Rate-Distortion Optimized Embedding

Jin Li and Shawmin Lei
Sharp Laboratories of America, Digital Video Division
5750 NW Pacific Rim Blvd., Camas, WA 98607, USA.
Tel. (360) 817-8486, Fax. (360) 817-8436, Email: lijin@sharplabs.com

ABSTRACT: *In this paper, we improve the performance of the embedded coder by reorganising its output bitstream in the rate-distortion (R-D) sense. In the proposed rate-distortion optimized embedding (RDE), the coding bit is first allocated to the coefficient with the steepest R-D slope, i.e. the biggest distortion decrease per coding bit. To avoid transmission of the position of the coded coefficient, RDE uses the expectation R-D slope that can be calculated by the coded bits that have already been transmitted to the decoder. RDE also takes advantage of the probability estimation table of the QM-coder so that the calculation of the R-D slope is just a lookup table operation. Extensive experimental results show that RDE significantly improves the coding efficiency.*

1. INTRODUCTION

The embedded image coding receives great attention recently. The representative works of embedding include the embedded zerotree wavelet coding (EZW) proposed by Shapiro [1] , the set partitioning in hierarchical trees (SPIHT) proposed by Said and Pearlman [2] , and the layered zero coding (LZC) proposed by Taubman and Zakhor [3] . In addition to providing a very good rate-distortion (R-D) performance, embedded coder has a desirable property that the bitstream can be truncated at any point and still decode reasonable quality image. Such embedding property is ideal for a number of applications such as progressive image transmission, rate control, scalable coding, etc.. EZW and its derivatives achieve embedding by organizing the quantization bit-plane by bit-plane and entropy encoding the significant status or refinement bit. However, there is no guarantee that the rate-distortion performance was optimized at the truncation point.

It is well known that the coding achieves optimality if the rate-distortion slopes for all coded coefficients are constant. Xiong and Ramchandran [4] fixed the rate-distortion slope, and applied tree-pruning to spatial frequency quantization to exactly encode each coefficient with the same rate-distortion slope. Although achieving good coding efficiency, the scheme was very complex, and it lost the bitstream embedding property which was very useful in many applications. Li and Kuo [5] showed that the R-D slopes of significance identification and refinement coding was different, and by placing the significance identification before refinement coding in each layer, the coding efficiency could be improved. However, the improvement of [5] was fairly limited.

In this research, we propose a rate-distortion optimized embedding (RDE) by allocating the available coding bits first to the coefficient with the steepest R-D slope, i.e. the biggest distortion decrease per coding bit. Considering the synchronization between the encoder and the decoder, the actual optimization is based on the expectation R-D slope that can be calculated on the decoder side. We take advantage of the probability estimation table of the QM-coder [6] [7] to simplify the calculation of the R-D slope and speed up the algorithm. The algorithm significantly improves the coding efficiency.

The paper is organized as follows. The initiative of the rate-distortion optimization is introduced in Section 2. The framework and the implementation detail of RDE are investigated in Section 3. We focus primarily on the two key steps of RDE, i.e., the R-D slope calculation and the coefficient selection. Extensive experimental results are shown in Section 4 to compare the performance of RDE with various other algorithms. Concluding remarks are presented in Section 5.

2. CODING OPTIMIZATION BASED ON THE RATE-DISTORTION SLOPE

For convenience, let us assume that the image has already been converted into the transform domain. The transform used in embedded coding is usually wavelet decomposition, but it can be DCT as well, as in [10] . Let the index position of the image be denoted as i=(x,y), let the coefficient at index position i be denoted as w_i. Suppose the coefficients have been normalized with absolute maximum value of 1. Because w_i is between –1 and +1, it can be represented by a binary bit stream as:

$$\pm 0.b_1 b_2 b_3 \ldots b_j \ldots \qquad (1)$$

where b_j is the jth most significant bit (MSB) of coefficient w_i. Traditional coding first determines the bit-depth n (or quantization precision), then sequentially encodes one coefficient by another. For each coefficient w_i, the most significant n bits are bound together with the sign

$$\pm 0.b_1 b_2 b_3 \ldots b_n \qquad (2)$$

and encoded by the entropy coder. The quantization precision is predetermined before coding. If the coding bitstream is truncated, the bottom half of the coefficients will be lost. Embedded coding is distinctive from traditional coding in the sense that the image is coded bit-plane by bit-plane rather than coefficient by coefficient. It first encodes bit b_1 of coefficient w_1, then bit b_1 of w_2, then bit b_1 of w_3, etc.. After the bit plane b_1 of all coefficients has been encoded, it moves on to bit plane b_2, and then bit plane b_3. In the embedded coding, each bit is an independent coding unit. Whenever a coefficient w_i becomes nonzero, its sign is encoded right after. The embedded coding bitstream can be truncated at any point with graceful quality degradation since at least part of each coefficient has been coded.

Although the bit-plane oriented embedded coding is better than the coefficient oriented coding, it is still not optimal in the rate-distortion (R-D) sense. We illustrate the initiative in Figure 1. Suppose there are five symbols a, b, c, d and e that can be coded independently. Coding each symbol requires a certain amount of bits and results in a certain amount of coding distortion decrease. Coding of all symbols sequentially gives an R-D curve shown as the solid line in Figure 1. If the coding order is reorganized so that the symbol with the steepest R-D slope is encoded first, we can get an R-D curve shown as the dashed line in Figure 1. Though both lines reach the same final R-D point, the algorithm that follows the dashed line performs better when the coding is truncated at an intermediate bit rate. Therefore, the initiative of the rate-distortion optimized embedding (RDE) is to allocate the available coding bits first to the coefficient with the steepest R-D slope, i.e., the coefficient with the biggest distortion decrease per coding bit. Note when all bits have been coded (i.e., the coding achieves lossless), RDE will have exactly the same performance as its counterpart without R-D optimization. However, in a practical wide bit rate range, RDE will outperforms traditional embedding significantly.

3. IMPLEMENTATION OF RATE-DISTORTION OPTIMIZED EMBEDDING (RDE)

The framework of the RDE can be shown in Figure 2. In general, RDE calculates the R-D slope, or the distortion decrease divided by the coding bits, for the candidate bit of all coefficients. It then encodes the coefficient w_i that has the steepest R-D slope, i.e., the biggest distortion decrease per coding bit. Such coding strategy achieves embedding with optimal R-D performance. If the coding bit stream is truncated, the performance of coding at that bit rate will be optimal. Since the actual R-D slope will not be available at the decoder, RDE uses the expectation R-D slope that can be calculated by the already coded bits. By doing so, the decoder can derive the same R-D slope and track the coefficient to be transmitted next just the same as the encoder. Therefore, the location information of w_i does not need to be transmitted.

3.1 Calculation of Rate-Distortion Slope

The two key steps of RDE are coefficient selection and R-D slope calculation. In this subsection, we develop a very efficient algorithm which calculate the R-D slope with just a lookup table operation.

For coefficient w_i, assume the most significant n_i-1 bits have already been coded, and the n_ith bit is the next bit to be processed. We call n_i as the current coding layer, and the n_ith bit of w_i as the candidate bit. The situation is shown in Figure 3, where the coded bits are marked with slashes, and the candidate bits are marked with horizontal or vertical bars. As traditional embedded coding, RDE classifies the coding of candidate bits into two categories – significance identification and refinement coding. For coefficient w_i, if all previous coded bits b_j are '0' for $j=1 \cdots n_i-1$, the significance identification mode is used for the n_ith bit. If either one of the previous coded bits is '1', the refinement mode is used. We show an example in Figure 3, where the bits undergone significance identification are marked by vertical bars, and the bits undergone refinement coding are marked by horizontal bars. The R-D slopes for the two modes are very different.

The significance identification mode classifies coefficient w_i from interval (-2T,2T) to interval (-2T,-T] of negative significance, (-T,T) of non-significance, and [T,2T) of positive significance, where $T=2^{-n_i}$ is the width of the coding interval determined directly by the current coding layer n_i. From the coding interval, we can easily derive the decoding reconstruction before significance identification as:

$$r_b=0 \tag{3}$$

and the decoding reconstruction for negative significance, non-significance, positive significance as:

$$r_{s-,a}=1.5T,\ \ r_{i,a}=0\ ,\ \ r_{s-,a}=1.5T \tag{4}$$

respectively. In significance identification, the coded symbol is highly biased towards non-significance. We encode the result of significance identification with a QM-coder, which estimates the probability of significance p_i with a state machine, and then arithmetic encodes the result. As shown in Figure 4, the QM-coder uses a context which is a 7-bit string with each bit representing the significant status of one spatial neighbor coefficient or one coefficient at the same spatial position but in the parent band of current coefficient w_i. By monitoring the pattern of past 0s ('insignificance') and 1s ('significance') under the same context (i.e., the same neighborhood configuration), the QM-coder estimates the probability of significance p_i of the current coding symbol. The probability estimation is very simple for QM-coder, as it is just a state table transition operation. For details of the QM-coder and its probability estimation, we refer to [6] [7] . Assuming the QM-coder performs closely to the Shannon bound for coding the result of significance identification, we may calculate the expectation coding rate as:

$$E[\Delta R]=(1-p_i)\,\Delta R_{insig}+p_i\Delta R_{sig}$$
$$=(1-p_i)\,[-\log_2(1-p_i)]+p_i(-\log_2 p_i+1)=p_i+H(p_i) \quad (5)$$

where $H(p_i)$ is the entropy of a binary symbol with probability of 1 equals to p_i. Note that in (5), when the symbol becomes significant, it needs one additional bit to encode the sign of significance. With the known probability of significance p_i, the expectation distortion decrease of significance identification can be calculated as:

$$E[\Delta D]=(1-p_i)\,\Delta D_{insig}+p_i\,\Delta D_{sig} \quad (6)$$

where ΔD_{sig} and ΔD_{insig} can be further calculated by taking the probability weighted average of the coding distortion decrease:

$$\Delta D_{insig}=\int_{-T}^{T}[(x-r_b)^2-(x-r_{i,a})^2]p(x)dx \quad (7)$$

$$\Delta D_{sig}=\int_{-2T}^{-T}[(x-r_b)^2-(x-r_{s-,a})^2]p(x)dx$$
$$+\int_{T}^{2T}[(x-r_b)^2-(x-r_{s+,a})^2]p(x)dx \quad (8)$$

By substituting (3) and (4) into (7), it becomes:

$$\Delta D_{insig}=0 \quad (9)$$

There is no coding distortion decrease if w_i is still insignificant. To calculate ΔD_{sig}, we need the *a priori* probability distribution of coefficient w_i within the significance interval (-2T,-T] and [T,2T). Note that the probability distribution of w_i within interval (-T,T) is irrelevant to the calculation of distortion decrease. Assuming that the *a priori* probability distribution within the significance interval is uniform, we can derive:

$$p(x)=\frac{1}{2T}, \quad \text{for } T<|x|<2T \quad (10)$$

Based on (3), (4), (8) and (10), we conclude that

$$\Delta D_{sig}=2.25T^2 \quad (11)$$

The expectation distortion decrease for significance identification is therefore:

$$E[\Delta D]=p_i\,2.25T^2 \quad (12)$$

It is straightforward to derive the R-D slope of significant identification w_i as:

$$\lambda_{sig}=\frac{E[\Delta D]}{E[\Delta R]}=\frac{2.25T^2}{1+H(p_i)/p_i} \quad (13)$$

Function

$$f_s(p_i)=\frac{1}{1+H(p_i)/p_i} \quad (14)$$

is plotted in Figure 5. It is apparent that the symbol with higher probability of significance has a larger rate-distortion slope and is thus favored to be encoded first.

We may similarly calculate the R-D slope of refinement coding mode. The refinement coding classifies coefficient w_i from interval [S,S+2T) to interval [S,S+T) or [S+T,S+2T), where $T=2^{-n_i}$ is again determined by the coding layer n_i, and S is the start of the refinement interval, which is the value indicated by the coded bits of the coefficients. The decoding reconstruction before refinement coding is:

$$r_b=S+T \quad (15)$$

Depending on the coding result, the decoding reconstruction after refinement becomes:

$$r_{0,a}=S+0.5T,\ r_{1,a}=S+1.5T \quad (16)$$

Because the refinement coding is equilibrium between ‘0’ and ‘1’, the expectation coding rate is close to one bit.

$$E[\Delta R]=1.0 \quad (17)$$

Similar to (6), the expectation distortion decrease can be calculated as:

$$E[\Delta D]=\int_{S}^{S+T}[(x-r_b)^2-(x-r_{0,a})^2]p(x)dx$$
$$+\int_{S+T}^{S+2T}[(x-r_b)^2-(x-r_{1,a})^2]p(x)dx \quad (18)$$

Assuming that the *a priori* probability distribution with in interval [S,S+2T) is uniform, i.e., p(x)=1/2T, we conclude for refinement coding:

$$E[\Delta D]=0.25\,T^2 \quad (19)$$

The R-D slope of refinement coding is thus:

$$\lambda_{ref}=\frac{E[\Delta D]}{E[\Delta R]}=0.25T^2 \quad (20)$$

Compare (13) and (20), it is apparent that for the same coding layer n_i, the R-D slope of refinement coding is smaller than that of significance identification whenever the significance probability p_i is above 0.01. Thus in one coding layer, significance identification should be in general placed before the refinement coding, as indicated in [5] .

We have also modeled the *a priori* probability function of coefficient w_i to be Laplacian. In such case, the R-D slope for significance identification and refinement coding becomes:

$$\lambda_{sig}=\frac{2.25T^2}{1+H(p_i)/p_i}\,g_{sig}(\sigma,T) \quad (21)$$

$$\lambda_{ref}=0.25T^2\,g_{ref}(\sigma,T) \quad (22)$$

where σ is the variance of Laplacian distribution which is again estimated from the already coded bits, and $g_{sig}(\sigma,T)$ and $g_{ref}(\sigma,T)$ are Laplacian modification factors in the form of:

$$g_{sig}(\sigma,T)=\frac{1}{2.25}\left(0.75+\frac{3\sigma}{T}-\frac{3e^{-T/\sigma}}{1-e^{-T/\sigma}}\right) \quad (23)$$

$$g_{ref}(\sigma,T)=4\left(0.75+\frac{2\frac{\sigma}{T}e^{-T/\sigma}-\frac{\sigma}{T}\left(1+e^{-2T/\sigma}\right)}{1-e^{-2T/\sigma}}\right) \quad (24)$$

However, experiments show that the additional performance improvement provided by the Laplacian model is minor. Since the uniform probability distribution model is much simpler to implement, it is used throughout the experiment.

Because the probability of significance p_i determined by the QM-coder state is discrete, and the width of interval T determined by the coding layer n_i is discrete, both R-D slope (13) and (20) have a discrete number of states. For fast calculation, (13) and (20) may be pre-computed and stored in a table indexed by the coding layer n_i and the probability state of QM-coder. Thus, computation of the R-D slope will be only a lookup table operation. The number of entries M of the lookup table can be calculated by:

$$M=2xNxL+N \quad (25)$$

where N is the maximum coding layer, L is the number of states in the QM-coder. In the current implementation of RDE, there are a total of 113 states in the QM-coder, and a maximum of 20 coding layers. This brings a lookup table of size 4540.

3.2 Coefficient Selection and Flowchart of Rate-Distortion Optimization

The second key step in RDE is selecting the coefficient with the maximum R-D slope. Since an exhaustive search or a sorting over all image coefficients is computational expensive, a threshold based selection approach is introduced in this subsection. We first set an R-D slope threshold λ. The whole image is scanned and those coefficients with R-D slope greater than λ are encoded. The R-D slope threshold λ is then reduced by a certain factor and the whole image is scanned again.

The coding operation of RDE can be shown in Figure 6. It can be described step by step as:

1) The image is decomposed by the wavelet transform.
2) The initial R-D slope threshold λ is set to λ_0, with:

$$\lambda_0 =0.25x0.5^2=0.03125 \quad (26)$$

3) The image is scanned and coded.

   The scan goes from the coarse scale to the fine scale, and follows the raster line order within the subband.
4) For each coefficient, its expectation R-D slope is calculated.

   Depending on whether the coefficient is already significant, the R-D slope is calculated by (13) and (20), respectively. Note that the calculation of the R-D slope is only a lookup table operation indexed by the QM-coder state and the coding layer n_i.
5) The calculated R-D slope is compared with λ.

   If the R-D slope is smaller than λ, the coding proceeds to the next coefficient. Only those coefficients with R-D slope greater than λ are encoded.
6) The coefficient is encoded with significant identification or refinement coding.

   The bit of significance identification is encoded by a QM-coder with context designated in Figure 4. The bit of sign and the bit of refinement are encoded by a QM-coder with fixed probability 0.5.
7) The coder checks if the assigned coding rate is reached. If not, it repeats step 4) until the entire image has been scanned.
8) The R-D slope threshold is reduced by a factor of α:

$$\lambda \leftarrow \lambda/\alpha \quad (27)$$

In our current implementation, α is set to 1.25. The coder then goes back to step 3) and scans the entire image again.

4. EXPERIMENT RESULTS

Extensive experiments are performed to compare RDE with two existing algorithms. One is a predictive embedded zerotree wavelet (PEZW) coding which is an improved version of the original EZW and is currently in the MPEG4 verification mode (VM) 6.0. We use PEZW as a reference for the state-of-the-art wavelet coding technique. The other one is the layered zero coding (LZC) proposed by Taubman and Zakhor. RDE encodes the bit of significance identification and the bit of refinement very similar to the one used by the LZC. In essence, RDE shuffles the embedded bitstream of LZC to improve its performance. We therefore compare the RDE with LZC to show the advantage of rate-distortion optimization. Two experimental images are used. One is the famous Lena of size 512x512, the other one is the first frame of Akiyo, which is a MPEG4 test image of size 176x144 (QCIF) or size 352x288 (CIF). The Lena image is decomposed by a 5-level 9-7 tap biorthogonal Daubechies filter [8] as in most of the other coding literature. The Akiyo image is decomposed according to the specification in MPEG4 VM 6.0 with a 9-3 tap biorthogonal Daubechies filter [8] of 4-levels for QCIF and 5-levels for CIF.

The performance of RDE versus LZC of Lena for bit rate range 0.15bpp to 1.0bpp is shown in Figure 7. We plot the R-D performance curve of LZC with dashed line, and plot the R-D performance curve of RDE with solid line. For the clarity of the result, we split the result into 3 subgraphs with bitrate range 0.15-0.30bpp, 0.40-0.60bpp and 0.70-1.00bpp. RDE outperforms LZC by 0.2-0.4dB over the entire bit rate range. The performance advantage becomes larger at higher bitrate. The R-D performance curve of RDE is also much smoother than that of LZC. In Table 1, we show the performance comparison of PEZW, LZC and RDE on Akiyo image at 10k, 20k and 30k bits for QCIF and 25k, 50k and 70kbits for CIF. The result of PEZW is obtained from document [9] . It is shown that the performance gap between LZC and RDE becomes larger for a low-detailed image as Akiyo. In general, RDE outperforms LZC for 0.4-0.8dB, and outperforms PEZW for 0.3-1.2dB for Akiyo image over a variety of bitrate.

5. CONCLUSIONS

In this paper, we propose a rate-distortion optimized embedding (RDE) algorithm. By reordering the embedding bitstream so that the available coding bit is first allocated to the coefficient with the steepest R-D slope, i.e. the biggest distortion decrease per coding bit, RDE substantially improves the performance of embedded coding. For synchronisation between the encoder and the decoder, RDE uses the expectation R-D slope which can be calculated at the decoder side with the already coded bits. RDE also takes advantage of the probability estimation table of the QM-coder so that the calculation of the R-D slope is just one lookup table operation. Extensive experimental results confirm that RDE significantly improves the coding efficiency.

6. REFERENCES

[1] J. Shapiro, "Embedded image coding using zerotree of wavelet coefficients", IEEE Trans. On Signal Processing, vol. 41, pp.3445-3462, Dec. 1993.

[2] A. Said, and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees", IEEE Trans. On Circuit and System for Video Technology, Vol. 6, No. 3, Jun. 1996, pp. 243-250

[3] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video", IEEE Trans. On Image Processing, Vol. 3, No. 5, Sept. 1994, pp.572-588.

[4] Z. Xiong and K. Ramchandran, "Wavelet packet-based image coding using joint space-frequency quantization", First IEEE International Conference on Image Processing, Austin, Texas, Nov. 13-16, 1994.

[5] J. Li, P. Cheng and C. -C. J. Kuo, "On the improvements of embedded zerotree wavelet (EZW) coding", SPIE: Visual Communication and Image Processing, vol. 2601, Taipei, Taiwan, May. 1995, pp. 1490-1501.

[6] W. Pennebaker and J. Mitchell, IBM, "Probability adaptation for arithmetic coders", US patent 5,099,440, Mar. 24, 1992.

[7] D. Duttweiler and C. Chamzas, "Probability estimation in arithmetic and adaptive Huffman entropy coders", IEEE Trans. On Image Processing, vol. 4 no. 3, Mar. 1995, pp. 237-246.

[8] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," IEEE Trans. on Image Processing, Vol. 1, pp. 205-230, Apr. 1992.

[9] J. Liang, "Results Report on Core Experiment T1: Wavelet Coding of I Pictures," ISO/IEC JTC1/SC29/WG11 MPEG97/M1796,Feb.1997.

[10] J. Li, J. Li, C.-C. Jay Kuo, "Layered DCT still image compression", IEEE Trans. On Circuit and System for Video Technology, Vol. 7, No. 2, pp.440-443, Apr. 1997.

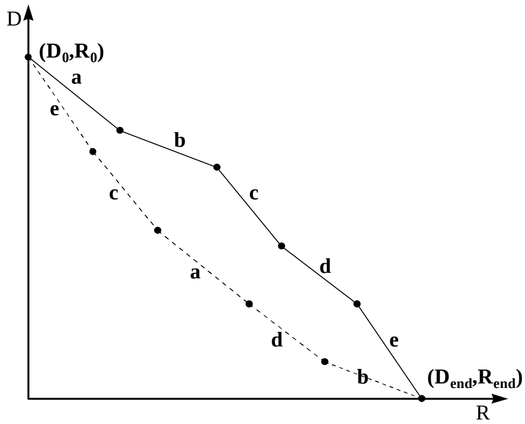


Figure 1 Initiative of rate-distortion optimization.

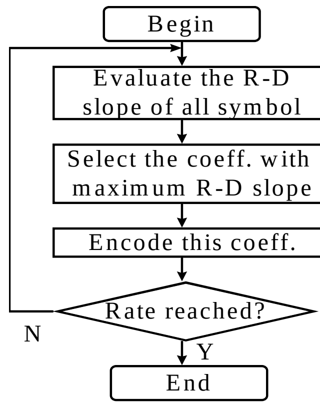


Figure 2 The framework of rate-distortion optimized embedding.

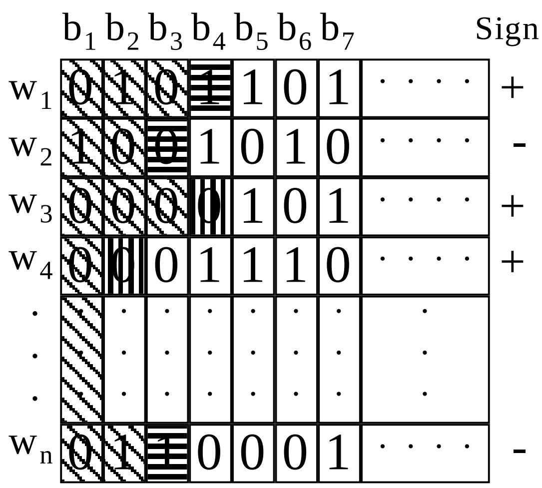


Figure 3 Element of rate-distortion optimized embedded. (The already coded bits are marked with slashes, the candidate bits of significance identification with vertical bars, the candidate bits of refinement coding with horizontal bars.)

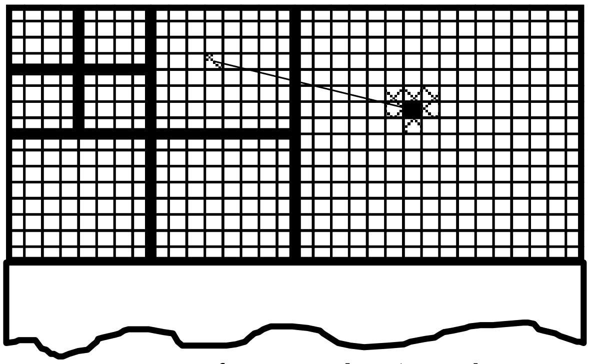


Figure 4 Context for QM-coder. (■ is the current coding coefficient w_i , ⊡ are its context coefficients.)

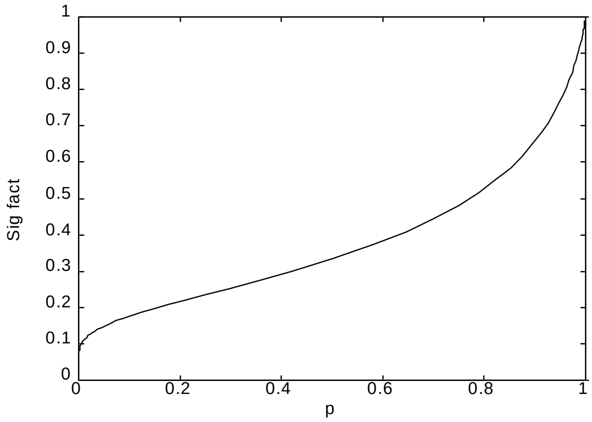


Figure 5 The rate-distortion slope modification factor for significance identification.

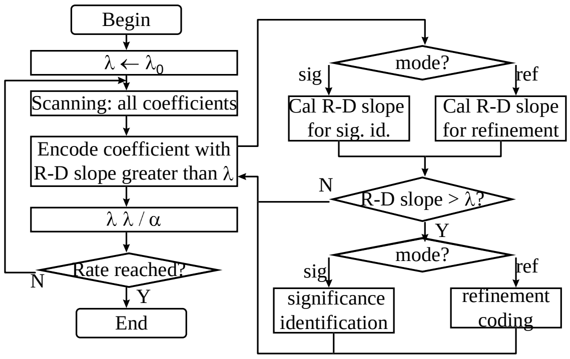
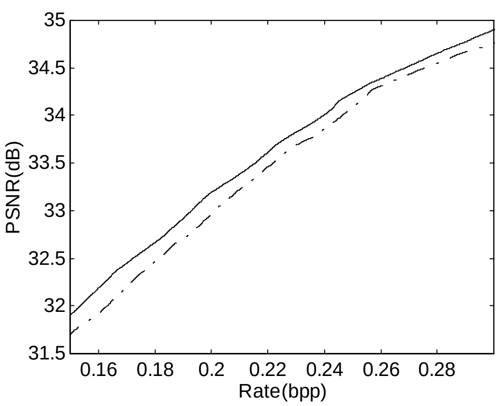
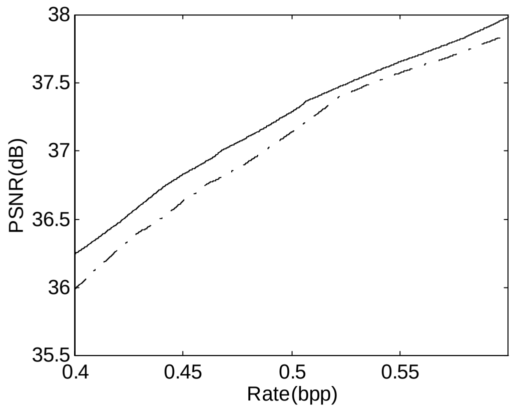


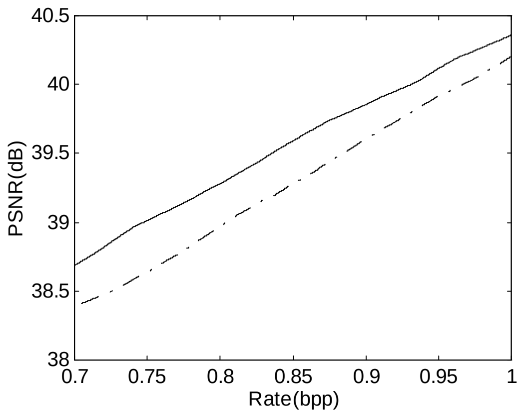
Figure 6 The flowchart of rate-distortion optimized embedding.



(a)



(b)



(c)

Figure 7 Comparison between RDE (solid line) and LZC (dashed line) for Lena image with bitrate range (a) 0.15-0.3bpp, (b) 0.4-0.6bpp, (c) 0.7-1.0bpp.

Table 1 Experimental result for image Akiyo.

	LZC		PEZW		RDE	
Size	Bit Rate (bits)	PSNR_Y PSNR_U PSNR_V (dB)	Bit Rate (bits)	PSNR_Y PSNR_U PSNR_V (dB)	Bit Rate (bits)	PSNR_Y PSNR_U PSNR_V (dB)
QCIF	10000	32.38 34.67 37.12	10256	32.3 34.2 36.9	10000	33.13 35.02 37.90
QCIF	20000	37.30 39.18 41.18	20816	37.5 39.1 41.0	20000	37.97 40.03 42.17
QCIF	30000	41.40 43.56 44.00	29240	40.8 41.5 42.6	30000	42.01 44.23 44.75
CIF	25000	34.69 38.25 40.38	25112	34.7 37.7 40.1	25000	35.03 38.23 41.03
CIF	50000	39.23 41.72 44.24	49016	39.3 41.3 43.6	50000	40.04 42.46 44.73
CIF	70000	42.35 44.58 46.12	70448	42.2 43.2 45.2	70000	43.03 44.35 46.39