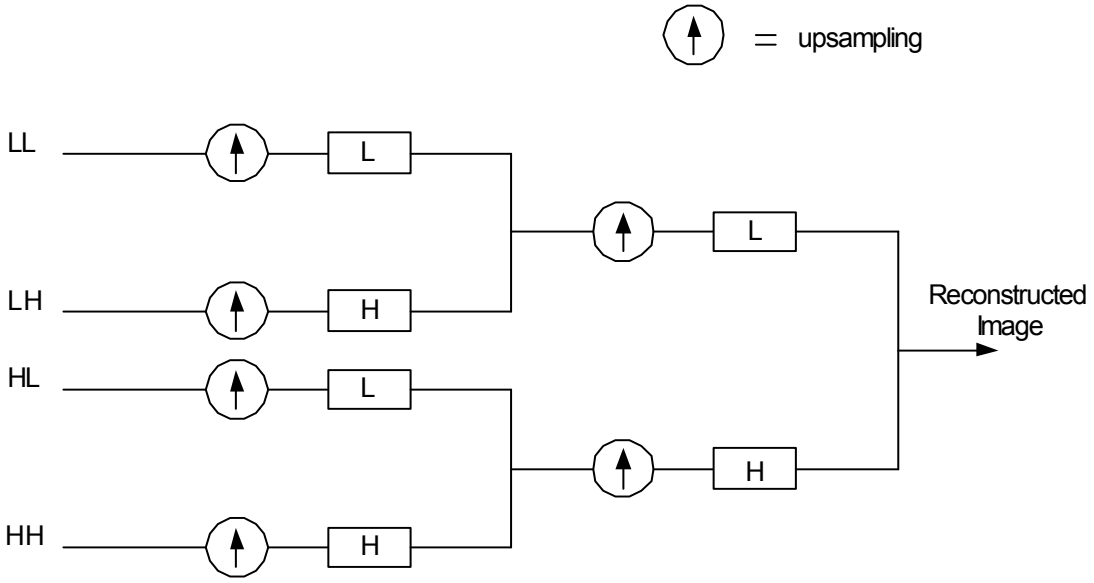

Fig. 5: Wavelet Reconstruction for 2D Image

During the reconstruction part of the image, we apply the transforms on the columns first and then on the rows [3]. Periodic symmetric extension is performed on all the four sub-bands obtained after the decomposition stage. The resultant array of all bands obtained after periodic extension are up sampled by a factor of two the reason being the coefficients were down sampled by the same factor 2 during decomposition. After up sampling by a factor 2, all the coefficients are zero padded followed by the low-pass and high-pass filtering. Same operations are iterated again but this time the transforms are applied on the rows to get the reconstructed image. Thus, we get the same original image after reconstruction and with the use of bi-orthogonal filters and the image quality after reconstruction is much better.

Implementation of algorithm on TMS320C6701

TMS320C6701 EVM (Evaluation Module) is a floating-point processor manufactured by TI [4]. Many real-time DSP applications processors chosen are dependent on many factors such as time, cost, power, efficiency, and maintenance. However for this particular application, we have chosen TMS320C6701 EVM for it's special feature of accepting floating-point instruction set. In the above algorithm, we first determined time-critical functions were identified first. The three functions that were time-critical in the entire algorithm were convolution, up sampling and down sampling. All the three functions were written in assembly language. The basic way of doing this was by assigning a pointer to the beginning of the data so that they can be treated as an array. Finally, simple functions are added in both C and assembly to illustrate how function calling

works. This method of placing data in memory is simple and easy to use and can be used for applications in which constants need to be in memory, such as filter coefficients.

Code composer studio (CCS) is a useful integrated development environment (IDE) that provides an easy to use software tool to build and debug programs with TMS320C6701 processor. In addition it allows, real time analysis of application programs. During its setup CCS is configured for target DSP boards as C6x EVM. This software includes an integrated editor for editing both C and assembly files. Another important compiler option is the target version option. For the floating-point target DSP TMS320C6701 application, the target version option of the compiler needs to be set to 6701 from the pull down menu list.

All of the files required for creating an executable file using code-composer are as follows:

1. A *C* file of the entire algorithm.
2. The *.asm* source files are nothing but the assembly files for the functions that were translated from c to assembly language.
3. A command file *(.cmd)* is used as it is very important in deciding the layout of entire data as to what portions of memory should it utilize on the processor.
4. The header files *(.h)* files are included in the project as it stores all the data details than would be used as the input samples for the algorithm.
5. The library files *(.lib)* used for this particular processor is the (rts6701.lib)
6. A *vector.asm* file is added every time when the chip/processor is reset.

The CCS code development begins with the creation of a Project to easily integrate and manage all these required files for generating and running an executable file [5]. The project view panel on the left hand side of the CCS window provides an easy mechanism for doing so. A project file can be created or opened to contain not only the source and library files but also the compiler, assembler and linker options for generating an executable file. As a result one need not type command lines for compilation, assembling and linking, as was the case with original software development tools.

After adding all the source files, command file and library file to the project, the project is built or to create an executable file for the target DSP. To do so, we need to choose the project → rebuild menu item. CCS compiles, assembles and links all the files in the project. When building process is completed without any errors, the *(.out)* file is generated. Once the *(.out)* file is generated, it is loaded on the processor by Load program option in the code composer studio. After this, the processor is run to get the desired results or output.

Result and Analysis

Two image were selected, and they are: *cameraman.jpg*, and *scenary.jpg*. Five different sizes have been tested for cameraman.jpg and they are: 8×8, 16×16, 32×32, 64×64 and 128×128 to comply with different features and hence provide a valid average values. The *cameraman.jpg* file has various transitions and areas of texture, and *scenary.jpg* file has all sharp features and curves.

As discussed earlier, for implementing this algorithm we have used a bi-orthogonal 4.4-filter that helped us achieve better reconstruction of the image for one-level 2-D wavelet transform.

After performing the wavelet decomposition and reconstruction of the 2-D images, we perform various tests to determine the quality of the reconstructed image. The various analysis measures included computing the Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) between the original and reconstructed image. An approximation of the original image in most of the cases is enough, as long as the error between the original and reconstructed image is tolerable.

Mean Square Error

One of the error metrics used to compare the various image quality techniques is the Mean Square Error (MSE) The MSE is the cumulative squared error between the reconstructed and the original image. A lower value of MSE is good as it indicates there is very little error between the original and reconstructed images.

Peak Signal to Noise Ratio

Signal to Noise ratios are estimates of the quality of a reconstructed image compared with original image. The basic idea is to compute a single number that reflects the quality of the reconstructed image. Reconstructed image with higher metric are judged better. The actual parameter we will compute is the peak signal-to-noise ratio (PSNR) for the reconstructed image. Assume we are given a source image *f(i,j)* that contains *N* by *N* pixels and a reconstructed image *F(i,j)* where F is reconstructed by decoding the encoded version of *f(i,j)*. Error metrics are computed on the luminance signal only so the pixel values *f(i,j)* range between black (0) and white (255). Therefore, we first compute the mean square error followed by the PSNR to determine the image quality.

The other important technique for displaying errors is to construct an error image that shows the pixel-by-pixel errors. The simplest computation of this image is to create an image by taking the difference between the reconstructed and original pixels. These images are hard to see because zero difference is black and most errors are small numbers that are shades of black. The typical construction of the error image multiples the difference by a constant to increase the visible difference and translates the entire image to a grey level.

Following are two test images *cameraman.jpg* and *scenary.jpg* that were tested and analyzed to plot the reconstructed image, error image compute the value of Mean Square Error and Pixel Signal to Noise Ratio. The test images size are:

- *cameraman* (128×128)
- *scenary* (128×128)

Fig.6: Original Image

Fig.7: Reconstructed Image

Fig. 8: Error Image (*cameraman.jpg*)

Fig.9: Original Image

Fig.10: Reconstructed Image

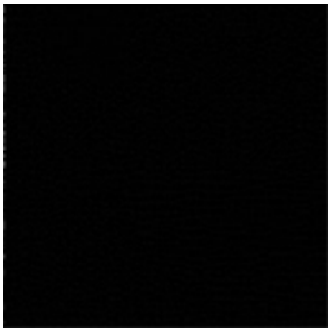

Fig. 11: Error Image (*scenary.jpg*)

The results of the empirical evaluation are calculated and shown in Fig.6 through Fig.14 respectively for a 128×128 size 2-D image.Fig.6 is the original image and Fig.7 is the one that we obtained after reconstruction of the image with this algorithm. Fig. 8 demonstrates the actual difference or the mean square error between the two images. Thus, the similarity between both the original and reconstructed images concludes the implementation of the algorithm. Since this was implemented on TMS320C6701 floating-point processor, we term the entire process as a real time implementation of the algorithm. The mean square error computed for all the three test images is tabulated below.

Test Images	Mean Square Error	PSNR (dB)
cameraman	207.8020	24.97
scenary	11.7445	37.45

Typical PSNR values range between 20 and 40. They are usually reported to two decimal points (e.g., 25.47). The actual value is not meaningful, but the comparison between two values for different reconstructed images gives one measure of quality. Also, the PSNR measure for a reconstructed and original image is given in decibels (dB).

Profiling of the functions

Performance is a key issue for embedded system development. As the program continues to grow in size and complexity, it becomes increasingly more difficult to isolate the subtle problems that can cause poor performance. Profiling helps to reduce the time it takes to identify and eliminate performance bottlenecks. The profiler analyzes program execution and shows where the program is spending its time. As discussed earlier, the time-critical functions in the entire algorithm were identified, as they were the most called functions in the entire C program. These identified functions in this algorithm were convolution, up sampling and down sampling. The number of clock cycles required for each function in C or assembly is calculated and compared by a method called profiling which is nothing but counting the clock cycles for a particular section, function or a piece of code.

There are different ways in which the profiling of these functions could be carried out. In our methodology of profiling, a particular area, range or section of code was selected to carry out this operation. The start line of the code to begin profiling from was selected and a break point was set at the end location. After the compilation of the program and execution of it on the chip, the number of clock cycles or CPU cycles needed to complete the operation was computed.

There was a significant reduction in the number of clock cycles using assembly language as compared to the C language, the reason being all the time critical functions were in hand assembly language, because of which the CPU registers on the processors are used for all the arithmetical and mathematical calculations for all the assembly based functions, optimization of the C code was achieved to a great extent. Hence, the purpose of using the TMS320C6X processor was satisfied for a great extent. Here, we have taken *cameraman.jpg* as the test image

and all of its sizes were analyzed and tested to determine the values of the clock cycles both in C and assembly languages and is tabulated as shown below.

Test Size of Image (*cameraman*)	No.of clock cycles (only C)	No.of Clock cycles (assembly)	No.of cycles (reduced)
8×8	2416140	1045663	1370477
16×16	9660000	4182652	5474348
32×32	38400000	17730608	20669392
64×64	151026145	70922432	80103713
128×128	600102765	289689728	310413037

Conclusion

The interplay of representation and approximation of signals was reviewed. The concepts of wavelets and multi-resolution analysis were studies and we have seen how with the use of wavelet transform, an efficient decomposition and reconstruction of signal is achieved. Implementation of the algorithm using TMS320C6701 processor and translating all the time-critical based functions in assembly language helped to optimize the code and make this entirely as a real-time based project.

Bibliography

[1] Zoltan Nagy and Kamil Vrba, "Transition from Mallat's Pyramidal algorithm to Lifting Scheme Based Wavelet Transform", www.electronicletters.com.

[2] Bryan E. Usevitch, "A tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG2000," *IEEE Signal processing Magazine*., vol. 18, pp 59-73, September 2001.

[3] M. Vetterli and J. Kovacevic, "Wavelets and Subband Coding", Prentice Hall, 1995.

[4] N. Kehtarnavaz and M. Keramat, "DSP System Design Using the TMS320C6000", Prentice Hall, 2000.

[5] R. Chassaing, "DSP application using C and the TMS320C6x DSK", John Willey & Sons, January 2002.

CHUNG S. LEUNG received the Ph.D. degree in Electrical Egnineering from the Florida Atlantic University in 1989. He has been a faculty in Texas A&M University-Kingsville since 1990. His research interests are in the area of real-time DSP, image compression and microprocessor interface.

SUNIL KULKARNI is an EE graduate student in Texas A&M University-Kingsville. His research interests are in the area of image compression and real-time DSP.