

Wireless Image Transmission Using Turbo Codes and Optimal Unequal Error Protection

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

Abstract—A novel image transmission scheme is proposed for the communication of set partitioning in hierarchical trees image streams over wireless channels. The proposed scheme employs turbo codes and Reed–Solomon codes in order to deal effectively with burst errors. An algorithm for the optimal unequal error protection of the compressed bitstream is also proposed and applied in conjunction with an inherently more efficient technique for product code decoding. The resulting scheme is tested for the transmission of images over wireless channels. Experimental evaluation clearly demonstrates the superiority of the proposed transmission system in comparison to well-known robust coding schemes.

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

I. INTRODUCTION

DURING the past few years, mobile/wireless systems have become a dominant means of communication. Their advantages, however, come at the cost of significant additional complexity in the design of suitable coding and transmission systems. The specific application of image transmission over wireless channels poses a challenging research problem, requiring among others, appropriate selection of the set of channel codes for effective forward-error correction (FEC).

A variety of error-resilient techniques have been recently proposed in the literature. Most are based on the state-of-the-art set partitioning in hierarchical trees (SPIHT) [1] source coder which generates embedded bitstreams, i.e., streams in which lower rates are prefixes of higher rates. In [2] an ad-hoc approach was formulated for the protection of SPIHT streams, using product codes that consist of concatenated rate compatible punctured convolutional codes/cyclic redundancy check codes (RCPC/CRC) [3], [4] and Reed–Solomon (RS) codes [5]. The resulting scheme was shown to perform well for image transmission over wireless and memoryless channels. In [6], a multiple description product code was presented, similar in many aspects with that in [7], appropriate for image transmission 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 code based on the SPIHT coder and RS erasure-correction codes. An algorithm for unequal error protection (UEP) was also proposed based on ideas initially explored in [8].

In [9], a modified SPIHT algorithm was used 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 modified source stream was channel coded using the techniques in [10]. The resulting coding methodology was tested for the transmission of images over channels which suffer bit errors as well as packet erasures.

In this paper, a novel methodology for the transmission of images over wireless channels is proposed. The proposed scheme is based on the SPIHT source coder applied in conjunction with the application of turbo codes [11] across rows of data in a product code consisting of turbo codes and RS codes. Due to the systematic form of turbo codes, immediate extraction and decoding of source information from the channel-coded stream is possible whenever the stream is not corrupted.

The optimal allocation of RS 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 blocks of constant size, i.e., blocks of information in which the source bytes vary but the sum of source and channel bytes is fixed. This approach admits a fast dynamic programming implementation. Further, the streams received at the decoder side are subjected to a successive decoding process which is inherently more efficient than the conventional decoding of product codes. 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.

This paper is arranged as follows. In Section II, the forward error correction methodology used with the present system is described. In Section III, a channel rate allocation technique is presented. The technique used for the efficient successive decoding of product codes is described in Section IV. Experimental results are reported in Section V, and, finally, conclusions are drawn in Section VI.

II. ERROR-RESILIENT CODING USING TURBO CODES

In this section, we describe a product code based on turbo codes and RS codes. A product code generates parity bits for data arrays in both horizontal and vertical directions (across

Manuscript received March 13, 2003; revised September 16, 2004. This work was supported in part by the European Commission under contract FP6-511568 3DTV and in part by the EU IST project OTELO. This work was presented at the IEEE International Conference on Image Processing 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. John Apostolopoulos.

N. Thomos and M. G. Strintzis are with the Information Processing Laboratory, Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Thessaloniki 541 24, Greece (e-mail: nthomos@iti.gr; strintzi@eng.auth.gr).

N. V. Boulgouris is with the Electronic Engineering Department, King's College, London WC2R 2LS, U.K. (e-mail: boulgouris@ieee.org).

Digital Object Identifier 10.1109/TIP.2005.854482



Fig. 1. Arrangement of data in a turbo-coded packet. Unlike RCPC/CRC-coded streams, info bits can be directly decoded if they are not corrupted.

rows and columns of the array, respectively). Product codes have been recently used in wireless transmission applications [2], [6]. In the present product code, the row code consists of a cyclic redundancy code (CRC) combined with a systematic turbo code [12] (see Fig. 1) while the column code is an erasure-correction punctured systematic RS code [5]. RS codes are effective when errors occur in bursts and their location in the transmitted symbol sequence is known. Moreover, RS codes are maximum distance separable (MDS) codes. RS codes generated using symbol puncturing are still MDS codes [13]. RS codes are a subset of BCH [5] codes and are denoted by a pair (N, v) , where v is the number of the information symbols and N is the length (in term of symbols) of the resulting codeword. The error correction capability of RS codes is $N - v$ when the positions of the erasures are known; otherwise, it is reduced to roughly half, since $\lfloor (N - v)/2 \rfloor$ erased symbols can be recovered and decoded [5].

The product code used for the protection of source symbols is depicted in Fig. 2. A different amount of protection, provided by RS codes, is allocated to each portion of the stream. A header is placed at the beginning of the stream in order to declare the RS protection levels of the blocks that follow. The header is essentially a small block, with three columns, and is protected using the same RS code rate that is used for the protection of the first block. The overhead has information about the number of RS symbols used for the protection of each block. All rows of the array are protected using systematic turbo codes of the same code rate. The resulting turbo-coded row will be hereafter termed “packet.” The CRC code is used because it is highly efficient for error detection during the decoding of a packet. Such errors occur quite often in wireless transmission due to deep fades. Concatenated turbo/CRC codes are preferable for transmission over slow fading channels compared to the commonly used RCPC/CRC combination because of the better overall performance of turbo codes, especially when large packets are used. On the other hand, the use of many short packets would demand a significant increase in the transmitted overhead, i.e., the CRC bytes.

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 corrupted information. If, however, the packet is found not to contain errors, due to the systematic form of the turbo codes, the source information can be directly extracted without the need for channel decoding. Specifically, after turbo coding of a packet, the coded information has the form shown in Fig. 1. As shown in [5] and [13], the CRC check indicates the corruption of the information bits with high reliability (99.99%). Therefore, if, during decoding, the CRC indicates no corruption of the information bitstream, then the information bits can be directly decoded using the source decoder without the need to perform turbo decoding. In this respect, the use of turbo codes in our system has a significant advantage over the RCPC/CRC coding approach followed

in [2] and [6], which requires decoding of convolutional codes even for uncorrupted streams. This feature of our system partially outbalances the additional decoding complexity that turbo codes have in the cases of corrupted streams. A more detailed analysis of the complexity issue is presented in Appendix I.

In the ensuing section, an algorithm for the optimal allocation of protection to the source stream is presented.

III. FAST CHANNEL RATE ALLOCATION

A. Basic Problem Formulation

A UEP algorithm for channel rate allocation is presented in this section. The amount of protection allocated to data varies in the vertical direction (columns) of a data sub-array which will be hereafter termed a “block” (see Fig. 2). The proposed algorithm takes into account the importance of each block and allocates more channel symbols (RS symbols) to blocks carrying important information and fewer to other blocks. In this way, blocks that achieve higher quality improvement are better protected than the rest.

The rate allocation algorithm determines the number of source symbols in each column of the block. Since a systematic RS codeword consists of information symbols and parity symbols, the remaining positions in each column of the block are filled with parity RS symbols generated after the RS encoding of the source symbols. The problem formulated as above can be solved optimally under a specific target rate constraint using dynamic programming. The optimization procedure is facilitated by the constant number of columns in each block due to which many paths in the trellis merge yielding a much simpler structure and, thus, reducing dramatically the computational cost of the dynamic programming algorithm. In this way, every block 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 [14] and was subsequently used in [6]–[8], [15], and [16].

Another important issue in the design of an efficient transmission scheme, based on the structure of Fig. 2, is the choice of the size of blocks. In general, the use of larger blocks would result in the reduction of the complexity of the allocation process since the appropriate protection would have to be determined for fewer blocks. On the other hand, blocks have to be small enough so that they contain, as much as possible, source information of equal importance. Consequently, we would like to form blocks in which, after subtracting the bytes reserved for RS protection, the remaining bytes are filled using information taken from the same source layer¹ (and, therefore, having the same importance). In the experimental results section, we use the above considerations in order to make specific choices on the size of the block.

As mentioned in the previous section, the decoding of the present scheme consists of a row-wise decoding of turbo-codes (in case the CRC check indicates a packet is corrupted) and then column-wise decoding of RS codes. Therefore, in order

¹A “source layer” is a portion of the bitstream which includes bits of approximately the same importance, i.e., their transmission produces the same image quality improvement. In many cases, a source layer coincides with an entire bit-plane of wavelet coefficients.

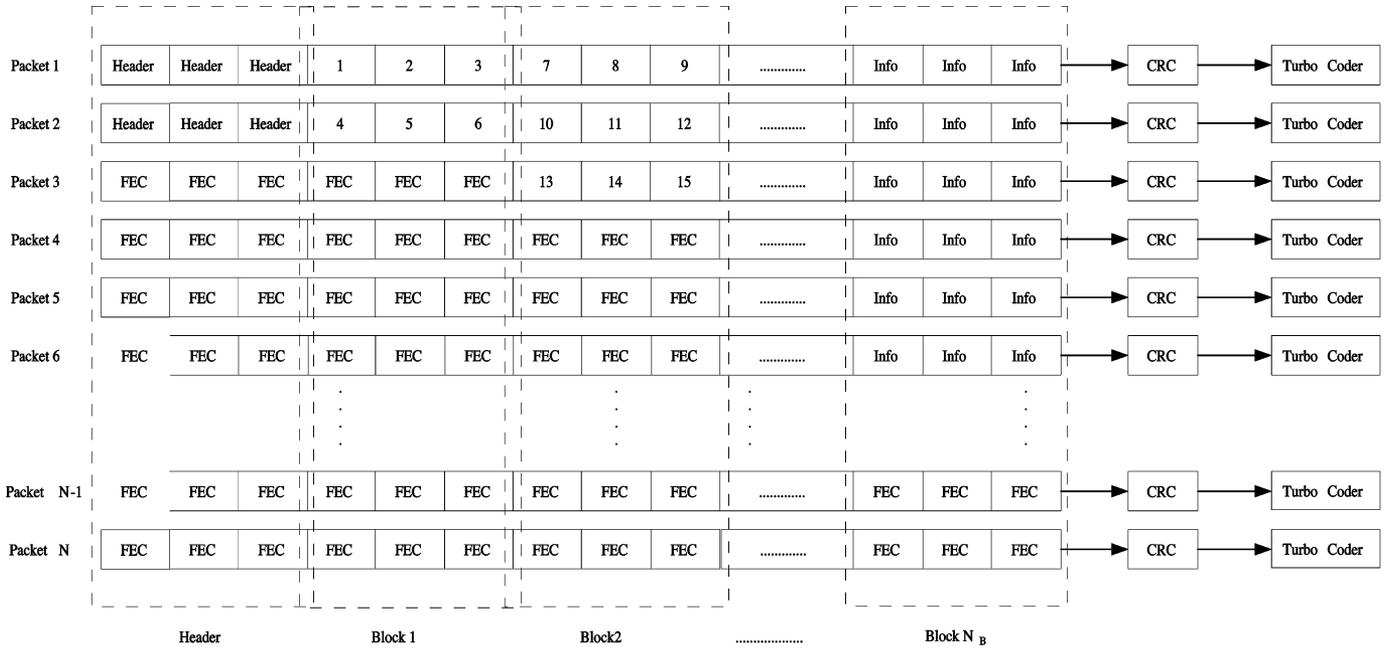


Fig. 2. Product code based on turbo codes and RS codes.

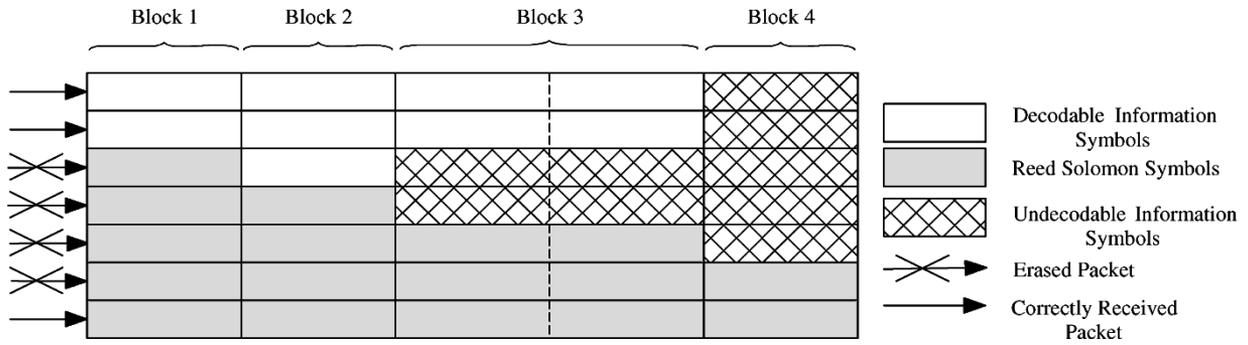


Fig. 3. Decodable bitstream when the number of errors in the stream exceeds the error correction capability of RS codes.

to perform efficient allocation of RS symbols, we assume that the probabilities $P(n)$ that exactly n rows, out of the N rows in the product code array, are erased due to transmission errors and turbo decoding failure are known in advance.²

Since RS codes are used, the probability that the k_{th} block is lost is equal to the probability that the number of lost rows in the k_{th} block is greater than the number of RS symbols in each column of the block

$$P_L(k) = \sum_{i=Q(k)+1}^N P(i) \quad (1)$$

where $Q(k)$ denotes the number of RS symbols in each column of the k_{th} block.

We shall assume that the total rate budget is $R_B = N_B \cdot R_{s+c}$, where N_B is the total number of blocks to be transmitted. In this way, the computed allocation policy is optimal for the specific target rate. A methodology for optimizing the allocation in intermediate rates (wherever an embedded coder is used) was proposed in [17].

²The probabilities are evaluated via simulation for each possible channel condition.

In order to facilitate the allocation of protection to the source stream, we initially assume that whenever a block is plagued by uncorrectable errors, all subsequent blocks are rendered useless and do not further lower the distortion (see Fig. 3). Suppose that n packets are lost. Then, the expected distortion is trivially seen to equal

$$\begin{aligned} \bar{D} = & D_0 \cdot P(n > Q(1)) + D_1 \cdot P(n > Q(2), n \leq Q(1)) \\ & + D_2 \cdot P(n > Q(3), n \leq Q(2)) \\ & + \dots + D_k \cdot P(n > Q(k+1), n \leq Q(k)) \\ & + \dots + D_{N_B} \cdot P(n \leq Q(N_B)) \end{aligned} \quad (2)$$

where D_k is the resulting distortion after the successful transmission of the first k blocks. The last term in the summation expresses the distortion when all blocks are decodable at the receiver side. Since the rate-distortion function for practical embedded coders (such as the SPIHT coder that is used in the present paper) is convex, we further assume that the RS protection allocated to the source bitstream is descending, i.e., for two arbitrary blocks k and k' , $k < k'$, the k_{th} block is always protected using at least as powerful codes as the codes used for the protection of the k'_{th} block. Due to the descending RS protection

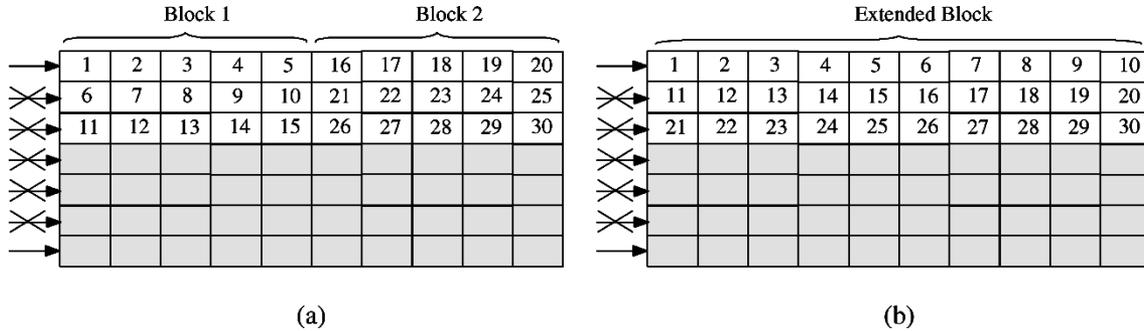


Fig. 4. If the second packet in (a) is unrecoverably corrupted, only source symbols 1–5 would be decodable. However, if the ordering in (b) is used, symbols 1–10 would be decodable. Note that, since both blocks are protected using the same number of RS symbols in each column, either both blocks will be corrupted or both will be received correctly.

level across blocks, the probability that all blocks are decodable is equal to the probability that the last (least protected) block is correctly decoded. Note that

$$\begin{aligned}
 & P(n > Q(k), n \leq Q(k-1)) \\
 &= P(Q(k) < n \leq Q(k-1)) \\
 &= \sum_{i=Q(k)+1}^{Q(k-1)} P(i) \\
 &= \sum_{i=Q(k)+1}^N P(i) - \sum_{i=Q(k-1)+1}^N P(i) \\
 &= P_L(k) - P_L(k-1). \tag{3}
 \end{aligned}$$

We define P_d as

$$P_d(k) = \begin{cases} P_L(k), & \text{if } k = 1 \\ P_L(k) - P_L(k-1), & \text{otherwise.} \end{cases} \tag{4}$$

Thus, $P_d(k)$ is the probability difference between the event of losing the k_{th} block and the event of losing the $(k-1)_{th}$ block. Thus, (2) can be equivalently expressed as

$$\bar{D} = D_0 \cdot P_d(1) + \sum_{k=2}^{N_B} D_{k-1} \cdot P_d(k) + (1 - P_L(N_B)) \cdot D_{N_B}. \tag{5}$$

The minimization of the above average distortion D can be achieved using dynamic programming based on the viterbi algorithm [18]. Whenever two or more paths merge in one trellis node, i.e., they represent the same cumulative total rate and the same cumulative channel rate, the paths with the greatest distortion are discarded and only the path having the minimum distortion is retained. The optimal solution of this problem will be studied later in this section.

B. More Accurate Approach

The above formulation disregards the fact that, even in the blocks containing uncorrectable errors, the information in the packets preceding the occurrence of a corrupted packet is useful since the information they contain is still decodable. This observation can be further exploited in practice by reordering information in blocks to which equal amounts of protection have been allocated. The blocks that are protected equally attain the obvious property that during transmission over wireless channel

either they will all be decoded successfully or will all fail. This is due to the fact that the number of erased packets that renders each block undecodable is the same for all such blocks. Rearranging the source bytes inside such blocks is advantageous over the initial information ordering and results in lower distortion, since more bytes are decodable in case of RS code failure, whereas there is no change in the distortion when the blocks are transmitted correctly. This is explained in the following.

Observation 1: The average distortion resulting in case of RS protection failure is always lower if consecutive blocks that are allocated equal amounts of protection are treated as a single extended block.

Proof: Suppose that, in two consecutive blocks, block 1 and block 2 that are allocated the same amount of protection, information is ordered so that the decoding of information in packet 1 that also belongs in block 2 demands prior decoding of the last source symbols in block 1 [see Fig. 4(a)]. This means that in case of RS failure, only the information from block 1 that lies in the first packets (until the first corrupted packet) will be decodable. If, however, in each packet, information is ordered so that block 2 contains symbols that appear right after the last source symbol of block 1 [see Fig. 4(b)], then this information would still be decodable if, say, the second or the third packet is unrecoverably corrupted.

In practice, the above reordering of information can be seen as merging the consecutive, equally protected blocks and then ordering information in a conventional raster-scan way. The size of the extended block L' obtained after merging of l consecutive elementary blocks of size L is

$$L' = l \cdot L \tag{6}$$

where L, L' are the lengths of the elementary and the extended blocks, respectively, and l is an integer greater or equal to 2. Since the distortion function $D(\cdot)$ is assumed to be monotonically decreasing with respect to the source rate (this is a reasonable assumption when the SPIHT source coder is used), the resulting distortion after the merging will always be lower, i.e.,

$$D(R_{k-1} + l \cdot L \cdot r) \leq D(R_{k-1} + L \cdot r) \tag{7}$$

where R_{k-1} is the source rate reached after the successful decoding of the $(k-1)_{th}$ block and $r = 0, 1, \dots, Q(k) - 1$ is the number of initial packets that are received errorlessly in cases

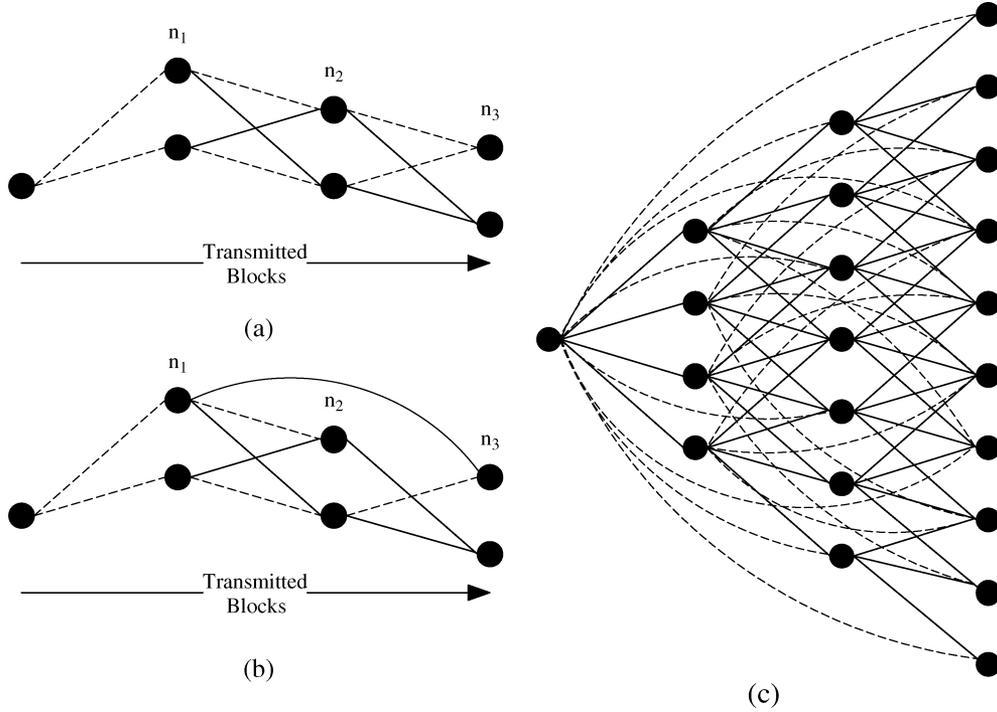


Fig. 5. (a) Conventional tree growing. (b) Proposed tree growing. (c) Modified trellis.

where the decoding of the k_{th} block fails. The equality holds for $r = 0$. ■

The above observation dictates the use of an optimization policy which is different in comparison to that expressed using (5). Assuming that the $(k + 1)_{th}$ block is the first one having uncorrectable errors, i.e., decoding stops at the $(k + 1)_{th}$ block, then taking into account the modified decodable rate for this block, the average distortion when k blocks are correctly received is

$$D'_k = \sum_{i=Q(k+1)+1}^{Q(k)} \frac{P(i)}{P_d(k+1)} \cdot \left\{ \sum_{j=1}^{N-i+1} P_{er}(i, j) \cdot D(R_k + (j-1) \cdot L') \right\} \quad (8)$$

where $P_{er}(i, j)$ denotes the probability that the j_{th} packet is the first packet (in a total of i erased packets) that is erased and is trivially seen to be given by

$$P_{er}(i, j) = \begin{cases} \frac{i}{N}, & \text{if } j = 1 \\ \frac{i}{N-j+1} \cdot \prod_{l=1}^{j-1} \left(1 - \frac{i}{N-l+1}\right), & \text{otherwise.} \end{cases} \quad (9)$$

The outer summation in (8) declares that the current block is unrecoverably corrupted but the previous blocks are correctly received. In this case, the number of erased packets exceeds the error correction capability $Q(k + 1)$ of the RS code used in the $(k + 1)_{th}$ block but is less than $Q(k)$, the error correction capability of the code applied to the k_{th} block. In (8), the term inside the brackets expresses the expected distortion when the number of erased packets is between $Q(k + 1) + 1$ and $Q(k)$.

C. Optimal Selection of Channel Rates Using Dynamic Programming

The problem of the channel rate allocation, as formulated above, can be solved optimally using dynamic programming techniques. We will first describe the solution of the basic problem of (5), i.e., disregarding partial block decoding and extended blocks, as an introduction to our more general methodology described later in this section. Our basic rate allocation formulation is similar in many aspects to that in [15].

The trellis diagram corresponding to the minimization of D in (5) given a target transmission rate is shown in Fig. 5(a). Each branch represents the application of a specific RS protection level for a block. Paths arriving to the same node represent channel allocation policies which yield the same channel code rates. In each node, only the path exhibiting the lowest distortion is retained whereas the rest of the paths are pruned. In fact, the nodes are the intermediate states in which a decision is taken regarding the *survivor* path. The *survivor* path is the one with the lowest total distortion cost reaching the examined node. The cost metric $D_{n_{12}}$ for a branch linking node n_1 and node n_2 in the trellis diagram is equal to

$$D_{n_{12}} = D'_k \cdot P(n > Q(k + 1), n \leq Q(k)) \quad (10)$$

where k is the block index and n is the number of erased packets.

The two paths displayed with a dashed line in Fig. 5(a) merged in node n_3 . The upper path corresponds to the allocation of equal amounts of protection for blocks 2 and 3, whereas the lower path corresponds to different RS protection. If block merging is not permitted, the conventional tree growing shown in Fig. 5(a) must be followed. If, however, block merging is assumed, the trellis diagram in Fig. 5(a) must be modified.

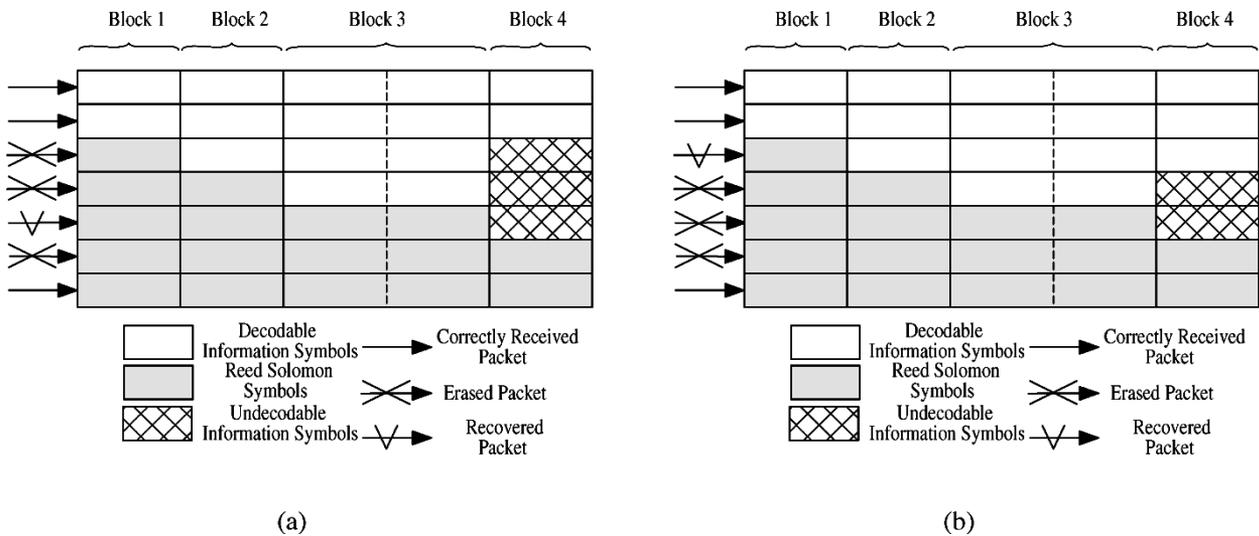


Fig. 6. Decodable bitstream after the application of successive decoding of product codes for the case shown in Fig. 3. (a) The fifth packet is recovered. (b) The third packet is recovered.

Fig. 5(b) shows the modifications occurring when two successive blocks are allocated the same protection. In this case, the node n_2 in the upper path is cancelled and the path linking n_1 and n_3 via n_2 is replaced by a new one connecting nodes n_1 and n_3 directly. This is due to the fact that blocks 2 and 3 share the same probability of failure. In a similar manner, whenever two or more consecutive blocks are allocated the same protection, the intermediate nodes are cancelled (only for these paths) and the initial and the final node are connected using a new branch.

According to (2), the added cost metric at node n_1 [see Fig. 5(b)] is equal to

$$D_{n_{12}} = D_0 \cdot P(n > Q(1)) \quad (11)$$

where n_{12} denotes the transition from node n_1 to n_2 . At node n_2 the metric added to the so far accumulated metric is

$$D_{n_{23}} = D_1 \cdot P(n > Q(2), n \leq Q(1)). \quad (12)$$

If the third block is allocated the same protection as the second block, then the cost $D_{n_{23}}$ representing the transmission of the previous second block alone is subtracted and a new cost $D_{n_{13}}$ is calculated corresponding to the transmission of an extended block having double the length of the elementary block. The extension of the above methodology to the case of three or more equally protected blocks is straightforward. The resulting modified trellis in its entirety is presented in Fig. 5(c). Using the approach described above, the optimal path in the sense of minimum reconstruction error at the decoder can be found directly. Results demonstrating the efficiency of our method are presented in the experimental evaluation section. Next, we describe a method to further improve the performance of our transmission schemes.

IV. SUCCESSIVE DECODING OF PRODUCT CODES

A. Successive Decoding

The product code scheme presented in the previous sections first attempts to correct transmission errors using turbo decoding. Whenever this is not possible, a packet erasure is

indicated by the CRC syndrome [5]. However, if across packets (i.e., along columns of the product code array) there is an appropriate number of redundant (RS) symbols, some erased blocks are corrected during the subsequent RS decoding stage. This fact can be exploited in a way similar to that used in [19] for improving the performance of a scheme based on concatenated turbo/RS codes. Specifically, using the coder proposed in the present paper, a successive decoding mechanism is invoked in cases where product decoding fails to correct all errors. This mechanism re-decodes turbo-coded packets that were initially undecodable using source information that was in the meantime restored by the intermediate RS decoding process. This technique is applicable only when some information is restored by the RS decoding stage, i.e., only when the number of initially undecodable packets is smaller than the number of redundant (RS) symbols in at least one of the transmitted blocks.

As seen in Fig. 1, the first segment of a turbo-coded packet contains source information and the other two are channel segments. An erased packet can be recovered using successive decoding if the source segment is partially recovered by means of information corrected during RS decoding. The injection of recovered information in a formerly undecodable packet may render the entire packet decodable using turbo decoding. The above technique is successively applied on erased packets and is terminated as soon as the turbo decoder is unable to recover any new source information in two successive iterations.

The result of successive decoding of product codes in the case described in Fig. 3 is illustrated in Fig. 6. After one iteration of the above technique, one packet initially indicated as erased in Fig. 3 can be recovered. This permits the successful decoding of the entire third block and the partial recovery of the fourth block. The decoded portion of the fourth block depends on the position of the first erased packet. For example, if the third packet is recovered, the decodable bitstream is depicted in Fig. 6(b). In any other case, the decoded bitstream is shown in Fig. 6(a). Note that despite the recovery of the entire fifth packet in Fig. 6(a), its rightmost information segment is still undecodable since it depends on the source segments above it which are corrupted.

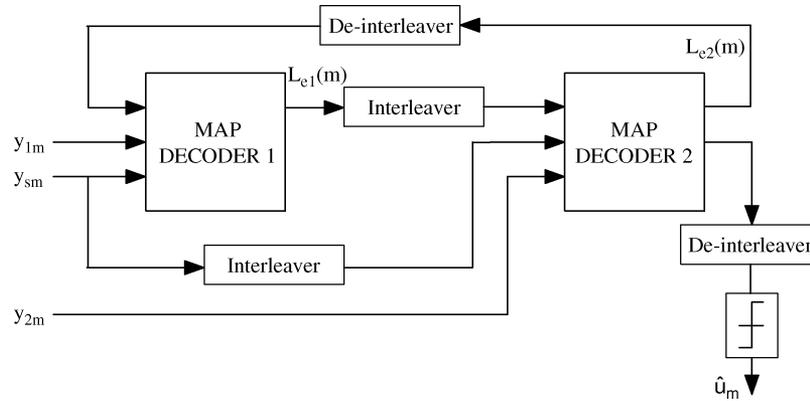


Fig. 7. Rate 1/3 turbo decoder.

It is noted that the above technique for the decoding of product codes is applicable only when UEP is used. This is due to the fact that *only* in the case of UEP, some initially uncorrectable packets may be partly recovered by the subsequent RS decoding and, thus, benefit from the reapplication of turbo decoding. In EEP cases, there are no partly decoded packets, and, therefore, the application of successive decoding is meaningless.

B. Exploitation of Recovered Information During Turbo Decoding

In usual turbo decoding applications, the entire stream to be decoded could possibly suffer from transmission errors. However, in the specific successive decoding process described above, parts of the stream may be *a priori* known to be uncorrupted. Therefore, a challenging task in the methodology described above is to determine an appropriate way for the exploitation of uncorrupted information in a turbo-coded packet, so as to optimize the turbo decoding process. This issue was formerly investigated in [20] in the context of convolutional decoding.

The turbo decoder used in the present paper consists of two soft-in/soft-out decoders which share probabilistic information cooperatively and iteratively [21]. The goal of this iterative decoding process is to estimate the *a posteriori* probabilities (APP) $\Pr(u_m|\mathbf{y})$ where u_m is the m th data bit, and \mathbf{y} is the received codeword (see Fig. 7). Each decoder receives from the other decoder *extrinsic* information for each bit, which serves as *a priori* information. The sign of a transmitted symbol is determined using a MAP rule [12]. Specifically, for the m th bit, the two extrinsic information terms, denoted $L_{e1}(m)$ and $L_{e2}(m)$, are computed in each iteration. The final decision on the value $\hat{u}(m)$ of the m th bit is taken using the formula

$$\text{sign}(\hat{u}(m)) = \begin{cases} +1, & \text{if } L_c \cdot y_s(m) + L_{e1}(m) + L_{e2}(m) > 0 \\ -1, & \text{if } L_c \cdot y_s(m) + L_{e1}(m) + L_{e2}(m) < 0 \end{cases} \quad (13)$$

where the L_c parameter depends on the channel [22].

Since, in our case, there exists an uncorrupted portion of the source segment, recovered during RS decoding, we disregard the contribution of $L_{e1}(m)$ and $L_{e2}(m)$ in (13) and base the decision for the value of these bits only on the term $L_c \cdot y_s(m)$.

In this way, bits known to be uncorrupted are always decoded correctly and are not affected by the turbo-decoding process. Furthermore, in the case of transmission of an uncorrupted bit sequence, the parameters $L_{e1}(m)$, $L_{e2}(m)$, and $y_s(m)$ always have the same sign. This can be exploited in our case by forcing the signs of $L_{e1}(m)$, $L_{e2}(m)$ to be the same as the sign of $y_s(m)$ for the bits known to be uncorrupted. The above modifications in the turbo decoding process improved its performance significantly.

In the ensuing section, the performance of our transmission scheme is experimentally evaluated and compared to the performance of other known methods.

V. EXPERIMENTAL RESULTS

The proposed scheme was experimentally evaluated for the transmission of the 512×512 test images “Lenna” and “Peppers” over a flat-fading Rayleigh channel [23] simulated using the Jakes model [24]. The channel model was selected so as to accurately simulate fading channels which are common in mobile and wireless communications. Using this model, the channel is characterized by two parameters, i.e., the average received signal-to-noise ratio (SNR) $\overline{\text{SNR}}$, which determines the average bit error rate, and the normalized Doppler spread³ (i.e., the Doppler spread normalized by division by the data rate), which determines how quickly the channel changes over time.

The method proposed in this paper was compared to the methods in [2], [6], for the case of normalized Doppler spread $f_D = 10^{-5}$ Hz/bps and $\overline{\text{SNR}}$ equal to 10 dB. The peak signal-to-noise ratio (PSNR) is used as a measure of the reconstruction quality.⁴ Our methodology, termed turbo-coded SPIHT (TCS), was implemented and tested in conjunction with two protection strategies, one using equal error protection (EEP) and the other using UEP. The TCS with successive decoding (termed TCSD) was also implemented and evaluated. All protection strategies were applied to the SPIHT source stream. A decoding example is shown in Fig. 8.

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

⁴Following the approach adopted in [2] and [6], the reported mean PSNR values are computed by averaging decoded MSE values and then converting the mean MSE to the corresponding PSNR value rather than averaging the PSNR values directly.

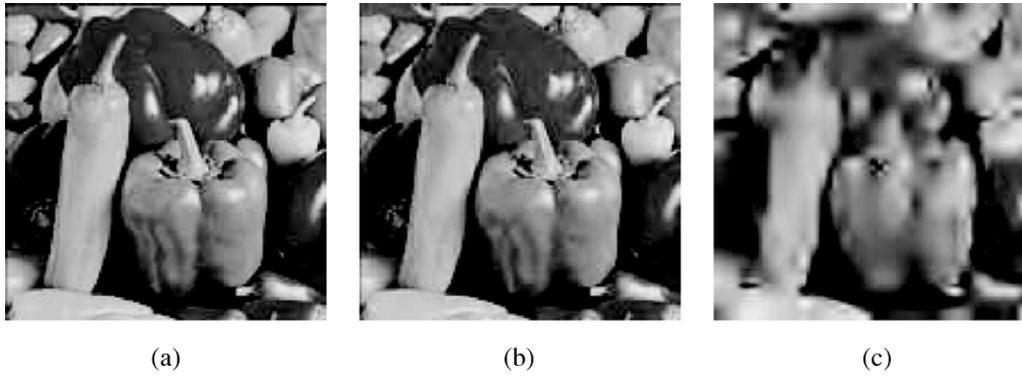


Fig. 8. Visual comparison for the “Peppers” image using the proposed method and the method in [2]. Comparison in terms of a random transmission (the images were transmitted and decoded using the exact same simulation of the channel). (a) Original. (b) TCS-UEP (28.95 dB). (c) Sherwood UEP2 (20.18 dB).

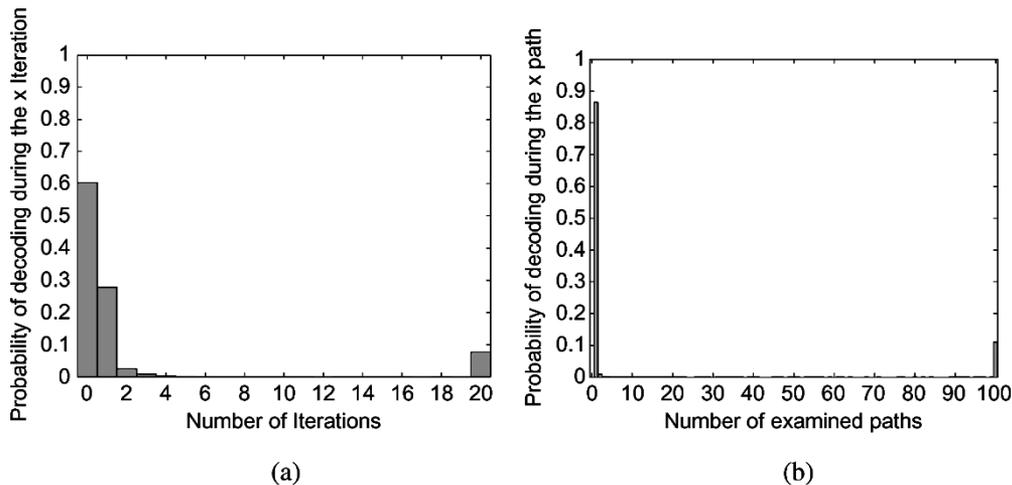


Fig. 9. Decoding iterations for turbo/CRC and number of examined paths for RCPC/CRC codes. (a) Turbo/CRC. (b) RCPC/CRC. Since, for all (eventually) undecodable cases, the maximum number of iterations/paths is required, the event of maximum number of iterations/paths occurs with a significant probability in both turbo and RCPC-based transmission schemes.

For the application of the proposed techniques, a product code consisting of 16 packets was used. In all cases, $(31, 27)_{\text{octal}}$ turbo codes [11] of rate $1/3$ were applied for the protection of symbols in the horizontal direction of the array, whereas a CRC-16 with generator polynomial $(254\ 465)_{\text{octal}}$ was used for the efficient detection of errors (see Fig. 2). The codes were punctured (with puncturing period 8) to yield a code rate equal to $1/2$. An S-random interleaver [25] with $S = 15$ was used with the turbo coding/decoding processes. The maximum number of turbo decoding iterations was 20. However, in practice, a few iterations usually suffice during decoding (see Fig. 9). The RS codes used with the proposed scheme are defined on $GF(2^8)$. This means that one RS symbol is equal to one byte. Assuming that 16 packets provide fine RS protection granularity, we reached the conclusion that for the sizes of layers generated by common bitplane coders (as is the SPIHT coder used in the present paper) the appropriate number of bytes in each row of a block is in the range 20–30. In the present work, we chose to use blocks with 25 bytes in each row. The use of even smaller blocks (i.e., more blocks in each row) would not be a good choice since the complexity of allocation would increase without a significant performance gain. Under the above assumptions, each source+channel packet included 511 bytes. In the case of EEP, $(16, 11)$ RS codes (determined by allowing only one protection level in the optimization algorithm

in Section III) were applied to the entire image stream which was organized in a row-by-row manner in the product code array. In the UEP case different amounts of protection were provided by RS codes in the vertical direction.

For the determination of the channel code rates in the UEP case, the algorithm of Section III was used. For a target rate of 0.25 bits/pixel, the number of elementary blocks were 10. The number of allocated RS symbols using our UEP-based method for the transmission of “Lenna” and “Peppers” are shown in Figs. 10 and 11, respectively. Figs. 10(a) and 11(a) present the allocation when the formation of extended blocks is not allowed. In Figs. 10(b) and 11(b), the allocation using the complete algorithm is shown. In general, the UEP algorithm without the extended block strategy overprotects the source information resulting in lower PSNR values in the case of errorless transmission. On the other hand, the allocation resulting from the application of an extended block strategy seems to enforce blocks use the same amount of protection. The extended block strategy yields a gain of approximately 0.4 dB in average PSNR. This gain comes about because the extended block approach not only results in more decodable stream in the case of unrecoverable corruption (as shown in Fig. 4) but also allows UEP to be more efficient. Thus, part of the gain results from the more efficient UEP.

The EEP and UEP schemes were evaluated for the transmission of images over wireless channels. Results for $f_D =$

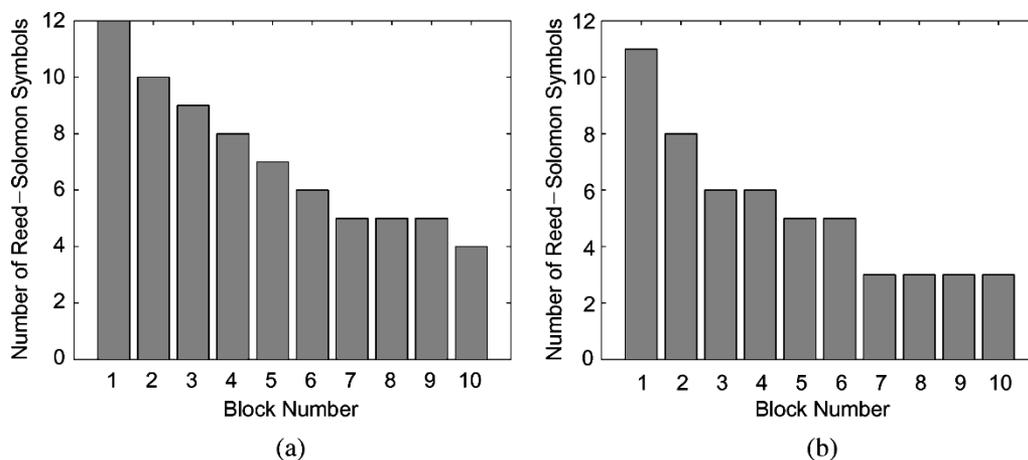


Fig. 10. Channel rates for the protection of "Lenna" as determined using the proposed algorithm for UEP. (a) Non-extended blocks are assumed. (b) Extended blocks are allowed.

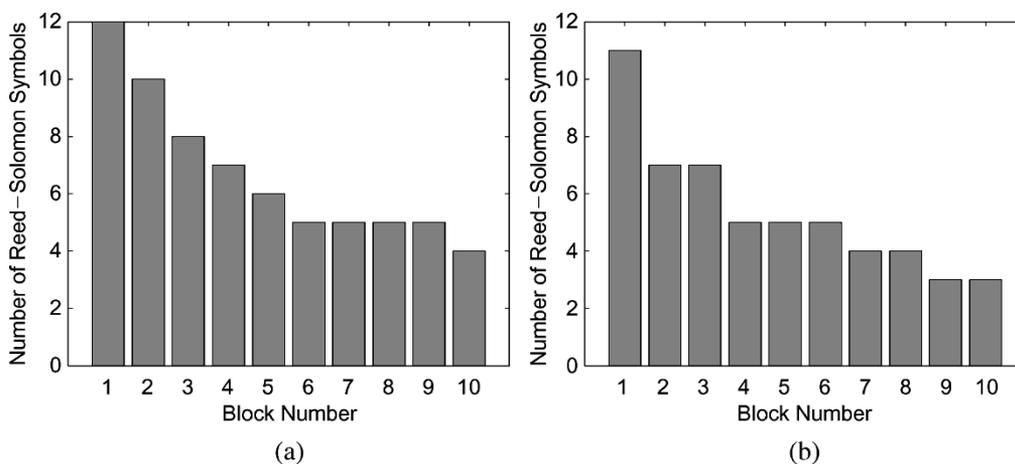


Fig. 11. Channel rates for the protection of "Peppers" as determined using the proposed algorithm for UEP. (a) Nonextended blocks are assumed. (b) Extended blocks are allowed.

TABLE I
PERFORMANCE COMPARISON FOR THE 512×512 "LENA" IMAGE (0.25 bpp). PSNR RESULTS ARE REPORTED IN DECIBELS. THE PROPOSED SCHEMES USING UEP OUTPERFORMS ALL OTHER SCHEMES IN THE COMPARISON IN TERMS OF AVERAGE QUALITY (MEAN PSNR)

Encoding	Code Rate	Mean PSNR
Sherwood and Zeger "UEP2" [2]	0.28	27.82
MDFEC rate-1/2 RCPC codes [6]	0.30	27.90
MDFEC rate-2/3 RCPC codes [6]	0.37	28.38
MDFEC rate-4/7 RCPC codes [6]	0.35	28.35
TCS-EEP	0.31	28.09
TCS-UEP	0.33	28.64
TCS-D	0.33	28.73

10^{-5} Hz/bps and $\overline{\text{SNR}}$ equal to 10 dB are reported in Tables I and II. As seen, our UEP scheme generally outperforms its EEP variant and yield average (expected) PSNR values, which are very close to the peak PSNR. Our coders were also compared to the SPIHT-based coders in [2] and [6]. In particular, our TCS-UEP scheme outperforms the system in [2] by 0.8 dB and

TABLE II
PERFORMANCE COMPARISON FOR THE 512×512 "PEPPERS" IMAGE (0.25 bpp). PSNR RESULTS ARE REPORTED IN DECIBELS. THE PROPOSED SCHEMES USING UEP OUTPERFORMS ALL OTHER SCHEMES IN THE COMPARISON IN TERMS OF AVERAGE QUALITY (MEAN PSNR)

Encoding	Code Rate	Mean PSNR
Sherwood and Zeger "UEP2" [2]	0.28	27.03
TCS-EEP	0.31	27.46
TCS-UEP	0.33	28.00
TCS-D	0.33	28.13

the best scheme in [6] by approximately 0.25 dB.⁵ The superior performance of our scheme is due to the use of turbo codes and the efficient channel-rate allocation. The TCS-D variant of our methodology provides additional improvement of about 0.1 dB.

⁵The method in [6] reports results at a target rate of 0.237 bits/pixel instead of 0.25 bits/pixel which is the target rate used in [2] and in this work. Therefore, the actual performance difference between the results in this work and the best results in [6] will be slightly lower.

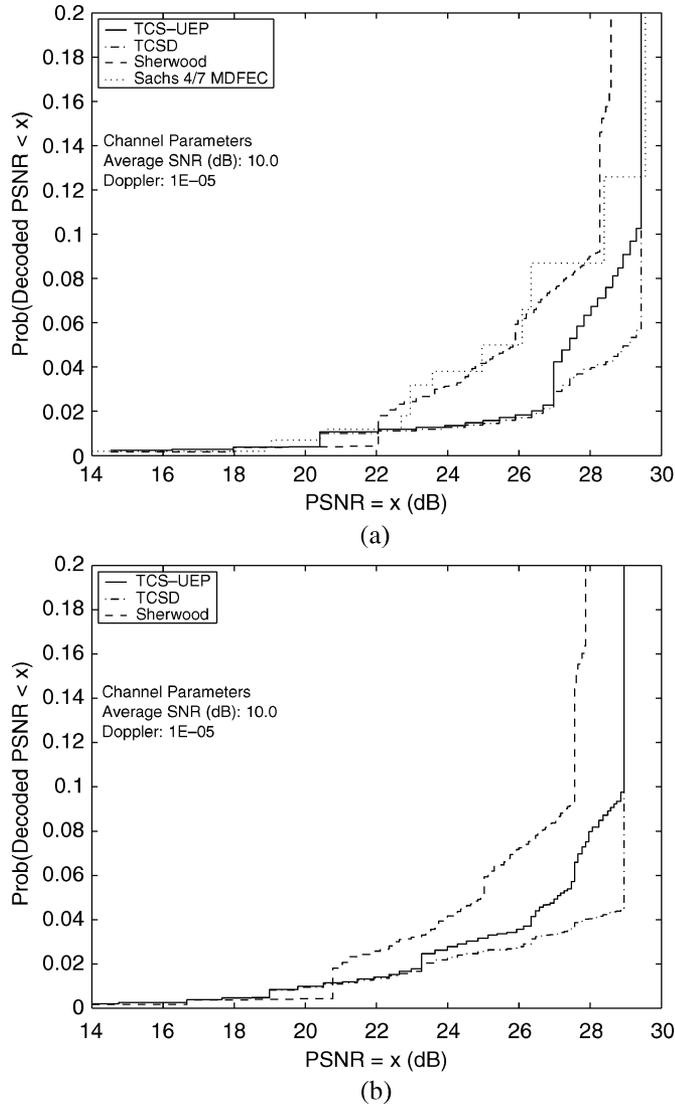


Fig. 12. Cumulative distribution of decoded PSNR for the 512×512 images (a) “Lenna” and (b) “Peppers,” transmitted at rate 0.25 bits/pixel over a Rayleigh fading channel with average SNR = 10.0 dB and doppler spread 10^{-5} .

Furthermore, the superiority of our method is seen even more clearly when comparisons are made in terms of *cumulative quality distributions* as reported in Fig. 12. As shown, the proposed UEP-based scheme produces low-quality images at the receiver far less often than the methods in [2] and [6]. The gain over the method in [2] in terms of the maximum achievable PSNR is approximately 1.0 dB, which is due to the higher code rate used by our scheme. This benefit comes without sacrificing the capability for robust transmission since our system achieve the maximum achievable PSNR more often. The TCSD scheme appears to be very efficient in cases where the quality of the reconstructed images is good. In such cases, successive decoding improves the reconstructed quality significantly.

The proposed schemes were also evaluated for transmission in channel mismatch conditions, i.e., when the actual channel parameters do not match the parameters assumed in the optimization process. Specifically, the stream was optimized for $f_D = 10^{-5}$ and $\overline{SNR} = 10$ dB, and transmitted over channels with \overline{SNR} other than that of the optimization (the f_D was not changed). Comparisons are reported in Tables III and IV. As seen, the UEP

TABLE III
AVERAGE MSE CONVERTED TO PSNR OF THE PROPOSED SCHEMES IN COMPARISON TO THE METHOD IN [2] FOR THE TRANSMISSION OF “LENNA” IN CHANNEL MISMATCH CONDITIONS. ALL SCHEMES WERE OPTIMIZED FOR $f_D = 10^{-5}$ AND $\overline{SNR} = 10$ dB. SIMULATIONS WERE CONDUCTED FOR SEVERAL MISMATCHED VALUES OF SNR. f_D WAS NOT VARIED

Encoding	\overline{SNR}			
	10	9	8	7
Sherwood “UEP2”[2]	27.82	27.52	26.91	26.61
TCS-EEP	28.09	27.42	26.76	25.90
TCS-UEP	28.64	28.19	27.64	26.96
TCSD	28.73	28.30	27.77	27.11

TABLE IV
AVERAGE MSE CONVERTED TO PSNR OF THE PROPOSED SCHEMES IN COMPARISON TO THE METHOD IN [2] FOR THE TRANSMISSION OF “PEPPERS” IN CHANNEL MISMATCH CONDITIONS. ALL SCHEMES WERE OPTIMIZED FOR $f_D = 10^{-5}$ AND $\overline{SNR} = 10$ dB. SIMULATIONS WERE CONDUCTED FOR SEVERAL MISMATCHED VALUES OF SNR. f_D WAS NOT VARIED

Encoding	\overline{SNR}			
	10	9	8	7
Sherwood “UEP2”[2]	27.03	26.69	26.04	25.71
TCS-EEP	27.46	26.73	26.02	25.10
TCS-UEP	28.00	27.50	26.87	26.06
TCSD	28.13	27.67	27.07	26.29

schemes perform better in channel mismatch conditions than the EEP schemes. The TCSD scheme in particular outperforms all other methods in the comparison. Moreover, the performance gap between TCSD and TCS-UEP widens to at least 0.2 dB. Another comparison is reported in Fig. 13. As seen, the scheme based on the TCS-UEP coder achieves significantly better performance in comparison to the scheme in [2].⁶ The TCSD system, based on successive decoding, further improves the results.

An instance in our simulations, which demonstrates well the reliability of our TCS-UEP methodology, is shown in Fig. 8. Our system is compared to that in [2] for the transmission of “Lenna” over a wireless channel with $f_D = 10^{-5}$ and average SNR = 10 dB. In this specific example, a deep fade at the beginning of transmission produces an uncorrectable error in the stream produced by the coder in [2] and renders most of the stream useless. On the contrary, when our robust methodology is used, all errors are corrected yielding superior reconstruction quality.

The application of successive decoding in a large number of cases provides useful insight regarding its effectiveness. In particular, in cases of long fades, the successive decoding cannot provide much assistance. However, in cases in which images encounter short fades during transmission, the probability of full restoration of an image is considerable. Experimentation also showed that this technique is most effective when applied to blocks in which the initially undecodable packets contain one or

⁶Comparison with [6] in the case of mismatch is not possible since such results are not included in [6].

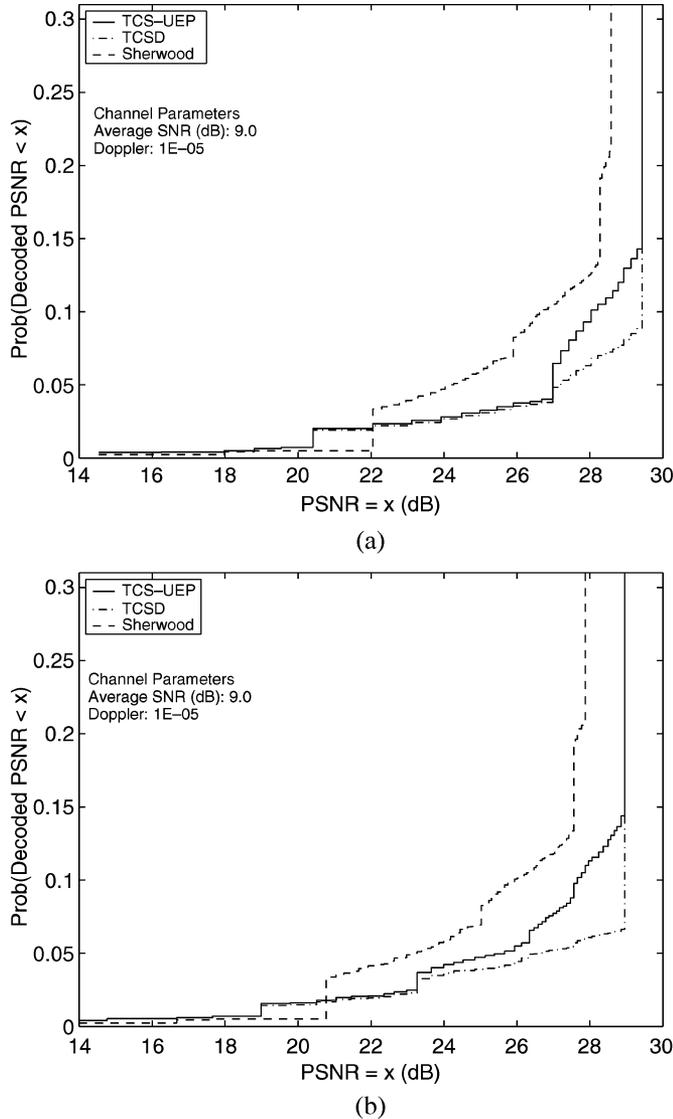


Fig. 13. Comparison in channel mismatch conditions. Cumulative distribution of decoded PSNR for the 512×512 images (a) “Lenna” and (b) “Peppers,” transmitted at rate 0.25 bits/pixel over a Rayleigh fading channel with average SNR = 9.0 dB and doppler spread 10^{-5} . SNR = 10 dB was assumed for the optimization of the rate allocation.

two redundant symbols fewer than the number of RS symbols in the first undecodable block. An obvious conclusion reached by the examination of Figs. 12 and 13 is that more than half of the cases in which there is a transmission error, the application of successive decoding yields concrete benefits in terms on PSNR gain.

VI. CONCLUSION

A novel image transmission scheme was proposed for the communication of compressed SPIHT image streams over wireless channels. The proposed scheme employs product codes consisting of turbo codes and erasure-correction codes in order to deal effectively with burst errors. A novel methodology for the optimal UEP of compressed streams was also proposed and applied in conjunction with an inherently more efficient technique for the decoding of product codes. The resulting system

was tested for the transmission of images over wireless channels. Experimental evaluation showed the superiority of the proposed schemes in comparison to well-known robust coding schemes.

APPENDIX I COMPLEXITY ANALYSIS

The utilized turbo/CRC codes, presented in detail in the experimental results section, employ a mother code with code rate $1/3$. The turbo codes were subsequently punctured with puncturing period 8, yielding an eventual turbo code rate equal to $1/2$. The turbo decoding was performed using two BCJR-MAP decoders with maximum number of iterations equal to 20. The parameters of the RCPC/CRC codes are identical to that of [2], i.e., the code rate of the RCPC code was $1/4$, the generator polynomial of the CRC was $(254\ 465)_{\text{octal}}$ and the RCPC generator polynomial was $(155\ 123\ 137\ 147)_{\text{octal}}$. For the decoding of the RCPC code the list Viterbi algorithm [4] was used, as in [2], with a list of 100 candidate paths. Both turbo/CRC and RCPC/CRC codes have identical code rates (equal to $1/2$) after puncturing.

As stated in [26], the computation of the α_t (or β_t) parameter of the BCJR-MAP decoder requires $M \cdot 2^{k_0}$ additions and $M \cdot 2^{k_0}$ multiplications per time instance t , where M denotes the number of states in the trellis diagram and k_0 the number of inserted bits in the shift register. The computation of the APP requires M multiplications and $k_0 \cdot M/2$ additions, while the computation of the γ_t parameter is accomplished by means of a lookup table. Thus, a BCJR-MAP decoder requires $2 \cdot 2^{k_0} \cdot M + k_0 \cdot M/2$ additions and $2 \cdot 2^{k_0} \cdot M + M$ multiplications. Since the turbo coder has two-component BCJR-MAP decoders, the total computational cost is $2 \cdot (2 \cdot 2^{k_0} \cdot M + k_0 \cdot M/2)$ additions and $2 \cdot (2 \cdot 2^{k_0} \cdot M + M)$ multiplications. On the other hand, Viterbi decoding requires $M \cdot 2^{k_0}$ additions and $M \cdot 2^{k_0}$ comparisons in each time instance.

For the turbo coder used by our product coder, $k_0 = 1$ and $M = 16$. Hence, in each time instance, 160 multiplications and 144 additions are required. The system of [2] has $k_0 = 1$ and $M = 64$. Therefore, 128 comparisons and 128 additions are performed in each time instant. The complexity of multiplication for 32-bit numbers can be as little as four times the complexity the addition (see, e.g., [27]). The comparisons have approximately the same complexity as the additions. Hence, for the parameters listed above, the complexity of turbo decoding is roughly three times that of the RCPC.

In order to estimate the complexity of decoding for the turbo codes in comparison to RCPC codes, we compute the probability density function (pdf) of the number of paths that are needed for the transmission of “Lenna” over a wireless channel with $\overline{\text{SNR}} = 10$ dB and normalized Doppler spread $f_D = 10^{-5}$ Hz/bps. The resulting pdfs, termed P_{RCPC} and P_{turbo} , are shown in Fig. 9. We define the following complexity functions:

$$C_{\text{RCPC}} = \sum_{i=1}^{100} P_{\text{RCPC}}(i) \cdot i$$

$$C_{\text{turbo}} = \sum_{i=0}^{20} P_{\text{turbo}}(i) \cdot 3 \cdot i.$$

Note that, for the RCPC case, the lowest path index i is equal to 1 since at least 1 path is required for RCPC decoding. On the con-

rary, the lowest iteration index for the turbo case is 0 since whenever no error is detected turbo decoding is not required. As seen in Fig. 9(a), for the test conditions described earlier, roughly 60% of the time, there is no need for turbo decoding. For the pdfs in Fig. 9, it is calculated that the average complexity of RCPC/CRC and turbo/CRC decoding is 12.44 and 5.99, respectively. Even if we assumed RCPC/CRC decoding with a reduced maximum number of paths (this would make the decoding less efficient in channel mismatch conditions), the associated complexity would be at least as much as that of the turbo-based scheme. Therefore, the proposed scheme, based on turbo codes, is of similar overall complexity while being more efficient than the RCPC-based schemes that are included in our comparisons.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive reviews.

REFERENCES

- [1] 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, pp. 243–250, Jun. 1996.
- [2] 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.
- [3] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC Codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1989.
- [4] 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.
- [5] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [6] D. G. Sachs, A. Raghavan, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated codes," presented at the Data Compression Conf., 2000.
- [7] 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.
- [8] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correcting codes," in *Proc. Asilomar Conf. Signals Syst. Comp.*, vol. 1, Pacific Grove, CA, 1999, pp. 342–346.
- [9] 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.*, vol. 9, no. 6, pp. 982–993, Jun. 2000.
- [10] G. Sherwood and K. Zeger, "Progressive image coding on noisy channels," *IEEE Signal Process. Lett.*, vol. 4, no. 7, pp. 189–191, Jul. 1997.
- [11] 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.
- [12] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo codes (1)," in *Proc. IEEE Int. Conf. Communications*, vol. 2, Geneva, Switzerland, May 1993, pp. 23–26.
- [13] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [14] A. Albanese, J. Bloemer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inf. Theory*, vol. 42, no. 11, pp. 1737–1744, Nov. 1996.
- [15] 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.
- [16] G. Davis and J. Danskin, "Joint source and channel coding for image transmission over lossy packet networks," in *Proc. SPIE*, Apr. 1996, pp. 376–387.
- [17] V. Chande and N. Farvardin, "Progressive transmission of images over memoryless noisy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 850–860, Jun. 2000.
- [18] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.
- [19] J. He, D. J. Costello, Y. F. Huang, and R. L. Stevenson, "On the application of turbo codes to the robust transmission of compressed images," in *Proc. Int. Conf. Image Processing*, Washington, DC, Oct. 1997, pp. 559–562.
- [20] O. M. Collins and M. Hizlan, "Determinate state convolutional codes," *IEEE Trans. Commun.*, vol. 41, no. 12, pp. 1785–1794, Dec. 1993.
- [21] Ö. Açikel and W. Ryan, "Punctured turbo-codes for BPSK/QPSK channels," *IEEE Trans. Commun.*, vol. 47, no. 9, pp. 1315–1323, Sep. 1999.
- [22] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.
- [23] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [24] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.
- [25] D. Divsalar and F. Pollara, "Turbo codes for pcs applications," in *Proc. ICC*, Seattle, WA, Jun. 1995, pp. 54–59.
- [26] L. R. Bahl, J. Cocke, F. Jelinek, and R. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.
- [27] D. E. Knuth, *The Art of Computing*, 2nd ed. Reading, MA: Addison-Wesley, 1981, vol. 2, *Seminumerical Algorithms*.



Nikolaos Thomos (S'02) was born in Cologne, Germany. He received the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2000, where he is currently pursuing the Ph.D. degree.

He holds research and teaching assistantship positions in the Electrical and Computer Engineering Department, Aristotle University of Thessaloniki. He is also a Graduate Research Assistant with the Informatics and Telematics Institute, Thessaloniki. His research interests include image/video coding and transmission, multimedia networking, wavelets, and digital filters.

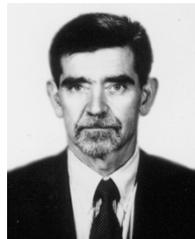
Mr. Thomos is a member of the Technical Chamber of Greece.



Nikolaos V. Boulgouris (S'96-M'04) received the Diploma and the Ph.D. degrees from the Electrical and Computer Engineering department of the 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, Thessaloniki. 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 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.