EMBEDDED ZERO–TREE CODING OF IMAGES EMPLOYING SOFT–THRESHOLDING

Frank Müller and Klaus Illgner

Institut für Elektrische Nachrichtentechnik
Aachen University of Technology (RWTH), 52056 Aachen, Germany
Tel.: +49-241-807681, Fax.: +49-241-8888196
e-mail: {mueller,illgner}@ient.rwth-aachen.de

ABSTRACT

Improvements of embedded zero-tree wavelet (EZW) coding by employment of soft-thresholding in the wavelet domain are reported. By proper adjustment of the thresholds, the attainable PSNR can be slightly improved (around 0.3 dB). Moreover, soft-thresholding can improve the visual appearance of the coded images, especially in the very low data rate case. The quality in the early stages of progressive image transmission can be improved by reduction of annoying artifacts resulting from coarse quantization of the wavelet coefficients. Adaptation of the thresholds to the "current" width of the quantization intervals preserves the embedded bit stream property of the coder.

1 INTRODUCTION

Lossy coding of images and image sequences is nowadays an important task. Especially multimedia applications strongly rely on the existence of efficient coding algorithms. Rate capacity as well as desired quality and image resolution may vary from application to application and from user to user. Therefore, coding methods which allow reconstructions (of lower quality) from predefined subsets of the bit stream gained much attention in the recent past. Such coders are termed "rate distortion scalable". If the coder sends information in a way, such that the decoder may truncate the bit stream at *any* point and still can utilize all of the data which has been arrived until this time to reconstruct the image, the coding method is termed "embedded".

Embedded coding is very attractive in a variety of applications. E.g. for browsing in digital image libraries, the user can judge from coarse approximations if the image content accords to what is searched for. If the user is satisfied with a coarser approximation than the full resolution image, he may stop transmission at the desired point. For update coding in hybrid video coding, the coder control can stop coding when a given bit budget is exhausted in order to keep the frame rate [1] .

One of the most efficient embedded image coding algorithms has been developed by Shapiro [2], and is called embedded zerotree wavelet (EZW) coding algorithm. It consists of a transformation of the image into a wavelet domain and subsequent embedded coding with the zerotree algorithm, which starts with coarse quantizations and refines the wavelet coefficients in subsequent passes.

Soft-thresholding in the wavelet domain has been proposed for noise removal by several authors. Recently, Donoho [3] gave a justification of these methods using estimation theory. He showed, that with proper choice of the thresholds, noise induced artifacts as "ripples" and oscillations can be removed with high probability without loosing too much in the mean squared error sense.

By observing, that coding in the wavelet domain introduces quantization errors (which can be regarded as noise), we expected improvements of the EZW coder by soft-thresholding. A close examination of [3] justifies this even for deterministic "noise" as e.g. quantization errors. Another observation is, that with adaptation of the thresholds to the "current" width of the quantization intervals the embedded bit stream property is preserved.

2 EZW CODING

EZW coding, introduced by Shapiro [2] consists of

1. an octave band "wavelet" decomposition,
2. layered quantization of the wavelet coefficients,
3. efficient zero coding with zerotrees at each layer,
4. adaptive arithmetic coding.

In this section the embedded zerotree wavelet (EZW) coding is reviewed as far as necessary for understanding the connection to wavelet thresholding. For this purpose it is useful to confine on the used quantization intervals and reconstruction levels. For a detailed description of the wavelet decomposition, the prediction of insignificant coefficients across scales and the lossless coding part employing zerotrees and adaptive arithmetic coding refer to the original paper of Shapiro [2].

2.1 Layered quantization

Before quantization the wavelet coefficients are scanned in a predefined order, resulting in a sequence of wavelet coefficients. SNR scalability is achieved by progressive quantization in a sequence of up to N layers.

Each layer can be described by the associated quantizer. It is convenient, to differentiate between two sets of quantizers $\mathcal{Q}_0^D, \ldots, \mathcal{Q}_{N-1}^D$ and $\mathcal{Q}_1^S, \ldots, \mathcal{Q}_{N-1}^S$. The first set of quantizers belongs to the *dominant passes* and the second to the *subordinate passes* of the coding algorithm. These passes alternate, starting with a dominant pass. Hence during the coding process the sequence $\mathcal{Q}_0^D, \mathcal{Q}_1^S, \mathcal{Q}_1^D, \mathcal{Q}_2^S, \mathcal{Q}_2^D, \ldots$ of quantizers is employed.

Each quantizer $\mathcal{Q}_n$ is defined by a set of disjoint quantization intervals $\mathcal{I}_{n,k_1}, \mathcal{I}_{n,k_2}, \ldots$ and the quantization function

$$\mathcal{Q}_n^D(x) = k, \text{ for } x \in \mathcal{I}_{n,k}^D \tag{1}$$

$$\mathcal{Q}_n^S(x) = k, \text{ for } x \in \mathcal{I}_{n,k}^S \tag{2}$$

which maps each input value x to an index k.

In order to allow efficient layered quantization, the sequence of quantization intervals must form a set of nested intervals.

The "dominant" intervals $\mathcal{I}_{n,k}^D$ used in EZW coding are symmetric around zero, uniformly spaced with a dead zone twice as large as all the other quantization intervals. From layer to layer each quantization interval is halved. Thus the intervals are specified with a single parameter Δ_0 as follows:

$$\mathcal{I}_{n,k}^D = \begin{cases} (-\Delta_n, \Delta_n) & \text{if } k = 0 \\ [k\Delta_n, (k+1)\Delta_n) & \text{if } k > 0 \\ ((k-1)\Delta_n, k\Delta_n] & \text{if } k < 0 \end{cases} \tag{3}$$

with

$$\Delta_n = \Delta_{n-1}/2 \quad \text{for } n > 0. \tag{4}$$

The "subordinate" intervals $\mathcal{I}_{n,k}^S$ are refinements of the intervals $\mathcal{I}_{n-1,k}^D$. More precisely, each interval $\mathcal{I}_{n-1,k}^D$ contains two intervals $\mathcal{I}_{n,k}^S$ of equal width, except for $k = 0$, where $\mathcal{I}_{n-1,0}^D = \mathcal{I}_{n,0}^S$. Hence, the "subordinate" intervals are symmetric around zero, uniformly spaced with a dead zone four times larger than the other intervals:

$$\mathcal{I}_{n,k}^S = \begin{cases} \mathcal{I}_{n-1,0}^D & \text{if } k = 0 \\ \emptyset & \text{if } k \in \{-1, 1\} \\ \mathcal{I}_{n,k}^D & \text{if } k \notin \{-1, 0, 1\} \end{cases} \tag{5}$$

Note, that each quantization interval $\mathcal{Q}_n^D$ is contained in some quantization interval of $\mathcal{Q}_{n-1}^D$. Moreover, the intervals $\mathcal{I}_{n,0}^D$ contain the *three* intervals $\mathcal{I}_{n+1,-1}^D, \mathcal{I}_{n+1,0}^D, \mathcal{I}_{n+1,1}^D$, whereas all other intervals $\mathcal{I}_{n,k}^D$ (with $k \neq 0$) contain *two* intervals $\mathcal{I}_{n+1,2k}^D, \mathcal{I}_{n+1,2k+1}^D$. (Similar properties exist for the "subordinate" intervals, since these are defined through (5) via the dominant intervals.) This is the only condition on the quantization intervals which must be imposed. However, specifying the intervals as in (3) is very convenient, because the set of quantization functions is defined by Δ_0 and N alone.

2.2 Layered zero coding using zerotrees

A wavelet coefficient c_{jk} is termed *insignificant in the n-th layer* if $\mathcal{Q}_n^D(c_{jk}) = 0$, and termed *significant* if $\mathcal{Q}_n^D(c_{jk}) \neq 0$. A *significance map* $\sigma_n(i,j)$ is defined for each layer n indicating the significance of the coefficients:

$$\sigma_n(i,j) = \begin{cases} 1 & \text{if } \mathcal{Q}_n^D(c_{jk}) \neq 0 \quad \text{(significant)} \\ 0 & \text{if } \mathcal{Q}_n^D(c_{jk}) = 0 \quad \text{(not significant)} \end{cases} \tag{6}$$

A tree structure imposed on the wavelet domain allows for efficient coding of the positions of insignificant coefficients at the different layers. Any subtree consisting only of insignificant values is encoded by means of the root of this tree (also called zerotree root). This method can be viewed as prediction of insignificance across scales or as exploitation of selfsimilarity of the significance maps.

However, the most important aspect for this work which is concerned with joint coding and denoising is that the coding algorithm can gain from insignificant coefficients (especially when the positions of these can be predicted well). On the other side, isolated significant values can turn out "expensive" in terms of rate (especially when their appearance disturbs the prediction performance).

2.3 Dequantization at the decoder

Until now the EZW encoder has been described. The decoder receives an arithmetically encoded bitstream, decodes it into a symbol stream and reconstructs from this symbol stream successively refined quantization intervals. For each coefficient a value of the current interval must be chosen as reconstruction value (dequantization). Then the inverse wavelet transform yields the decoded image.

There are several possibilities for designing the dequantization function. In Shapiro's original work the reconstruction values are chosen in the middle of the current interval. Other dequantizers can be employed as well and improve average performance, because the actual values (before quantization) are not equally distributed within all quantization intervals.

3 DENOISING BY THRESHOLDING

Noise removal by thresholding in a transform domain is nowadays a standard technique. It works in three steps:

1. Apply an orthogonal transform to the noisy image.
2. Threshold the transform coefficients.
3. Transform back in the original domain.

Any orthogonal transform is allowed. However, depending on the transform the method may introduce annoying ringing artifacts. Especially Fourier based denoising suffers from the Gibbs-phenomenon and DCT based denoising from discontinuities at the block boundaries. If the transform is into the wavelet domain, the method is also termed *wavelet shrinkage* and has been discussed in detail in [4].

For the second step there are two options: the thresholding may applied "hard" or "soft". Hard–thresholding of a set of transform coefficients c_{jk} with a positive threshold λ means

$$c^*_{jk} = \begin{cases} c_{jk} & \text{if } |c_{jk}| \geq \lambda \\ 0 & \text{if } |c_{jk}| < \lambda. \end{cases} \tag{7}$$

Soft–thresholding with threshold λ means application of the nonlinearity

$$c^*_{jk} = \text{sign}(c_{jk})(|c_{jk}| - \lambda)_+, \tag{8}$$

where $(\cdot)_+$ means "set the value to zero, if negative".

Finally, a proper value for the threshold λ has to be chosen.

Hard–thresholding is a straight forward approach for denoising. It resembles the idea of statistical hypothesis testing. The underlying assumptions are: (i) only some coefficients carry information about the original (noiseless) image and (ii) the noise is identically independently distributed (iid) in the wavelet domain. Then comparison with the threshold λ tests the hypothesis if a particular wavelet coefficient carries information about the original image. Hard–thresholding then sets every coefficient to zero which is assumed to belong to noise only. All other coefficients (containing signal *and* noise information) remain unchanged.

Soft-thresholding does exactly the same with the coefficients below the threshold, but differs in the way it deals with the remaining coefficients. These stay not unchanged, their magnitude is decreased by the value of the threshold. A discussion justifying this operation can be found in Donoho's work, in which statistical properties of this estimator are derived (e.g. [3] [4]). One fortunate aspect is that in contrast to the hard-thresholding case the input space and the output space are identical.

4 JOINT THRESHOLDING AND CODING

Denoising by wavelet shrinkage can be combined with EZW coding in several ways. In case of an noisy input image it is beneficial to denoise the image prior to coding. This improves coding efficiency, because the rate otherwise spent for encoding the noise can be utilized for coding of the image contents. This approach will be described in the next subsection. However, even in the case, that the input image is regarded as noise free, the encoding process can benefit from wavelet shrinkage by attenuating quantization noise. Because the quantization noise decreases with increasing quantization layers, the thresholds must be adapted to preserve the embedded bit stream property. An efficient implementation of this adaption which affects the dequantization only is given in the second subsection.

4.1 Thresholding as preprocessing

Wavelet shrinkage prior to EZW coding means performing the steps (1) forward transform, (2) threshold, (3) backward transform, (4) forward transform, (5) zerotree coding, (6) zerotree decoding, (7) backward transform. Here steps (1-3) belong to wavelet shrinkage, (4-5) to EZW encoding, and (6-7) to EZW decoding. By using the same wavelet decomposition for denoising and coding steps (3) and (4) can be dropped. Hence, denoising as preprocessing can be incorporated into EZW coding as thresholding prior to the layered quantization.

Hard thresholding with a constant threshold λ has no impact on coding performance in "early" layers. This is due to the fact, that the EZW coder already employs (implicitly) hard thresholding of the coefficients. As long as $\lambda < \Delta_n$, the code is not affected up to layer Q^S_{n+1}. To some extent this explains why the EZW coder shows good performance even at low rates.

The situation changes, when soft thresholding is employed. Soft thresholding affects all coefficients, hence the performance differs at all bit rates. We found out, that soft thresholding prior to quantization can improve the average coding performance. However, the rate-distortion curves of the compared methods (with and without thresholding) were intersecting each other and at high rates the thresholding lead to oversmoothing of fine details. Therefore, we recommend this method only in case of noisy images (when the denoised images look "nicer" than the originals) or in case of predetermined bitrates. If one wants to preserve the embedded bitstream property, the adaptive method below is recommended.

4.2 Adaptive thresholding

Shapiro's EZW coder works with varying quantization layers in the wavelet domain as explained above. Since the resolution of each transmitted wavelet coefficient is refined over time, the quantization "noise" decreases with increasing length of the bitstream. Naturally, the threshold should decrease adaptively with the quantization noise.

Choosing the threshold λ depending on the current worst case error is theoretically sound (see [3]) and can be implemented without affecting the encoder. If the dequantizer chooses always the boundary point of the interval which is closest to zero, this is equivalent to the original EZW coding scheme with subsequent soft thresholding. In this case the threshold λ equals half the width of the "current" quantization interval. By delegating the denoising to the decoder the embedded bitstream property is preserved.

Figure 1: Frame #01 of CIF-sequence *Miss America,* coded with 500 bit. Top: Image obtained with the original EZW algorithm. Down: Image obtained with the proposed algorithm.

Figure 2: Frame #01 of CIF-sequence *Miss America,* coded with 1000 bit. Top: Image obtained with the original EZW algorithm. Down: Image obtained with the proposed algorithm.

5 SIMULATION RESULTS

A comparison of Shapiro's original algorithm and results obtained by soft-thresholding is given in the figures for very low bit rates. Note, that 1000 bit for one frame (CIF) corresponds to 0.01 bit per pixel or a compression ratio of 1:811. The improvements are especially visible in figure 2, where the original algorithm introduced artifacts in the face region, but also in the 500 bit case (figure 1). The shoulders can be guessed here from the thresholded coded image. Note, that incorporation of this coding scheme into a hybrid video coder can gain much from the artifact removal, because the artifacts disturb motion estimation.

References

[1] F. Müller and K. Illgner. Embedded pyramid coding of displaced frame differences. *IEEE International Conference on Image Processing, vol. III,* pp. 216-219, Washington D.C., USA, Nov. 1995.

[2] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Processing,* 41(12):3445–3462, 1993.

[3] D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Information Theory,* 41(3):613–627, 1995.

[4] D. L. Donoho et al. Wavelet shrinkage: Asymptopia. *J. Roy. Statist. Soc.,* B57(2):301–369, 1995.