

Image Compression via Edge-Based Wavelet Transform

Alfred Mertins

University of Wollongong, Australia

Abstract

In this paper, an edge-based wavelet transform is proposed for image compression. In the encoder, first the dominant edges of an image are detected and coded as side information. Then, the wavelet transform (WT) is carried out in such a way that no filtering over previously detected edges is performed. Compression examples show that the edge-based WT achieves good reproduction of sharp edges - even at very low bit rates. Because of the additional side information, the PSNR is typically slightly lower than for standard wavelet coders. On the other hand, the picture quality scale, an objective quality measure that better reflects the subjective impression, shows superior results for the proposed coder.

Keywords: Wavelet Transform, Image Compression, Edges, Signal Processing

Alfred Mertins

University of Wollongong

School of Electrical, Computer and Telecommunications Engineering

Wollongong NSW 2522, AUSTRALIA

Email: mertins@uow.edu.au

Telephone +61 2 4221-3410

Facsimile +61 2 4221-3236

1 Introduction

The discrete wavelet transform (DWT) is known to be one of the most efficient tools for image compression [1]. The principle of this transform is to hierarchically decompose a signal into a multiresolution pyramid, where the signal is split into a coarse approximation and some detail information at each resolution level [2]. The approximation will be further decomposed in the next stage. The attractiveness of the DWT results from the fact that it provides very good compaction properties for many classes of natural images while the implementation cost is low.

Each lossy compression scheme shows its typical artifacts when coding is performed at very low bit rates. For the DWT, the main artifacts are “blurring” and “ringing”. Blurring results from neglecting most of the high-frequency details of the image and cannot be avoided at very low bit rates. However, one can try to allocate bits in such a way that the produced blurring is not very annoying to human observers. Ringing is related to the Gibbs phenomenon and occurs in the vicinity of sharp edges. The amount of ringing depends on the wavelets in use and also on the bit allocation. However, at extremely low rates, ringing usually cannot be avoided.

The idea behind the coding technique presented in this work is to overcome the ringing problem by separating images at positions where sharp edges are located. The DWT is then carried out in such a way that no filtering over sharp edges is performed. This type of transform will be called the edge-based DWT throughout this paper. The approach includes, but is not limited to so-called shape adaptive wavelet transforms, where images are partitioned into a set of regions and each region is transformed separately. The proposed algorithm is related to the work in [3], but the way of considering edges and the way of processing finite-length signals are entirely different.

The basic requirement for the edge-based WT is the ability to perform support preservative subband decompositions of finite-length signals. Various techniques for this purpose are known in the literature [4–13], but most of them are not suitable for the edge-based transform, because they only allow the processing of even-length signals. Due to the open structure of edges that may occur at any position and do not necessarily surround closed regions, special requirements on the wavelet transform arise. Especially, it is required to carry out wavelet transforms with

arbitrarily located wavelets on arbitrary-length intervals that allow perfect reconstruction in the absence of quantization and that do not introduce redundancy. Suitable schemes providing PR can be found in [6, 8, 11]. However, as will be outlined in Section 3, they suffer from the drawback that the energies of the boundary analysis and synthesis filters may differ significantly from the energies of the original filters. This means, the propagation of quantization errors from a subband to the output depends on the position of the errors within the subband. On the analysis side, the spatially varying energies of the analysis filters result in spatially varying variances of the subband signals (even for a stationary input process). This makes an optimal bit allocation difficult. Moreover, white quantization noise being introduced in the subbands may appear as highly correlated noise at the output. This problem has recently been addressed in [14], where general solutions for the optimization of the boundary filters and the equalization of their energies have been presented. However, the optimization approach presented in this paper differs from the one in [14] in the parameterization and requires a lower number of operations for the implementation.

The paper is organized as follows. Section 2 presents the general idea of the proposed coder. In Section 3, the boundary processing scheme is presented, and in Section 4, further implementation details are given. In Section 5, the coding performance of different schemes is evaluated, and in Section 6, coding examples are presented. Finally, Section 7 gives a brief conclusion.

2 The Edge-Based Wavelet Transform

The general principle of the edge-based DWT is shown in Figure 1. In both directions, filtering is interrupted in front of an edge and is continued behind the edge, so that no filtering over sharp edges is performed. The edge information is then forwarded to the next resolution level. In order to avoid the introduction of redundancy, and in order to maximally exploit the correlation between pixels, the following requirements shall be formulated:

- (i) The decomposition scheme must be support preservative and must allow perfect reconstruction.

- (ii) The edge-based and the standard DWT of an image should differ only in the vicinity of edges. This ensures that the horizontal/vertical filtering introduces a minimum amount of discontinuity in the vertical/horizontal direction.

Since edges may occur at any position in an image, strategies are required for handling the decomposition process in any possible situation, and also strategies for decomposing the edge information and for passing it on to lower resolution levels are needed. The proposed scheme is based on the definition of edges between the pixels, rather than on the pixels. This results in a simple decomposition scheme for the edges, which is uniquely defined once the global alignment of the wavelets to the image and the sequence of the horizontal and vertical decomposition are fixed.

For the discussion of the filtering and edge decomposition process, we focus on the horizontal decomposition, which is carried out prior to the vertical one. Since we need to distinguish between segments that start or stop at even or odd positions, let us say that the pixel in the upper left corner of an image is at position $(0,0)$. We will consider the example of an 8×8 image in Figure 2. The first two rows in Figure 2(a) do not contain edges, so that they are decomposed into four lowpass and four highpass coefficients as usual. The first segment in the third row has length two. It is decomposed into one lowpass and one highpass coefficient. Consequently, the edge in the L and H images in Figure 2(b) occurs after the first pixel of row three. The first segment of the fifth row in Figure 2(a) has length three. This segment, which starts at an even position, is decomposed into two lowpass and one highpass coefficient. Thus, the edge in the L band in Figure 2(b) occurs after the second pixel, whereas it occurs in the H band after the first pixel. Now the second segment of row five has to be split into two lowpass and three highpass coefficients. This shows that, although both segments are of odd length, they have to be processed in different ways. Proceeding in this manner for the last three rows leads to the two sub-images shown in Figure 2(b). After decomposing the two sub-images in the vertical direction, one gets the four sub-images depicted in Figure 2(c) together with the corresponding edge information. The lowest band can now be further decomposed in the same way the original image in Figure 2(a) has been processed. Note that the edge information proceeds to the subbands in such a way that no gaps are introduced. Edges that completely

surround closed regions in the original will also be found as closed contours in the sub-images. In the example in Figure 2, two pixels occurred which deserve special attention. These are the pixels marked with a circle and with a square, which form horizontal segments of length one that cannot be decomposed into lowpass and highpass components. For the pixel marked with the circle, the processing is simple, because it naturally falls into the lowpass band (see Figure 2(b)). All we do here is to scale the pixel with the DC amplification of the analysis lowpass, so that a DC input signal will lead to the same DC value in the low-frequency band for any segment length. The pixel marked with the square has a lowpass characteristic, but it falls into the high-frequency band. This means, it has completely different statistical properties as the surrounding highpass coefficients in the right sub-image in Figure 2(b). If this pixel would be treated like a normal highpass coefficient during the next vertical filtering, its (potentially) high energy would smear out on several pixels after subband decomposition. In order to avoid this problem, the single pixel will be marked as a “single”, and the vertical filtering will be stopped in front of this pixel and will be continued behind the pixel. Note that the same problem occurred in the region-based coding scheme proposed in [15], and the name “single” has been adapted from there.

In the reconstruction stage we have to strictly follow the decomposition procedure backwards, which means that we have to perform the vertical synthesis prior to the horizontal one. The reason is that the edge-based wavelet decomposition is not separable. This can be seen from Figure 2(a) under the assumption that the vertical decomposition is carried out first. In this case, the two marked pixels would not cause any problem and would not lead to singles.

3 Decomposition of Finite-Length Signals

3.1 Initial Boundary Processing Method

Decomposition schemes that satisfy all the requirements formulated in Section 2 can be found for even and odd-length linear-phase filters, and also for non-symmetric filters [6, 8, 11]. For the reason of conciseness, we concentrate on linear-phase filters here.

A processing scheme for odd-length biorthogonal linear-phase filters is depicted in Figure 3. The upper rows show the extended input signal, where the given input samples are shown in solid boxes. The lowpass and highpass subband samples, c_n and d_n , respectively, are computed by taking the inner products of the impulse responses in the displayed positions with the corresponding part of the extended input signal. As we see, in all cases in Figure 3, only the desired number of lowpass and highpass coefficients occurs and has to be transmitted. Figure 4 shows a processing scheme for even-length linear-phase filters. As before, only the desired number of coefficients occurs.

A common description of all possible two-band decompositions is given by

$$\mathbf{y} = \mathbf{H} \mathbf{x} \quad (1)$$

The vector \mathbf{x} represents a length- N input signal, which can be considered as an arbitrary-length segment of a row or column of an image. \mathbf{y} contains the subband samples for the segment in increasing order (e.g. $\mathbf{y}^T = [c_0, d_0, c_1, d_1, \dots]$). The rows of the $N \times N$ matrix \mathbf{H} contain the time-shifted analysis impulse responses in reversed order. In the upper left and the lower right corner of the quadratic matrix \mathbf{H} , one finds the so-called boundary filters, which are needed for the processing of finite-length signals. Useful boundary filters for odd-length linear-phase filters can easily be designed by using the reflection scheme described above. For example, for the situations shown in Figures 3(a) and (b) the corresponding matrices \mathbf{H} are depicted in Figure 5.

3.2 Optimization of Boundary Filters

This section shows how the performance of boundary filters can be improved, regardless of their construction. In order to achieve a compact description, we first partition \mathbf{y} and \mathbf{H} such that

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix}, \quad \mathbf{y}_k = \mathbf{H}_k \mathbf{x}, \quad (2)$$

where \mathbf{y}_1 and \mathbf{y}_3 contain the subband samples produced by the boundary filters. Figure 5 illustrates the partitioning. In general, the partitions are taken in such a way that \mathbf{H}_2 , which

contains the regular impulse responses, has maximum size. The only exception occurs when a matrix \mathbf{H}_1 or \mathbf{H}_3 degenerates to a single row. In such a case the number of rows is chosen to be two. Notice that the actual number of boundary filters and thus the number of rows of \mathbf{H}_1 and \mathbf{H}_3 depends on the length of the filters in use.

The boundary filter manipulation may be written in the following matrix form, where the vectors \mathbf{v}_k contain the new subband samples:

$$\mathbf{v}_k = \mathbf{U}_k \mathbf{H}_k \mathbf{x}, \quad \mathbf{U}_2 = \mathbf{I}. \quad (3)$$

This means that, apart from the boundaries, the signal is processed as usual (indicated by the identity matrix). In the boundary regions, the impulse responses of the boundary filters are linearly combined (\mathbf{U}_1 and \mathbf{U}_3) in order to result in better boundary filters. Clearly, the inverse operation is required on the synthesis side.

Properties of Standard Boundary Filters. The boundary filters designed by the symmetric extension method have the important property that a DC input signal results in a DC signal at the output of the lowpass, while the highpass output is zero. This allows the efficient compression of very-low frequency signals without annoying artifacts at the image boundaries or at edge locations where the image is split according to the proposed method. However, continuity of the DC component is not the only important issue. For example, it turns out that the boundary analysis and synthesis filters usually have energies that differ from those of the original filters. This is easily seen by considering the left boundary in Figure 3. The first boundary lowpass has the impulse response $\{2B, 2A, C\}$ while the normal impulse response in steady state is $\{A, B, C, B, A\}$. The energies differ by $2A^2 + 2B^2$. Since the complete 2-D wavelet decomposition is carried out by cascading several 1-D decompositions, this effect can accumulate and result in iterated filters and associated wavelets with energies that may differ significantly from their nominal value. Both, the analysis and the synthesis bank are affected.

Large differences in the energies of filters belonging to the same subband are critical for the bit allocation, because quantization errors introduced in the subbands will propagate to the output with amplifications that are equal to the associated synthesis filter energy. Thus, white

quantization noise introduced in the subbands may occur at the output as extremely colored noise when some of the boundary synthesis filters have very high energies. For example, in a four-band decomposition using the 9-7 filters from [1], the noise amplification factor varies between 0.67 and 2.0, while the nominal value is one. The effect occurs in all frequency bands, and it becomes more dominant when more decomposition levels are used. In a three-level wavelet decomposition using the same filters, it ranges between 0.4 and 2.8. The effect occurs at all edges where the signal is split prior decomposition and also at the frame boundaries. Thus, it is also present in standard wavelet coders to some extent.

Constraints during Optimization. It is useful to preserve the approximation properties for size-limited DC signals throughout the optimization. Thus, U_1 and U_3 have to be restricted in a certain way. In order to derive suitable parameterizations of U_1 and U_3 , we consider the vector

$$\mathbf{b}_k = \mathbf{H}_k \mathbf{1}, \quad (4)$$

where $\mathbf{1}$ stands for a vector of appropriate size containing ones. Thus, \mathbf{b}_k contains the sum of the filter coefficients in \mathbf{H}_k . Note that due to the continuity of the reflection method, all lowpass filters sum up to the same constant, while all highpass coefficients sum up to zero. This important property shall be kept throughout the optimization. Thus, U_k has to satisfy

$$\mathbf{b}_k = U_k \mathbf{b}_k. \quad (5)$$

It is easily verified that U_k can be parameterized as

$$U_k = I_k + P_k Q_k, \quad (6)$$

where I_k is an identity matrix of appropriate size and Q_k contains the basis of the nullspace of \mathbf{b}_k^T , such that $Q_k \mathbf{b}_k = \mathbf{0}$. The matrix P_k contains the design parameters, which can be optimized in an unconstrained way. A generalization of this approach that allows to simultaneously satisfy certain moment conditions has been presented in [14]. However, the preservation of (5), as guaranteed by (6), turns out to be the most important issue in compression applications. Moreover, compared to a general matrix U_k as in [14], the parameterization (6) provides the

advantage that efficient implementations of \mathbf{U}_k can easily be found by choosing sparse matrices \mathbf{Q}_k and \mathbf{P}_k .

The Objective Function. For any \mathbf{P}_k the property (4) is satisfied and the elements of \mathbf{P}_k can be used freely as design parameters. The following points are of major importance in a 2-D shape-adaptive scheme and shall be incorporated in the objective function:

1. The energies of the analysis and synthesis boundary filters should be equal to the energies of the original filters which are used in the interior.
2. Boundary filters operating on adjacent rows (columns) which start or stop at different positions should be well aligned in the vertical (horizontal) direction.

An objective function which includes both requirements can be stated as follows:

$$C_k(\mathbf{P}_k) = \lambda_1 C_k^{(1)}(\mathbf{P}_k) + \lambda_2 C_k^{(2)}(\mathbf{P}_k) + \lambda_3 C_k^{(3)}(\mathbf{P}_k) \quad (7)$$

with

$$\begin{aligned} C_k^{(1)}(\mathbf{P}_k) &= \left\| \text{diag} \left\{ \mathbf{U}_k \mathbf{H}_k \mathbf{H}_k^T \mathbf{U}_k^T \right\} - \mathbf{1} \right\|^2 \\ C_k^{(2)}(\mathbf{P}_k) &= \left\| \text{diag} \left\{ (\mathbf{U}_k^{-1})^T \mathbf{G}_k^T \mathbf{G}_k \mathbf{U}_k^{-1} \right\} - \mathbf{1} \right\|^2 \\ C_k^{(3)}(\mathbf{P}_k) &= E \{ \|\mathbf{v}_k - \mathbf{w}_k\|^2 \} \end{aligned}$$

Herein, $\text{diag}\{\cdot\}$ is a vector containing the diagonal elements of a matrix. The first term refers to the energies of the analysis boundary filters, which should be close to one. The second term states the same requirement for the synthesis side (\mathbf{G}_k are the corresponding partitions of the synthesis matrix $\mathbf{G} = \mathbf{H}^{-1}$). The third term states that the output \mathbf{v}_k should be as close as possible to a desired output \mathbf{w}_k , where both \mathbf{v} and \mathbf{w} are considered to be generated from the same stochastic input process. In order to explain this part of the objective function in more detail, let us consider an infinite-length 1-D stochastic process \mathbf{x}_∞ . The vector of subband samples computed from \mathbf{x}_∞ can be written as $\mathbf{w}_\infty = \mathbf{F}\mathbf{x}_\infty$, where \mathbf{F} describes the analysis operation. By defining a matrix \mathbf{C}_w as a partition of an infinitely sized identity matrix, we can describe a length- N segment of \mathbf{w}_∞ as $\mathbf{w} = \mathbf{C}_w \mathbf{w}_\infty = \mathbf{C}_w \mathbf{F} \mathbf{x}_\infty$. On the other hand, we may write a length- N segment of \mathbf{x}_∞ as $\mathbf{x} = \mathbf{C}_x \mathbf{x}_\infty$. Based on \mathbf{x} we can then compute the length- N

subband vector $\mathbf{v} = \mathbf{U}\mathbf{H}\mathbf{x}$. By choosing \mathbf{U} in such a way that \mathbf{v} becomes as close as possible to \mathbf{w} for a typical input process, \mathbf{x}_∞ , the alignment of the boundary filters can be improved. Partitioning \mathbf{w} into \mathbf{w}_1 , \mathbf{w}_2 , and \mathbf{w}_3 and requiring $E\{\|\mathbf{v}_k - \mathbf{w}_k\|^2\} \stackrel{!}{=} \min$ yields the third term of the objective function. In the experiments discussed below, the process \mathbf{x}_∞ was chosen as an AR(1) process with correlation coefficient 0.95.

The pre-factors λ_1 , λ_2 , and λ_3 are introduced in order to allow weighting of the different criteria. For $\lambda_1 \rightarrow 0$ all synthesis filters will have an energy close to one, but some of the analysis filters may have extremely high energies. Accordingly, for $\lambda_2 \rightarrow 0$ the analysis filters will almost perfectly satisfy the requirements, but the behavior of the synthesis side becomes uncontrolled. Finally, for $\lambda_3 = 0$ the filters may or may not be well aligned. The values $\lambda_1 = 0.75$, $\lambda_2 = 1$, and $\lambda_3 = 2$ turned out to be a good compromise between the different criteria. The bias $\lambda_1/\lambda_2 = 3/4$ has been introduced in order to make convergence to local minima with $C_k^{(1)}(\mathbf{P}_k) \ll C_k^{(2)}(\mathbf{P}_k)$ more difficult. The overall result depends on the starting point and the minimum found.

For extremely short segments, the boundary filters for the left and right boundary merge and \mathbf{H}_2 vanishes. In these cases, we replace the matrices \mathbf{U}_1 and \mathbf{U}_3 by a common matrix \mathbf{U} , such that $\mathbf{v} = \mathbf{U}\mathbf{H}\mathbf{x}$, where $\mathbf{U} = \mathbf{I} + \mathbf{P}\mathbf{Q}$. Then, for each segment length and starting position (even or odd), a different matrix $\mathbf{U}\mathbf{H}$ will be implemented. The optimization of the matrices \mathbf{U} is carried out in the same way as the optimization of \mathbf{U}_k .

Optimization Results. Table 1 shows the value of the objective function (7) for various cases of interest. Segments of length one and two are not included in the table, because they require a fixed solution that does not offer further design freedom (apart from scaling the highpass). As we can see in Table 1, optimization allows significant improvement for both odd and even-length filters. For the 9-7 filters, the only case where an undesirable behavior occurs is for segments of length three, starting at odd positions. The 6-10 filters generally have some difficulties in cases where a segment starts at an odd position. For both even and odd-length filters, the noise amplification factor varies between 0.98 and 1.02 after optimization, provided that the segment length is larger than three. Thus, it can be regarded as being independent of the location. However, if more decomposition levels are used, the factors may slightly vary from their

nominal value, because in the biorthogonal case, the energy of an upsampled and interpolated filter is not necessarily equal to the energy of the filter itself. That is, $\sum_n g_0^2(n) \neq \sum_n f^2(n)$, where $f(2n) = \sum_m g_0(n-m)g_0(2m)$, $f(2n+1) = \sum_m g_0(n-m)g_0(2m+1)$.

In all cases tested, the resulting matrices \mathbf{U}_k turned out to be well-behaved. This is intuitively clear, because they transform well-behaved matrices \mathbf{H}_k into well-behaved matrices $\mathbf{U}_k \mathbf{H}_k$ with almost-orthonormal rows.

4 Implementation Details

In the experiments discussed in the next sections, the edge detection has been carried out using the Sobel edge detector [16]. In order to reduce the number of gaps within detected edges, the images were pre-smoothed using a lowpass filter with impulse response $(z + 2 + z^{-1})^4/256$ in both directions. For input data in the range $[0, 255]$ the sensitivity threshold for accepting edges was fixed at 15. Thus, independent of the image content, only the dominant edges were detected. In a further edge processing step, only edges consisting of more than eight elements were considered as important ones, and the others were removed from the edge map. The compression of the edge information was carried out with chain coding [17] followed by arithmetic coding.

Edge detectors typically search for maxima of local gradients (e.g. the Sobel operator), or they look for zero crossings of second derivatives (e.g. the Marr-Hildreth detector [18], [17]). One useful property of the Marr-Hildreth detector is that edges are easily defined between pixels, as required for the edge-based transform. However, also edge detectors that look for maxima of local gradients and produce binary edge maps may be used. Then, in a first step, the binary edge information has to be converted into edge maps where edges are defined between pixels and still form closed curves.

The filters used in the experiments are the 9-7 filters from [1] and the 6-10 filters from [19], which are known for their good coding properties [19, 20]. The lossy compression of wavelet coefficients was based on the embedded wavelet coder proposed in [21] whose performance is

comparable to the embedded zerotree coder in [22].

The edge-based DWT requires additional complexity for edge detection and the implementation of optimal boundary processing schemes. The major part of these additional operations is needed for edge detection. It amounts to 70% of the complexity of a standard DWT. The number of additional operations for optimal boundary processing depends on the number of edges. For the cameraman image it amounts to 20% of the complexity of a standard DWT. Thus, the edge-based DWT itself, as required in the receiver in form of an inverse edge-based DWT, means no significant increase of the number of operations.

5 Objective Quality Measures

In order to compare the coding performance of the edge-based and the standard DWT, we consider two different objective quality measures: (a) the PSNR and (b) the picture quality scale (PQS) [23]. The PQS is designed to approximate the mean opinion score in visual tests. It reflects the psychovisual impression of human observers much better than the PSNR. A PQS of 5 means that the distortion is imperceptible, while for a decreasing PQS the distortion becomes increasingly annoying.

Table 2 shows the rate-distortion tradeoff for the “camera man” image, using the edge map in Figure 6(b). The abbreviations in Table 2 have the following meaning: “standard”= standard DWT; “edge, init.” = edge based DWT without optimization; “edge, opt.” = edge based DWT with optimization. As the results show, the PSNR for the standard DWT is slightly better than for the edge-based one. This is due to the fact that the latter requires a relatively large amount of side information for the transmission of the edge map (0.052 bpp for this image). If we do not take this overhead into account, the edge-based coder gives the better PSNR. The results further show that the initial boundary processing scheme for odd-length filters performs almost optimally, while the one for even-length filters performs relatively poorly. When using optimized boundary processing schemes, the results for even and odd-length filters are of equal quality.

As for the second quality measure, Table 3 shows the PQS versus the bit rate for the same test image. According to this measure, the edge-based transform gives superior performance. This is also reflected by the impression of human observers that were asked to comment on the quality of the coders. Odd-length filters yield slightly better results than the even-length ones.

6 Coding Examples

Figure 6 shows coding results for the camera-man image. It turns out that the standard DWT leads to distinct ringing artifacts at sharp edges while the edge-based method is able to reproduce extremely sharp edges, even at very low rates. Since edge-based coding requires the transmission of the edge map, less bits can be allocated for texture, and we have a less detailed reproduction of texture. This property is clearly visible in Figure 6(f), where less bits are used for texture than for edge coding.

A possible strategy to reduce the disproportion in the bit allocation at extremely low rates is to use a standard DWT for the first stage of signal decomposition and to define the edge map on a lower resolution level. However, one cannot gain very much from such an approach, because the smaller edge map will typically contain much less redundancy than the full-resolution map, and therefore it cannot be compressed very efficiently. Moreover, at extremely low rates, a low-resolution edge map will cause step-like edges in the reconstructed image.

Figure 7 shows a second coding example. The motivation for this example is the fact that many images are composed from real-world contents and text (e.g. internet applications). In Figure 7(a) we see the original Lena image, but with text added. Figure 7(b) shows the edge information. For detecting the edges around the letters, the sensitivity threshold was lowered to 5 in this particular image region (assuming that the text location is known). This image (without text) is known to be easily compressed with the DWT. Now, because of the text, extreme ringing artifacts occur. With the edge-based method, only very few artifacts are visible and we have a much better overall performance. Even the PSNR is slightly better for the edge-based method.

7 Conclusion

In this paper, image coding via an edge-based wavelet transform has been presented. It turned out that this coding technique allows a significant reduction of the ringing effect, which usually occurs in the vicinity of sharp edges. Since the technique requires the transmission of the edge map as side information, less bits can be assigned for texture coding, and the reduction of ringing is not necessarily accompanied by an improvement of the PSNR. However, the picture quality scale, which better reflects the impression of human observers, shows superior results for the proposed method. Moreover, the coding technique is well-suited for database applications where the edge map, which forms an integral part of the compressed bit stream, can be utilized for sketch based image retrieval and browsing.

References

- [1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Trans. Image Processing*, vol. 1, pp. 205–220, April 1992.
- [2] S. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 11, pp. 674–693, July 1989.
- [3] K. Cinkler and A. Mertins, “Edge sensitive subband coding of images,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, USA, pp. 2365–2368, May 1996.
- [4] J. Woods and S. O’Neil, “Subband coding of images,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 1278–1288, May 1986.
- [5] M.J.T. Smith and S.L. Eddins, “Analysis/synthesis techniques for subband coding,” *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 1446–1456, August 1990.
- [6] H.J. Barnard, J.H. Weber, and J. Biemond, “Efficient signal extension for subband/wavelet decomposition of arbitrary length signals,” in *Proc. SPIE, VCIP*, vol. 2094, pp. 966–975, November 1993.

- [7] L. Chen, T. Nguyen, and K.P. Chan, "Symmetric extension methods for M-channel PR LP FIR analysis/synthesis systems," in *Proc. IEEE Int. Symp. Circuits and Systems*, London, pp. 2.273 – 2.276, June 1994.
- [8] C. Herley, "Boundary filters for finite-length signals and time-varying filter banks," *IEEE Trans. Circuits and Systems II*, vol. 42, pp. 102–114, February 1995.
- [9] A. Mertins, "Time-varying and support preservative filter banks: Design of optimal transition and boundary filters via SVD," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Detroit, pp. 1316–1319, May 1995.
- [10] V. Nuri and R.H. Bamberger, "Size limited filter banks for subband image compression," *IEEE Trans. Image Processing*, vol. 4, pp. 1317–1323, September 1995.
- [11] M. Coffrey, "Boundary compensated wavelet bases," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, München, pp. 2129–2132, May 1997.
- [12] R.L. de Queiroz, "Subband processing of finite length signals without border distortions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, USA, vol. IV, pp. 613 – 616, March 1992.
- [13] R.L. de Queiroz and K.R. Rao, "Optimal orthogonal boundary filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Detroit, USA, vol. II, pp. 1296 – 1299, May 1995.
- [14] A. Mertins, "Optimized biorthogonal shape adaptive wavelets," in *Proc. ICIP'98*, Chicago, October 1998.
- [15] H.J. Barnard, J.H. Weber, and J. Biemond, A Region-Based Discrete Wavelet Transform for Image Coding. In *Advances in Image Communication: Wavelets in Image Communication*, M. Barlaud (ed.), vol. 5, pp. 189–227. Amsterdam: Elsevier, 1994.
- [16] R.C. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1977.

- [17] S. Carlsson, "Sketch based coding of grey level images," *Signal Processing*, vol. 15, no. 1, pp. 57–83, 1988.
- [18] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc., London, Ser. B*, vol. 207, pp. 197–217, 1980.
- [19] J.D. Villasenor, B. Belzer, and J. Liao, "Wavelet filter evaluation for image compression," *IEEE Trans. Image Processing*, vol. 4, pp. 1053–1060, August 1995.
- [20] E.A.B. da Silva and M. Ghanbari, "On the performance of linear phase wavelet transforms in low bit-rate image coding," *IEEE Trans. Image Processing*, vol. 5, pp. 689–704, May 1996.
- [21] R.W. Buccigrossi and E.P. Simoncelli, "Progressive wavelet image coding based on a conditional probability model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, München, Germany, vol. 4, pp. 2957–2960, April 1997.
- [22] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, December 1993.
- [23] M. Miyahara, K. Kotani, and V.R. Algazi, "Objective picture quality scale (PQS) for image coding," *submitted to IEEE Trans. on Communications*, 1996.

List of Tables

Table 1

Objective function (7) for various cases (initial value prior optimization and final value; 9-7 filters from [1] and 6-10 filters from [19]).

Case of interest			9-7 filters		6-10 filters	
Start position	Stop position	Segment length	initial	final	initial	final
even	-	-	0.58	0.05	0.24	0.06
odd	-	-	0.85	0.13	3.55	0.59
-	even	-	0.85	0.13	0.24	0.06
-	odd	-	0.58	0.05	1.38	0.15
even	even	3	0.99	0.08	1.42	0.14
odd	odd	3	0.79	0.63	3.69	1.39
even	odd	4	0.91	0.04	0.35	0.10
odd	even	4	0.91	0.04	3.63	0.92
even	even	5	0.77	0.04	1.42	0.15
odd	odd	5	1.13	0.13	3.58	0.64
even	odd	6	0.95	0.05	0.35	0.08
odd	even	6	0.95	0.05	3.93	0.64
even	even	7	0.77	0.04	1.42	0.15
odd	odd	7	1.13	0.13	3.58	0.56

Table 2

PSNR versus bit rate for the cameraman image.

Rate [bpp]	0.1	0.2	0.3	0.4	0.6	0.8	1.0
6-10 standard	23.28	25.60	27.27	28.48	30.62	32.46	34.10
6-10 edge, init.	22.01	24.52	26.04	27.58	29.55	31.51	32.94
6-10 edge, opt.	22.63	25.00	26.82	28.11	30.42	32.10	33.81
9-7 standard	23.29	25.58	27.21	28.49	30.54	32.47	34.22
9-7 edge, init.	22.35	24.86	26.71	27.78	30.21	32.00	33.80
9-7 edge, opt.	22.39	24.89	26.78	28.02	30.35	32.22	33.94

Table 3

Picture quality scale versus bit rate for the cameraman image.

Rate [bpp]	0.1	0.2	0.3	0.4	0.6	0.8	1.0	2.0	4.0
6-10 standard	-3.28	-2.10	-0.94	-0.37	0.72	1.32	1.93	3.60	4.74
6-10 edge, init.	-2.68	-1.56	-0.86	-0.25	0.66	1.30	1.78	3.44	4.65
6-10 edge, opt.	-2.05	-1.20	-0.60	0.05	0.78	1.47	2.04	3.63	4.71
9-7 standard	-3.10	-1.41	-0.70	-0.13	0.76	1.60	1.91	3.63	4.74
9-7 edge, init.	-1.96	-1.07	-0.32	0.23	0.92	1.58	2.06	3.64	4.72
9-7 edge, opt.	-1.88	-0.74	-0.25	0.28	0.99	1.65	2.13	3.64	4.73

List of Figures

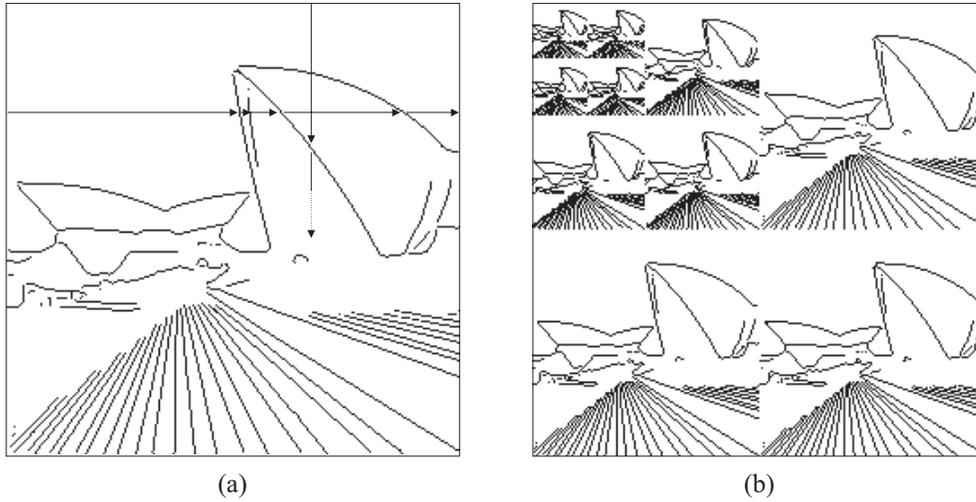


Figure 1: Edge map on different resolution levels. (a) original edge map and processing intervals; (b) decomposed map.

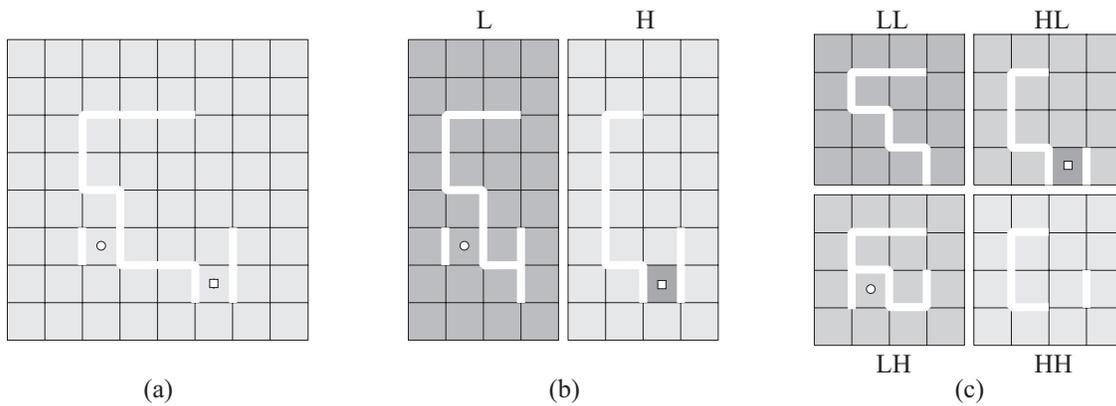
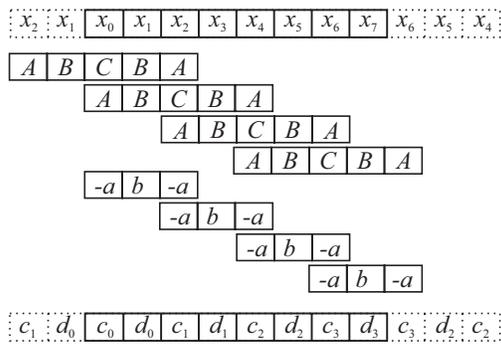
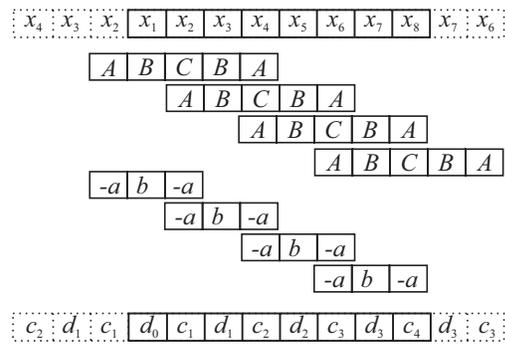


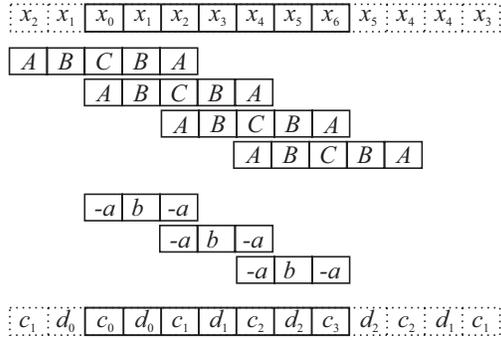
Figure 2: Example for the decomposition of an image and the edge information into four sub-bands. The edges are shown in white. (a) original image with 8×8 pixels and edge information; (b) lowpass and highpass band after horizontal decomposition; (c) four subbands after vertical decomposition.



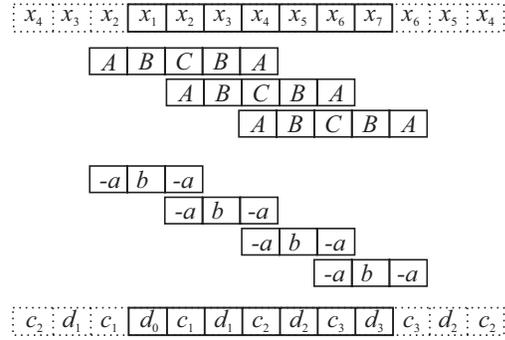
(a)



(b)



(c)



(d)

Figure 3: Symmetric reflection using odd-length symmetric filters with impulse responses $\{A, B, C, B, A\}$ for the lowpass and $\{-a, b, -a\}$ for the highpass; (a) even-length segment starting at an even position; (b) even-length segment starting at an odd position; (c) odd-length segment starting at an even position; (d) odd-length segment starting at an odd position.

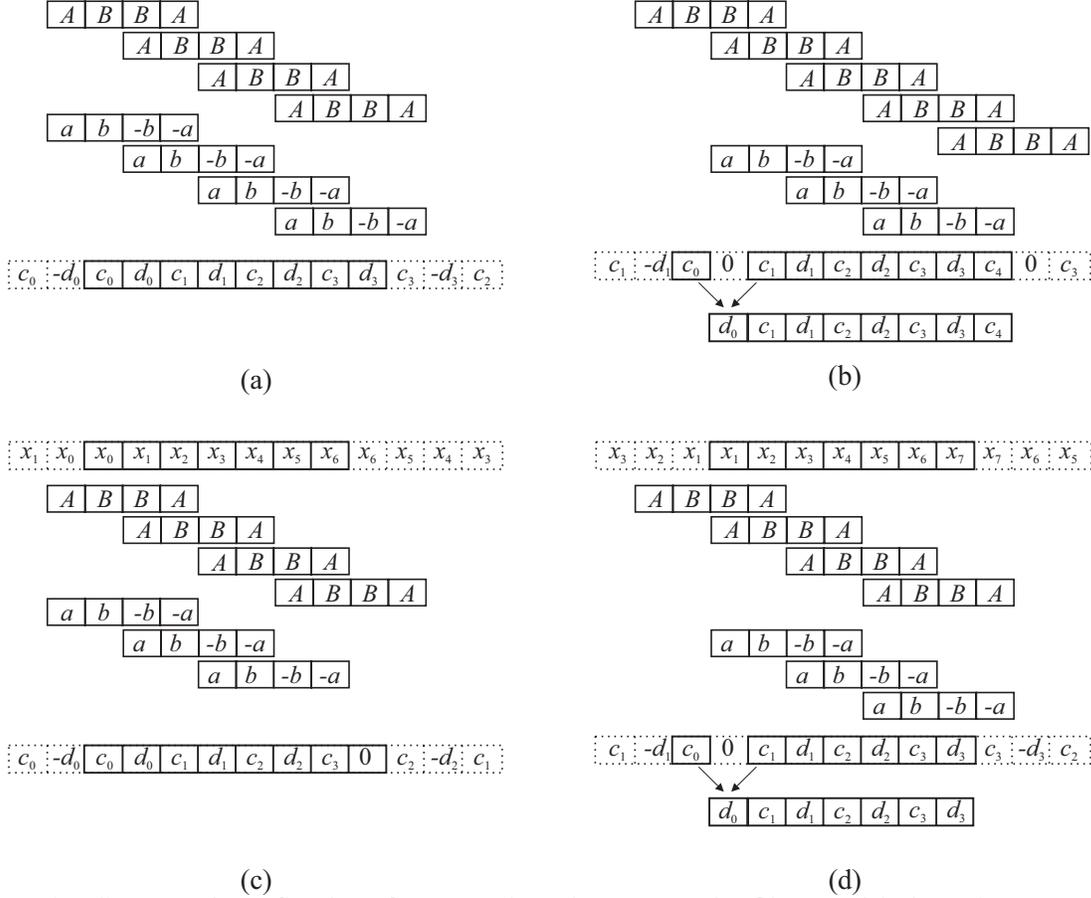


Figure 4: Symmetric reflection for even-length symmetric filters with impulse responses $\{A, B, B, A\}$ for the lowpass and $\{-a, -b, b, a\}$ for the highpass; (a) even-length segment starting at an even position; (b) even-length segment starting at an odd position. (c) odd-length segment starting at an even position; (d) odd-length segment starting at an odd position.

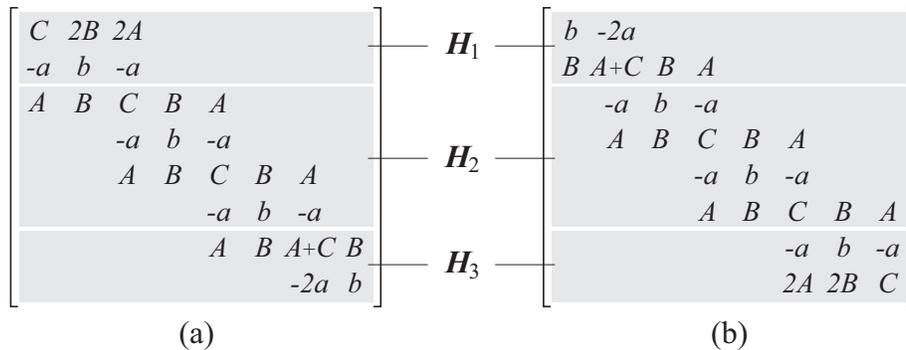
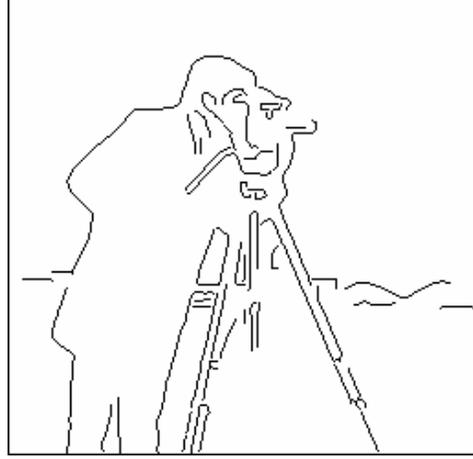


Figure 5: Matrix H for the examples in Figures 3(a) and (b).



(a)



(b)



(c)



(d)



(e)



(f)

Figure 6: Coding results for the 256×256 camera man image. The DWT is based on the 9-7 filters. (a) original; (b) edge map, coded at 0.052 bpp; (c) standard DWT at 0.4 bpp; (d) edge-based DWT at 0.4 bpp (0.348 bpp for texture); (e) standard DWT at 0.1 bpp; (f) edge-based DWT at 0.1 bpp (0.048 bpp for texture).



(a)



(b)



(c)



(d)



(e)



(f)

Figure 7: Coding results for Lena with text. The DWT is based on the 9-7 filters. (a) original; (b) edge map, coded at 0.026 bpp; (c) and (e) standard DWT at 0.2 bpp, PSNR = 28.8 dB; (d) and (f) edge-based DWT at 0.2 bpp (0.174 bpp for texture), PSNR = 28.9 dB.