An Optimal Wavelet Filter-Based Method that Reduces Execution Time for Unstructured Image Compression Algorithms.

V. Kustov, P. Srinivasan, S. Mitra, and S. Shishkin, Department of Electrical Engineering, Texas Tech University, Lubbock, TX, 79409, MS-3102
Telephone: (806)742-1381
E-mail: smitra@coe.ttu.edu

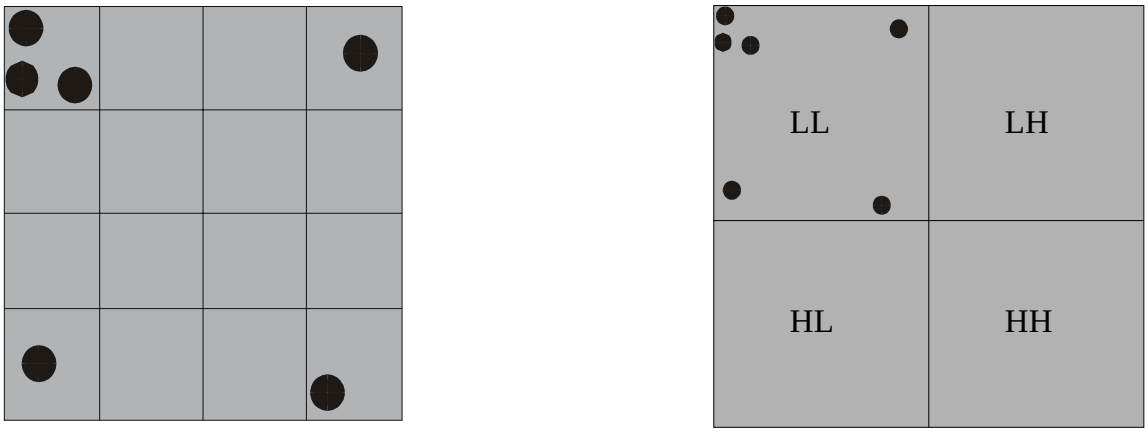
Abstract – **The dramatic increase in the computational speed provided by state-of-the-art high-end DSP's offers the possibilities of processing large high-resolution images in real time. The authors have experience in high-fidelity lossy compression of high-resolution color medical images (up to 2048x4096 pixels, eight bits per pixel) using TI's TMS320C6201 DSP's. The amount of on-chip memory in C'6000 DSP cores is insufficient for processing such images using internal memory only. Allocating the data to the external memory can increase the execution time significantly. The use of direct memory access channels (DMA) does not lead to a substantial decrease of the execution time for unstructured algorithms, i.e. the algorithms in which the data processing order is highly dependent on the results of the previous steps in the algorithm. We have developed and tested a technique, which decreases execution time for such algorithms by applying them to only one subband of a signal dependent wavelet transform.**

1. Introduction

Advanced lossy image compression algorithms utilizing vector quantization [1,2,3] offer low distortion in the reconstructed images at high compression ratios (even above 100:1). Such low distortion rate makes these algorithms a good choice for effective storage and internet transmission of large high-resolution images for many applications. While storage space has become a lesser problem in recent years, fast internet transmission is an acute problem due to increasing internet traffic and the limited channel bandwidths available for most consumers. Obviously, if the original image is compressed a hundred times, it takes one hundred less time to receive the image given the same transmission channel conditions. While the compressed image can be transmitted for viewing fast, the decoding time depends on the type of the processor employed, the decoding algorithm, and the total size of the data used during decoding. If the compression takes place upon the client's request (only original uncompressed images are stored at the transmitting end), then the total time that takes to transmit and display the image includes the encoding time as well. Since the main goal in compressing images for transmission is to transmit and display the received images faster, the total coding/decoding time should be less than the time for transmitting the compressed images. This last requirement is frequently not met by the vector-quantization based algorithms mentioned above; the coding/decoding time can be in excess of one minute for 6-24Mbytes original image sizes using Pentium 400 MHz processors.

The use of faster processors such as Texas Instruments' C'6000 DSP's can decrease the encoding/decoding time. However, this decrease can be dramatic for some algorithms while only marginal for others. A common problem faced by many algorithms when applied to large data sets is a slow data access rate. For example, 512 Kbits of data memory available on TI 'C6201 DSP and 4 Mbits on C'6203 are insufficient to perform

compression storing the images (6-24Mbytes) and intermediate data entirely in the internal memory. Allocating the data to the external memory can increase the execution time significantly (the new ‘C64x core is expected to run at 1.1 GHz, while a typical SDRAM device, the most common inexpensive type of memory, runs at 100-120 MHz). The common solution to this problem is the use of direct memory access channels. This solution is very effective when the algorithm is well structured, i.e. it is known in advance which block of data needs to be transferred from the external to the internal memory, while the current block of data is being processed. A good example of such an algorithm is the DCT-based JPEG, which divides the input image into 16Kbyte independent blocks of data and processes them separately. Fig. 1 a illustrates such a technique.



(a) (b)

Fig 1.
Only three blobs are within the same independent block (a)
All the blobs are within the same LL subband of a wavelet transform (b)

In that figure the entire data set is divided into 16 separate blocks of data of such sizes that each of them can be processed entirely in the internal memory. Then, while the processor is encoding the first block, the second block is fetched from the external to the internal memory using DMA. The processor is freed from data trafficking and concentrates on “crunching numbers”. Unfortunately, such use of DMA leads to a significant execution time reduction only if it is known in advance which block of data should be processed next after the current block is done. The basis of vector quantization algorithms is in finding similar vectors (blocks of the image/image transform matrix) and then approximating (clustering) them by some vector (centroid vector). Fig. 1 shows six blobs that form similar vectors. If the 16 blocks of data are encoded independently, then only the three blobs that are in the same block will be approximated by one vector, while encoding the entire data set at the same time approximates the remaining three blobs by the same vector thus achieving a higher compression.

The solution presented in this paper is based on encoding only one subband of the output of a multirate filter bank (commonly referred to as the discrete wavelet transform).

Fig. 1 b. illustrates such a technique. The original image is divided into four subbands marked LL, HL, LH, and HH in the figure. Each subband contains incomplete (in general) information about the entire image. Thus, as it can be seen, in Fig 1. b, if we process one subband (for example LL) the size of the data set is reduced by four, yet all the six blobs will be approximated by the same vector thus achieving a maximum compression for this example.

2. Background

Unless otherwise specified all functions used in this paper are assumed to be real and compactly supported. Functions' arguments and indices are assumed to be integer, if not specified otherwise. All functions are assumed to be bounded. If the upper and lower limits of a summation are omitted, the operation is performed on all values of the argument(s) for which the expression inside the summation is supported. Where the extension to two-dimensions is straightforward one-dimensional notations are used.

The part of the wavelet decomposition created by applying the low-pass filter in both directions is called the low-frequency subband/low-resolution part (denoted LL) of the decomposition. The rest of the decomposition forms the higher frequency subbands/detail components (denoted LH, HL, and HH).

Let us denote by $h_0(n)$ and $h_1(n)$ the low-pass and high-pass filters, respectively, that form a two-channel filter bank. The filter bank is orthogonal if the following equations hold true[4]:

$$\sum_n h_0(n+2m)h_0(n) = \delta(m), \tag{2.1}$$

$$\sum_n h_1(n+2m)h_1(n) = \delta(m), \tag{2.2}$$

$$\sum_n h_1(n+2m)h_0(n) = 0, \tag{2.3}$$

The image matrix can be convolved with $h_0(n)$ and $h_1(n)$ and downsampled by two in both dimensions. The result is a matrix of the same as the original image matrix dimensions, which schematically depicted in Fig. 1 b. The resulting matrix is called a discrete wavelet transform of the original image. If (2.1) – (2.3) hold then by applying the inverse procedure to the wavelet transform we get the original image without any distortion. This condition is called the perfect reconstruction condition (PR). Because of downsampling by two in two dimensions each subband contains one fourth of the number of elements in the original image. If the downsampling by two and convolution are applied again just to one of the subbands, we will get the second level wavelet transform. Each subband in the second level transform will contain one sixteenth of the number of elements in the original image. Applying the same procedure iteratively will lead to subbands containing $\frac{1}{4^l}$ of the elements in the original image. Here l denotes the lth level of the transform.

If the following equation is true the amplitude spectrum of $h_0(n)$ and $h_1(n)$ are symmetrical (mirrored) with respect to $\frac{\pi}{2}$

$$h_0 = (-1)^{k+1} h_1(N-k), \qquad k = 0..N-1 \tag{2.4}$$

and $h_0(n)$ and $h_1(n)$ form a filter bank called a quadrature mirror filter (QMF) bank[5]
Note that if (2.4) and (2.2) are satisfied then (2.1) and (2.3) are always satisfied. Thus, if we find $h_1(n)$ that satisfies (2.2), we can always find $h_0(n)$. Filters satisfying either (2.2) or (2.1) are called Nyquist (2) filters. If either $h_1(n)$ or $h_0(n)$ is a Nyquist (2) filter and (2.4) holds, then $h_0(n)$ and $h_1(n)$ form a perfect reconstruction QMF bank (PR-QMF).
Note, since there are infinitely many solutions to obtain Nyquist (2) filters for filter lengths greater than two, we can design PR-QMF filter banks with some desired properties. Finally, we require that the Fourier transform of $h_1(n)$ have a zero DC component, or in the time/spatial domain:

$$\sum_n h_1(n) = 0 \tag{2.5}$$

If (2.5) is satisfied, and $h_0(n)$ and $h_1(n)$ form a QMF bank, then $h_0(n)$ has a zero at π.
This last requirement is due to the fact that typical images contain a large DC component, which we want to put entirely in one of the subbands.

3. Description of the execution time reduction technique

Briefly, our technique can be described in several steps.
In the first step, some statistics of the entire image is amassed (typically the autocorrelation sequence). The first step uses the external memory if the image does not fit into the internal memory, but this step is not computationally expensive.
In the second step our new interior-point-based wavelet optimization algorithm finds the optimal wavelet filter bank that compacts the energy (for a given input image) in the low-resolution part of the wavelet decomposition. This part does not operate on a large data set and uses the internal memory only. The execution time for this step is small and independent of the size of the input image.
In the third step, using the optimal filter, a discrete wavelet transform is performed. This step uses the external memory, but it is not computationally expensive.
Finally, the detail component of the wavelet decomposition that contains small coefficients is completely discarded (in practice it is not even computed). Only the low-resolution part of the wavelet decomposition that contains one fourth (because we use two-channel filter banks in two dimensions) of the elements in the original image is used as an input to the main compression algorithm.
A block diagram of our technique is shown in Fig 2

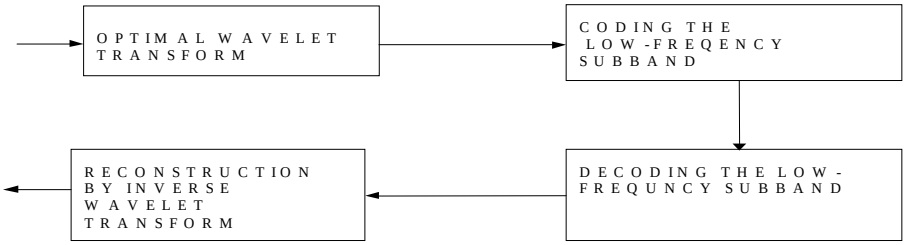


Fig 2.
A block diagram of our optimal wavelet-based method.
The main compression algorithm acts on ¼ of the transform

The decoding is performed in reverse order. First the main compression algorithm decodes the low-resolution part of the transform. Then the detail components of the wavelet transform are approximated by zeros, and the inverse wavelet transform is performed.

Since the goal of this technique is reduction of the execution time, a fast procedure for finding "the best" PR-QMF filter bank is required.

Some previous work[6] in this area was aimed at finding an optimum filter bank analytically. There are two drawbacks in that technique. Firstly, it does not work for every autocorrelation sequence – for some autocorrelation sequences the algorithm completely fails. Secondly, the final step of that algorithm requires spectral factorization of a polynomial, which is not an analytical procedure (for $N > 4$).

Our algorithm uses an interior-point optimization algorithm[7] to find "the best" PR-QMF filter bank. Interior-point methods are fast global optimization algorithms for constrained problems that exhibit a polynomial convergence rate with respect to the number of variables and constraints. They also possess a logarithmic convergence rate with respect to the initial distance from starting point to the solution, i.e. $\propto \log(\frac{R}{\varepsilon})$,

where R is the distance and ε is the desired accuracy of the solution.

Formally, the optimization problem is stated as follows:

given the autocorralation matrix $C \in R^{NxN}$, where N is the length of the optimal filter $h_1(n)$, find $h_1(n)$, such that

$$h_1^T C h_1 \to \min \qquad (3.1)$$

under the constraints (2.2). Here h_1 is the column vector that represents the high-pass filter $h_1(n)$. In particular, we have implemented the algorithm for $N = 6$. (for $N = 2$ the orthonormal filter bank is unique, and for $N = 4$ it was shown in [8] that Daubechies' Maxflat filters are optimal in terms of (3.1)) under frequently met assumptions on the autocorrelation sequence. For $N = 6$ Daubechies' Maxflat filters are near optimal[8] with respect to (3.1) (under some assumptions on the autocorrelation sequence), and therefore provide a good benchmark for comparison.

The main problem in using interior-point methods for (3.1) is that (3.1) is not a convex/concave function, and interior-point methods can only be used directly for convex/concave functions. We have been able to reformulate the problem without loss of generality into a convex optimization problem and then solve the latter by an interior-point method (to find filter $h_1(n)$ that puts the smallest amount of energy into the detail components of the wavelet decomposition). Let us describe briefly the main stages of our method. First using (2.2), (2.5), we parameterize the vectors $h_0(n)$ and $h_1(n)$ by the vector $y \in R^5$ to remove the linear equation (2.5) and we rewrite Eq (2.2), and (2.5) using vector notations:

$q^T h_1 = 0$, $\quad q = [1,1,1,1,1,1]^T$, then:

let $y = Gh_1$, $\quad G \in R^{6x5} : q^T G = 0$

and write the optimization problem in the form:

$$y^T A_0 y \to \min \tag{3.2}$$

$$y^T A_i y = 0, \quad i = 1,2; \quad y^T A_3 y = 1, \tag{3.3}$$

where matrix $A_0 = G^T C G$ is obtained from C by parameterization by G, and A_1, A_2, A_3 are obtained from (2.2) after the parameterization by G (to remind: (2.1) and (2.3) follow from (2.2) and (2.4)).

Then we consider the auxiliary optimization problem:

$$x^T (A_0 - \mu A_3) x \to \min \tag{3.4}$$

$$x^T A_i x = 0, \quad i = 1,2 \quad x \in R^5 \tag{3.5}$$

and adjust $\mu \in R$ in such a way that the optimal value of (3.4) and (3.5) are equal to zero. When such μ_* is found and the optimal solution x_* is located, the solution of (3.2) and (3.3) can be obtained by

$$y_* = \frac{x_*}{\sqrt{x_*^T A_3 x_*}} \tag{3.6}$$

It remains to describe the solution method for (3.4) and (3.5) when μ is fixed. For any $\lambda \in R^2$, let us denote

$$B = A_0 - \mu A_3 \qquad B = \begin{matrix} \tilde{B} & b \\ b^T & \beta \end{matrix} \qquad A_i = \begin{matrix} \tilde{A}_i & a_i \\ a_i^T & \alpha_i \end{matrix}$$

$$\tilde{A}_\lambda = \tilde{B} + \lambda_1 \tilde{A}_1 + \lambda_2 \tilde{A}_2, \quad a_\lambda = b + \lambda_1 a_1 + \lambda_2 a_2, \quad \alpha_\lambda = \beta + \lambda_1 \alpha_1 + \lambda_2 \alpha_2$$

$$f(\lambda) = a_\lambda^T \tilde{A}_\lambda^{-1} a_\lambda - \alpha_\lambda,$$

and consider the optimization problem:

$$\text{minimize } (f(\lambda) : \tilde{A}_\lambda \geq 0) \tag{3.7}$$

If λ_* is the solution of (3.7), then $x_\lambda = \begin{matrix} -\tilde{A}_\lambda^{-1} a_\lambda \\ 1 \end{matrix} \in R^5$ is the solution of (3.4) and (3.5). It is easy to verify that the problem (3.7) is convex. Then, an interior-point algorithm described below is used to find the global minimum:

$\mu = 0$;

repeat:

$B = A_0 - \mu A_3$;

$\lambda_* = \arg\min(f(\lambda)/\tilde{A}_\lambda \geq 0)$;

$\mu_* = \arg\max(\gamma / A_\lambda - \gamma A_3 \geq 0)$;

$\mu = \mu + \xi\mu_*$;

until: $\mu_* < \varepsilon$;

$$y_* = \frac{x_\lambda}{\sqrt{x_\lambda^T A_3 x_\lambda}}$$

where ξ, ε are the algorithm's parameters.
On all of the test images the interior-point algorithm converges to a global minimum after 14 iterations in 0.2-second (Pentium II 400MHz CPU) independent of the input image size.

4. Results and discussion

We tested our algorithm by applying it to different images. The results below present both the performance of the overall algorithm (of Fig. 2) and an important intermediate figure of merit – the amount of energy in the higher-frequency subbands, which are discarded. Since distortion in the reconstructed image is caused by discarding these subbands (approximating them by zero), a small amount of energy in those subbands lead to a small distortion. To evaluate the energy compaction capability of our algorithm let us define the figures of merit used below (we do not use the coding gain as the figure of merit, because in general optimization for coding gain and energy compaction are not the same[9]). Let us denote by E_d the amount of energy in the higher subbands of a one level decomposition using Daubechies' Maxflat filters and by E_o using optimal filters.

$$T = \frac{E_d - E_o}{(E_d + E_o)/2} x100\%$$

Table 4.1 below shows the algorithm's performance when it was applied to four test images.

Table 4.1 Energy decrease for gray scale images

Image	Improvement (T) in the detail part of the decomposition.
Lena	7.07
Visible Human	3.75
Calgary	18.95
Banff	56.44

Figure 4.1 shows the effect of discarding the detail part on the reconstructed (Calgary) image.

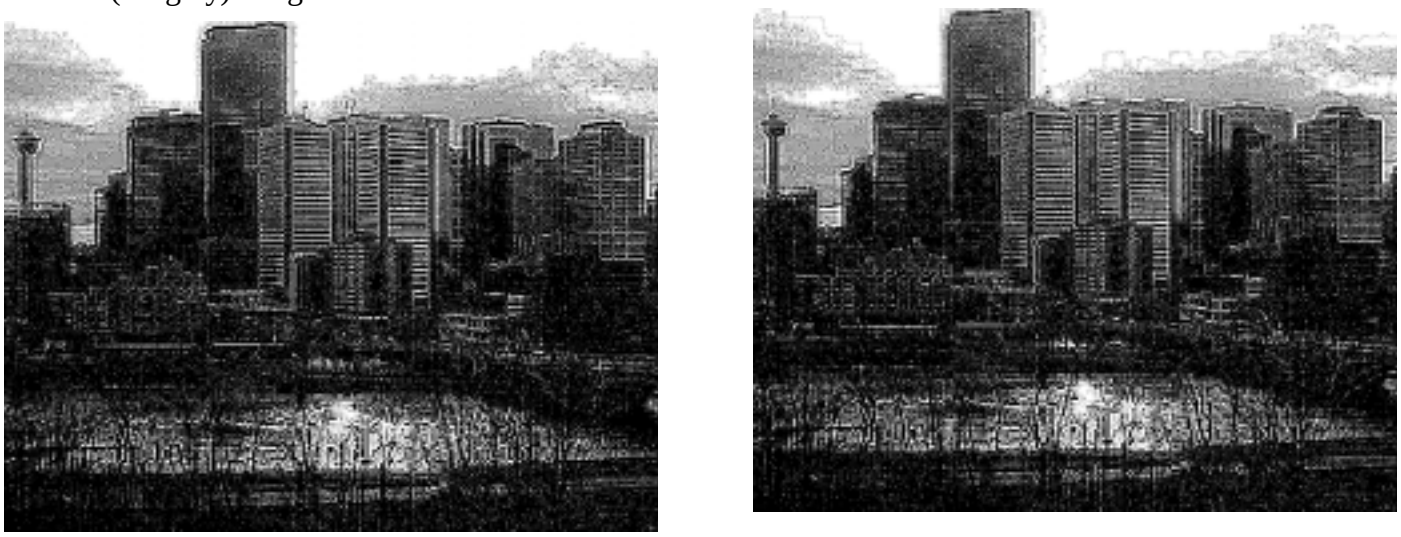


Fig 4.1
Edge enhanced image of the Calgary skyline reconstructed from only the low-frequency subband using optimal QMF bank of length 6 (left) and Daub 6 (right)

As it can be seen in Fig. 4.1 the image on the right suffers considerably from the ringing artifact (the right side of the tall building and the clouds). The image on the left does not have such a well-pronounced artifact.

The evaluation of the overall performance of our algorithm was conducted using the EZW algorithm[10] as the main compression algorithm of Fig 2. Although EZW employs scalar quantization, the results can be extended to vector quantization algorithm without loss of generality. Table 4.2 shows the distortion for the Lena image when the EZW algorithm used the entire wavelet decomposition, and when it used only the low-frequency subband of the optimal transform. The optimal filters were used only for the first level decomposition. The main (EZW) algorithm used Duabechies' Maxflat filters of length six.

Table 4.2 Distortion for the standard Lena image at 100:1 compression ratio

Technique Used	MSE	PSNR
EZW	330.93	22.93
EZW with preprocessing*	319.82	23.08

***Preprocessing means EZW is applied to the low-frequency subband of the wavelet transform instead of being applied to the original image**

As it can be seen the distortion is comparable (there is some increase in PSNR when our technique was used). The overall execution time, however, was reduced 3.9 times compared with the EZW applied to the entire decomposition.

However, at lower compression ratios (30:1) our technique introduces more distortion compared with the coding algorithm applied to the entire image. When lower compression ratios can be tolerated, (a high bandwidth is available or transmission time is not crucial), the DCT-based JPEG is fast and introduces low distortion at low compression rates.

The above results show that our execution time reduction technique can be used to reduce encoding/decoding time of advanced image compression algorithms without an appreciable increase in distortion at high compression ratios. One straightforward way of further decreasing distortion introduced by discarding of the detail components is to increase the filter length. One can view wavelet filters as basis vectors in N dimensional space. An image is reconstructed from its wavelet transform can be viewed as a superposition of these basis vectors. Longer basis vectors will be able to represent the image using fewer significant coefficients, if the basis vectors are selected appropriately. The fact that PR-QMF banks were used in our algorithm has its advantages as well as drawbacks. If most of the image's energy is distributed at spatial frequencies $\varpi < \frac{\pi}{2}$ (which is almost always the case), the reduction in the amount of energy passed by $h_1(n)$ will lead to the reduction of aliasing between the subbands. This leads to an overall improvement in the distortion rate (since aliasing also introduces a significant distortion when some of the subbands are discarded).

There are two major drawbacks in using PR-QMF banks. Firstly, if some filter $h_1(n)$ is optimal, it may not be a Nyquist (2) filter, thus we have to be satisfied with some other $h_1(n)$, which is Nyquist (2), but suboptimal. In case of biorthogonal filter banks, given an optimal $h_1(n)$, a biorthogonal filter bank can almost always be computed to satisfy the PR condition (in the biorthogonal case $h_1(n)$ does not have to be Nyquist (2)) . Secondly, an iterative numerical procedure could not be avoided. Thus, computing longer filters takes significantly more time.

5. Conclusion

We have developed and tested a technique, which decreases execution time for image compression algorithms by applying them to only one subband of a signal

dependent wavelet transform. The EZW algorithm has been used as the main compression algorithm. Further tests will be carried out with AFLC-VQ^2. In the most general case the reduction of the execution time is achieved due to the reduction of the number of elements that need to be processed. If the reduction is sufficient to put all the data entirely into the internal memory, an additional reduction is achieved, since the data no longer should be fetched from the external memory. This is particularly important for unstructured algorithms, for which the use of DMA does not offer significant execution time reduction. If the reduced data set cannot fit the on-chip memory entirely, the number of elements residing in the external memory is reduced leading to fewer accesses of the external memory.

Our current research is conducted on biorthogonal filter banks with embedded aliasing reduction to replace PR-QMF filter banks in order to reduce the computation time for finding optimal filters and increase the solution space.

6. References

1. A. Gersho, and R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, (1995).
2. S. Mitra, and Shu-Yu Yang, "High Fidelity Adaptive Vector Quantization at Very Low Bit Rates for Progressive Transmission of Radiographic Images", *Journal of Electronic Imaging*, Vol.8, No. 1, pp 23-35, 1999.
3. M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, "Image coding using wavelet transform," IEEE Transactions on Image Processing, 1(2), 205-220 (1992).
4. G. Strang, T. Nguyen, "*Wavelets and Filter Banks*", Wellesley-Cambridge Press 1997.
5. M Vetterli and J. Kovacechic, "*Wavelets and Subband Coding*", Prentice Hall, 1995.
6. A. Kirac and P. Vaidyanathan, "Theory and Design of Optimal FIR Compaction Filters", *IEEE Transactions on Signal Processing*, VOL. 46, NO. 4, April 1998.
7. Y. Ye, "*Interior Point Algorithms: Theory and Analysis*", John Wiley & Sons Inc., 1997.
8. B. Usevitch and M. Orchard, "Smooth Wavelets, Transform Coding, and Markov-1 Processes", *IEEE Transactions on Signal Processing*, VOL. 43, NO. 11, November 1995.
9. P. Moulin and M. Mihcak, "Theory and Design of Signal-Adapted FIR Paraunitary Filter Banks", *IEEE Transactions on Signal Processing*, VOL. 46, NO. 4, April 1998.
10. J. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Signal Processing*, VOL. 41, NO. 12, December 1993.