EXPERIMENTS ON JOINT SOURCE-CHANNEL FRACTAL IMAGE CODING WITH UNEQUAL ERROR PROTECTION

Youssef Charfi,[*] *Vladimir Stanković,*[†] *Raouf Hamzaoui, Dietmar Saupe*

University of Leipzig
Department of Computer Science
Augustusplatz 10/11, D-04109 Leipzig, Germany
charfi,stankovi,hamzaoui,saupe@informatik.uni-leipzig.de

Ameur Haouari

University Mohammed V
LEESA, Faculty of Science
B.P. 1014, Rabat, Morocco
haouari@fsr.ac.ma

ABSTRACT

We propose a joint source-channel coding system for fractal image compression. We allocate the available total bit rate between the source code and a range of error-correcting codes using a Lagrange multiplier optimization technique. The principle of the proposed unequal error protection strategy is to partition the information bits into sensitivity classes and to assign one code from a range of error-correcting codes to each sensitivity class in a nearly optimal way. Experimental results show that joint source-channel coding with fractal image compression is feasible, leads to efficient protection strategies, and outperforms previous works in this field that only covered channel coding with a fixed source rate.

1. INTRODUCTION

Robust and efficient transmission of images over noisy channels has recently attracted a widespread interest because of the increasing popularity of the internet and wireless multimedia devices and services. According to Shannon's separation theorem, the source coder and the channel coder may be designed independently [1]. However, the theorem holds under asymptotic assumptions which are not fulfilled in practice [2]. This motivates the design of joint source-channel coders. Efficient solutions for the allocation of the total transmission rate between the source coder and the channel coder were proposed for many compression systems, including vector quantizers, DCT- and wavelet-based coders [3, 4, 5, 6]. In this paper, we consider the problem of joint source-channel coding for fractal image compression. Moreover, in fractal compression the bits of the code have unequal importance and hence we also take into account unequal error protection techniques. Rate-distortion optimization is applied to the source and all parts of the channel code.

Fractal image compression provides satisfactory rate-distortion results, fast decoding, and resolution independence. A fractal code is a binary representation of an image operator whose unique fixed point is close to the original image. The image operator is based on an image partition in blocks and an affine similarity between the blocks in the partition and other blocks from the same image. The decoder recovers the fixed point by iterating the image operator. Fractal codes are very sensitive to channel errors because errors propagate during the decoding. Although more than 650 papers have been dedicated to fractal image compression [7], only three considered transmission through noisy channels. The first work about error protection for fractal image codes is due to Streit and Hanzo [8]. They used a two-level error protection scheme with BCH codes [9] to send a fractal code over a Rayleigh-fading channel. The authors observed that unequal error protection (UEP) has a significantly better performance than equal error protection (EEP). However, the proposed UEP scheme was given for a poor coder based on uniform partitions, and the unequal error protection was not optimized. Novak [10] introduced a model-residual fractal coder and showed that it is less sensitive to errors than a standard fractal coder. Error protection was not considered in this work. Finally, Noh, Kim, and Kim [11] 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.

Recently, three authors of this paper (Stanković, Hamzaoui, Saupe) proposed a UEP technique that computes an allocation of the information bits into a set of protection classes [12]. The goal of the algorithm is to minimize the expectation of the reconstruction error for a given bit error rate (BER) in a binary symmetric channel (BSC) and for a fixed source rate subject to a constraint on the number of protection bits. The performance of the system at a target transmission rate can be improved significantly by determining the optimal corresponding source rate. Unfortunately, the optimal trade-off between the source rate and the channel rate is not obvious because of the great number of parameters in the fractal source coder and the UEP scheme.

This study continues and extends the discussion of [12]. We introduce an unequal error protection strategy for fractal image codes that improves our previous results without increasing the computing requirements. Moreover, we provide techniques for joint source-channel coding yielding far superior results than unequal error protection for fixed source rates.

This paper is organized as follows. In Section 2, we describe a generic fractal coder. Our main contribution is Section 3, in which we present a fast unequal error protection and joint source-channel coding system. We discuss the tradeoff betweeen source and channel rates and provide an analysis of the sensitivity of the individual bits in the fixed length codewords of the fractal code, upon which we base a UEP scheme. Our proposed technique efficiently yields a suboptimal source-channel code pair, and we compare its performance to corrresponding full search algorithms. In Section 4, we

[*] Support of the German Academic Exchange Service (DAAD) is gratefully acknowledged.

[†] Thanks to the Graduate College *Wissensrepräsentation,* University of Leipzig, of the Deutsche Forschungsgesellschaft (DFG) for funding.

give simulation results which show that our system outperforms all previous schemes where the source rate is fixed and only the channel rate is optimized. In Section 5 we discuss two further studies on the relation of source and channel distortion for fractal image coding.

2. FRACTAL IMAGE CODING

In fractal compression, the encoder finds a contractive image operator T whose fixed point f_T approximates the original image $f^\star$. 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. For example, in this paper the operator T is given by a quadtree 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} from $D_1, ..., D_{n_D} \subset \mathcal{I}$,
- an isometry of the square $I_{k_i} \in \{I_1, I_2, ..., I_8\}$,
- a *scaling factor* $s_{l_i} \in \{s_1, ..., s_{n_s}\} \subset (-1, 1)$,
- and an *offset* $o_{m_i} \in \{o_1, ..., o_{n_o}\} \subset \mathbb{R}$.

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

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

is the best l_2 approximation of $\mathbf{R}_i$. Here boldface capital letter like, e.g., $\mathbf{B}$ denote the array of pixel intensities of $f^\star$ in the corresponding subset $B \subset \mathcal{I}$, 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.

We used Fisher's quadtree coder [13], which we modified according to Øien's orthogonalization technique [14]. Thus, the approximation (1) was replaced by

$$\widehat{\mathbf{R}}_i = s_{l_i} I_{k_i} A_i(\mathbf{D}_{j_i} - \mu(\mathbf{D}_{j_i})\mathbf{1}) + o_{m_i}\mathbf{1}. \tag{2}$$

Here $\mu(\cdot)$ denotes the mean of a block. This yields a more robust code because in contrast to equation (1), errors in the scaling factor do not affect the mean of a reconstructed block.

Using the method of least squares, the optimal nonquantized scaling factor and offset associated to a domain D and an isometry I are

$$\begin{aligned} s &= \frac{|R_i|\langle C, \mathbf{R}_i\rangle - |R_i|^2\mu(\mathbf{D})\mu(\mathbf{R}_i)}{|R_i|\langle C, C\rangle - \mu(\mathbf{D})^2}, \\ o &= \mu(\mathbf{R}_i). \end{aligned}$$

Here $|R_i|$ is the number of pixels in R_i, $C = IA_i\mathbf{D}$, 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 decoder constructs $f_T = \lim_{k\to\infty} f^{(k)}$ by

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

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 $d = \|f^\star - f_T\|_2^2$ is called the *attractor error*.

The fractal code associated to T consists of bits for the quadtree and a codeword for each range block R in the partition. In the following, we set $n_s = 2^5$ and $n_o = 2^7$. For each range, the candidate domains consist of blocks that are twice the range size and whose upper-left pixels are located at coordinates (i, j), where $i \equiv 0$ (mod 4) and $j \equiv 0$ (mod 4). For an image of size 512 $\times$ 512 this leads to $n_D = 2^{14}$ domains. We denote the (binary) range codeword, consisting of 29 bits, by

$$\omega_R = s_4 \cdots s_0 o_6 \cdots o_0 i_2 \cdots i_0 d_{13} \cdots d_0. \tag{3}$$

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. For such a range the corresponding codeword would comprize of less than 29 bits. However, to ensure synchronization in our transmission system, we must send bits for the zero scaling factor, an arbitrary isometry, and a domain block address. Thus, all codewords have a length of 29 bits.

3. JOINT SOURCE-CHANNEL CODING SYSTEM

3.1. Tradeoff between source and channel rates

We consider the quadtree fractal coder of the previous section and assume that the header bits of the fractal code that include the definition of the quadtree image partition are perfectly protected. That is not a severe limitation since the header that constitutes less than 2/100 of the total code, can be perfectly protected with a negligible increase of the bitrate. Therefore we consider only unequal error protection of the n_R codewords $\omega_{R_i}, i = 1, ..., n_R$, see (3).

We define a finite set of channel code rates $\mathcal{R} = \{r_1, ..., r_l\}$. For example, one can use a range of rate-compatible punctured convolutional (RCPC) code rates [15]. A code of a fractal transform T is protected by a strategy S that assigns a code rate to each bit of ω_R. We assume that all ranges are protected with the same strategy. We denote the set of possible strategies by Ω and the set of available fractal transforms by $\mathcal{T}$. These sets Ω and $\mathcal{T}$ are finite sets.

Suppose that the original image $f^\star$ is encoded with a transform T and protected with a strategy S. Let $S(T)$ be the transform obtained after channel decoding. Then $D(T, S) = E\{\|f^\star - f_{S(T)}\|_2^2\}$ is the expected distortion due to both the source quantization and the channel noise. Let $R_s(T)$ be the source bit rate associated to T and $R_c(T, S)$ be the channel bit rate associated to the protection of T by S. An optimal joint source-channel coder finds a pair (T^*, S^*) that solves the constrained minimization problem

$$\min_{(T,S)\in\mathcal{T}\times\Omega} D(T, S) \ \text{ subject to } \ R_s(T) + R_c(T, S) \leq R_t,$$

where R_t is the available transmission rate. This constrained optimization problem can be simplified by converting it to the unconstrained problem

$$\min_{(T,S)\in\mathcal{T}\times\Omega} L(T, S, \lambda) \tag{4}$$

where

$$L(T, S, \lambda) = D(T, S) + \lambda(R_s(T) + R_c(T, S)) \tag{5}$$

and $\lambda \geq 0$ is a Lagrange multiplier [16]. In practice we choose a set $\mathcal{T}$ of fractal transforms yielding image codes with differing source rates and minimize $L(T, S, \lambda)$ for each $T \in \mathcal{T}$,

$$\min_{T\in\mathcal{T}} \left[\min_{S\in\Omega} L(T, S, \lambda)\right]. \tag{6}$$

Let us call this optimization *Algorithm JSCC-0*. To meet the constraint on the total rate, the bisection method can be applied requiring that the optimization be done for several values of the Lagrange multiplier λ.

By separating the two nested minimizations in (6) we obtain an algorithm that is faster than JSCC-0, which, however, sacrifices some quality for the sake of speed. For initialization we choose a medium strength equal error protection strategy, $S_0 = (r_{\lfloor l/2 \rfloor}, ..., r_{\lfloor l/2 \rfloor})$. Then a sequence of pairs $(T_k, S_k)_{k=1,...,k_{\max}}$ is obtained in $k_{\max}$ iterations as follows,

$$T_k = \arg\min_{T \in \mathcal{T}} L(T, S_{k-1}, \lambda), \quad (7)$$

$$S_k = \arg\min_{S \in \Omega} L(T_k, S, \lambda). \quad (8)$$

As final result we obtain the transform $T_{k_{\max}}$, the strategy $S_{k_{\max}}$, and the associated Lagrangian cost $L(T_{k_{\max}}, S_{k_{\max}}, \lambda)$. Let us call this modified procedure *Algorithm JSCC-1*.

Let $(T_k, S_k)_{k=1,2,...}$ be the sequence of fractal transforms and protection strategies produced in Algorithm JSCC-1. Then it is clear that the sequence $L(T_k, S_k, \lambda)_{k=1,2,...}$ is monotonically decreasing and converging.

The computing times are dominated by the simulation time required to evaluate the total distortion, t_D, for a choice of (T, S, λ). For JSCC-0 this amounts to a total time of $|\mathcal{T}|t_D$ and for JSCC-1 this is $k_{\max}t_D$.

Following the next subsection on bit sensitivity analysis we discuss the minimization of the Lagrangian cost $L(T, S, \lambda)$ for a fixed transform T which is required in (8). We will arrive at three unequal error protection algorithms with different qualities and complexities, which we call *Algorithms UEP-0, UEP-1,* and *UEP-2* all of which can be combined with JSCC-0 and JSCC-1.

3.2. Bit sensitivity analysis

In this subsection we study the importance of the bits of the codewords, which can be determined by corrupting the bits at the same position in the range codewords $\omega_{R_i},\ i = 1, \ldots, n_R$ with a certain bit error rate and computing the degradation of the PSNR in the reconstructed image. Figure 1 shows the resulting PSNR degradations for the 512×512 Lenna image at source rate 0.21 bpp and and a binary symmetric channel bit error rate (BER) of 10^{-1}. Each PSNR is averaged over 50 experiments with the same BER. The figure shows that the codeword bits have unequal importance. We obtained similar results for other BERs and other images at several source rates.

Figure 2 shows the PSNR reconstruction quality as a function of the BER when *all* the bits of only one fractal parameter (offset, scaling, isometry, domain address) were corrupted. Again, each PSNR value 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.

Each of the 29 bits in the codewords ω_{R_i} may be protected using a different channel rate. To reduce the complexity of the optimization we suggest to partition the 29 codeword bits into $m \leq 29$ sensitivity classes $C_1, ..., C_m$ and to protect all bits in a class C_k with the same channel rate $r_{j_k} \in \mathcal{R}$. The partition of the bits into the classes is based on the importance of the bits. For the clustering of the 29 codeword bits into m classes we used the k-means algorithm applied to the set of 29 values of PSNR degradation shown in Figure 1. The result for $m = 13$ is given in Table 1.

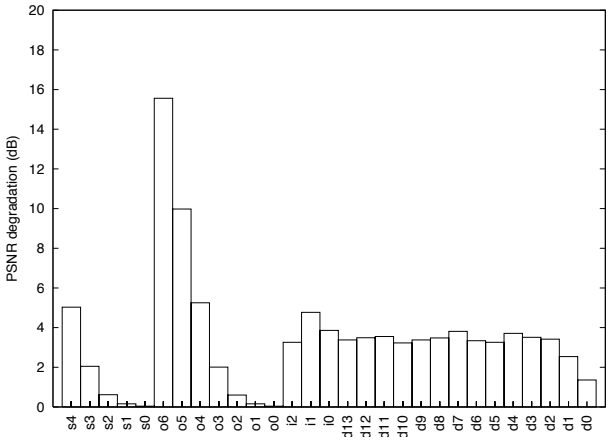

Fig. 1. PSNR degradation caused by the corruption of a single range codeword bit with probability 10^{-1} separately. The test image was the 512×512 Lenna image at source rate 0.21 bpp.

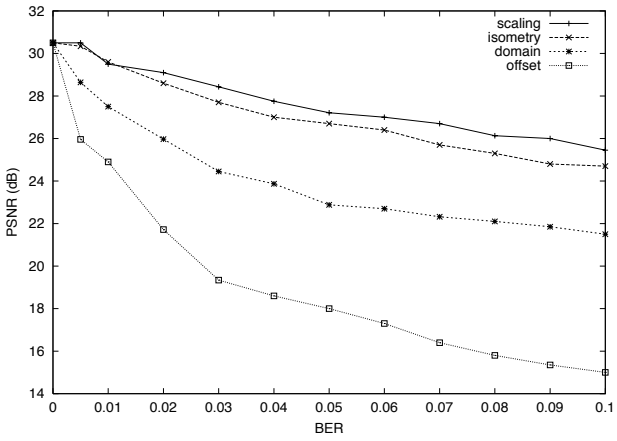

Fig. 2. PSNR vs. BER in all bits of a single fractal parameter for the Lenna image at source rate 0.21 bpp.

3.3. Unequal error protection of fractal image codes

We recall our goal, ie., to compute $\min_{S \in \Omega} L(T, S, \lambda)$ for fixed $T \in \mathcal{T}$ and $\lambda \geq 0$. Throughout this report we assume that the used error protection scheme has the property that increasing the number of protection bits increases the correction capability. In other words, the residual bit error rate is assumed to be a monotonically increasing function of the code rate.

In this subsection we present three methods for unequal error protection of fractal codes, in which codeword bits from the same

sensitivity classes	range codeword bits
C_1	o_6
C_2	o_5
C_3	o_4
C_4	s_4
C_5	i_1
C_6	i_0, d_4, d_7
C_7	$d_2, d_3, d_8, d_{10}, d_{11}, d_{12}, d_{13}$
C_8	i_2, d_5, d_6, d_9
C_9	d_1
C_{10}	o_3, s_3
C_{11}	d_0
C_{12}	s_2, o_2
C_{13}	s_0, s_1, o_0, o_1

Table 1. Partition of the range codeword bits into 13 sensitivity classes with decreasing sensitivity for a fractal code of the 512 × 512 image Lenna.

class C_k are equally protected. The channel rate $R_c(T, S)$ is given by

$$R_c(T,S) = n_R \sum_{k=1}^{m} \left(\frac{1 - r_{j_k}}{r_{j_k}} \right) n_k \tag{9}$$

where n_R denotes the number of range blocks of the image. For $k = 1, ..., m$ the number of bits of a range codeword assigned to the sensitivity class C_k is n_k. These bits are protected by the same code rate r_{j_k} in all codewords.

For $k = 1, 2, \ldots, m-1$ the bits in class C_k are more sensitive to channel errors than the bits in class C_{k+1}. Thus, it is reasonable to use a protection strategy such that $r_{j_k} \leq r_{j_{k+1}}$. With this convention we give the following definition.

Definition 1 *Let $\mathcal{R} = \{r_1, ..., r_l\}$ be the set of channel code rates. The set Ω of protection strategies is*

$$\Omega = \{(r_{j_1}, ..., r_{j_m}) \in \mathcal{R}^m \mid r_{j_1} \leq r_{j_2} \leq ... \leq r_{j_m}\}.$$

For convenience let us order the rates in $\mathcal{R}$ to be increasing, $r_1 < r_2 < \cdots < r_l$. Thus, the set Ω of protection strategies is

$$\Omega = \{(r_{j_1}, ..., r_{j_m}) \in \mathcal{R}^m \mid j_1 \leq j_2 \leq ... \leq j_m\}.$$

The requirement of increasing code rates in a strategy $S \in \Omega$ reduces the size of the space $\mathcal{R}^m$ of strategies from l^m to $f(m, l) = \binom{m+l-1}{m}$, as shown by the next lemma. Although $f(m, l)$ is still of order l^m the reduction of size is relevant in practice. For the example of $m = 13$ and $l = 6$, a practical setting used in our experiments, $f(m, l) = 8568$, which is less than a millionth of $l^m = 6^{13}$.

Lemma 1 *Let $A = \{1, 2, ..., l\}$. The number of m-tupels $(j_1, ..., j_m) \in A^m$ with $j_1 \leq \cdots \leq j_m$ is $f(m, l) = \binom{m+l-1}{m}$.*

Proof The proof is straightforward by induction on m [12].

We call the minimization $\min_{S \in \Omega} L(T, S, \lambda)$ by enumeration of all $f(m, l)$ (increasing) channel code rate allocations *Algorithm UEP-0*.

We propose to replace the enumeration in the minimization $\min_{S \in \Omega} L(T, S, \lambda)$ by a faster heuristic algorithm. For this purpose we introduce some more notation.

Definition 2 *Let m be the number of protection classes and $\mathcal{R} = \{r_1, ..., r_l\}$ the set of channel code rates. For $K = 1, ..., m$ and $r_i \in \mathcal{R}$ the set of protection strategies in Ω which prescribe to protect the first K classes $C_1, ..., C_K$ by the same rate r_i and the remaining classes by higher rates is denoted by*

$$\Omega_K(r_i) = \quad \{ \quad (r_{j_1}, ..., r_{j_m}) \in \Omega \mid \\ r_i = r_{j_1} = \cdots = r_{j_K} < r_{j_{K+1}} \leq \cdots \leq r_{j_m} \ \}.$$

Since the channel code rates are ordered by increasing numbers, $\Omega_K(r_i)$ is the set of tupels $(r_{j_1}, ..., r_{j_m}) \in \Omega$ with

$$i = j_1 = \cdots = j_K < j_{K+1} \leq \cdots \leq j_m.$$

If we replace the strict inequality $r_{j_K} < r_{j_{K+1}}$ by $r_{j_K} \leq r_{j_{K+1}}$ we obtain the set of protection strategies in Ω which prescribe to protect the classes $C_1, ..., C_K$ by rate r_i. In other words, this set of strategies is

$$\overline{\Omega}_K(r_i) = \bigcup_{k=K,...,m} \Omega_k(r_i).$$

We note that the strategy sets $\Omega_K(r_i)$ as well as the sets $\overline{\Omega}_1(r_i)$ make up a partition of the set Ω of all allowed strategies.

Lemma 2

$$\Omega = \bigcup_{\substack{i=1,...,l \\ K=1,...,m}} \Omega_K(r_i) = \bigcup_{i=1,...,l} \overline{\Omega}_1(r_i)$$

where $\Omega_K(r_i) \cap \Omega_{K'}(r_{i'}) = \emptyset$ if $K \neq K'$ or $i \neq i'$ and $\overline{\Omega}_1(r_i) \cap \overline{\Omega}_1(r_{i'}) = \emptyset$ if $i \neq i'$.

Also note that $\Omega_K(r_l) = \emptyset$ for $K = 1, ..., m-1$. This means that if source code bits from the most sensitive class C_1 are protected by the weakest channel code rate, i.e., by the highest rate $r_l \in \mathcal{R}$, then all bits from the remaining classes $C_2, ..., C_m$ can only be protected by the same channel code rate.

For each protection strategy $S = (r_{j_1}, ..., r_{j_m}) \in \overline{\Omega}_K(r_i)$, $i = 2, ..., l$, $K = 1, ..., m$ we can define a successor strategy $\mathrm{succ}(S)$ by putting the next highest protection on the first K classes, i.e., by replacing the first K rates r_i by the nearest lower rate r_{i-1}.

Definition 3 *For $i = 2, ..., l$, $K = 1, ..., m$ and $S = (r_{j_1}, ..., r_{j_m}) = (r_i, ..., r_i, r_{j_{K+1}}, ..., r_{j_m}) \in \overline{\Omega}_K(r_i)$, let*

$$\mathrm{succ}(S) = (r_{i-1}, ..., r_{i-1}, r_{j_{K+1}}, ..., r_{j_m}) \in \Omega_K(r_{i-1}).$$

We remark that the successor function yields a bijection between the sets $\overline{\Omega}_K(r_i)$ and $\Omega_K(r_{i-1})$ for all $i = 2, ..., l$, $K = 1, ..., m$.

Let us consider two protection strategies $S', S'' \in \overline{\Omega}_K(r_i)$ with $i \geq 2$. *Then we may assume for the Lagrangian costs that $L(T, S', \lambda) \geq L(T, S'', \lambda)$ if and only if $L(T, \mathrm{succ}(S'), \lambda) \geq L(T, \mathrm{succ}(S''), \lambda)$.* In the successors only the first K channel code rates r_i are replaced by r_{i-1}. It is reasonable to assume that the expected distortion $D(T, S)$ is decreased by about the same amount in both cases. Since also the rate changes by an (exactly) equal amount, we obtain our assumption. Our results in Section 4 empirically justify this assumption. If we considere a model in which distortion due to errors in different protection classes is additive, we can also prove our assumption.

Lemma 3 *Assume there exist distortion functions $D_{T,k}(r)$ for $k = 1, ..., m$, $r \in \mathcal{R}$, and $T \in \mathcal{T}$, such that the total distortion can be written as*

$$D(T, S) = D(T, (r_{j_1}, ..., r_{j_m})) = \sum_{k=1}^{m} D_k^T(r_{j_k}).$$

Then, for $K = 1, ..., m$, $r_i \in \mathcal{R}\backslash\{r_1\}$, $\lambda \geq 0$ and any two given protection strategies $S' = (r'_{j_1}, ..., r'_{j_m}) \in \overline{\Omega}_K(r_i)$ and $S'' = (r''_{j_1}, ..., r''_{j_m}) \in \overline{\Omega}_K(r_i)$ we have that $L(T, S', \lambda) \geq L(T, S'', \lambda)$ if and only if $L(T, \mathrm{succ}(S'), \lambda) \geq L(T, \mathrm{succ}(S''), \lambda)$.

Proof Let $K \in \{1, ..., m\}$ and $\lambda \geq 0$. For $i = 1, ..., l$ let

$$L_i^T = \lambda R_s(T) + \sum_{k=1}^{K} \left[D_k^T(r_i) + \lambda n_R \left(\frac{1 - r_i}{r_i} \right) n_k \right].$$

Then, by construction, for $S = (r_{j_1}, ..., r_{j_m}) \in \overline{\Omega}_K(r_i)$,

$$L(T, S, \lambda) = L_i^T + \sum_{k=K+1}^{m} \left[D_k^T(r_{j_k}) + \lambda n_R \left(\frac{1 - r_{j_k}}{r_{j_k}} \right) n_k \right].$$

Therefore, for $S', S'' \in \overline{\Omega}_K(r_i)$ with $i > 1$ we obtain

$$L(T, \mathrm{succ}(S'), \lambda) = L(T, S', \lambda) - L_i^T + L_{i-1}^T,$$
$$L(T, \mathrm{succ}(S''), \lambda) = L(T, S'', \lambda) - L_i^T + L_{i-1}^T.$$

The proof of the lemma is completed by subtracting these two equations.

For $K = 1, ..., m$ and $i = 1, ..., l$ let us denote by $S_K(r_i)$, $\overline{S}_K(r_i)$ and $L_K(r_i, \lambda)$, $\overline{L}_K(r_i, \lambda)$ optimal protection strategies and corresponding Lagrangian costs,

$$\begin{aligned} S_K(r_i) &= \arg \min_{S \in \Omega_K(r_i)} L(T, S, \lambda), \\ L_K(r_i, \lambda) &= L(T, S_K(r_i), \lambda), \end{aligned}$$

and, correspondingly,

$$\begin{aligned} \overline{S}_K(r_i) &= \arg \min_{S \in \overline{\Omega}_K(r_i)} L(T, S, \lambda), \\ \overline{L}_K(r_i, \lambda) &= L(T, \overline{S}_K(r_i), \lambda). \end{aligned}$$

From the definition of $\overline{\Omega}_K(r_i)$ we conclude that with $k^\star = \arg\min_{k=K,...,m} L_k(r_i, \lambda)$ we have

$$\overline{S}_K(r_i) = S_{k^\star}(r_i), \quad \overline{L}_K(r_i, \lambda) = L_{k^\star}(r_i, \lambda).$$

Thus, we can compute an optimal strategy in $\overline{\Omega}_K(r_i)$ from those in $\Omega_K(r_i)$, $i = K, ..., m$.

Using the basic heuristic assumption from above we can obtain the best strategy in $\Omega_K(r_i)$ from that in $\overline{\Omega}_K(r_{i+1})$ simply by replacing the first K rates r_{i+1} by r_i, i.e., by taking the successor of $\overline{S}_K(r_{i+1})$.

The algorithm for computing all optimal protection strategies $S_K(r_i)$ and $\overline{S}_K(r_i)$ therefore may proceed in stages $i = l - 1, l - 2, ..., 1$. After initializing the trivial optimal strategies $\overline{S}_K(r_l) = (r_l, ..., r_l)$ for $K = 1, ..., m$ one may write down in stage i strategies $S_K(r_i)$ as $\mathrm{succ}(\overline{S}_K(r_{i+1}))$ and $\overline{S}_K(r_i) = S_{k^\star}(r_i)$ with $k^\star$ as defined above. Finally, the overall optimal strategy is the best one of $\overline{S}_1(r_i), i = 1, ..., l$. We call this procedure *Algorithm UEP-1*. The pseudo code in Table 2 presents an implementation. The number of evaluations of Lagrangian costs is $m(l - 1) + 1$, which compares favorably with $f(m, l)$.

Initialization
`for` $K := 1, ..., m \quad \overline{S}_K(r_l) := (r_l, ..., r_l)$
Iteration
`for` $i := l - 1, l - 2, ..., 1$
  `for` $K := 1, ..., m$ // *compute* $S_K(r_i)$
    $S_K(r_i) := \mathrm{succ}(\overline{S}_K(r_{i+1}))$
    $L_K(r_i, \lambda) := L(T, S_K(r_i), \lambda)$
  $\overline{S}_m(r_i) := S_m(r_i)$ // *compute* $\overline{S}_K(r_i)$
  $\overline{L}_m(r_i, \lambda)) := L_m(r_i, \lambda))$
  `for` $K := m - 1, ..., 1$
    $\overline{S}_K(r_i) := S_K(r_i)$
    $\overline{L}_K(r_i, \lambda)) := L_K(r_i, \lambda))$
    `if` $L_K(r_i, \lambda)) > \overline{L}_{K+1}(r_i, \lambda))$
      $\overline{S}_K(r_i) := \overline{S}_{K+1}(r_i)$
      $\overline{L}_K(r_i, \lambda)) := \overline{L}_{K+1}(r_i, \lambda))$
Result
$i^\star := 1$
`for` $i := 2, ..., l$
  `if` $\overline{L}_1(r_i, \lambda)) < \overline{L}_1(r_{i^\star, \lambda)})$ $i^\star := i$
$S^\star := \overline{S}_1(r_i^\star)$ // *optimal strategy*
$L^\star := \overline{L}_1(r_i^\star, \lambda))$ // *minimal Lagrangian cost*

Table 2. Pseudo code for Algorithm UEP-1, computing an optimal protection strategy for a given code of the fractal transform T and Lagrange parameter $\lambda \geq 0$.

An alternative to Algorithm UEP-1 was given in [12]. It is based on the assumption in Lemma 3, minimizing $L(T, S, \lambda)$ over arbitrary rate allocations $S \in \mathcal{R}^m$. Thus, with the ansatz

$$L(T, S, \lambda) = \sum_{k=1}^{m} D_k^T(r_{j_k}) + \lambda n_R \left(\frac{1 - r_{j_k}}{r_{j_k}} \right) n_k$$

the algorithm computes $S = (r_{j_1}, ..., r_{j_m}) \in \mathcal{R}^m$ by

$$r_{j_k} = \arg \min_{r_i \in \mathcal{R}} D_k^T(r_i) + \lambda n_R \left(\frac{1 - r_i}{r_i} \right) n_k$$

for $k = 1, ..., m$. We call this method *Algorithm UEP-2*. (It was named UEP2 in [12]). It requires ml simulations for the evaluation of distortions. Thus, the complexity is similar to that of Algorithm UEP-1.

The algorithms UEP-0 to UEP-2 are designed to minimize $L(T, S, \lambda)$ for fixed λ and T over all strategies $S \in \Omega$ as needed in the joint source-channel coding method JSCC-1.

4. RESULTS

We consider the fractal coder described in Section 2 as a source coder. The 29 information bits of the range codewords were partitioned in $m = 13$ sensitivity classes, see Table 1.

Our methods for UEP and joint source-channel fractal image codes works independently of the choice of channel coder. For channel coding we used RCPC codes [15]. We chose RCPC codes because changing their rates is very simple and the same Viterbi

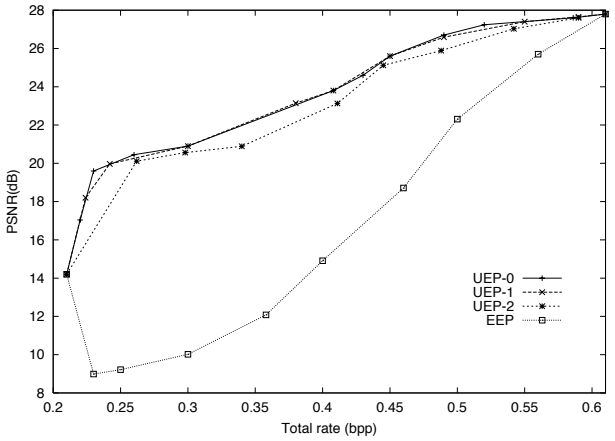

Fig. 3. Results for the 512×512 Lenna image at source rate 0.21 bpp and BER of 10^{-1}. The curve for EEP dips below the PSNR of about 14 dB achieved without error protection (total rate 0.21 bpp) at the first four points corresponding to code rates 8/9, 8/10, 8/12, 8/14. This is explained by the fact that for these code rates the residual bit error rate is higher than the channel BER, see Table 3.

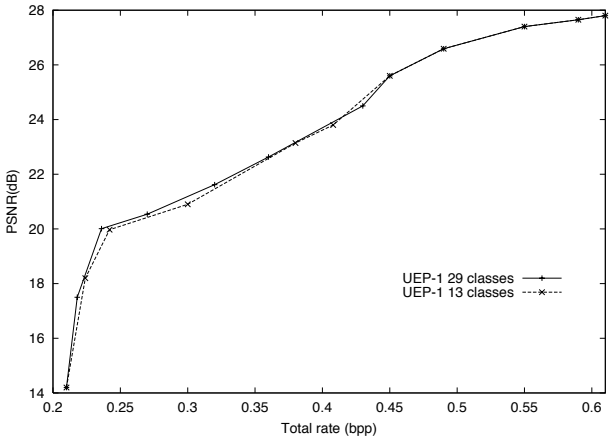

Fig. 4. Comparison of error protection performance at fixed source rate 0.21 bpp, BER 10^{-1}, and using 13 resp. 29 protection classes.

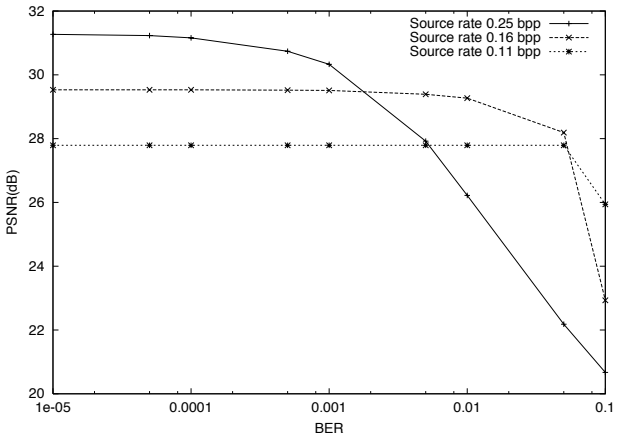

Fig. 5. PSNR vs. BER for the 512×512 Lenna image at total bit rate 0.31 bpp. Results for UEP-1 at various source rates are shown.

decoder can be used for all code rates which makes the implementation of a UEP scheme easy. The range of RCPC codes used is described by the convolutional mother code of rate $\frac{1}{N} = \frac{1}{3}$ and memory $M = 6$. Together with N, the puncturing period $P = 8$ determines the range of code rates: $R = \frac{p}{p+l}, l = 1, \ldots, (N-1)p$, from which we retained the rates $\frac{8}{24}, \frac{8}{22}, \frac{8}{20}, \frac{8}{18}, \frac{8}{16}, \frac{8}{14}, \frac{8}{12}, \frac{8}{10}, \frac{8}{9}$, and 1. For these rates we experimentally determined the residual bit error rates and collected them in Table 3.

Error-correcting code rates whose residual error probabilities were greater than the BER of the channel were removed. Also, all rates were removed that yielded a zero residual error probability except for the highest one of them. Thus, the set of used rates was $\mathcal{R} = \{\frac{8}{24}, \frac{8}{22}, \frac{8}{20}, \frac{8}{18}, \frac{8}{16}, 1\}$ with $l = 6$ for BER 10^{-1}, and $\mathcal{R}_c = \{\frac{8}{14}, \frac{8}{12}, \frac{8}{10}, 1\}$ with $l = 4$ for BER 10^{-2}.

In the first experiment we considered the 512×512 Lenna image at source rate of 0.21 bpp and BER of 10^{-1}. The reconstruction errors were estimated in the optimization using averaged mean square collage error over 50 simulations each. Using the collage error instead of the original reconstruction error is not a limitation of the method. Many tests showed that when using true reconstruction errors in the optimization the improvements of the resulting source-channel code were less than 0.02 dB of PSNR.

code rate	BER 10^{-1}	BER 10^{-2}
8/9	0.49	0.045
8/10	0.46	0.0044
8/12	0.35	0.000065
8/14	0.20	0.0
8/16	0.09	0.0
8/18	0.036	0.0
8/20	0.014	0.0
8/22	0.0056	0.0
8/24	0.0027	0.0

Table 3. Residual bit error rates for RCPC codes, estimated using an information sequence of 4 million random bits for each measurement.

Figure 3 shows the performance of equal error protection (EEP), where for each point the same rate from the total set of 10 retained rates was used for protection of all bits. This is compared with the performance of the three proposed UEP algorithms UEP-0, UEP-1, and UEP-2.

The results showed that unequal error protection performed much better than equal error protection. Of the three UEP schemes the last one, UEP-2, gave significantly worse results than UEP-0 and UEP-1, whereas the difference between UEP-0 and UEP-1 was negligible. Moreover, the CPU time was 8113.6s for UEP-0, and 63.7s for UEP-1. This is in line with our theoretical complexity estimation. For $m = 13$ and $l = 6$ we have to consider $f(m, l) = 8568$ rate allocations in UEP-0, while in UEP-1 the number is $m(l-1)+1 = 66$, which is 0.77% of 8568. This percentage agrees almost exactly with the ratio of our timings. The CPU times were measured on a 270 MHz MIPS R12000 processor

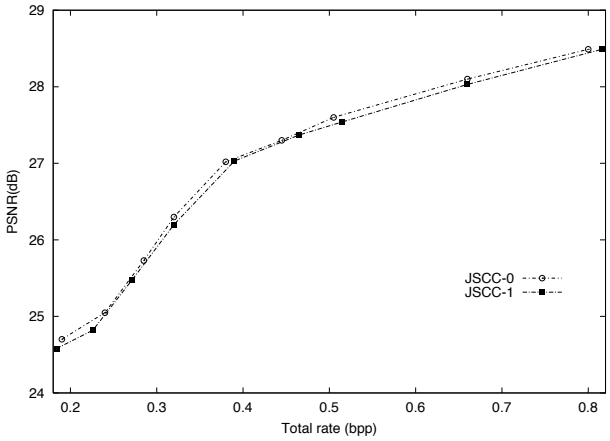

Fig. 6. Results for the 512×512 Lenna image at BER 10^{-1}. The two curves are for joint source-channel coding.

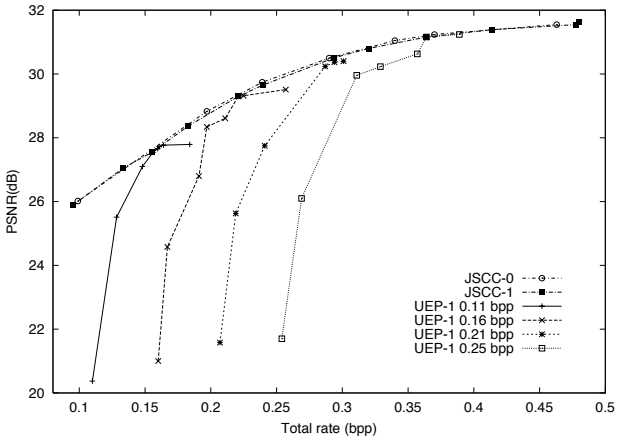

Fig. 7. Results for the 512×512 Lenna image at BER 10^{-2}. The two top curves are for joint source-channel coding. The four bottom curves are for encodings with fixed source rates and varying channel rates.

of a SGI Origin200 server with main memory size of 1.5 Gbytes. We conclude that the algorithm UEP-1 gave the best performance when considering both PSNR and time. Therefore, in all following tests we used only UEP-1.

We remark that the computing times can safely be reduced by half simply by using only 25 simulations for the distortion estimates instead of 50. We observed a maximal loss in PSNR of no more than 0.07 dB.

Our next experiment showed that the restriction to $m = 13$ classes did not significantly reduce the quality of the error protected codes. Again we set the source rate to 0.21 bpp and used a BER of 10^{-1}. With the maximum of 29 sensitivity classes only marginally better PSNR values were obtained, see Figure 4. Therefore, we continued further experiments only with $m = 13$.

Figure 5 shows the result of an experiment using three encodings with source rates 0.11, 0.16, and 0.25 bpp, and with error protection chosen such that the same total rate of 0.31 bpp was achieved. The optimization of the rate allocation was done for BER 10^{-1}. As the bit error rate was increased from 10^{-5} to 10^{-1} the performance of the three encodings deteriorated in different ways. Of course, encodings with higher source rates performed better than with lower source rates under nearly error-free conditions. But, the degradation in performance for encodings with higher source rates occurred much more rapidly than for encodings with lower source rates when the BER increases. Notice that for the source rate 0.11 bpp, there was no significant degradation in performance for BERs up to $5 \cdot 10^{-2}$. Even for the maximal BER of 10^{-1}, the drop in performance for the source rate 0.11 bpp was less than 2 dB while it was more than 10 dB for the source rate 0.25 bpp. This can be explained by the fact that for higher source rates more bits are exposed to channel errors, while only weak protection is allowed.

In the previous simulations the source rate was fixed and the optimal allocation of channel code rates for the data in the m sensitivity classes was searched under the constraint on the channel bit rate. However, joint source-channel coding may enhance the performance of our system. We present results for the proposed algorithms JSCC-0 and the less complex algorithm JSCC-1 for rate allocation using Lagrangian optimization and UEP-1 as the core algorithm. We used a range of 20 source rates from 0.07 bpp to 0.45 bpp obtained by different quadtree partitions. Thus, the set $\mathcal{T}$ consisted of 20 fractal codes. For Algorithm JSCC-1 we found convergence already after $k_{\max} = 3$ iterations. Results for test image Lenna at BER 10^{-1} are shown in Figure 6. The reconstruction quality for the codes from the fast algorithm JSCC-1 was only slightly worse than that of JSCC-0, up to about 0.1 dB in PSNR.

For the smaller bit error rate 10^{-2} the performance gap between JSCC-0 and JSCC-1 was even smaller, see Figure 7. The figure also displays curves for the performance of encodings with fixed source rates from 0.11 to 0.25 bpp. These findings again demonstrate the importance of joint source-channel coding as well as the efficiency of our proposed algorithms.

5. SOURCE AND CHANNEL DISTORTION FOR FRACTAL IMAGE CODING

In this section we report on further studies on the relation of source and channel distortion for fractal image coding. For a memoryless binary symmetric channel the channel errors are independent of the source parameters. Therefore, the expected distortion $D(T, S)$ can be approximated by the sum of source and channel distortion, $D_s(T) + D_c(T, S)$, where $D_s(T) = \|f^\star - f_T\|_2^2$ and $D_c(T, S) = E\{\|f_T - f_{S(T)}\|_2^2\}$. The following experiment illustrates the quality of this approximation. The test image was the 512 $\times$ 512 Lenna image and the bit error rate (BER) of the BSC was 10^{-1}. The channel coder was an RCPC coder with code rates $\{\frac{8}{24}, \frac{8}{22}, \frac{8}{20}, \frac{8}{18}, \frac{8}{16}, 1\}$. We randomly picked 100 pairs $(T, S) \in \mathcal{T} \times \Omega$ and computed the relative difference

$$\Delta = \frac{D(T, S) - [D_s(T) + D_c(T, S)]}{D(T, S)}.$$

The histogram in Figure 9 and the small root-mean-square value of Δ of only 0.0053 confirms our approximation.

The channel distortion $D_c(T, S)$ does depend on the fractal transform $T \in \mathcal{T}$ to some extent. However, experiments show that the channel distortion depends on the fractal transform T only in a very mild form. This can be seen in Figure 11, which shows the channel distortion as a function of the source rate.

	left	right	noiseless
BER	0.10	0.10	0.00
R_s (bpp)	0.07	0.07	0.07
R_t (bpp)	0.07	0.19	0.07
PSNR (dB)	12.82	24.70	26.07

	left	right	noiseless
BER	0.10	0.10	0.00
R_s (bpp)	0.28	0.28	0.28
R_t (bpp)	0.28	0.80	0.28
PSNR (dB)	14.11	28.49	31.43

Fig. 8. 512×512 Lenna image at different source rates R_s, transmission rates R_t, and bit error rate 10^{-1}, with and without joint source-channel coding using unequal error protection. The left column images are without error protection ($R_t = R_s$). PSNR results are also given for transmission through a noiseless channel.

	left	right	noiseless
BER	0.01	0.01	0.00
R_s (bpp)	0.09	0.09	0.09
R_t (bpp)	0.09	0.13	0.09
PSNR (dB)	20.11	27.04	27.16

	left	right	noiseless
BER	0.01	0.01	0.00
R_s (bpp)	0.25	0.25	0.25
R_t (bpp)	0.25	0.37	0.25
PSNR (dB)	21.56	31.24	31.29

Fig. 10. Results as in Figure 8 for BER 0.01.

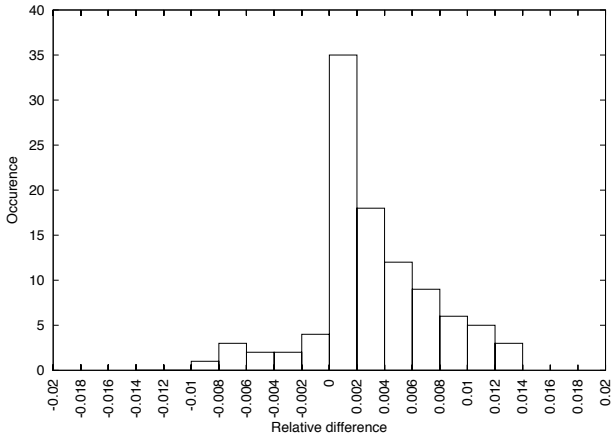

Fig. 9. Histogram of the relative difference between the sum of source and channel distortion $D_s(T) + D_c(T, S)$ and the overall distortion $D(T, S)$ obtained from 100 random measurements.

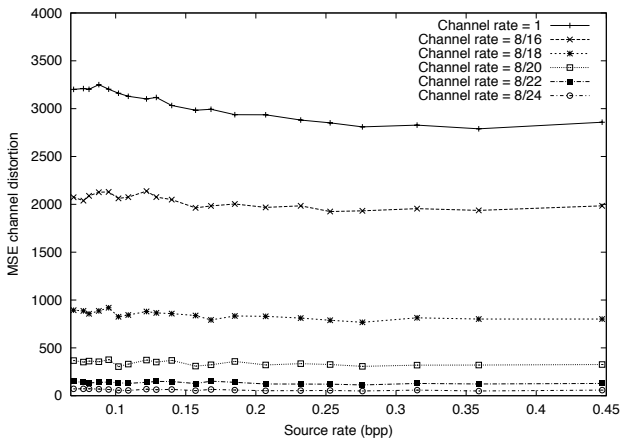

Fig. 11. MSE channel distortion versus source rate for the 512×512 Lenna image at various channel rates and equal error protection for a BSC with a BER of 10^{-1}.

6. CONCLUSIONS

We developed and studied a set of algorithms for unequal error protection of fractal image codes in the context of joint source-channel coding. The proposed methods allocate the total transmission rate between the source code and a range of error-correcting codes in a nearly optimal way. We conclude that in practice the combination of our algorithm JSCC-1 with UEP-1 at the core gives the best performance when considering both expected image reconstruction quality and time complexity. The approach is not limited to fractal coders. It can be modified for other source coders that output fixed length code words such as some types of vector quantizers. Many simulations using a quadtree-based fractal encoder and a range of RCPC codes showed that our system is efficient. The results proved that joint source-channel coding with fractal image coding is feasible, leads to efficient protection strategies, and outperforms previous works in this field that only covered channel coding with a fixed source rate.

7. REFERENCES

[1] Shannon C. E., "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and 623–656, 1948.

[2] Viterbi A. J. and Omura J. K., *Principles of Digital Communications and Coding*, McGraw-Hill, New York, 1979.

[3] Modestino J. W. and Daut D. G., "Combined source-channel coding of images," *IEEE Transactions on Communications*, vol. COM-27, pp. 1644–1659, 1979.

[4] Modestino J. W., Daut D. G., and Vickers A. L., "Combined source-channel coding of images using the block cosine Transform," *IEEE Transactions on Communications*, vol. COM-29, pp. 1261–1274, 1981.

[5] Hochwald B. and Zeger K., "Tradeoff between source and channel coding," *IEEE Transactions on Information Theory*, vol. IT-43, pp. 1412–1424, 1997.

[6] Ruf M. J. and Modestino J. W., "Operational rate-distortion performance for joint source and channel coding of images," *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 305–320, 1999.

[7] Saupe D. and Hamzaoui R., "The Leipzig Paper Collection on Fractal Image Compression," ftp://shear.informatik.uni-leipzig.de/pub/Fractal/papers/pdf/README.html.

[8] Streit J. and Hanzo L., "A fractal video communicator," in *Proc. IEEE Veh. Tech. Conf., Stockholm*, 1994.

[9] Clark G. C. and Cain J. B., *Error-correction coding for digital communications*, Plenum Press, 1981.

[10] Novak M., "Transmission error robust fractal coding using a model-residual approach," in *Proc. DCC-98, IEEE Comp. Soc. Press*, 1998, pp. 349–358.

[11] Noh Y. H., Kim S. H., and Kim N. C., "Block loss recovery using fractal extrapolation for fractal coded images," in *Proc. IEEE ICIP-98, Chicago*, 1998.

[12] Stankovic V., Hamzaoui R., and Saupe D., "Rate-distortion unequal error protection for fractal image codes," in *Proc. IEEE ICIP-2001, Thessaloniki*, 2001.

[13] Fisher Y., *Fractal Image Compression — Theory and Application*, Springer-Verlag, New York, 1994.

[14] Øien G. E. and Lepsøy S., "A class of fractal image coders with fast decoder convergence," in *Fractal Image Compression – Theory and Application*, Y. Fisher, Ed. Springer-Verlag, 1994.

[15] Hagenauer J., "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Comm.*, vol. 36,4, pp. 389–400, 1988.

[16] Everett H., "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, pp. 399–417, 1963.