Optimized Transmission of JPEG2000 Streams Over Wireless Channels

Nikolaos Thomos, *Student Member, IEEE*, Nikolaos V. Boulgouris, *Member, IEEE*, and Michael G. Strintzis, *Fellow, IEEE*

Abstract—The transmission of JPEG2000 images over wireless channels is examined using reorganization of the compressed images into error-resilient, product-coded streams. The product-code consists of Turbo-codes and Reed–Solomon codes which are optimized using an iterative process. The generation of the stream to be transmitted is performed directly using compressed JPEG2000 streams. The resulting scheme is tested for the transmission of compressed JPEG2000 images over wireless channels and is shown to outperform other algorithms which were recently proposed for the wireless transmission of images.

Index Terms—Channel coding, error-resilience, JPEG2000, turbo codes, unequal error protection (UEP).

I. INTRODUCTION

A VARIETY of error-resilient techniques for image transmission have been recently proposed in the literature. Most are based on the state-of-the-art SPIHT [2] source coder which generates embedded bitstreams, i.e., streams in which lower rates are prefixes of higher rates. However, the use of SPIHT appears to have some disadvantages due to its zerotree-based coding methodology. For this reason, other source coding schemes often appear to be preferable [1], [3]. The specific application of image transmission over wireless channels has deservedly attracted much attention since it requires not only careful design of the coding methodology for the compression of images, but also appropriate selection of the set of channel codes for effective forward-error correction (FEC).

In [4], the protection of SPIHT streams was achieved using product codes that consist of concatenated RCPC/CRC codes [5] and Reed–Solomon codes [6]. The resulting scheme was tested for image transmission over wireless and memoryless channels. In [7], a multiple description product code was introduced for the transmission of images over fading channels. The product codes used consisted of an RCPC/CRC concatenated channel code applied on a row-wise basis and a source-channel column based on the SPIHT coder and Reed–Solomon

erasure-correction codes. An algorithm for unequal error protection (UEP) was also proposed based on ideas initially explored in [8]. In [9] and [10], a methodology based on Turbo-codes was proposed. The resulting scheme was shown to outperform the method in [7] for image transmission over wireless channels.

In [11], the SPIHT streams were modified for the robust transmission of images over varying channels. Specifically, the embedded zerotree stream was reordered into packets containing a small set of wavelet coefficient trees. The techniques in [12] were used for the channel coding of the modified source streams. The resulting coding methodology was tested for the transmission of images over channels which suffer bit errors as well as packet erasures.

In [13], a real-time optimization algorithm was presented for the transmission of independently decodable packet streams over varying channels. The system utilizes the packetization scheme presented in [14]. In [15], the system of [13] was improved by replacing the ad-hoc selection of RS and RCPC codes in [4] with an EEP algorithm for fast allocation. A more complete presentation of these works appeared recently in [16].

In [17], a general framework was presented for image transmission over packet-erasure network. The presented methodology takes into consideration the dependencies between information in the compressed stream in order to cluster dependent layers and protect them according to their importance.

The system proposed in the present paper is based on the JPEG2000 coder, which is able to generate error-resilient streams. The considered transmission scenarios are over wireless channels producing bit errors according to the Rayleigh distribution. The JPEG2000 coder is used in conjunction with the application of a product code consisting of Turbo codes [18] and Reed–Solomon codes. Due to the systematic form of Turbo codes, the immediate extraction and decoding of source information from the channel-coded stream is possible, if the stream is not corrupted. Whenever the stream is corrupted, the product codes will correct several errors. Uncorrectable errors are localized and the corrupted portion of the stream is discarded.

The optimal allocation of Reed–Solomon symbols is also examined in the present paper and an algorithm for efficient UEP is proposed. The UEP algorithm is based on the formulation of channel packets of constant size, i.e., packets in which the source bytes vary but the sum of source and channel bytes is fixed. This approach admits a fast dynamic programming solution. The resulting robust transmission system is evaluated and is shown to outperform the best-performing known schemes for the transmission of images over wireless channels.

The paper is arranged as follows. In Section II, the error-resilient packetization of JPEG2000 streams is described. A
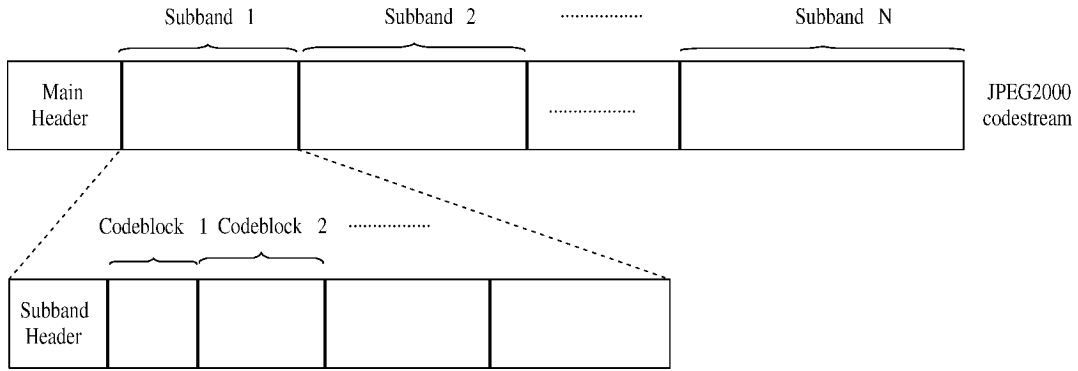
Fig. 1.   Modified JPEG2000 stream for the transmission of image over wireless channels.

channel rate allocation technique is described in Section III. An additional process for source rate reallocation is presented in Section IV. In Section V, experimental results are reported. Conclusions are drawn in Section VI.

II. ERROR-RESILIENT CODING IN THE WAVELET DOMAIN

A. JPEG2000 Coder

The JPEG2000 image coder is free of the disadvantages of the SPIHT coder. Unlike SPIHT, which uses hierarchical tree structures for the coding of wavelet coefficients, JPEG2000 is based on independent block coding of wavelet coefficients [19]. The JPEG2000 bitstream is composed by a succession of layers corresponding to codeblocks which are independent, in that their decoding does not require prior decoding of other codeblocks.

In order to achieve an error-resilient bitstream packetization, the initial JPEG2000 bitstream organization was modified so that each packet in the product-code contains information from only one subband and has its own separate header. To this end, the start of packet (SOP) marker [20] was appropriately modified in order to indicate the start of the new packet formation. Furthermore, only one quality layer was considered. The modified stream is schematically depicted in Fig. 1. It should be noted that after transmission, the decoder will reassemble the modified stream into a fully compatible JPEG2000 stream.

Since the transmission scenario of images is over wireless channels, which may have large number of errors concentrated in small bitstream segments, the initial header containing the specifications of the JPEG2000 coder is assumed to be fixed, and, thus, it need not be transmitted. The reason behind this choice is that this information should be correctly received prior to the decoding process and because it is very difficult for any forward error correction scheme to guarantee the recovery of this packet with a reasonable amount of channel protection.

Although the JPEG2000 coder offers excellent image quality while achieving high compression ratios, the generated bitstream is vulnerable to errors and a few bit errors may render the stream undecodable. The robustness of the JPEG2000 bitstream can be improved by activating the error resilient tools offered by the JPEG2000 coder. Specifically, using arithmetic coding termination at the end of each coding pass, the bit errors are localized and the decoding process continues with the unaffected code-blocks. Moreover, the arithmetic coder is initialized at the start of each code-block coding pass. The above error resilient tools, provided by JPEG2000, allow independent coding/decoding of the resulting code-blocks. In addition to the exploitation of the above features, we modified the MQ decoder in order to be able to partially decode the first corrupted coding pass. This modification further improves the efficiency of the overall system without sacrificing JPEG2000 compliance.

B. Independent Source Ordering

One of our primary goals during the design and implementation of the proposed system was the transmission of JPEG2000 streams in such a way so that the corrupted portion of the stream can be discarded and the rest of the information can be decodable. For this reason, we propose the division of the wavelet coefficients to be transmitted into N disjoint sets J_n, $n = 1, \ldots, N$ in the wavelet domain so that

$$J_{n_1} \bigcap J_{n_2} = \emptyset, \quad \text{for} \quad n_1 \neq n_2, \quad \text{and} \quad \bigcup_n J_n = \mathcal{W}$$

where $\mathcal{W}$ is the set including *all* coefficients of the wavelet representation. If the disjoint sets of coefficients are channel-coded appropriately into channel packets, then the erasure of a packet during transmission will not prevent the uncorrupted information from being decoded. Although numerous combinations of disjoint sets of coefficients can be conceived, in practice, since blockwise coding is performed, the subbands of the wavelet decomposition were chosen in the present paper as a reasonable compromise (see Fig. 2) between coding efficiency and information decoupling.

C. Forward Error Correction

For the protection of the above independent information packets, we use a product code, i.e., a code which generates parity bits for data arrays in both horizontal and vertical directions (across rows and columns of the array, respectively). The row code consists of a cyclic redundancy code (CRC) combined with a systematic Turbo code [21] (see Fig. 3), while the column code is an erasure-correction shortened systematic Reed–Solomon code [6]. Reed–Solomon codes are effective when errors occur in bursts and their location in the transmitted symbol sequence is known. They are denoted by a pair (N, m), where m is number of source symbols and N is the total number of symbols in a codeword (N coincides with the number of packets in the product code array). When the position of the
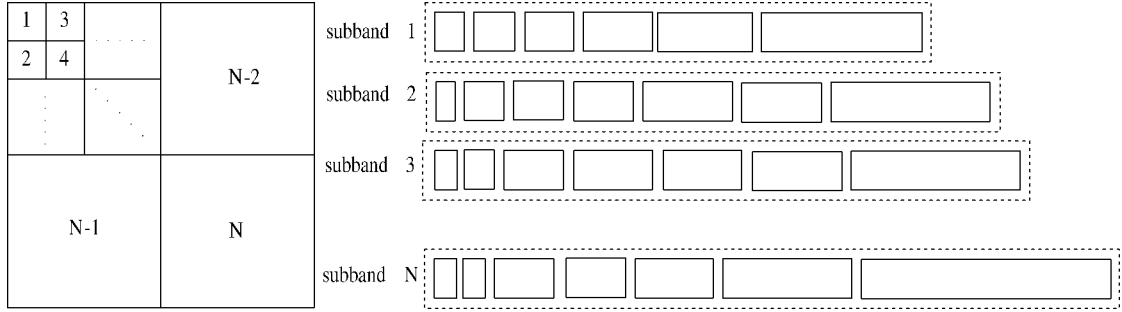
Fig. 2.   Each subband forms a group of layers that can be independently protected and decoded. A five-level decomposition is assumed.

Fig. 3.   Arrangement of data in a Turbo-coded row. Unlike in RCPC/CRC rows, data bits can be directly decoded if they are not corrupted.

erasures is known, the error correction capability of these codes is $N - m$.

A different amount of protection, provided by Reed–Solomon codes, is allocated to each layer. Some overhead information is added to the stream to describe the Reed–Solomon policy and the size of the layers in bytes. The product code used for the protection of source symbols is depicted in Fig. 4. All rows are protected using systematic Turbo codes. During the Turbo decoding of a received packet, the CRC indicates if the packet is corrupted. On the occurrence of a corrupted packet, the Turbo codes are used to recover the information. If, however, the packet is not corrupted, due to the systematic form of the Turbo codes, the source information can be directly extracted without the need for channel decoding.

III. CHANNEL RATE ALLOCATION

A. General Problem Formulation

Since the bitstreams generated from the JPEG2000 coder consist of layers with unequal importance, UEP should generally be applied for their efficient protection from channel errors. An equal error protection (EEP) policy using Turbo codes and Reed–Solomon codes can also be applied in order to protect the source symbols. Such a rate allocation strategy has the advantage that no overhead need be transmitted stating the channel codes used for each layer. However, its performance is significantly inferior compared to schemes utilizing UEP [22].

A UEP algorithm for channel rate allocation is presented in this section. The proposed algorithm takes into account the importance of each packet (as defined in Section II) and allocates more channel symbols (Turbo code bytes and Reed–Solomon symbols) to packets carrying important information and fewer to other packets. In this way, packets that contribute with higher distortion improvement to the eventual image quality are better protected than the rest.

The rate allocation algorithm determines the number of channel symbols in each row and column of the product code array so that the remaining positions are available to source symbols. The problem formulated as above can be solved optimally under a specific target rate constraint by assuming that every product array includes the same number of source+channel bytes, namely R_{s+c}. The idea of keeping the size of channel packets constant and varying the amount of protection according to the importance of source information was originally proposed in [23] and was subsequently used in [7], [8], and [24]–[26].

As explained in the preceding section, the dependent layers are placed in the same packet, i.e., in the same row in the product code array. Since the channel codes are variable in both horizontal and vertical direction, the beginning of the first source byte in a packet is placed immediately after the last RS symbol. On the other hand, the Turbo-code stream begins after the last source symbol. Specifically we assume that in the nth row of the product code array, there are r_n RS symbols, s_n source bytes, and c_n Turbo-code bytes. The resulting product code array is schematically shown in Fig. 5.

The transmission of each packet in Fig. 5 stimulates a reduction in the average (expected) distortion of the image reconstructed after transmission. Since transmitted packets are independent of each other, the eventual *distortion reduction* D is the cumulative sum of the reductions achieved by the transmission of each packet separately, i.e.,

$$D = \sum_{n=1}^{N} D^n \qquad (1)$$

where D^n is the average distortion reduction caused by the transmission of the nth packet. Our intention is to determine the optimal r_n, s_n, c_n for $n = 1, \ldots, N$ by maximization of the average distortion reduction D subject to the constraint

$$r_n + s_n + c_n \leq R_p \Rightarrow s_n \leq R_p - r_n - c_n, \text{ for } n = 1, \ldots, N \qquad (2)$$

where $R_p = R_{s+c}/N$. In order to simplify our optimization formulation, we make two assumptions regarding the allocation of channel rate. Since the most important subbands are placed in the first packets we assume that the Turbo protection is nonincreasing with n, i.e.,

$$c_1 \geq c_2 \geq \ldots \geq c_N.$$

Similarly, we assume that the number of RS symbols in a packet is nondecreasing with n in the product code array, i.e.,

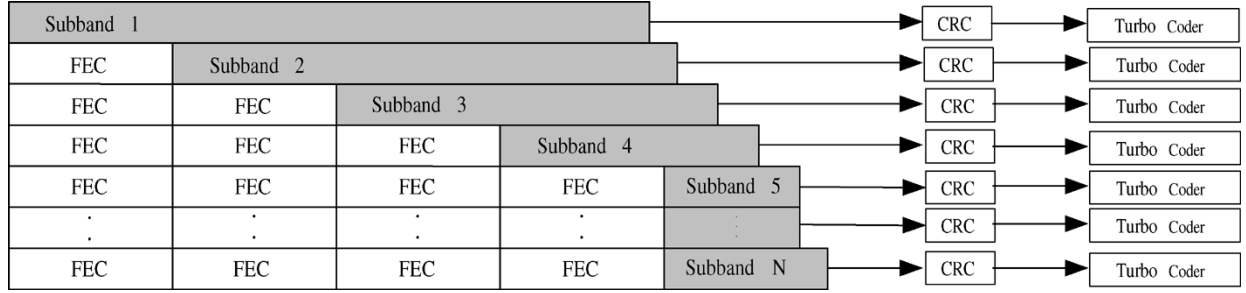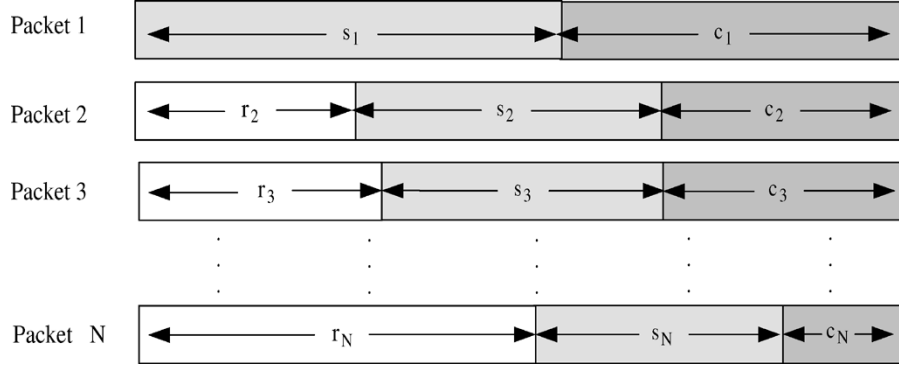$$r_1 \leq r_2 \leq \ldots \leq r_N.$$

Fig. 4.   Product code based on Turbo codes and Reed–Solomon codes.

Fig. 5.   Location of RS bytes, source bytes, and Turbo code bytes (denoted r_n, s_n, c_n, respectively) in the product-code array.

The probability that the nth packet is erased is denoted by $p_n(c_n)$ since, for given channel conditions, it depends on the code-rate of the Turbo code that was used for its protection. Moreover, let $D_n(y)$ denote the distortion reduction achieved by the transmission of the first y *source* bytes on the nth packet. Although $D_n(y)$ is, in practice, a discrete function our treatment here assumes a continuous extension of it. $D_n(y)$ is computed for each packet based on the wavelet coefficients that are included in the packet. This calculation is performed using information that is directly extracted from the JPEG2000 stream. Note that, in practice, $D_n(y)$ is meaningful only in the interval $[0, s_n]$ equivalently we can consider that $D_n(y) = 0$ for $y \notin [0, s_n]$.

The expected distortion depends on the number of packets that are erased during transmission. In practice, an erasure occurs when the Turbo decoder is unable to recover the information in a corrupted packet. The average distortion reduction caused by the transmission of the nth packet is given by

$$D^n = (1 - p_n(c_n)) \cdot D_n(s_n) + p_n(c_n) \cdot \overline{D}_n. \tag{3}$$

The term $D_n(s_n)$ in (3) expresses the distortion reduction in the case of fully recovering the nth packet by means of Turbo decoding, while $\overline{D}_n$ is the average distortion reduction in case the nth packet is corrupted and cannot be recovered by Turbo decoding. Assume that $p_e(x)$, $x = 0, \dots, N$ is the probability that x packets are erased. Then the average distortion reduction *when the nth packet is corrupted* is

$$\overline{D}_n = \sum_{x=1}^{N} p_e(x) \cdot D_n(r_{N-x+1} - r_n). \tag{4}$$

To gain insight regarding the term $D_n(r_{N-x+1} - r_n)$, assume that x packets are erased (see Fig. 6) during transmission. This

means that only the source symbols in product code columns in which there are at least x RS symbols can be recovered. Since the RS rate is monotonically nondecreasing with n, the portion of the stream that can be recovered is determined by the end of the RS stream in the $N - x + 1$ packet. Thus, in every column on the left of the axis in Fig. 6, there are at least x packets carrying RS symbols which guarantee the recovery of the erased information.

The probability $p_e(x)$ that exactly x packets, out of N packets in total, are erased is associated with the number x_j, $j = 1, \dots, Q$, of erased packets at the jth protection level, where Q is the number of *applied* Turbo protection levels[1] and t_j the total number of packets protected at the jth level. We define

$$u_j = \begin{cases} \min(t_j, x_j), & \text{if } j = 1 \\ \min\left(t_j, x - \sum_{k=1}^{j-1} x_k\right), & \text{otherwise.} \end{cases} \tag{5}$$

Then

$$\begin{aligned} p_e(x) &= P\{x_1 + x_2 + \dots + x_Q = x\} \\ &= \sum_{x_1=0}^{u_1} p_{t_1}(x_1) \cdot P\{x_2 + \dots + x_Q = x - x_1\} \\ &= \sum_{x_1=0}^{u_1} p_{t_1}(x_1) \cdot \sum_{x_2=0}^{u_2} p_{t_2}(x_2) \\ &\quad \cdot P\{x_3 + \dots + x_Q = x - x_1 - x_2\} \\ &= \sum_{x_1=0}^{u_1} \sum_{x_2=0}^{u_2} \cdots \sum_{x_{Q-1}=0}^{u_{Q-1}} p_{t_1}(x_1) \cdot p_{t_2}(x_2) \cdot \dots \\ &\quad \cdot p_{t_{Q-1}}(x_{Q-1}) \cdot P\left\{x_Q = x - \sum_{l=1}^{Q-1} x_l\right\} \end{aligned} \tag{6}$$

[1]Q should not be mistaken for the total number of *available* protection levels.
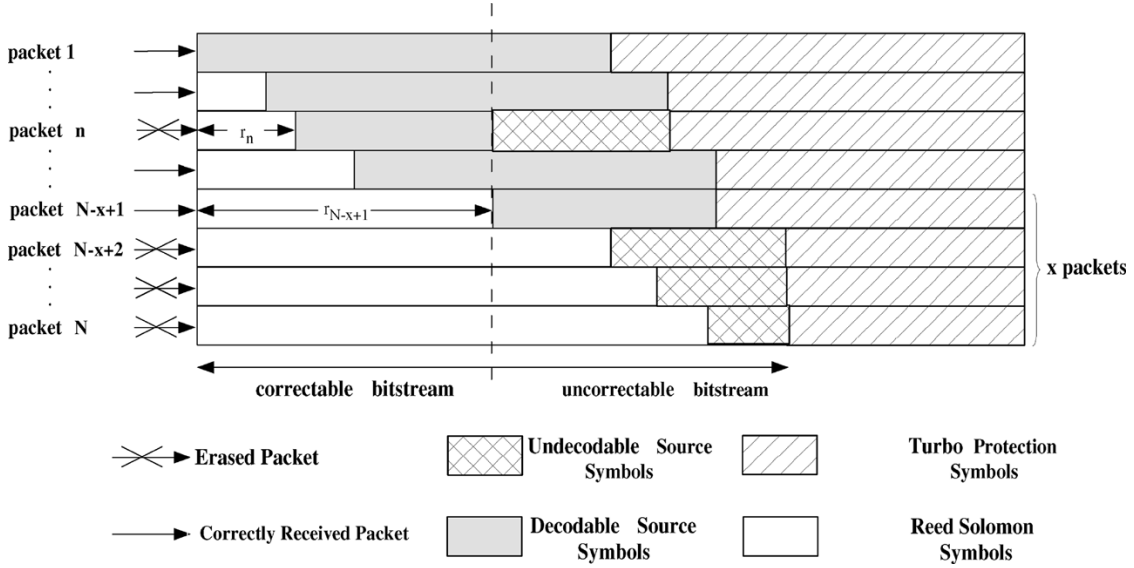
Fig. 6.　When x packets are erased, the correctable portion of the bitstream lies on the left of the axis defined by the end of the RS stream in the $N - x + 1$ packet (since symbols in this stream are protected by at least x RS symbols). Information symbols that lie on the right of the axis are decodable only if they are part of uncorrupted packets. Note that the first packet does not contain RS symbols.

where the variable $p_{t_j}(x_j)$ represents the probability that x_j packets are erased among the t_j packets at the jth protection level. These probabilities are experimentally evaluated for the examined channel condition, and, therefore, the calculation of $p_e(x)$ from (6) is straightforward.

Using (2)–(4), (1) is seen to be equivalent to

$$D = \sum_{n=1}^{N} \left\{ (1 - p_n(c_n)) \cdot D_n(R_p - r_n - c_n) + p_n(c_n) \right.$$
$$\left. \cdot \sum_{x=1}^{N} (p_e(x) \cdot D_n(r_{N+1-x} - r_n)) \right\}. \quad (7)$$

However, since $D_n(r_{N+1-x} - r_n)$ is zero for $N + 1 - x \le n \Rightarrow x \ge N + 1 - n$, only cases in which $x = 1, \ldots, N - n$ need be considered. Thus, (7) is equivalently expressed as

$$D = \sum_{n=1}^{N} \left\{ (1 - p_n(c_n)) \cdot D_n(R_p - r_n - c_n) + p_n(c_n) \right.$$
$$\left. \cdot \sum_{x=1}^{N-n} (p_e(x) \cdot D_n(r_{N+1-x} - r_n)) \right\}. \quad (8)$$

Note that the alteration of the RS rate r_n for the nth packet produces a change in the distortion reduction of the nth packet and additionally affects the distortion of other packets, as well. This is due to the fact that, in practice, the RS rate in a packet varies the correction capability of the RS code across all packets. Note, also, that the first packet does not contain any RS symbols, and, therefore, $r_1 = 0$.

Our purpose is to maximize the distortion reduction given by (8). For the efficient solution of the maximization problem, a two-stage procedure is followed. First, the RS code is kept constant and the Turbo-coding stream is optimized. Subsequently, the Turbo stream determined in the previous step is kept constant and the RS stream is optimized. The above procedure is repeated several times until convergence. It should be noted that

since the appropriate amounts of RS and Turbo protection are determined using a two-step process, and not jointly, the above algorithm does not guarantee global optimization. In practice, however, the proposed allocation algorithm yields very satisfactory results.

The source/channel allocation algorithm can be summarized in the following pseudocode.

1) Initialize the Turbo stream by allocating random, nonincreasing, amounts of Turbo protection.
2) Initialize the RS stream by allocating random, nondecreasing, amounts of RS protection.
3) Calculate $p_e(x)$ based on the corresponding probabilities $p_n(c_n)$ for $n = 1, \ldots, N$.
4) Optimize the Turbo stream keeping the RS stream constant.
5) Optimize RS allocation keeping the Turbo stream constant.
6) If the distortion is lower than that of the previous iteration then go to step 3); else, terminate.

Summarizing the above, the resulting channel allocation algorithm can be seen as a recursion of a two-step process. The first step is intended to coarsely determine the Turbo protection by assuming constant RS protection. Similarly, the second step is to determine the strength of the RS protection. The algorithm stops when in two consecutive iterations there is no further decrease in the distortion. Although it is certain that the distortion will stop decreasing after a few iterations, the determined code-rates using this strategy may not be theoretically optimal since the optimization is performed in two stages and not jointly. In practice, however, this approach yields excellent allocations.

Despite the fact that the division of the original optimization problem in two distinct problems is much simpler, the solution of each of the two optimizations themselves is not straightforward. The Turbo optimization problem, described in the sequel, is treated by exhaustive search among all possible combinations of channel rate allocation. The RS optimization problem is treated using the algorithm in Section III-C.
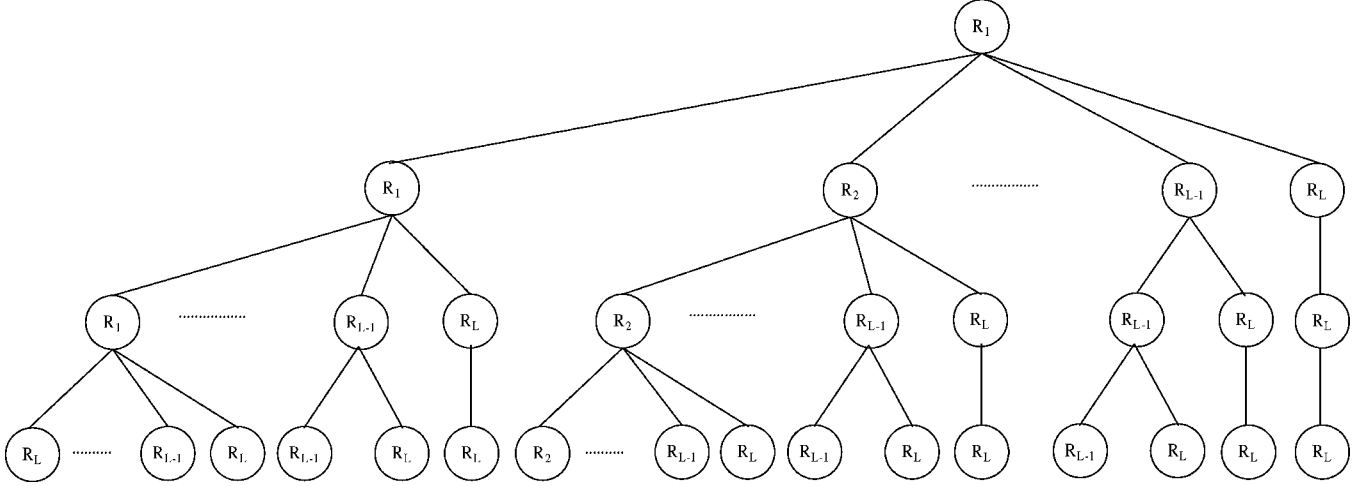
Fig. 7. Set of choices for the Turbo code-rates under the constraint given by (9).

B. Turbo-Code Rate Allocation

The search space of all possible combinations of Turbo protection can be significantly reduced by taking into account a constraint on the allowable protection levels for Turbo-coded packets. This constraint is derived from the fact that the allocated Turbo code-rates must be nondecreasing if the packets are ordered according to their importance prior to application of the optimization algorithm. Thus, if the code-rate (i.e., the ratio of the source over the channel bits) of the Turbo code applied to the nth packet is denoted R_n, the channel constraint is expressed as

$$R_n \leq R_{n+1}, \quad n = 1, \ldots, N-1 \tag{9}$$

i.e., the most important layers receive more Turbo bits.

This constraint does not allow codestreams corresponding to high frequency subbands (lower decomposition levels) to take more channel symbols than these of lower frequency subbands (higher decomposition levels). In general, the initial set of Turbo code-rates includes $\{R_1, R_2, \ldots, R_L\}$ with $R_1 < R_2 < \cdots < R_L$. Going from a packet to the next packet, this set can either remain unchanged or be reduced to a subset with lower protection (higher code rate) than the protection of the current node. The above procedure is illustrated in Fig. 7. Thus, if the nth packet is protected using Turbo channel code-rate R_L then the only possible code-rate for the protection of all subsequent packets is also R_L. In case the nth packet is protected using code-rate R_{L-1}, the $(n+1)$th packet is always protected using rates from the set $\{R_{L-1}, R_L\}$. If, however, this packet selects R_L the following packets will also select R_L; otherwise, the available set for the selection of $(n+2)$th packet protection is not reduced.

The constraint of nondecreasing Turbo code-rates reduces dramatically the number of eligible combinations for Turbo code allocation. For example, if the packets are grouped into three sets that are protected using four available Turbo code-rates R_1, R_2, R_3, R_4, which is a reasonable assumption in practice, then the number of possible combinations is reduced from a total of $3^4 = 81$ down to 15. Thus, in the case of a limited number of Turbo code-rates, the optimal allocation can

be determined by exhaustive search without imposing a heavy additional computational burden.

C. Reed–Solomon Rate Allocation

For the calculation of the optimal RS allocation, we visit the packets in the product-code block from the last packet to the first. During our search for the optimal amount of RS protection, the expected distortion reduction D is calculated using (8). The RS allocation achieving the highest distortion reduction is regarded as the initial RS allocation for the next iteration. Below, we prove a theorem that will help us formulate a fast algorithm for the optimization of the allocation of RS protection to the product-code packets. Note that, although our general packet/subband index is n, henceforth, we also use k to denote the index of the *specific* packet that is optimized.

Theorem: It can be shown that the function of the total expected distortion improvement with respect to the **RS rate** r_k in the kth packet is concave if the functions of the individual distortion improvements for each subband are concave with respect to the **source** bitrate.

Proof: From (8), the total expected distortion improvement can be expressed as

$$
\begin{aligned}
D &= \sum_{n=1}^{N} (1 - p_n(c_n)) \cdot D_n(R_p - r_n - c_n) \\
&\quad + \sum_{n=1}^{N} p_n(c_n) \cdot \sum_{x=1}^{N-n} p_e(x) \cdot D_n(r_{N+1-x} - r_n) \\
&= (1 - p_k(c_k)) \cdot D_k(R_p - r_k - c_k) \\
&\quad + \sum_{\substack{n-1 \\ n \neq k}}^{N} (1 - p_n(c_n)) \cdot D_n(R_p - r_n - c_n) + p_k(c_k) \\
&\quad \cdot \sum_{x=1}^{N-k} p_e(x) \cdot D_k(r_{N+1-x} - r_k) + \sum_{\substack{n-1 \\ n \neq k}}^{N} p_n(c_n) \\
&\quad \cdot \sum_{x=1}^{N-n} p_e(x) \cdot D_n(r_{N+1-x} - r_n). \tag{10}
\end{aligned}
$$

We denote $D(r_k)$ as the function of the above distortion with respect to r_k. By analyzing its rightmost term, (10) is expressed as

$$
\begin{aligned}
D(r_k) = {} & (1 - p_k(c_k)) \cdot D_k(R_p - r_k - c_k) \\
& + \sum_{\substack{n-1 \\ n \neq k}}^{N} (1 - p_n(c_n)) \cdot D_n(R_p - r_n - c_n) + p_k(c_k) \\
& \cdot \sum_{x=1}^{N-k} p_e(x) \cdot D_k(r_{N+1-x} - r_k) + \sum_{\substack{n-1 \\ n \neq k}}^{N} p_n(c_n) \\
& \cdot \left[\sum_{x=1}^{N-k} p_e(x) \cdot D_n(r_{N+1-x} - r_n) \right. \\
& \quad + p_e(N+1-k) \cdot D_n(r_k - r_n) \\
& \quad \left. + \sum_{x=N-k+2}^{N-n} p_e(x) \cdot D_n(r_{N+1-x} - r_n) \right]. \quad (11)
\end{aligned}
$$

Let us assume that $D(r_k)$ and $D_n(s_n)$, which are functions of the *RS rate* and the *source rate*, respectively, are continuous and differentiable (up to second order derivative). Then, the derivative[2] of the total expected distortion improvement with respect to r_k is

$$
\begin{aligned}
\frac{\partial D(r_k)}{\partial r_k} = {} & (1 - p_k(c_k)) \cdot \frac{\partial D_k(R_p - r_k - c_k)}{\partial r_k} + p_k(c_k) \\
& \cdot \sum_{x=1}^{N-k} p_e(x) \cdot \frac{\partial D_k(r_{N+1-x} - r_k)}{\partial r_k} + p_e(N+1-k) \\
& \cdot \sum_{\substack{n-1 \\ n \neq k}}^{N} p_n(c_n) \cdot \frac{\partial D_n(r_k - r_n)}{\partial r_k} \quad (12)
\end{aligned}
$$

and since $D_n(r_k - r_n) = 0$ for $n \geq k$, the above expression becomes

$$
\begin{aligned}
\frac{\partial D(r_k)}{\partial r_k} = {} & (1 - p_k(c_k)) \cdot \frac{\partial D_k(R_p - r_k - c_k)}{\partial r_k} + p_k(c_k) \\
& \cdot \sum_{x=1}^{N-k} p_e(x) \cdot \frac{\partial D_k(r_{N+1-x} - r_k)}{\partial r_k} + p_e(N+1-k) \\
& \cdot \sum_{n=1}^{k-1} p_n(c_n) \cdot \frac{\partial D_n(r_k - r_n)}{\partial r_k}. \quad (13)
\end{aligned}
$$

By studying (13), we can gain insight about the RS rate allocation process. It is obvious that the three terms in the right-hand side of (13) determine the distortion improvement as r_k changes. Since $D_n(\cdot)$ is an increasing function of the source rate, the first of the three terms indicates that as r_k grows larger, i.e., the source rate decreases, $D_k(R_p - r_k - c_k)$ decreases when the kth packet is not corrupted. Equivalently, this means that $(\partial D_k(R_p - r_k - c_k)/\partial r_k) < 0$. Similarly, the second term indicates that, as r_k grows larger, $D_k(r_{N+1-x} - r_k)$ decreases when the kth packet is corrupted. However, when x packets are corrupted in total and

$k = N - x + 1$, then the decoding of *all* other packets $0, \cdots, k-1$ benefits by the increase in the RS rate. This is expressed by the positiveness of the third term in the rightmost side of (13).

The second derivative of $D(\cdot)$ with respect to r_k is

$$
\begin{aligned}
\frac{\partial^2 D(r_k)}{\partial r_k^2} = {} & (1 - p_k(c_k)) \cdot \frac{\partial^2 D_k(R_p - r_k - c_k)}{\partial r_k^2} + p_k(c_k) \\
& \cdot \sum_{x=1}^{N-k} p_e(x) \cdot \frac{\partial^2 D_k(r_{N+1-x} - r_k)}{\partial r_k^2} + p_e(N+1-k) \\
& \cdot \sum_{n=1}^{k-1} p_n(c_n) \cdot \frac{\partial^2 D_n(r_k - r_n)}{\partial r_k^2}. \quad (14)
\end{aligned}
$$

Since we assumed that $D_n(s_n)$, $n = 1, \ldots, k, \ldots, N$ is concave everywhere, then $\partial^2 D_n(s_n)/\partial s_n^2$ is negative for $n = 1, \ldots, k, \ldots, N$. Trivially, all terms in the right-hand side of (14) are seen to be negative. Therefore, $(\partial^2 D(r_k)/\partial r_k^2) < 0$, and, thus, $D(r_k)$ is concave [27].

In practice, the assumption that $D_n(s_n)$, $n = 1, \ldots, k, \ldots, N$, is concave for all N subbands will not always hold. However, as seen from (8), due to the fact that D is a sum of $D_n(\cdot)$ terms, it is almost always concave with respect to r_n even if one or more of the $D_n(s_n)$ are not concave with respect to the source rate. We optimize the allocation of RS bytes, i.e., the determination of r_k for $k = 2, \cdots, N$, by maximizing $D(r_k)$ for each packet k. Specifically, for each k, the optimal r_k, in the sense of maximizing the total expected distortion improvement, is calculated by searching in the interval $[r_{k-1}, r_{k+1}]$. However, since $D(r_k)$ was previously shown to be concave, it is not actually necessary to apply exhaustive search techniques, but instead a systematic way in which the maximum is calculated can be applied. There are only three alternative cases.

Case 1: The maximum is at r_{k+1} [see Fig. 8(a)].
Case 2: The maximum is at r_{k-1} [see Fig. 8(b)].
Case 3: The maximum is in-between $[r_{k-1}, r_{k+1}]$ [see Fig. 8(c)].

The first two cases can be identified by calculating the gradient of D at r_{k+1} and r_{k-1}, respectively. The gradient D' is calculated as $D'(r) = (D(r + \delta r) - D(r - \delta r))/2\delta r$ where δr is a sufficiently small rate quantity. On the interval boundaries: $D'(r_{k-1}) = (D(r + \delta r) - D(r))/\delta r$ and $D'(r_{k+1}) = (D(r) - D(r - \delta r))/\delta r$. Thus, if $D'(r_{k-1})$ and $D'(r_{k+1})$ have the same sign, then the maximum of the distortion improvement coincides with $D(r_{k-1})$ or $D(r_{k+1})$. Otherwise, a bisection process must be applied to locate the appropriate r_k.

The above are summarized as follows.

- If $D'(r_{k-1}) > D'(r_{k+1}) > 0$, then the maximum is at r_{k+1}.
- If $D'(r_{k+1}) < D'(r_{k-1}) < 0$, then the maximum is at r_{k-1}.
- If $D'(r_{k-1}) \cdot D'(r_{k+1}) < 0$, then the maximum is in-between and bisection has to be applied.

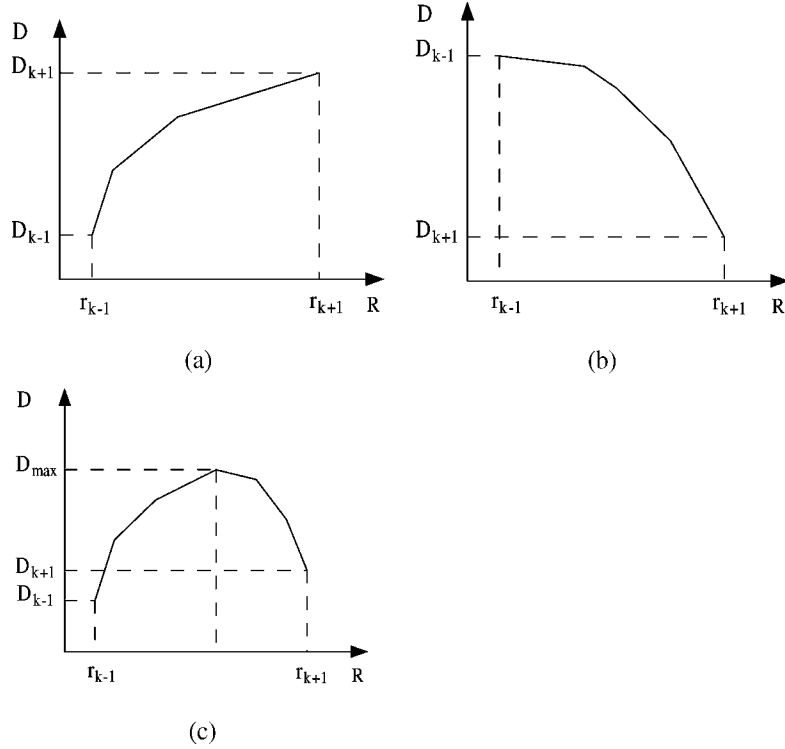A process which is complementary to the processes detailed in this section is described next.

Fig. 8. Search for the RS rate that maximizes the distortion improvement. (a) Case 1: The optimal rate is equal to r_{k+1}. (b) Case 2: The optimal rate is equal to r_{k-1}. (c) Case 3: The optimal rate is in the interval $[r_{k-1}, r_{k+1}]$.

IV. REALLOCATION OF JPEG2000 SUBSTREAMS

In the preceding section, a channel rate allocation algorithm was presented consisting of a two-stage procedure for the allocation of Turbo and Reed–Solomon symbols. The solution reached using this algorithm would be satisfactory if the JPEG2000 streams corresponding to each subband were of equal importance and their source lengths were equal to the assumed packet's length. In practice, this situation never occurs. Instead, some subbands will contain significant information that cannot be accommodated in a single packet and will be left out, whereas some other subbands will contain less significant information with which they will fill their corresponding packet. Apparently, assigning a packet to each subband will not lead to the inclusion of the most significant information in the eventual transmitted stream. In order to tackle the above problem, a postprocessing procedure is applied for the reallocation of source symbols among subbands.

The procedure is composed by two successive processes. First, the slopes $\lambda(k, j)$ are calculated for each layer j in a subband k

$$\lambda(k, j) = \frac{\Delta D(k, j)}{\Delta R(k, j)}$$

where $\Delta D(k, j)$ is the distortion reduction caused by the transmission of the jth layer in subband k and $\Delta R(k, j)$ is the layer's length (in bytes). If j'_t layers from subband k' have been included in packet k' and $\lambda(k', j'_t + 1) > \lambda(k, j_t)$, where j_t is the number of layers included from subband k in packet k, then $\Delta R(k, j)$ bytes are dropped from subband k and an equal number of bytes is added to subband k'. The procedure terminates when no further distortion improvements can be achieved

by making room for significant layers by discarding insignificant ones.

In order to further refine the above allocation, a second procedure is invoked which searches for possible exchanges of terminating *bytes* between subbands that would lead to an increase in the distortion improvement. This procedure is terminated if no exchange is found to improve the overall distortion.

After the reallocation procedure, the subband source rates determined using the algorithm of the previous section will not correspond to the new subband source rates since some subbands will now need more source space in their packet and others will need less. However, the summations of the subband source rates before and after the reallocation process will be equal. Thus, the source symbols from a subband are placed in the beginning of the available source space in their corresponding packet leaving any remaining space for symbols from other subbands whose new (after reallocation) source rate cannot be accommodated in their own corresponding packet. This procedure made the resulting allocations even more efficient.

V. EXPERIMENTAL RESULTS

The proposed scheme for the transmission of JPEG2000 streams over wireless channels (henceforth termed JPW) was experimentally evaluated for the transmission of the 512×512 test images "Lena" and "Peppers" over a flat-fading [28] Rayleigh channel simulated using the Jakes model [29]. Using this model, the channel is characterized by two parameters, i.e., the average received signal-to-noise ratio $\overline{\text{SNR}}$, which determines the average bit error rate, and the normalized Doppler

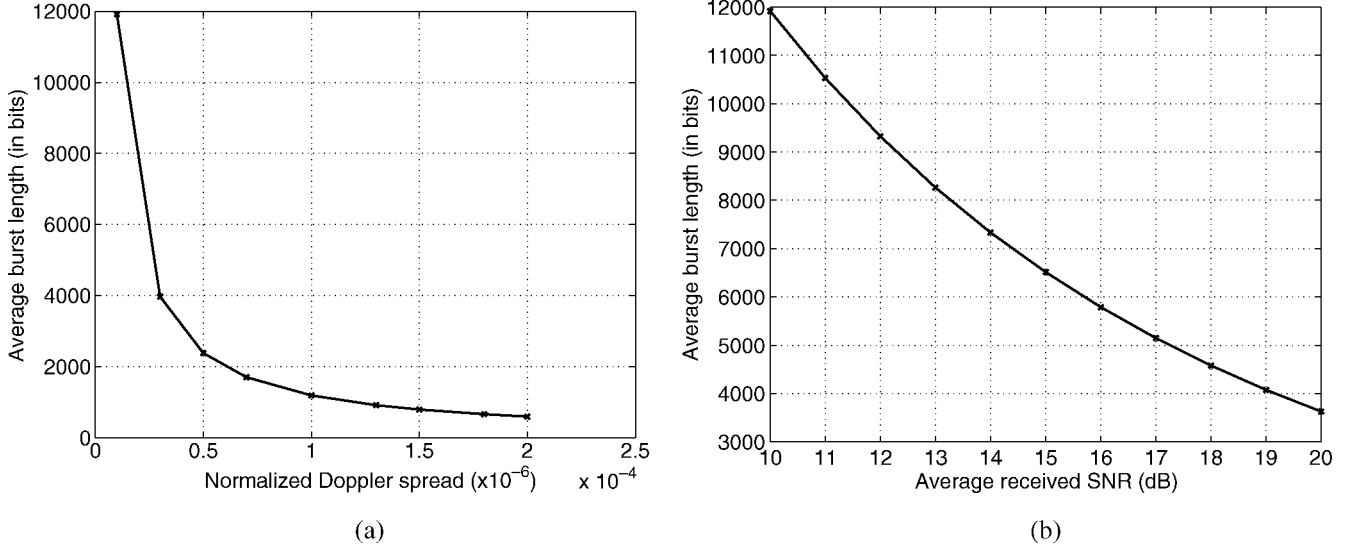(a)   (b)

Fig. 9.   (a) Average burst length with respect to f_D when $\overline{\text{SNR}} = 10$ dB. (b) Average burst length with respect to $\overline{\text{SNR}}$ when $f_D = 10^{-5}$.

spread[3] (i.e., the Doppler spread normalized by division by the data rate), which determines how quickly the channel changes over time. Therefore, in a practical transmission scenario, the decoder need only know the values of these two parameters that were used during encoding. According to [30] the burst length of a fade depends on f_D, the $\overline{\text{SNR}}$ and the received (SNR_r), which is related to the error rate as

$$P_e = \frac{1}{2} \cdot erfc(\sqrt{\text{SNR}_r}).$$

The average burst length is computed using the following formula:

$$\overline{\tau} = \frac{e^{\rho^2} - 1}{\rho \cdot f_D \cdot \sqrt{2 \cdot \pi}}$$

where ρ is the received amplitude normalized by the RMS amplitude and given by

$$\rho = \sqrt{\frac{\text{SNR}_r}{\overline{\text{SNR}}}}.$$

In Fig. 9(a), the average burst length is presented as a function of f_D when the $\overline{\text{SNR}} = 10$ dB. Respectively, in Fig. 9(b), the average burst length is reported for several $\overline{\text{SNR}}_s$ when $f_D = 10^{-5}$. All results are for $Pe \geq 0.1$ $(\text{SNR}_r = -0.8556$ dB).

For the application of the proposed techniques, fivel level wavelet decomposition of images was performed and a product code consisting of 16 packets was used. In all cases a 1/3 code rate Turbo coder was used with generator polynomials $(31, 27)_{\text{octal}}$ [18]. The Turbo codes were applied for the protection of symbols in the horizontal direction of the array. The output of the Turbo coder was punctured in order to achieve higher code rates. The resultant set of available Turbo code-rates was $\{8/9, \ldots, 8/24\}$. An S-random interleaver [31] with $S = 15$ was used with the Turbo coding/decoding processes. The maximum number of Turbo decoding iterations

[3]Spectral broadening observed in the transmission of a pure frequency tone.

TABLE I
PERFORMANCE COMPARISON FOR THE 512×512 "LENA" IMAGE (0.25 bpp).
MEAN PSNR RESULTS ARE REPORTED IN DECIBELS. THE PROPOSED SCHEME
OUTPERFORMS THE OTHER SCHEMES IN THE COMPARISON

Encoding	Code Rate	Mean PSNR
JPW	0.34	28.57
Sherwood [4]	0.28	27.82
Stankovic [15]	N/A	27.50

was 20. A CRC-16 with generator polynomial $(254\,465)_{\text{octal}}$ was also used for the efficient detection of corrupted packets. Each source + channel packet had approximately 512 bytes. For the determination of the Turbo code-rates and the RS protection, the algorithm of Section III was used. In order to further improve the performance of our scheme, the approach in [32] was followed during decoding.

The proposed scheme was evaluated for the case of normalized Doppler spread $f_D = 10^{-5}$ Hz/bps and average SNR equal to 10 dB. The JPW scheme was compared to the methods in [4] and [15] for the transmission of "Lena." The method in [4] was implemented using the List Viterbi algorithm [33] with a list of 100 paths. The peak-signal-to-noise ratio (PSNR) is used as a measure of the reconstruction quality.[4] All reported results are averages over 50 000 simulations. The results in Table I clearly demonstrate the superiority of the proposed scheme. In particular our system outperforms the methods in [4] and [15] by 0.75 and 1.07 dB, respectively. In Fig. 10 the proposed scheme is compared to the method in [4] in terms of *cumulative distribution function*. As shown, the proposed scheme achieves approximately the same number of error-free cases as [4]. From Fig. 10, it is obvious that the proposed system guarantees the recovery of higher PSNR images. We should emphasize that for

[4]Following the approach adopted in [4] and [7], the reported mean PSNR values were computed by averaging decoded MSE values and then converting the mean MSE to the corresponding PSNR value rather than averaging the PSNR values directly.
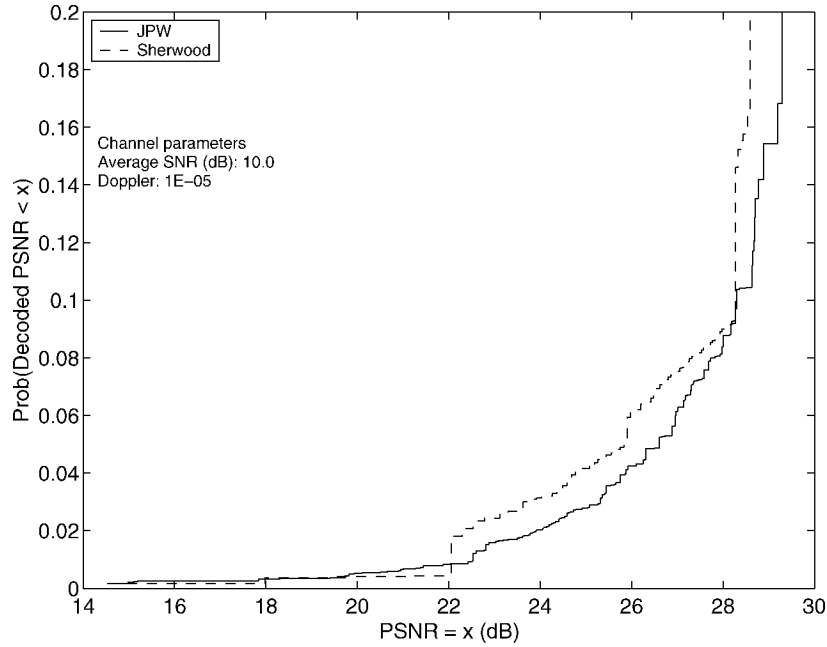
Fig. 10. Cumulative distribution of decoded PSNR for the 512×512 "Lena" image (0.25 dB), transmitted at rate 0.25 bits/pixel over a Rayleigh fading channel with $\overline{\text{SNR}} = 10.0$ dB and $f_D = 10^{-5}$.
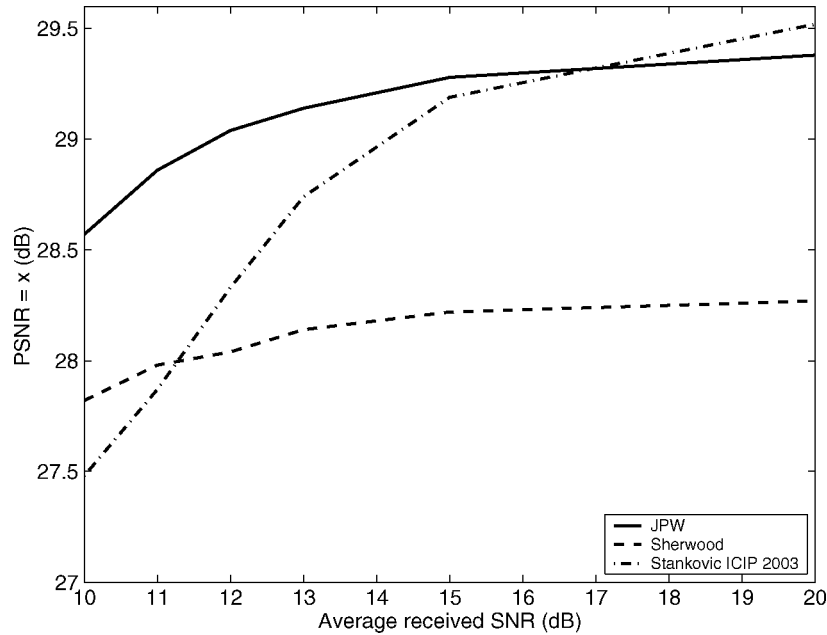

Fig. 11. PSNR comparison for the transmission of the 512×512 "Lena" image (0.25 bpp) as a function of the $\overline{\text{SNR}}$. The scheme was optimized for $\overline{\text{SNR}} = 10$ dB and $f_D = 10^{-5}$ and tested for various $\overline{\text{SNR}}$s and $f_D = 10^{-5}$.

all transmission scenarios that we consider in this section, the transmission of unprotected JPEG2000 bitstreams would lead to unacceptable image quality (below 20 dB) in the vast majority of the cases.

The proposed scheme was also evaluated for transmission in channel mismatch conditions. In Table II and Fig. 11, results are presented for "Lena" for the case where the scheme is optimized for normalized Doppler spread $f_D = 10^{-5}$ and $\overline{\text{SNR}} = 10$ dB and transmission is over a wireless channel with $f_D = 10^{-5}$ and variable $\overline{\text{SNR}}$. The results show that the proposed method performs better than the methods presented in [4] and [15]. Specifically, the gain of our scheme over the method in [15] becomes smaller for higher average SNRs (less noisy channels) since, in this case, error-free transmission occurs more often,

TABLE II
AVERAGE MSE CONVERTED TO PSNR OF THE PROPOSED SCHEMES IN COMPARISON TO THE METHODS IN [4] AND [15] FOR THE TRANSMISSION OF "LENA" IN MISMATCH CONDITIONS. ALL SCHEMES WERE OPTIMIZED FOR $f_D = 10^{-5}$ AND $\overline{\text{SNR}} = 10$ dB AND TESTED FOR VARIOUS $\overline{\text{SNR}}$s AND $f_D = 10^{-5}$

Encoding	$\overline{SNR}(dB)$					
	10	11	12	13	15	20
JPW	28.57	28.86	29.04	29.14	29.28	29.38
Sherwood [4]	27.82	27.98	28.04	28.14	28.22	28.27
Stankovic [15]	27.48	27.87	28.33	28.74	29.19	29.52

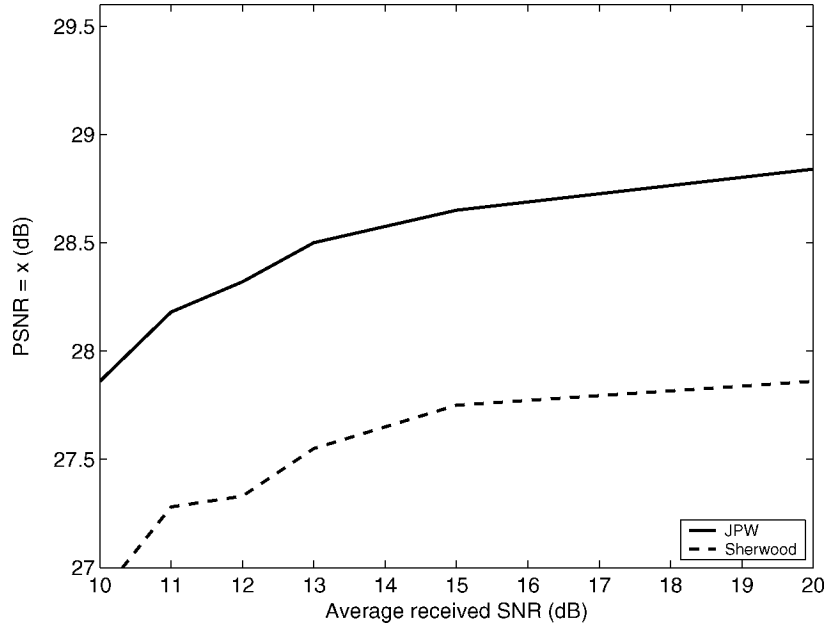and the method in [15] has a larger achievable peak PSNR in error free cases. On the contrary, the performance gain of the

Fig. 12.   PSNR comparison for the transmission of the 512×512 "Peppers" image (0.25 bpp) as a function of the $\overline{SNR}$. The scheme was optimized for $\overline{SNR} = 10$ dB and $f_D = 10^{-5}$ and tested for various $\overline{SNR}$s and $f_D = 10^{-5}$.

TABLE III

AVERAGE MSE CONVERTED TO PSNR OF THE PROPOSED SCHEMES IN COMPARISON TO THE METHODS IN [7], [13], [15] FOR THE TRANSMISSION OF "LENA" IN CHANNEL MISMATCH CONDITIONS. ALL SCHEMES WERE OPTIMIZED FOR $f_D = 10^{-4}$ AND $\overline{SNR} = 13$ dB AND TESTED FOR VARIOUS DOPPLER SPREADS AND $\overline{SNR} = 10$ dB

Encoding	$f_D (Hz/bps)$								
	10^{-5}	$3 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$7 \cdot 10^{-5}$	10^{-4}	$1.3 \cdot 10^{-4}$	$1.5 \cdot 10^{-4}$	$1.8 \cdot 10^{-5}$	$2 \cdot 10^{-4}$
JPW	27.80	29.11	29.45	29.34	29.06	28.67	28.43	27.93	27.63
Sachs [7]	21.45	22.69	24.04	25.39	26.99	27.96	28.42	28.61	28.44
Stankovic [15]	25.84	27.09	27.43	27.66	27.89	27.87	27.84	27.57	27.38
Stankovic [13]	27.08	27.22	27.24	27.21	27.17	27.11	27.23	27.02	27.00

present method over the method in [4] becomes larger for higher SNRs, due to the lower achievable peak PSNR of this method in error-free cases. In Fig. 12, additional results are presented for "Peppers." The new results verify the conclusions reached using "Lena."

For the sake of comparison, another mismatch scenario was considered. The proposed system was optimized for normalized Doppler spread $f_D = 10^{-4}$ and $\overline{SNR} = 13$ dB and transmitted over a wireless channel with $\overline{SNR} = 10$ dB and variable f_D. The results for "Lena" are presented in Table III and Fig. 13(a). The comparisons are with the methods of [7], [13], [15]. The proposed scheme has significantly better performance for slow fading channels due to the use of large packets. Although the performance of our scheme in channel mismatch conditions is less impressive for fast fading channels, it remains close to that of [7], [15]. The performance of [7] is poor for slow fading channels. However, the system in [7] has excellent performance for fast fading channels, which is attributed more to the higher PSNR than to the finer UEP protection. Results for the transmission of "Peppers" over the predescribed channel conditions are presented in Fig. 14(a). Although the method in [4] was proposed for channel conditions ($\overline{SNR} = 10$ dB and $f_D = 10^{-5}$)

which were closer to the actual testing conditions, the present scheme in most of the cases attains superior performance.

Finally, our scheme was tested for transmission over wireless channels with $f_D = 10^{-5}$ and variable $\overline{SNR}$, when the optimization conditions were as previously $f_D = 10^{-4}$ and $\overline{SNR} = 13$ dB. The results for "Lena" are presented in Table IV and Fig. 13(b). The present method has again the best performance for noisier channels. Specifically, the proposed method outperforms the methods [13], [15] by a significant margin under these conditions, while the method in [7] collapses. Results for the transmission of "Peppers" over the predescribed channel conditions are presented in Fig. 14(b).

In Fig. 15, we present a visual comparison of decoded images using our method and the method in [4].

VI. CONCLUSION

A novel image transmission scheme was proposed for the communication of JPEG2000 images over wireless channels. The proposed system reorganizes the compressed JPEG2000 stream in a product-code scheme consisting of Turbo codes and Reed–Solomon codes. An algorithm for the optimization of the
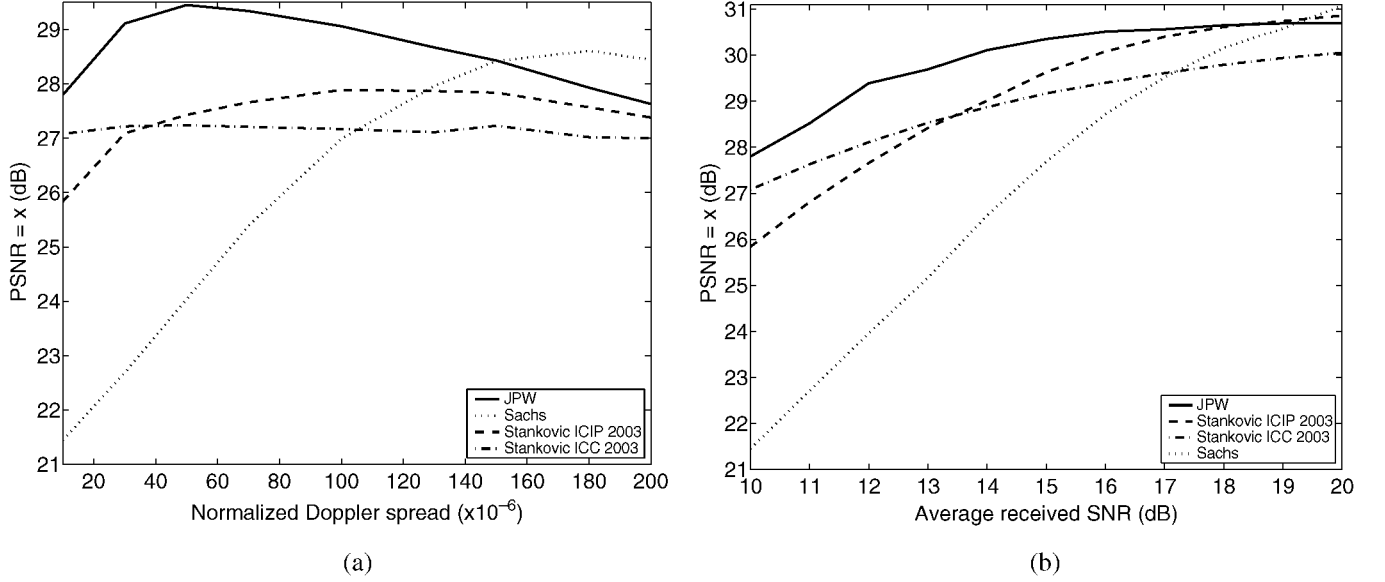
(a)

(b)

Fig. 13. PSNR comparison for the transmission of the 512×512 "Lena" image (0.25 bpp) as a function of the normalized Doppler spread f_D. The scheme was optimized for $\overline{SNR} = 13$ dB and $f_D = 10^{-4}$ and tested (a) for $\overline{SNR} = 10$ dB and various Doppler spreads and (b) for various $\overline{SNR}$s and $f_D = 10^{-5}$.
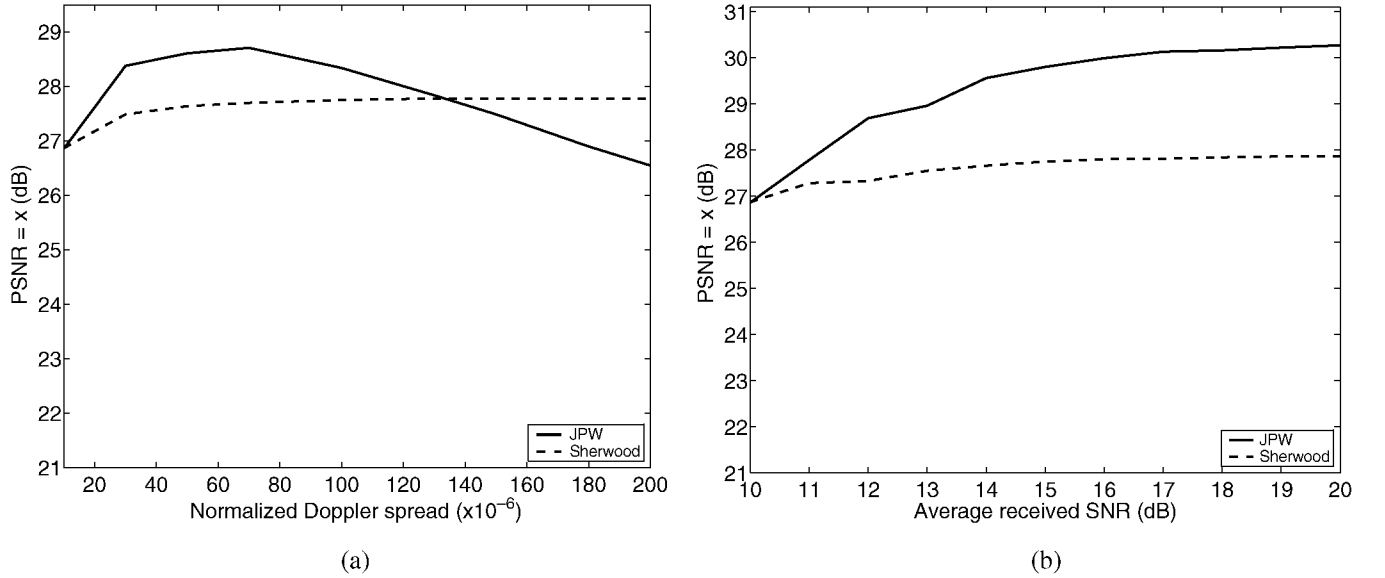

(a)

(b)

Fig. 14. PSNR comparison for the transmission of the 512×512 "Peppers" image (0.25 bpp) as a function of the normalized Doppler spread f_D. The proposed scheme was optimized for $\overline{SNR} = 13$ dB and $f_D = 10^{-4}$ while the scheme in [4] was optimized for $\overline{SNR} = 10$ dB and $f_D = 10^{-5}$. Both schemes were tested (a) for $\overline{SNR} = 10$ dB and various Doppler spreads and (b) for various $\overline{SNR}$s and $f_D = 10^{-5}$.

TABLE IV

AVERAGE MSE CONVERTED TO PSNR OF THE PROPOSED SCHEMES IN COMPARISON TO THE METHODS IN [7], [13], [15] FOR THE TRANSMISSION OF "LENA" IN MISMATCH CONDITIONS. ALL SCHEMES WERE OPTIMIZED FOR $f_D = 10^{-4}$ AND $\overline{SNR} = 13$ dB AND TESTED FOR VARIOUS $\overline{SNR}$s AND $f_D = 10^{-5}$

Encoding	$\overline{SNR}$										
	10	11	12	13	14	15	16	17	18	19	20
JPW	27.80	28.52	29.39	29.69	30.11	30.35	30.51	30.56	30.65	30.69	30.70
Sachs [7]	21.45	22.71	23.96	25.16	26.52	27.69	28.72	29.52	30.16	30.58	31.07
Stankovic [15]	25.84	26.81	27.66	28.42	29.01	29.63	30.08	30.40	30.61	30.74	30.86
Stankovic [13]	27.08	27.63	28.11	28.53	28.87	29.17	29.40	29.61	29.79	29.94	30.05

product-code was also proposed. The resulting schemes were tested for the transmission of images over wireless channels.

Experimental evaluation showed the superiority of the proposed scheme in comparison to other transmission schemes.

Fig. 15.   Visual comparison for the "Peppers" image using the proposed method and the method in [4]. The optimization conditions were $\overline{\mathrm{SNR}} = 10$ dB and $f_D = 10^{-5}$ for both methods. Comparison in terms of average quality:. (a) Proposed scheme (27.86 dB). (b) Sherwood method [4] (26.87 dB). Comparison in terms of a random transmission (the images were transmitted and decoded using the exact same simulation of the channel). (c) Proposed scheme (27.65 dB). (d) Sherwood method [4] (20.18 dB). It should be noted that the performance difference between the two methods would be much more pronounced for transmission in channel mismatch conditions.

REFERENCES

[1] N. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Wireless transmission of images using JPEG2000," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 2523–2526.

[2] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 6, pp. 243–250, Jun. 1996.

[3] N. V. Boulgouris, N. Thomos, and M. G. Strintzis, "Transmission of images over noisy channels using error-resilient wavelet coding and forward error correction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 12, pp. 1170–1181, Dec. 2003.

[4] G. Sherwood and K. Zeger, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Trans. Commun.*, vol. 46, no. 12, pp. 1555–1559, Dec. 1998.

[5] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1989.

[6] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[7] D. G. Sachs, A. Raghavan, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated codes," *Proc. SPIE*, pp. 300–311, Jan. 2000.

[8] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correcting codes," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 1999, pp. 342–346.

[9] N. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Wireless image transmission using turbo codes and optimal unequal error protection," in *Proc. IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sep. 2003, pp. 73–76.

[10] ——, "Wireless image transmission using turbo codes and optimal unequal error protection," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1890–1901, Nov. 2005.

[11] P. Cosman, J. Rogers, P. Sherwood, and K. Zeger, "Combined forward error control and packetized zerotree wavelet encoding for transmission of images over varying channels," *IEEE Trans. Image Process.*, no. 6, pp. 982–993, Jun. 2000.

[12] G. Sherwood and K. Zeger, "Progressive image coding on noisy channels," *IEEE Signal Process. Lett.*, vol. 4, no. 7, pp. 189–191, Jul. 1997.

[13] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Fast forward error protection algorithms for transmission of packetized multimedia bitstreams over varying channels," in *Proc. IEEE Int. Conf. Communication*, Anchorage, AK, May 2003, pp. 40–44.

[14] X. Wu, S. Cheng, and Z. Xiong, "On packetization of embedded multimedia bitstreams," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 132–140, Mar. 2001.

[15] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Product code error protection of packetized multimedia bitstreams," in *Proc. IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sep. 2003, pp. 77–80.

[16] V. Stankovic, R. Hamzaoui, and X. Zixiang, "Efficient channel code rate selection algorithms for forward error correction of packetized multimedia bitstreams in varying channels," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 240–248, Apr. 2004.

[17] J. Thie and D. Taubman, "Optimal erasure protection assignment for scalable compressed data with small packets and short channel codewords," *EURASIP J. Appl. Signal Process.*, vol. 2, pp. 207–219, Feb. 2004.

[18] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[19] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[20] *JPEG2000 Part I Final Draft International Standard*, ISO/IEC JTC 1/SC 29/WG1, 2000.

[21] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo codes (1)," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 23–26.

[22] T. Zhang and Y. Xu, "Unequal packet loss protection for layered vido transmission," *IEEE Trans. Broadcasting*, vol. 45, no. 2, pp. 243–252, Jun. 1999.

[23] A. Albanese, J. Bloemer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1737–1744, Nov. 1996.

[24] B. A. Banister, B. Belzer, and T. R. Fisher, "Robust image transmission using JPEG2000 and turbo codes," *IEEE Signal Process. Lett.*, vol. 9, no. 4, pp. 117–119, Apr. 2002.

[25] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 819–828, Jun. 2000.

[26] G. Davis and J. Danskin, "Joint source and channel coding for image transmission over lossy packet networks," in *Proc. SPIE Conf. Wavelet Applications to Digital Image Processing*, Denver, CO, Apr. 1996, pp. 376–387.

[27] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[28] K. N. Ngan, C. W. Yap, and K. T. Tan, *Video Coding for Wireless Communication Systems*. New York: Marcel Dekker, 2001.

[29] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.

[30] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[31] D. Divsalar and F. Pollara, "Turbo codes for pcs applications," in *Proc. IEEE Int. Conf. Communication*, Seattle, WA, Jun. 1995, pp. 54–59.

[32] L. Cao and C. W. Cheng, "A novel product coding and recurrent alternate decoding scheme for image transmission over noisy channels," *IEEE Trans. Commun.*, vol. 51, no. 9, pp. 1426–1431, Sep. 2003.

[33] N. Seshadri and C.-E. Sundberg, "List Viterbi decoding algorithm with applications," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 2/3/4, pp. 313–323, Feb./Mar./Apr. 1994.

Nikolaos Thomos (S'02) received the Diploma and the Ph.D. degrees from the Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2000 and 2005, respectively.

During his studies, he held teaching and research assistantship positions with the Electrical and Computer Engineering Department, Aristotle University of Thessaloniki. He was also Postgraduate Research Fellow with the Informatics and Telematics Institute, Thessaloniki, where he is currently a Postdoctoral Research Fellow. His research interests include image/video coding and transmission, multimedia networking, distributed source coding, space time coding, and digital filters.

Dr. Thomos is a member of the Technical Chamber of Greece.

Nikolaos V. Boulgouris (S'96–M'04) received the Diploma and the Ph.D. degrees from the Electrical and Computer Engineering Department, University of Thessaloniki, Greece, in 1997 and 2002, respectively.

Since December 2004, he has been a Lecturer with the Department of Electronic Engineering, Division of Engineering, King's College, London, U.K. From September 2003 to November 2004, he was a Postdoctoral Fellow with the University of Toronto, Toronto, ON, Canada. Previously, he was with the Informatics and Telematics Institute, Greece. He has participated in several research projects in the areas of image/video communication, pattern recognition, multimedia security, and content-based indexing and retrieval.

Dr. Boulgouris is a member of the British Machine Vision Association.
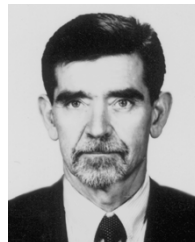
Michael G. Strintzis (S'68–M'70–SM'80–F'04) received the Diploma in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1967 and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1969 and 1970, respectively.

He joined the Electrical Engineering Department, University of Pittsburgh, Pittsburgh, PA, where he served as an Assistant Professor from 1970 to 1976 and an Associate Professor from 1976 to 1980. During that time, he worked in the area of stability of multidimensional systems. Since 1980, he has been a Professor of electrical and computer engineering at the Aristotle University of Thessaloniki, Thessaloniki, Greece. He has worked in the areas of multidimensional imaging and video coding. Over the past ten years, he has authored over over 100 journal publications and over 200 conference presentations. In 1998, he founded the Informatics and Telematics Institute, currently part of the Centre for Research and Technology Hellas, Thessaloniki.

Dr. Strintzis was awarded the Centennial Medal of the IEEE in 1984 and the Empirikeion Award for Research Excellence in Engineering in 1999.