

# RATE-DISTORTION UNEQUAL ERROR PROTECTION FOR FRACTAL IMAGE CODES

Vladimir Stankovic, Raouf Hamzaoui, Dietmar Saupe

Universität Leipzig, Institut für Informatik, Augustusplatz 10–11, 04109 Leipzig, Germany

## ABSTRACT

Fractal image codes are very sensitive to bit errors because the decoding of a block is dependent not only on the code information associated to this block but also on the code information associated to other blocks. We analyze the sensitivity of a fractal code to transmission errors in a binary symmetric channel. We provide two rate-distortion unequal error protection techniques that allocate the code bits to protection classes in a nearly optimal way. We give an implementation for BCH and RCPC channel codes and show that RCPC codes are preferable. For a binary symmetric channel bit error probability of 0.1 and a total code rate of 0.5 bpp, the loss in reconstruction quality with our best implementation was about 3.9 dB for the  $512 \times 512$  Lenna image, yielding a PSNR of 26.6 dB.

## 1. INTRODUCTION

Sending an image code through a noisy channel is a challenging problem because a single bit error may make the reconstructed image unacceptable. Fractal image codes [3], which consist of an image partition into blocks and of parameters that specify an affine similarity between the blocks in the partition and other blocks from the same image are particularly sensitive to bit errors because an error in a single parameter can effect the reconstruction of many parts of the image due to the iterative nature of the decoding and the complex dependency between the blocks. Although more than 650 papers have been dedicated to fractal image compression (see [9]), only three considered transmission through noisy channels. Streit and Hanzo [10] studied the sensitivity of a fractal code for videophone applications with a mobile Rayleigh-fading channel. They used unequal error protection (UEP) and BCH codes. However, the results were given for a poor coder based on uniform partitions, and the unequal error protection was not optimized. Novak [7] introduced a model-residual fractal coder that is more robust than the standard fractal coder. But he did not consider error protection. Finally, Noh, Kim, and Kim [6] used an interpolation technique for reconstructing blocks lost during the transmission of a fractal code in broadband integrated-services digital networks with asynchronous transfer mode protocols.

This paper is organized as follows. In Section 2, we introduce our terminology. In Section 3, we analyze the sensitivity of a quadtree-based fractal image scheme to errors in a binary symmetric channel. In Section 4, we propose two rate-distortion unequal error protection techniques based on discrete Lagrangian optimization. In Section 5, we compare unequal error protection to equal

error protection for BCH and RCPC channel codes. In the last section, we discuss the results and suggest future work.

## 2. FRACTAL CODING

In fractal compression, the encoder finds a contractive image operator  $T$  whose fixed point  $f_T$  approximates the original image  $f^*$ . The decoder constructs  $f_T$  as the limit of  $\{f^{(k)}\}_{k \geq 0}$ , where  $f^{(k+1)} = T(f^{(k)})$  and  $f^{(0)}$  is an arbitrary initial image. The operator  $T$  is given by a partition of the image support  $\mathcal{I}$  into  $n_R$  disjoint square blocks called *ranges* and by *fractal parameters* associated to each range  $R_i$ ,  $1 \leq i \leq n_R$ , and consisting of

- a square block (*domain*)  $D_{j_i} \in \{D_1, \dots, D_{n_D}\} \subset \mathcal{I}$ ,
- an isometry of the square  $I_{k_i} \in \{I_1, I_2, \dots, I_8\}$ ,
- a *scaling factor*  $s_{l_i} \in \{s_1, \dots, s_{n_s}\} \subset (-1, 1)$ ,
- and an *offset*  $o_{m_i} \in \{o_1, \dots, o_{n_o}\} \subset \mathbb{R}$ .

The parameters  $D_{j_i}$ ,  $I_{k_i}$ ,  $s_{l_i}$ , and  $o_{m_i}$  are selected from their respective sets such that

$$\widehat{\mathbf{R}}_i = s_{l_i} I_{k_i} A_i \mathbf{D}_{j_i} + o_{m_i} \mathbf{1} \quad (1)$$

is the best  $l_2$  approximation of  $\mathbf{R}_i$ . Here  $\mathbf{B}$  denotes the array of pixel intensities of  $f^*$  in  $B \subset \mathcal{I}$ , the operator  $A_i$  downsamples  $\mathbf{D}_{j_i}$  via pixel averaging to match the range size, and  $\mathbf{1}$  is the block with intensity 1 at every pixel. Using the method of least squares, the optimal nonquantized scaling factor and offset associated to a domain  $D$  and an isometry  $I$  are

$$s = \frac{|R_i| \langle C, \mathbf{R}_i \rangle - |R_i|^2 \mu(\mathbf{D}) \mu(\mathbf{R}_i)}{|R_i| \langle C, C \rangle - \mu(\mathbf{D})^2} \quad (2)$$

and

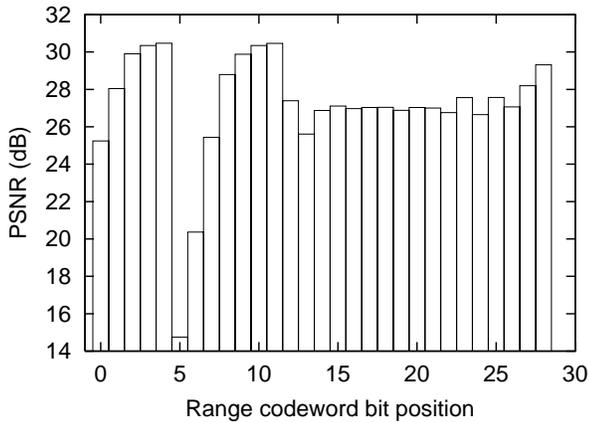
$$o = \mu(\mathbf{R}_i) - s \mu(\mathbf{D}). \quad (3)$$

Here  $|R_i|$  is the number of pixels in  $R_i$ ,  $C = I A_i \mathbf{D}$ ,  $\mu(\mathbf{B})$  denotes the mean of  $\mathbf{B}$ , and  $\langle \cdot, \cdot \rangle$  denotes the inner product. The error  $\sum_{i=1}^{n_R} \|\mathbf{R}_i - \widehat{\mathbf{R}}_i\|_2^2$  is called *collage error*. The code for the original image consists of bits for the image partition and bits for the indices of the fractal parameters. The decoder constructs  $f_T = \lim_{k \rightarrow \infty} f^{(k)}$  by

$$\mathbf{R}_i^{(k+1)} = s_{l_i} I_{k_i} A_i \mathbf{D}_{j_i}^{(k)} + o_{m_i} \mathbf{1}, \quad 1 \leq i \leq n_R,$$

where  $\mathbf{R}_i^{(k+1)}$  (respectively  $\mathbf{D}_{j_i}^{(k)}$ ) is the array of pixel intensities of  $f^{(k+1)}$  in  $R_i$  (respectively of  $f^{(k)}$  in  $D_{j_i}$ ). The reconstruction error  $d = \|f^* - f_T\|_2^2$  is called the *attractor error*.

To appear in the Proceedings of ICIP-2001, IEEE Int. Conf. on Image Processing, Thessaloniki, Oct. 2001. This document corrects the published paper, which, due to a bug in the used program, contains errors in Table 1, Table 2, Fig. 3, Fig. 4, and Fig. 5.



**Fig. 1.** PSNR for the  $512 \times 512$  Lenna image at 0.1 BER and source rate 0.21 bpp. Each box gives the PSNR when only one of the 29 bits in each range is subject to corruption.

### 3. BIT SENSITIVITY ANALYSIS

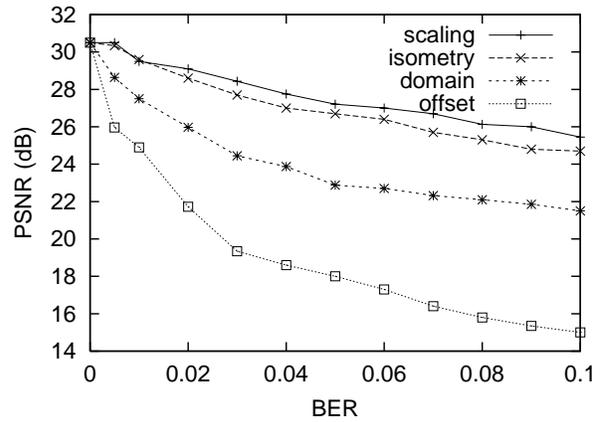
We used Fisher's quadtree coder [4], which we modified according to Øien's orthogonalization technique [8]. Thus, the approximation (1) was replaced by

$$\widehat{\mathbf{R}}_i = s_{l_i} I_{k_i} A_i (\mathbf{D}_{j_i} - \mu(\mathbf{D}_{j_i})\mathbf{1}) + o_{m_i} \mathbf{1}. \quad (4)$$

Now the optimal nonquantized offset is  $\mu(\mathbf{R}_i)$ , and the optimal nonquantized scaling factor and optimal collage error are the same as in the usual case. This yields a more robust code because in contrast to equation (3), errors in the scaling factor do not affect the mean of a reconstructed block.

The scaling factor and the offset were uniformly quantized with 32 and 128 quantization levels, respectively, and their indices were encoded with five and seven bits, respectively. The isometry index was encoded with three bits. The smallest range size was  $8 \times 8$ , and the largest range size was  $32 \times 32$ . For each  $2^n \times 2^n$  range, the candidate domains consisted of the  $2^{n+1} \times 2^{n+1}$  blocks of the image whose upper-left pixels are situated on locations  $(i, j)$ , where  $i \equiv 0 \pmod{4}$  and  $j \equiv 0 \pmod{4}$ . Thus, the domain index was encoded with 14 bits for images of size  $512 \times 512$ . This yields a total of 29 bits per range. In Fisher's original coder, if the optimal scaling factor for a range is equal to zero, then no domain and isometry bits are sent because they are redundant. However, to ensure synchronization in our transmission system, we must send these bits.

We denote the part of the code associated to range  $R$  by  $\omega_R = s_4 s_3 \cdots s_0 o_6 o_5 \cdots o_0 i_2 i_1 i_0 d_{13} d_{12} \cdots d_0$ , where  $s_4 s_3 \cdots s_0$ ,  $o_6 o_5 \cdots o_0$ ,  $i_2 i_1 i_0$ , and  $d_{13} d_{12} \cdots d_0$  are the binary representations of the scaling factor, the offset, the isometry, and the domain indices, respectively. For each range, we corrupted the bit at the same position in  $\omega_R$  with a certain probability and computed the PSNR of the reconstructed image. We did not introduce errors in the bits that specify the quadtree partition because those bits, which constitute less than 2/100 of the total code, can be perfectly protected with a negligible increase of the bitrate. Figure 1 gives the results for the  $512 \times 512$  Lenna image and a binary symmetric channel bit error rate (BER) of 0.1. We obtained similar results for other BERs and other images. Figure 2 shows the



**Fig. 2.** PSNR vs. BER in all bits of a single fractal parameter for the Lenna image at source rate 0.21 bpp.

PSNR as a function of the BER when all the bits of only one fractal parameter were corrupted. In both figures, each PSNR is averaged over 50 experiments with the same BER.

The analysis shows that the different fractal parameters had unequal importance. The most sensitive parameter was the offset, followed by the domain, the isometry, and the scaling factor. Moreover, for the offset and the scaling factor, a more significant bit of a given parameter was more sensitive than a less significant bit. For the domain, all bits except for the two least significant ones had about the same sensitivity. This was expected because for the used domain pool an error in the two least significant bits selects a domain that overlaps the original one, whereas an error in any of the other bits selects an unrelated domain.

### 4. ERROR PROTECTION

Since not all ranges have the same contribution to the reconstruction error, it would be advantageous to use unequal protection of the ranges. However, this requires overhead information, which depends not only on the image but also on the bitrate. Moreover, the encoder would have to wait until all ranges are encoded before starting sending the code to the channel. Therefore, we protected all ranges in the same way. On the other hand, the bits of the range codeword  $\omega_R$  were unequally protected because they do not have the same importance.

Let  $\mathcal{C}_1, \dots, \mathcal{C}_m$  denote  $m$  protection classes with respective rates  $r_1 < \dots < r_m$ , where

$$r_i = \frac{\text{information bits}}{\text{information bits} + \text{error protection bits}}.$$

For a given BER our goal is to assign the bits of each range codeword  $\omega_R$  into the protection classes such that the expectation of the reconstruction error  $d$  is minimum under a constraint on the total number of protection bits, which is computed as

$$n_R \sum_{i=1}^m \frac{1 - r_i}{r_i} n_i,$$

where  $n_i$  is the number of a range codeword bits assigned to  $\mathcal{C}_i$  (thus  $\sum_{i=1}^m n_i = 29$ ). Using the Lagrange multiplier method [2],

BCH	UEP2
0.278	$s_3, s_4, i_0, i_1, i_2, o_3, o_4, o_5, o_6$
0.48	<b>d</b>
1.0	$s_0, s_1, s_2, o_0, o_1, o_2, d_0$
RCPC	UEP2
0.33	$s_4, o_4, o_5, o_6, i_0, i_1$
0.36	$s_3, i_2$
0.4	$o_3, \mathbf{d}$
1	$s_0, s_1, s_2, o_0, o_1, o_2, d_0$

**Table 1.** Bit allocation for UEP2 at BER 0.1, source rate 0.21 bpp, and total rate 0.46 bpp for BCH and RCPC codes. Here **d** denotes the domain bits  $d_1, \dots, d_{13}$ .

an optimal allocation can be found by minimizing the Lagrangian

$$L = E\{d\} + \lambda n_R \sum_{i=1}^m \frac{1-r_i}{r_i} n_i$$

over all possible allocations. To meet the constraint on the total number of protection bits, the optimization has to be done for several values of the Lagrange multiplier  $\lambda$  by the bisection method.

Because the number of possible bit allocations is equal to  $m^{5+7+3+14}$ , it is not possible to find the optimal solution by exhaustive search. The complexity of the search was reduced by minimizing the expectation of the collage error,  $E\{c\}$ , instead of the expectation of the reconstruction error,  $E\{d\}$ , and by noting that the collage error for range  $R_i$  is

$$\|\mathbf{R}_i - \mu(\mathbf{R}_i) \mathbf{1} - s_i I_{k_i} A_i (\mathbf{D}_{j_i} - \mu(\mathbf{D}_{j_i}) \mathbf{1})\|_2^2 + |R_i| (\mu(\mathbf{R}_i) - o_{m_i})^2.$$

This approach allows us to split the Lagrangian into two independent parts. The first part,  $L_1$ , which does not include the offset contribution, has to be minimized over  $m^{5+3+14}$  bit allocations, and the second part,  $L_2$ , over  $m^7$  possible allocations of the offset bits. But this is still too complex for practical computations. Therefore, motivated by the sensitivity analysis of the previous section, we made the following reasonable simplifications:

- All domain bits, with the exception of  $d_0$ , were constrained to be in the same class.
- For a given fractal parameter, less sensitive bits were not protected better than more sensitive bits.

We call this technique UEP1. Now the number of bit allocations considered by  $L_1$  reduces to  $\binom{m+4}{5} \binom{m+2}{3} \binom{m+1}{2}$ , and the number of bit allocations considered by  $L_2$  reduces to  $\binom{m+6}{7}$  (see the appendix for a justification). But when  $m$  is large, this is still impractical. Thus, we propose a faster algorithm (UEP2) as follows. Let **d** denote the domain bits  $d_{13} \dots d_1$ . Then UEP2 assigns parameter  $i \in \{s_4, \dots, s_0, o_6, \dots, o_0, i_2, i_1, i_0, \mathbf{d}, d_0\}$  of codeword  $\omega_R$  to a protection class independently of the other parameters. So the index of the selected class for parameter  $i \neq \mathbf{d}$  is

$$\arg \min_{k=1, \dots, m} E\{d\} + \lambda n_R \frac{1-r_k}{r_k}, \quad (5)$$

where the reconstruction error  $E\{d\}$  is computed such that only parameters  $i$  (and their protection bits) are subject to channel error. The case  $i = \mathbf{d}$  is similar with  $\frac{1-r_k}{r_k}$  in (5) replaced by  $\frac{13(1-r_k)}{r_k}$ .

## 5. RESULTS

We present coding results for UEP2 with BCH( $n, k, t$ ) codes [1] and RCPC codes [5]. For BCH codes, we defined  $m = 25$  protection classes by setting the code length  $n$  to 255 and letting the information sequence length  $k$  take all values in  $\{255, 247, 239, \dots, 199, 191, 187, 179, \dots, 79, 71\}$ . We used Berlekamp's algorithm for the decoding. The constraint length of the RCPC code was  $K = 7$ , the puncturing period was  $p = 8$ , the generator polynomials were  $(G_1, G_2, G_3) = (133, 171, 145)$ , and the convolutional mother code had the rate  $\frac{1}{n} = \frac{1}{3}$ . This provided the code rates  $\frac{p}{l}$ ,  $l = p, p+1, \dots, np$ , from which we retained the  $m = 10$  rates  $8/24, 8/22, 8/20, 8/18, 8/16, 8/14, 8/12, 8/10, 8/9$ , and 1. The Lagrange optimization for the bit allocation was done for BER = 0.1, where  $E\{d\}$  was estimated by repeating the computations fifty times. Table 1 shows the resulting bit allocations for the  $512 \times 512$  Lenna image at source rate 0.21 bpp and total rate 0.46 bpp. Figure 3 compares UEP2 to equal error protection (EEP) for BCH codes. Results for RCPC codes are shown in Figure 4. The simulations show that both EEP and UEP2 can efficiently protect the fractal code. Moreover UEP2 was much better than EEP. Figure 5 compares BCH and RCPC channel codes. RCPC codes were superior to BCH codes.

Table 2 shows the PSNR degradation versus channel error for UEP2 optimized at BER 0.1 and for the scheme of Streit and Hanzo [10]. In [10], QCIF images, Rayleigh-fading channels, 16-level quadrature amplitude modulation, and BCH codes were used. We show results for comparable signal to noise ratio (SNR) channel error and BER pairs.

[10]		Our algorithm		
SNR	BCH	BER	BCH	RCPC
15	-2	0.05	-1.48	-0.79
11	-9	0.06	-2.13	-0.96
10	-12	0.075	-5.4	-1.43

**Table 2.** Comparison of PSNR degradation in dB for the scheme in [10] (source rate 0.28 bpp, total rate 0.51 bpp) and UEP2 (source rate 0.21 bpp, total rate 0.5 bpp).

## 6. CONCLUSION

Although fractal image codes are very sensitive to bit errors, we showed that they can be efficiently protected by unequal error protection with RCPC channel codes. We proposed two rate-distortion unequal error protection algorithms that try to minimize the expectation of the reconstruction error subject to a constraint on the total number of protection bits. Because the partition bits should be perfectly protected, our technique is efficient only when these bits form a small amount of the total code.

Future work will include joint source-channel coding, which can improve our results considerably. For example, for the  $512 \times 512$  Lenna image and RCPC codes, the PSNR was 25.42 dB at source rate 0.21 bpp, total rate 0.46 bpp, and BER 0.1. But for the same total rate and BER, we obtained a PSNR of 27 dB for source rate 0.16 bpp by a slight adjustment of the optimal bit allocation found for 0.21 bpp.

## 7. APPENDIX

Let  $f(m, n)$  be the number of possible allocations of the bits of codeword  $b_{n-1} \dots b_1 b_0$  into  $m$  classes  $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_m$  such that for  $i < j$  bit  $b_i$  is not protected better than bit  $b_j$ . Then  $f(m, n) = \binom{m+n-1}{n}$ .

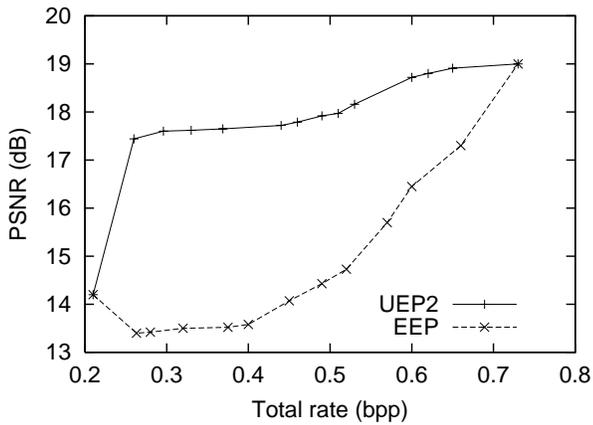
**Proof.** We prove the result by induction on  $n$ . We have  $f(m, 1) = m = \binom{m}{1}$ . We assume now that the result holds for  $n$  and prove it for  $n + 1$ . The total number of bit allocations is equal to the sum of the bit allocations where the most sensitive bit is fixed in one of the classes. Thus,

$$f(m, n + 1) = \sum_{k=1}^m f(k, n) = \sum_{k=1}^m \binom{k+n-1}{n} = \binom{m+n}{n+1}.$$

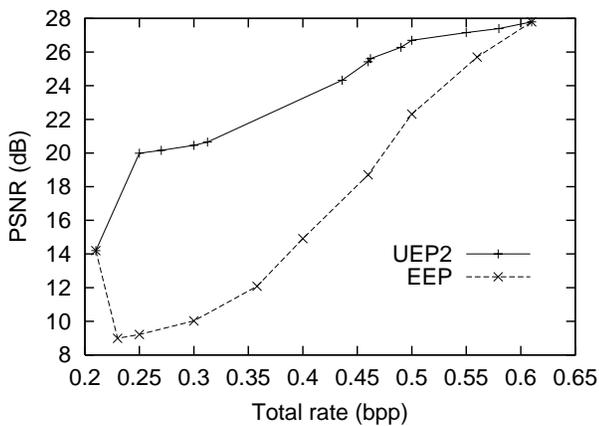
**Acknowledgment.** We thank Youssef Charfi for pointing out errors in the experimental results of our original paper. This work was supported by the Graduate Study Program “Knowledge Representation” and the grant Sa449/8-1, both of the Deutsche Forschungsgemeinschaft (DFG).

## 8. REFERENCES

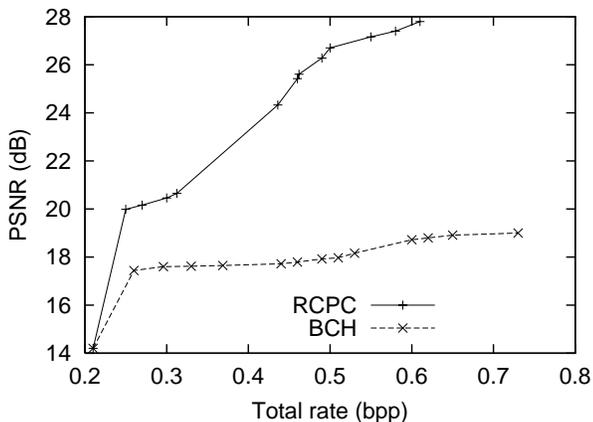
- [1] Clark, G. C., Cain, J. B., *Error-correction coding for digital communications*, Plenum Press, 1981.
- [2] Everett, H., *Generalized Lagrange multiplier method for solving problems of optimum allocation of resources*, Operations Research 11 (1963) 399–417.
- [3] Fisher, Y., *Fractal Image Compression — Theory and Application*, Springer-Verlag, New York, 1994.
- [4] Fisher, Y., *Fractal image compression with quadrees*, in: *Fractal Image Compression — Theory and Application*, Y. Fisher (ed.), Springer-Verlag, New York, 1994.
- [5] Hagenauer, J., *Rate-compatible punctured convolutional codes (RCPC codes) and their applications*, IEEE Trans. Comm. 36,4 (1988) 389–400.
- [6] Noh, Y. H., Kim, S., H., Kim, N. C., *Block loss recovery using fractal extrapolation for fractal coded images*, in: *Proc. IEEE ICIP-98*, Chicago, Oct. 1998.
- [7] Novak, M., *Transmission error robust fractal coding using a model-residual approach*, in: *Proc. DCC'98*, IEEE Comp. Soc. Press, pp. 349–358, 1998.
- [8] Øien, G. E., Lepsøy, S., *A class of fractal image coders with fast decoder convergence*, in: *Fractal Image Compression — Theory and Application*, Y. Fisher (ed.), Springer-Verlag, New York, 1994.
- [9] Saupe, D., Hamzaoui, R., *The Leipzig Paper Collection on Fractal Image Compression*, <ftp://shear.informatik.uni-leipzig.de/pub/Fractal/papers/pdf/README.html>
- [10] Streit, J., Hanzo, L., *A fractal video communicator*, in: *Proc. IEEE Veh. Tech. Conf.*, Stockholm, May 1994.



**Fig. 3.** PSNR vs. total rate for the  $512 \times 512$  Lenna image at source rate 0.21 bpp and BER 0.1. BCH channel codes were used. EEP corresponds to equal error protection.



**Fig. 4.** PSNR vs. total rate for the  $512 \times 512$  Lenna image at source rate 0.21 bpp and BER 0.1. RCPC codes were used. EEP corresponds to equal error protection.



**Fig. 5.** BCH vs. RCPC codes for UEP2 at various total bitrates for the  $512 \times 512$  Lenna image at source rate 0.21 bpp and BER 0.1.