Efficient Rate-Distortion Optimal Packetization of Embedded Bitstreams into Independent Source Packets

Sorina Dumitrescu, Jiayi Xu

Abstract

This work addresses the rate-distortion optimal packetization (RDOP) of embedded bitstreams into independent source packets, in order to limit error propagation in transmission of images over packet noisy channels. The input embedded stream is assumed to be an interleaving of K independently decodable basic streams. To form N ($N < K$) independent source packets, the set of basic streams is partitioned into N groups. The streams within each group are then interleaved to generate a source packet. Error/erasure protection may be further applied along/across source packets, to produce the channel packets to be transmitted.

The RDOP problem previously formulated by Wu *et al.* has the goal of finding the partitioning which minimizes the distortion when all source packets are decoded. We extend the problem formulation such that to also include the minimization of the expected distortion for general transmission scenarios which may apply uneven erasure/error protection. Further, we show that the dynamic programming algorithm of Wu *et al.* can be extended to solve the general RDOP problem.

The main contribution of this work is a fast divide and conquer algorithm to find the globally optimal solution, under the assumption that all basic streams have convex rate-distortion curves. Instrumental in obtaining the fast solution is our result which proves that the problem can be formulated as a series of matrix search problems in totally monotone matrices. The proposed divide and conquer algorithm reduces the running time from $O(K^2LN)$ achieved by the dynamic programming solution, to $O(NKL\log K)$. Experiments on SPIHT coded images demonstrate that the speed up is significant in practice.

Index Terms

The authors are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada L8S 4K1. Their email addresses are sorina@mail.ece.mcmaster.ca, xujiayi@grads.ece.mcmaster.ca

Rate-distortion optimization, partitioning in independent source packets, robust image/video transmission, totally monotone matrix, divide and conquer algorithm.

I. Introduction

The problem of robust data transmission over packet noisy channels has been the subject of intense research during the past fifteen years. A plethora of techniques to alleviate the effect of packet loss or corruption have been proposed, based on retransmission, forward error correction (FEC), error concealment or combinations of these.

Progressive image or video coders such as SPIHT [1], EBCOT [2], or 3DSPIHT [3], are very attractive due to their ability to easily adapt to rate variations. However, they are highly susceptible to error propagation. In an embedded (or progressive) data stream the quality of the reconstruction increases gradually as more symbols are decoded. On the other hand, a symbol can be decoded only if all the previous symbols are available. Thus, a lost or corrupted packet inevitably disables the decoding of other source symbols from received and correct packets. A natural way to confine the error only within the packet it occurs, is to break the decoding dependence between packets. This can be done by partitioning the data samples (for instance, the wavelet coefficients) into groups, then encoding each group independently and transmitting the resulting progressive stream within a packet. This idea has been used in [4]-[12] in conjunction with FEC or error concealment and in several multiple description schemes designed for image or video transmission [13]-[18].

On the other hand, progressive image coders already employ the partitioning of the set of samples into basic subsets, which are independently encoded producing embedded substreams. These substreams are further interleaved in optimal rate-distortion (R-D) sense to generate a single coded stream for the entire data. Specifically, in SPIHT, each basic subset is a spatial orientation tree of coefficients, while in JPEG2000 each basic subset is a block of coefficients of fixed dimension, within a subband. We will use the term *basic* or *primary* streams for these embedded bitstreams encoding each basic subset of samples. This feature can be exploited in the construction of independent source packets. In particular, in the case when the number K of basic streams is higher than the required number N of packets, the primary streams can be grouped into N pairwise disjoint groups. Next the streams within each group are interleaved to generate a separate source packet. Adopting this procedure reduces the task of partitioning the whole set of samples to the simpler task of partitioning the smaller set of basic subsets, or in other words, into partitioning the set of basic streams. Many researchers have taken this approach to generate independent source packets, however most have used ad-hoc methods to choose the partition. One such example is the work of Rogers and Cosman [5]. They developed a scheme for robust transmission of

SPIHT coded images over packet lossy channels, where no channel coding is applied to the independent source packets, but error concealment is used at the decoder to recover from losses. The basic streams (i.e., the code streams corresponding to individual trees) are assumed to be in a fixed order known to both the encoder and decoder. The groups are formed with consecutive basic streams until approximately reaching the capacity of the fixed size packet. To fit exactly the size of the packet, the interleaved stream is either truncated or extended, as necessary.

The first work which clearly addresses the problem of optimal packetization of embedded streams into independent source packets, is the work of Wu *et al.* [9]. The construction of source packets is also based on partitioning the set of basic streams into groups, but the packetization constraints are more relaxed than in [5] by also allowing a group which contains only one basic stream, to occupy more than one packet. On the other hand, the condition that each group is formed only from consecutive packets in the fixed ordering, is maintained in order to reduce the amount of side information needed to specify the composition of each group. The goal of the optimization problem is, given a fixed budget of N packets, each containing L symbols, to minimize the distortion when all the packets are decoded. The authors of [9] propose a globally optimal dynamic programming solution. However, as their simulations show, this solution is impractical due to the high overload of precomputations.

This work reexamines the problem of R-D optimal packetization (RDOP) of embedded streams into independent source packets based on partitioning the set of basic streams. In our formulation we impose the constraint of [5] that each group is formed out of consecutive primary streams according to a fixed ordering, and that the interleaved stream of each group occupies a single packet. We assume a general transmission scenario where the source packets may be further protected against bit errors and/or erasures. We consider a general formulation of the RDOP problem which includes as special cases: 1) the minimization of the expected distortion and 2) the minimization of the total distortion when all source packets are decoded.

Due to similarities in the problem formulation, the dynamic programming algorithm of [9] can be extended to solve the general RDOP problem. Our main contribution is a more efficient globally optimal solution to the problem. To make possible the accelerated solution, we assume that the R-D curve of each basic stream is a convex function. This assumption has been widely used in previous work and is a good approximation for practical situations. Based on this assumption we prove that the problem can be reduced to a series of matrix search problems in totally monotone matrices[1]. Leveraging this result we devise an efficient globally optimal solution based on the divide and conquer strategy. The proposed algorithm

[1]This notion was introduced in [19] and will be defined precisely in Section IV.

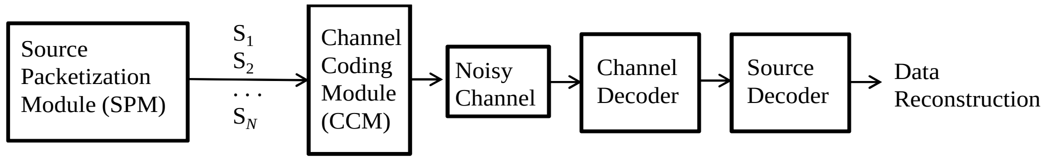


Fig. 1. Block diagram of the general transmission scheme.

reduces the running time from $O(K^2LN)$, achieved via dynamic programming, to $O(NKL\log K)$. We mention that the divide and conquer solution for the special case of the RDOP problem which minimizes the distortion when all source packets are decoded, appeared first in the conference publication [20].

The structure of the paper is as follows. In section II the RDOP problem is formulated for a general transmission scenario which consists of a source packetization module followed by a channel coding module. Section III presents the globally optimal dynamic programming solution algorithm. The next section contains the main contribution of this work. First the RDOP problem is equivalently cast as a series of matrix search problems and it is proved that each component matrix is totally monotone. Based on this result the efficient globally optimal divide and conquer algorithm is developed. Further, Section V discusses the application of the RDOP problem in specific transmission schemes using uneven erasure/error protection, including the schemes introduced in [12]. Experimental results with SPIHT coded images are presented in Section VI and Section VII concludes the paper.

II. Problem Formulation

A. General Transmission Scheme

We consider a general scheme for image transmission over noisy channels, which is composed of a source packetization module (SPM) followed by a channel coding module (CCM). The block diagram of such a system is depicted in Fig. 1. The SPM generates N independent source packets, where each source packet is an embedded bitstream of some given size. The CCM applies FEC to protect against erasures and/or bit errors, and outputs a number of channel packets, which are further transmitted over the noisy channel. At the receiver side all or only a subset of the transmitted channel packets may arrive at destination. Additionally, if the channel also incurs bit errors, some of the received packets may be corrupted. From the received data the channel decoder recovers part or all of the transmitted source symbols using the channel error protection. Further, the source decoder decodes the available source symbols providing a reconstruction for each data sample.

Let us now give some more details about the SPM. It is assumed that the set of samples is divided into non-overlapping *basic subsets* $\mathcal{B}_1, \cdots, \mathcal{B}_K$, for some $K, K > N$. The SPM partitions the total set of samples into N non-overlapping groups $\mathcal{G}_1, \cdots, \mathcal{G}_N$, such that each group to be the union of some basic subsets. Further, a progressive coder is used to encode the samples in each group. We will denote by G_n the progressive codestream encoding group $\mathcal{G}_n$ at the highest possible resolution. Further, for each $n, 1 \leq n \leq N$, source packet S_n consists of a prefix of G_n, if necessary also including a header to specify the grouping.

In practical examples for image transmission where the progressive encoder is SPIHT or JPEG2000, the data samples are the wavelet coefficients. In the case of SPIHT, each basic subset of samples can be a spatial orientation tree of coefficients, while for JPEG2000, each basic subset can be a block of coefficients of fixed dimension, within a subband.

B. Assumptions

Our problem formulation is based on some assumptions which hold approximately in practice. Specifically, for each $k, 1 \leq k \leq K$, let B_k denote the bitstream encoding subset $\mathcal{B}_k$ at the highest resolution, by using the specified progressive coder. We will refer to bitstreams $B_1, \cdots, B_K$, as *basic streams*. Notice that when the progressive coder is SPIHT, the bitstream G_n is actually an interleaving of the basic bitstreams B_k corresponding to the basic sets $\mathcal{B}_k$ which form the group $\mathcal{G}_n$. This statement is also true for the JPEG2000 coder if we disregard the side information which needs to be inserted in $\mathcal{G}_n$ to specify the composition of each quality layer. In this work rather then regarding B_k and G_n as sequences of bits, we will regard them as sequences of symbols, where a symbol is a sequence of a fixed number of bits (8 bits in our tests). We will consider a stronger assumption, namely that individual symbols are basic data units, which are not broken in the interleaving. This assumption is formulated next.

A1. For each $n, 1 \leq n \leq N$, the sequence of symbols G_n is an interleaving of the sequences of symbols B_k corresponding to the basic sets $\mathcal{B}_k$ which form the group of samples $\mathcal{G}_n$.

The next assumption is based on the practical observation that the R-D curve of a codestream output by SPIHT or JPEG2000 is nearly convex.

A2. Each basic codestream B_k, $1 \leq k \leq K$, has a convex R-D curve.

For each $k, 1 \leq k \leq K$, let l_k denote the number of symbols in B_k, and let $\Delta_k(i)$ denote the decrease in distortion induced by decoding the i-th symbol of B_k, given that all previous $i-1$ symbols have been decoded, for $1 \leq i \leq l_k$. Then Assumption A2 is equivalent to

$$\Delta_k(i) \geq \Delta_k(i+1), \quad \text{for all } 1 \leq i \leq l_k - 1, 1 \leq k \leq K. \tag{1}$$

Further, the assumption that the R-D curve of a codestream output by SPIHT or JPEG2000 is convex, together with A1 and A2 lead naturally to assumption A3 stated next.

A3. For each $n, 1 \leq n \leq N$, the interleaving of symbols of basic streams in G_n is performed in optimal R-D manner, in other words, such that the symbols are ordered in non-increasing order of their associated distortion reductions.

C. Constraint on Source Packetization

In order to reduce the amount of side information needed to describe the grouping to the decoder we impose the same constraint as in [5] and [9], specifically, that each group of samples $\mathcal{G}_n$ is the union of consecutive basic sets according to some fixed ordering known to both encoder and decoder. Let us assume that the basic sets $\mathcal{B}_k$ are labeled according to this ordering. Then this constraint can be equivalently formulated as follows.

C1. There are $N+1$ integers $p_0, p_1, \ldots, p_N$, such that $0 = p_0 < p_1 < p_2 < \cdots < p_N = K$, and for each $n, 1 \leq n \leq N$, the following holds

$$\mathcal{G}_n = \mathcal{B}_{p_{n-1}+1} \cup \mathcal{B}_{p_{n-1}+2} \cup \cdots \cup \mathcal{B}_{p_n}. \quad (2)$$

Finally, source packet S_n is formed out of the first L_n source symbols of codestream G_n, to which a header of $2\lceil \log_2 K \rceil$ bits is appended. The header encodes the pair of integers $(p_{n-1}+1, p_n)$.

D. RDOP Problem

The aim of this section is to formulate an R-D optimization problem for the SPM. A commonly used criterion in R-D optimization is the minimization of expected distortion. In this work we will also consider the expected distortion as a performance measure. However, rather than formulating the problem with this specific goal in mind, we consider a more general formulation which includes the case when the cost function is the expected distortion, but also includes other cases where the cost function may have different meanings. The reason for this generality is to be able to accommodate cases when the expected distortion of the whole transmission scheme depends on more variables which may be unknown to the SPM.

We will assume that the values $L_n, 1 \leq n \leq N$, are fixed and that some positive coefficients $\gamma(n,r), 1 \leq$

$r \leq L_n, 1 \leq n \leq N$, are given[2], satisfying the condition[3]

$$\gamma(n,r) \geq \gamma(n,r+1), \text{ for all } r,n, 1 \leq r \leq L_n, 1 \leq n \leq N. \tag{3}$$

We will further assume that statements A1, A2 and A3 hold. Then the problem of R-D optimal source packetization (RDOP) is formulated as

$$\begin{aligned} &\text{minimize}_{\mathcal{G}_1,\mathcal{G}_2,\cdots,\mathcal{G}_N} \quad \left(D_0 - \textstyle\sum_{n=1}^{N}\sum_{r=1}^{L_n}\gamma(n,r)\Delta D_{G_n}(r)\right), \\ &\qquad\qquad \text{subject to} \qquad\qquad \text{constraint C1} \end{aligned} \tag{4}$$

where D_0 is the distortion achieved by the signal reconstruction when no information is available, and $\Delta D_{G_n}(r)$ denotes the distortion decrease incurred by decoding the r-th symbol of codestream G_n.

The meaning of the cost function in (4) depends on the choice of the coefficients $\gamma(n,r)$. If the channel statistics are known and the redundancy allocation (RA) in the CCM is decided independently of the SPM, then after determining the RA, the values L_n can be computed and each $\gamma(n,r)$ can be chosen as the probability that the r-th symbol of codestream G_n, is decoded by the source decoder. Then the objective function of (4) represents the expected distortion of the signal reconstruction at the receiver. Notice that with this choice of the coefficients, $\gamma(n,r)$ is actually the probability that all first r source symbols of source packet S_n are correctly recovered by the channel decoder, because the r-th source symbol cannot be successfully decoded by the source decoder if any of the previous $r-1$ symbols is missing. Therefore, relations (3) hold. Recall that we use the expected distortion as a measure of the system performance. Therefore, for such a choice of coefficients $\gamma(n,r)$, the RDOP problem finds the source packetization which maximizes the system performance. This formulation of the RDOP problem is possible when the RA in the CCM is determined without needing the knowledge of the partitioning performed in the SPM. This is the case of the FMUEP scheme introduced in [12], which will be discussed in more detail in Section V.

Now notice that by letting $\gamma(n,r) = 1$ for all $1 \leq r \leq L_n, 1 \leq n \leq N$, the cost function in (4) becomes the distortion achieved when all source packets are decoded. This cost function was used in [9] to optimize the SPM for a transmission scheme over packet lossy channels, which does not contain the CCM, but uses error concealment at the source decoder. Such a criterion was used for SPM

[2]Examples of choices of such coefficients and the corresponding meaning of the cost function will be discussed shortly.

[3]This condition is essential for achieving the fast solution algorithm and, as it will be seen shortly, it is satisfied in the cases of interest, in particular when the cost function of (4) is the expected distortion.

optimization in [12] too, for both MUEP and FMUEP schemes, before optimizing the RA[4]. In both works [9] and [12] the optimized packetization under this criterion showed consistent performance improvement comparing with non-optimized packetization. Thus, the results of [9] and [12] support the hypothesis that the RDOP problem with a heuristic choice of coefficients $\gamma(n,r)$ can also be useful for increasing the system performance in transmission scenarios where the formulation of the RDOP problem for minimum expected distortion is not possible, such as schemes where the RA in the CCM is decided depending on the output of the SPM. On the other hand, for such schemes there is another obvious way to use the RDOP problem to improve the performance, as follows. After an initial choice of the source packetization, optimize the RA. Then, for the given RA optimize the SPM for minimum expected distortion by solving the RDOP problem. The process may stop here or continue iteratively by optimizing again the RA for the fixed source packetization, then the source packetization for the fixed RA and so on, until no additional performance improvement is achieved.

E. Equivalent Formulation of the RDOP Problem

In this subsection we provide an equivalent formulation of the RDOP problem which is more convenient for the description of the solution algorithms.

In view of assumptions A1, A2 and A3, and of relation (2), codestream G_n is the interleaving in optimal R-D manner of codestreams $B_{p_{n-1}+1}, B_{p_{n-1}+2}, \cdots, B_{p_n}$. Thus, the SPM problem of partitioning the data samples into groups can be actually regarded as the problem of partitioning the set of basic streams into groups. Further, notice that this partitioning is completely specified by the $(N+1)$-tuple of integers $\mathbf{p} = (p_0, p_1, \ldots, p_N)$. To make more clear the dependence of the cost function on $p_0, p_1, \ldots, p_N$, we will introduce the following notation. For each pair j, k, $1 \le j \le k \le K$, let $B(j,k)$ denote the sequence of symbols obtained by interleaving in optimal R-D manner the basic streams $B_j, B_{j+1}, \cdots, B_k$. We will refer to such a codestream as a *composite stream*. We point out again that, based on assumption A3, $B(j,k)$ is actually the codestream encoding the set of samples $\mathcal{B}_j \cup \mathcal{B}_{j+1} \cup \cdots \cup \mathcal{B}_k$ at the highest resolution. Now, for each composite stream $B(j,k)$, $1 \le j \le k \le K$, and for each packet index $n, 1 \le n \le N$, let the quantity $w_n(j,k)$ be defined as

$$w_n(j,k) = \sum_{r=1}^{L_n} \gamma(n,r) \Delta_{j,k}(r), \tag{5}$$

[4]In MUEP and FMUEP each channel packet includes symbols from a source packets along with additional redundancy symbols. The size of a channel packet is fixed, thus, the size of a source packet is determined by the RA. However, in the RDOP problem formulation considered in [12], source packets were assumed to be equal in size with channel packets.

where $\Delta_{j,k}(r)$ denotes the distortion decrease incurred by decoding the r-th symbol of $B(j,k)$. It is clear now that the objective function of (4) equals

$$D_0 - \sum_{n=1}^{N} w_n(p_{n-1}+1, p_n).$$

We conclude that the RDOP problem (4) can be equivalently cast as

$$\max_{\mathbf{p}} \sum_{n=1}^{N} w_n(p_{n-1}+1, p_n), \tag{6}$$

where the maximization is performed over all $(N+1)$-tuples of integers $\mathbf{p} = (p_0, p_1, \ldots, p_N)$ satisfying $0 = p_0 < p_1 < p_2 < \cdots < p_N = K$.

III. Dynamic Programming Solution

This section presents the globally optimal solution to the RDOP problem, obtained by adapting the dynamic programming algorithm of [9]. The algorithm complexity is analyzed and a special case which leads to computational savings is pointed out.

Notice that each source packet n contributes one term in the objective function of (6). Let $C_{opt}(n,k)$ denote the maximumum contribution due to the first n source packets, assuming that they are formed by partitioning the subset of basic streams $B_1, B_2, \ldots, B_k$, i.e.,

$$C_{opt}(n,k) = \max_{0=p_0<p_1<p_2<\cdots<p_n=k} \sum_{i=1}^{n} w_i(p_{i-1}+1, p_i).$$

Then the following recursion holds

$$C_{opt}(n,k) = \max_{0 \le j < k} \{C_{opt}(n-1,j) + w_n(j+1,k)\}, \tag{7}$$

for $1 \le n \le N$, $1 \le k \le K$, where $C_{opt}(0,j) = C_{opt}(n,0) = 0$ for $0 \le j \le K-1$, $1 \le n \le N$. The dynamic programming algorithm computes the values $C_{opt}(n,k)$ for all $1 \le n \le N$ and $1 \le k \le K$, in lexicographical order of the pairs n,k. At the end, the quantity $C_{opt}(N,K)$ is the optimal value of the objective function of (6). If all values $w_n(j,k)$ are already known, then the number of operations to solve (7) for fixed pair n,k is $O(K)$, amounting to a total of $O(K^2N)$ operations over all pairs n,k.

On the other hand, the computation of the N upper triangular matrices W_n with elements $w_n(j,k)$, $1 \le j \le k \le K$, $1 \le n \le N$, has to be accounted for as well. For this, matrix positions (j,k) with $1 \le j \le k \le K$, are visited starting at the top left corner and proceeding through the columns from left to right. For each column k, its positions are visited starting at the main diagonal ($j=k$) and progressing up (i.e., decreasing j). When $j=k$ clearly, $B(j,k) = B_k$, while when $j<k$, the composite stream $B(j,k)$ is computed by merging B_j with $B(j+1,k)$, which was computed previously. Then the N

weights $w_1(j,k),\cdots,w_N(j,k)$ are evaluated, the memory for $B(j+1,k)$ is deallocated, and $B(j,k)$ is stored to be used at the next step. Notice that only the prefix of length L of each composite stream $B(j,k)$ needs to be evaluated, where $L=\max_{1\leq n\leq N} L_n$. The merge procedure to accomplish this goal requires only $O(L)$ running time, while the number of multiplications and additions used to evaluate each element $w_n(j,k)$ amounts to $O(L)$ operations as well. Thus, the total time needed for all N upper triangular matrices is $O(K^2LN)$.

It is clear that in the case when all functions $w_n(\cdot)$ for $1\leq n\leq N$, are identical, computational savings can be achieved since only one matrix needs to be evaluated. This special case appears when $L_1=L_2=\cdots=L_N$ and $\gamma(1,r)=\gamma(2,r)=\cdots=\gamma(N,r)$, for each $1\leq r\leq L_1$. We will say that the RDOP problem is *uniform* if the above conditions are satisfied.

In conclusion, the time complexity of the dynamic programming solution is $O(K^2LN)$ for the general RDOP problem and $O(K^2(L+N))$ for the uniform RDOP problem. As for the space complexity, notice that, in order to store temporary results, only $O(\max(K,L))$ storage space is enough, while $O(K^2)$ memory locations are required to store each matrix W_n. Thus, the algorithm space complexity is $O(\max(K^2N,L))$ for the general RDOP and $O(\max(K^2,L))$ for the uniform RDOP.

IV. Fast Divide and Conquer Solution Based on Total Monotonicity

This section contains the main contribution of this work. We first show that the RDOP problem obeys a nice monotonicity property and then exploit it to develop an efficient divide and conquer globally optimal solution algorithm.

The first observation is that the RDOP problem can be regarded as the problem of finding the column maxima in N matrices. To see this, for each $1\leq n\leq N$, let M_n denote the $K\times K$-dimensional upper triangular matrix with elements defined as follows

$$M_n(j,k)=C_{opt}(n-1,j)+w_n(j+1,k), \tag{8}$$

for all $0\leq j<k\leq K$. It is clear now that solving (7) for fixed n and all k, is equivalent to finding the maximum value in each column of the matrix M_n. We will also refer to this problem as the matrix search problem. This observation alone is not sufficient to speed up the computations, however, as we will soon see, each matrix M_n has a special property, termed total monotonicity, which serves this purpose. Specifically, an upper triangular matrix M_n is said to be totally monotone [19] if the following relation holds

$$M_n(j,k)\leq M_n(j',k)\Rightarrow M_n(j,k')\leq M_n(j',k'),\qquad j<j'<k<k'. \tag{9}$$

The following property represents the key result toward the fast solution algorithm.

Proposition 1. For every $n, 1 \leq n \leq N$, the upper triangular matrix M_n is totally monotone.

To avoid disrupting the flow of the exposition we defer the proof of Proposition 1 to the appendix. However, we emphasize that two crucial factors in the proof are the assumption that the R-D curves of the basic streams are convex and the fact that coefficients $\gamma(n, r)$ are non-increasing in r, stated in (3).

Due to Proposition 1, the matrix search algorithm, which normally requires all matrix elements to be examined, can be accelerated by leveraging a pleasing property of the row indices where the maxima occur. Specifically, let $j(k)$ denote the row index where the maximum occurs on column k. If the maximum occurs on several rows we take the largest index. Clearly, property (9) implies that $j(1) \leq j(2) \leq \cdots \leq j(K)$. This property enables a fast divide and conquer algorithm to solve the matrix search problem. Notice that if the value $j(\lfloor K/2 \rfloor)$[5] is known, then, in order to find the maxima for the columns 1 through $\lfloor K/2 \rfloor - 1$, one only needs to look at the elements on the rows 0 through $j(\lfloor K/2 \rfloor)$. Likewise, to find the maxima on the columns with index higher than $\lfloor K/2 \rfloor$ one only needs to check the rows with indices higher or equal than $j(\lfloor K/2 \rfloor)$. Therefore, we start by determining the maximum in the middle column (i.e., $\lfloor K/2 \rfloor$), which further reduces the problem of searching matrix M_n into two subproblems of smaller sizes, namely searching submatrices $M_n(0 : j(\lfloor K/2 \rfloor), 1 : \lfloor K/2 \rfloor - 1)$ and $M_n(j(\lfloor K/2 \rfloor) : K - 1, \lfloor K/2 \rfloor + 1 : K)$, where $M_n(j : j', k : k')$ denotes the submatrix containing the rows in the range j to j' and columns in the range k to k', inclusive. Further, the column maxima problem in each submatrix is solved recursively using the same procedure, i.e. first the maximum on the middle column is found, then the problem is divided into two subproblems, and so on. Notice that the divide and conquer algorithm does not examine all elements of M_n, and therefore not all the weights $w_n(j, k)$ need to be computed. Actually, it turns out that, only $O(K \log K)$ matrix elements need to be checked, thus only $O(K \log K)$ weights $w_n(j, k)$ have to be evaluated. We will see that the computations can be organized in such a way so that only $O(L)$ operations per checked matrix element to suffice, leading to a running time of $O(KL \log K)$ for searching matrix M_n.

The pseudocode of the divide and conquer algorithm (D&C, for short) is presented in Fig. 2. Note that D&C$(M_n, j_1 : j_2, k_1 : k_2)$ denotes the recursive procedure to solve the matrix search in submatrix $M_n(j_1 : j_2, k_1 + 1 : k_2)$. In order to search matrix M_n, first all composite streams $B(k, k)$ are initialized as B_k, for $1 \leq k \leq K$, then D&C$(M_n, 0 : K - 1, 0 : K)$ is invoked. Referring to the pseudocode in Fig.2, it is easy to see that, if $j_2 \leq k_1 - 1$, then $B(j_2 + 1, k_1)$ has already been evaluated during one of the

[5]For any real value x, $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x.

D&C$(M_n, j_1 : j_2, k_1 : k_2)$

01. **if** $k_1 + 1 > k_2$ **stop**
02. **else** **do** $k := \lfloor (k_1 + k_2 + 1)/2 \rfloor$
03. **if** $j_2 \leq k_1 - 1$ **do**
04. $\{$**for** $i := k_1 + 1$ **to** k **do**
05. $\{B(j_2 + 1, i) :=$ **merge**$(B(j_2 + 1, i - 1), B_i)$
06. **compute** $w_n(j_2 + 1, i)$ and $M_n(j_2, i)$ via (8)$\}$
07. **for** $j := j_2 - 1$ **downto** j_1 **do**
08. $\{B(j + 1, k) :=$ **merge**$(B(j + 2, k), B_{j+1})$
09. **compute** $w_n(j + 1, k)$ and $M_n(j, k)$ via (8)$\}\}$
10. **else** **do**
11. $\{$**for** $j := k - 2$ **downto** j_1 **do**
12. $\{B(j + 1, k) :=$ **merge**$(B(j + 2, k), B_{j+1})$
13. **compute** $w_n(j + 1, k)$ and $M_n(j, k)$ via (8)$\}\}$
14. $j(k) := \arg\max_{j_1 \leq j \leq \min(j_2, k-1)} M_n(j, k)$
15. D&C$(M_n, j_1 : j(k), k_1 : k - 1)$
16. D&C$(M_n, j(k) : j_2, k : k_2)$

Fig. 2. Divide and conquer algorithm to search submatrix$M_n(j_1 : j_2, k_1 + 1 : k_2)$.

previous invocations, therefore the operation on line 5 is correctly defined. The merge procedure on lines 5, 8 and 12 takes only $O(L)$ operations. Lines 6, 9 and 13 also require $O(L)$ operations. The number of composite streams evaluated when $j_2 \leq k_1 - 1$ (lines 4-9), is $\lfloor (k_2 - k_1 + 1)/2 \rfloor + j_2 - j_1$, while for the case $j_2 \geq k_1$, it is $\lfloor (k_1 + k_2 + 1)/2 \rfloor - j_1 - 1 < (k_2 - k_1 + 1)/2 + j_2 - j_1$. Further, since the number of operations needed to execute line 14, is proportional to $j_2 - j_1$, it follows that the overhead needed to reduce the problem $M_n(j_1 : j_2, k_1 + 1 : k_2)$ into two subproblems is at most $c_1 L(k_2 - k_1 + j_2 - j_1)$ for some $c_1 > 0$.

In order to evaluate the total running time needed to execute D&C$(M_n, 0 : K-1, 0 : K)$, it is instructive to analyze the associated recursion tree, where each node represents a subproblem solved recursively. In particular, the root represents the initial problem $M_n(0 : K - 1, 1 : K)$. For every tree node, the left,

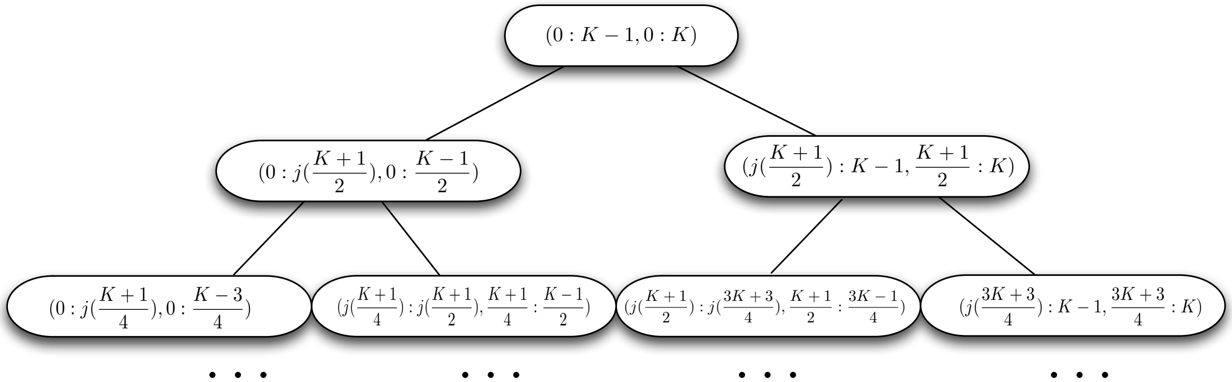


Fig. 3. Recursion tree associated to the divide and conquer algorithm.

respectively right, child represents the first, respectively second, subproblem invoked recursively at that node. Fig. 3 illustrates levels $0, 1$ and 2 of the recursion tree.

To simplify the running time evaluation we assume that K is a power of 2. The total run-time can be obtained by adding up the overhead needed to reduce the problem at every tree node. An important observation to aid this task is that the summation of the overheads for all nodes on any given level of the tree is at most c_2LK for some $c_2 > 0$. To see this, notice first that any level l has 2^l nodes and each node corresponds to a submatrix with $K/2^l$ columns, i.e., $k_2 - k_1 = K/2^l$. Thus, the summation of values $k_2 - k_1$ over all nodes at this level is K. On the other hand, as we scan the nodes on level l from left to right, the value j_2 for a node (the last row in the submatrix) is identical to the value of j_1 for the next node (the first row). Thus, the summation of $j_2 - j_1$ over all nodes is $K - 1$. Now, having established that the total overhead on any level is $O(LK)$, and since the height of the recursion tree is $\log_2 K$, it follows that the number of operations to solve the matrix search problem is $O(LK \log K)$ as claimed. The space complexity of the algorithm is $O(K^2 + LK \log K)$ due to the need to store the composite streams.

Finally, to solve the RDOP problem (6), the divide and conquer algorithm is applied to search each of the matrices $M_1, \cdots, M_N$, yielding an $O(NLK \log K)$ time complexity algorithm. On the other hand, for the uniform RDOP problem the weights $w_n(j, k)$ evaluated for some matrix M_n can be reused for subsequent matrices, thus their computation on lines 6, 9 and 13, and the invocation of the "merge" routine on lines 5, 8 and 12, are not performed all the time. Thus, the number of operations to search matrix M_n decreases as n increases. This observation suggests that for the uniform RDOP the running time estimate of $O(NLK \log K)$ is a loose upper bound, claim which is supported by our experimental results.

V. Applications of the RDOP Problem

In this section we present specific examples of transmission scenarios which fit into the framework of Section II-A and discuss the application of the RDOP problem. Specifically, we describe first the MUEP scheme (Unequal Erasure Protection for progressive Multi-streams) and its particular case termed FMUEP, which were introduced in [12]. Then we adress the extension of MUEP/FMUEP to a product code.

A. MUEP and FMUEP

MUEP ([12]) is a transmission scheme as described in Section II-A, where the CCM applies unequal erasure protection using *permuted* systematic Reed-Solomon (RS) codes across source packets. The term "permuted" refers to the fact that the systematic symbols of the RS codeword do not necessarily appear at the beginning or end of the codeword, as is usually assumed in unequal erasure protection (UEP) schemes ([21]-[24]), but are interleaved with the redundancy symbols, according to some specified permutation. The CCM produces N channel packets of L' symbols each, for some value L'. The packetization strategy of the CCM can be visualized by means of the the L'-by-N packetization array, where each column represents a channel packet and each entry represents a channel symbol. Such an array is illustrated in Fig. 4. The n-th column contains the symbols of source packet S_n interleaved with redundancy symbols, which come from the RS codes applied across packets. More specifically, each row i of the array is a permuted systematic $(N, m(i))$ RS codeword, thus, it contains $m(i)$ source symbols and $N - m(i)$ redundancy symbols. Such a code is able to recover all symbols from at most $N - m(i)$ erasures. The strength of the RS codeword decreases as i increases, i.e., $m(i) \leq m(i+1)$, in order to provide higher erasure protection to source symbols which appear earlier in the source packets. The rows consisting of (N, m) RS codewords form layer m.

The MUEP strategy is designed for packet lossy channels, where packets are either lost or delivered uncorrupted. If only a subset of k channel packets out of all N, are received at destination, then all source symbols in these packets can be recovered and decoded, while additionally, the source symbols situated in layers 1 through k from the missing packets are recovered using the erasure protection, and further decoded.

An important problem in this scenario is the optimal redundancy allocation (RA). This problem was addressed in [12] with the goal of minimizing the expected distortion. For $1 \leq n, m \leq N$, let x_m denote

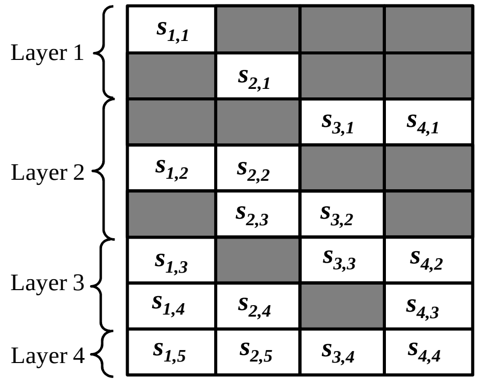


Fig. 4. Example of MUEP packetization array with $N = 4$ and $L' = 8$. Gray boxes represent redundancy symbols and white boxes source symbols. Symbol $s_{n,i}$ denotes the i-th symbol of source packet S_n.

the number of rows in layer m, and let $x_m^{(n)}$ be the number of source symbols in the n-th source packet, which are situated in layer m. Then the expected distortion at the receiver is

$$D_0 - \sum_{m=1}^{N} \sum_{n=1}^{N} C(n,m) \left(\sum_{r=1+\sum_{\ell=1}^{m-1} x_\ell^{(n)}}^{\sum_{\ell=1}^{m} x_\ell^{(n)}} \Delta D_{G_n}(r) \right), \tag{10}$$

where $C(n,m)$ denotes the probability that a source symbol in packet n situated in layer m, is recovered and decoded at the receiver. Clearly, (10) has the form of (4), by letting $\gamma(n,r)$ be equal to $C(n,m)$ if the r-th symbol of G_n resides in layer m. Due to non-increasing strength of the RS code applied to consecutive source symbols in each source packet, once a source symbol is recovered, it is guaranteed that all previous source symbols in the packet are recovered. Thus, $C(n,m)$ equals the probability that one of the following events occurs: 1) channel packet n is received at the destination, or 2) packet n is lost and at least m other packets are received.

The special case of MUEP termed FMUEP, was considered in [12], in order to decrease the amount of side information needed at the decoder to specify the RA. The additional constraint in FMUEP is that the number of source symbols in each layer m is the same for all packets: $x_m^{(n)} = mx_m/N$ for all m, n.

The optimization of RA for MUEP and FMUEP in [12] was formulated as the problem of minimizing (10) under the assumption that the size L' is fixed and that the partitioning of the set of samples into groups $\mathcal{G}_1, \cdots, \mathcal{G}_N$ is given. It is shown that for general channels the globally optimal solution is computationally demanding, and an efficient suboptimal algorithm is proposed for the case of *symmetric* channel, i.e., a channel where the probability that the packets in some subset $\mathcal{I} \subset \{1, 2, \cdots, N\}$ are lost, while the rest of packets are received, is the same for all subsets $\mathcal{I}$ of equal size. The experiments reported in [12] use SPIHT as the progressive encoder in the SPM and compare two grouping strategies: 1) a predefined grouping where each group contains K/N consecutive basic subsets of samples; 2) an optimized grouping, determined by solving the RDOP problem with $L_n = L'$ for all n, and $\gamma(n,r) = 1$ for all n and r. For both strategies the fixed ordering of the basic subsets is the same. According to the results of [12], for MUEP the two strategies can have comparable performance for small values of N, but the optimized grouping becomes consistently better as N increases. Moreover, the optimized grouping has performance very close to the MUEP upper bound determined in [12]. For FMUEP the optimized strategy is also better than the predefined one for higher N, except for the case of exponential packet loss channel with high loss rate[6].

It is important to note that the RA optimization algorithm used in [12] for FMUEP actually does not use the knowledge of the grouping in the SPM since it computes the values $x_1, \cdots, x_N$ only based on the R-D curve of the encoded stream for the whole set of data samples. After computing the RA the size of each source packet and the probability for each symbol to be decoded are determined. Then clearly, the partitioning can be optimized for minimum expected distortion by solving the RDOP problem after the RA is optimized. We will test this approach in the experimental section. Unfortunately, for MUEP the SPM cannot be directly optimized for minimum expected distortion, since the RA algorithm in [12] uses the knowledge of the R-D curves of individual codestreams $G_1, \cdots, G_n$. On the other hand, the empirical results of [12] suggest that by using an approximation for the size of the source packets and further solving the RDOP problem for minimum total distortion when all source packets are decoded, is useful to increase the performance. A further performance improvement for MUEP is possible by solving again the RDOP problem for minimum expected distortion after the RA optimization based on the initial SPM. However, we expect the additional improvement to be small since the initial SPM already achieves performance very close to the MUEP upper bound.

B. Product Code based on MUEP/FMUEP

The MUEP or FMUEP transmission schemes can be extended to incorporate bit error protection within each packet in the case when the channel incurs packet losses and bit errors, or only bit errors. For this, after constructing the MUEP/FMUEP packetization array as described above, each column (of L' symbols) is further applied a bit error correction code of some length L'' ($L'' > L'$). We will refer to this code as the intra packet channel code. The N codewords formed in this way are the channel packets. At destination, upon receiving each channel packet, the intra packet channel code is used first to correct bit errors, if possible. We assume that the decoder of the intra packet channel code either corrects all

[6]In the exponential channel model the probability of losing k packets out of N decreases exponentially in k.

errors or declares a decoding failure[7]. Thus, if the bit errors within the packet cannot be corrected , then the packet is declared unavailable. Further, all the packets which are corrected enter the MUEP/FMUEP decoder. Thus, assuming that channel packets $i_1, i_2, \cdots, i_k$ are received and corrected, then columns $i_1, i_2, \cdots, i_k$ in the channel packetization array are recovered, hence source packets $S_{i_1}, S_{i_2}, \cdots, S_{i_k}$ are recovered. Additionally all source symbols in layers 1 through k from the other packets will be further recovered by the MUEP/FMUEP decoder.

Clearly, the expected distortion still has the form of (10), where $C(n, m)$ equals the probability that one of the following events occurs: 1) channel packet n is received at the destination and is recovered from bit errors, or 2) channel packet n is either lost or not recovered from bit errors, but at least m other channel packets are correctly recovered. Thus, $C(n, m)$ depends on the channel statistics, but also on the error correction ability of the intra packet channel code.

Optimizing the RA comprises optimizing the code rate of the intra packet channel code and the RA for MUEP/FMUEP, assuming fixed N and L''. One possibility to perform the optimization is to consider different values for the rate of the intra packet channel code, for each such value optimize the RA in the MUEP/FMUEP scheme, and then take the best overall. Notice that for given rate of the intra packet channel code, the value L' and coefficients $C(n, m)$ are determined, thus the optimization of the RA for the MUEP/FMUEP scheme can proceed as in [12]. This means that the overall optimization of the RA does not depend on the SPM in the case of FMUEP, thus the SPM can be further optimized for minimum expected distortion by solving the RDOP problem. On the other hand, for MUEP, we speculate that the SPM optimized by solving the RDOP problem with $L_n = L'$ and $\gamma(n, r) = 1$, for all n, r, can be used to increase the system performance.

Such a product code scheme was used in [10], with the notable difference that the RS codes used across packets are *strictly* systematic codes, i.e., the redundant symbols appear only after the information symbols.

VI. Experimental Results

The goal of this section is twofold. First we assess empirically the increase in speed of the proposed divide and conquer (D&C) algorithm for the RDOP problem, versus the dynamic programming (DP) counterpart. Next we attest the practical impact on performance of using an R-D optimized partitioning via the D&C algorithm, versus a non-optimized one, in an FMUEP scenario.

[7] This can be realized in practice via a concatenated code with an outer error detection code and an inner error correction code.

N		8	16	24	32	40
L		819	409	273	204	163
uRDOP	D&C (s)	0.184	0.112	0.082	0.069	0.055
	DP (s)	17.237	8.647	5.836	4.472	3.518
gRDOP	D&C (s)	0.253	0.174	0.165	0.167	0.171
	DP (s)	19.077	10.535	7.810	6.620	5.979

TABLE I

RUNNING TIME COMPARISON D&C VERSUS DP; IMAGE LENA, $R_s = 0.2$ BPP.

N		8	16	24	32	40
L		819	409	273	204	163
uRDOP	D&C (s)	0.193	0.118	0.087	0.072	0.059
	DP (s)	17.267	8.660	5.885	4.482	3.530
gRDOP	D&C (s)	0.259	0.177	0.172	0.177	0.185
	DP (s)	19.160	10.595	7.877	6.719	6.075

TABLE II

RUNNING TIME COMPARISON D&C VERSUS DP; IMAGE PEPPERS, $R_s = 0.2$ BPP.

A. General Experimental Setting

Two 512×512 images (Lena and Peppers) were tested. In both cases, after subtracting the mean[8] from the original image, a 5-level Cohen-Daubechies-Feauveau 9/7 wavelet transform was applied. The wavelet coefficients constitute the data samples. The progressive source coder employed in our tests is the SPIHT encoder without arithmetic coding. We have used the QccSPIHT encoder from QccPack library [25]. We consider $K = 256$ basic subsets of samples, 192 corresponding to spatial orientation trees and 64 to non-root coefficients. Since the proposed D&C algorithm relies on the convexity of the R-D curves, we approximate the R-D curve of each primary codestream B_k by its convex hull. Moreover, a symbol consists of 8 bits. In our experiments, the fixed ordering of the basic streams is the subband dispersed order introduced in [12]. The packet header encoding the composition of the group contains $\log_2 K = 16$ bits, in other words, 2 symbols.

[8]We assume that this value is available at the decoder.

N		8	16	24	32	40
L		2048	1024	682	512	409
uRDOP	D&C (s)	0.586	0.268	0.187	0.146	0.124
	DP (s)	45.899	21.365	14.548	10.962	8.699
gRDOP	D&C (s)	0.758	0.487	0.378	0.363	0.371
	DP (s)	51.622	25.889	19.055	15.614	13.981

TABLE III

RUNNING TIME COMPARISON D&C VERSUS DP; IMAGE LENA, $R_s = 0.5$ BPP

N		8	16	24	32	40
L		2048	1024	682	512	409
uRDOP	D&C (s)	0.647	0.285	0.203	0.159	0.137
	DP (s)	46.155	21.279	14.602	11.111	8.771
gRDOP	D&C (s)	0.807	0.507	0.396	0.383	0.394
	DP (s)	52.235	26.021	19.186	15.828	14.117

TABLE IV

RUNNING TIME COMPARISON D&C VERSUS DP; IMAGE PEPPERS $R_s = 0.5$ BPP.

B. *Speed Assessment of D&C Algorithm*

For the speed comparison we have run tests for two source coding rates R_s: $R_s = 0.2$ bpp and $R_s = 0.5$ bpp, and number of packets $N = 8, 16, 24, 32, 40$. We assume that all source packets have the same number L of symbols, thus, for each N and R_s, L is computed such that $R_s = 8 \cdot N(L+2)/(512 \times 512)$ (notice that $L+2$ is the length of the packet after including the header).

For the purpose of speed assessment we have considered the general RDOP problem, as well as the uniform RDOP problem where the matrices $W_1, \cdots, W_N$ defined in section III, are identical. The coefficients $\gamma(n,r)$ used in the RDOP problems are computed as follows. For each N and L we first solve the RA problem for an FMUEP scenario using the algorithm of in [12], assuming that the size of a channel packet is $L' = L$. We assume an exponential packet loss model for the channel, i.e., where the probability $P_N(k)$ of losing k packets out of the transmitted N packets, decreases exponentially with k. As shown in [12], the probability $C(n,m)$ that a source symbol in channel packet n situated in layer m, is recovered and decoded by the source decoder, is

$$C(n,m) = 1 - \mu + \sum_{k=0}^{N-m} \frac{k}{N} P_N(k), \tag{11}$$

for $1 \le n \le N, 1 \le m \le N$, where μ is the mean packet loss rate. The RA algorithm determines the values x_m, $1 \le m \le N$, based on the R-D curve of the SPIHT coded stream for the whole image. Then we set $x_m^{(n)} = \lfloor m x_m / N \rfloor$ for $1 \le n \le N - j$, and $x_m^{(n)} = \lceil m x_m / N \rceil$ for $N - j \le n \le N$, where j is the remainder of the division of $m x_m$ by N. Let L_n denote the number of source symbols in the n-th channel packet obtained according to this solution, after subtracting the two symbols used in the header. In other words, $L_n = \sum_{m=1}^{N} x_m^{(n)} - 2$. The coefficients $\gamma(n,r)$ for the general RDOP problem are then set as $\gamma(n,r) = C(n,m)$ if $\sum_{k=1}^{m-1} x_k^{(n)} < r + 2 \le \sum_{k=1}^{m} x_k^{(n)}$, for some $m, 1 \le m \le N$, (i.e., if the r-th symbol of G_n is in layer m), and $\gamma(n,r) = \gamma(n, L_n)$ when $L_n < r \le L$. Notice that the above procedure produces coefficient vectors $(\gamma(n,r))_{1 \le r \le L}$ very similar or even identical for different values of n. Thus, the corresponding RDOP problem is actually very close to a uniform RDOP problem. However, in order to evaluate the running time for the general RDOP problem, we deliberately disregard similarities of the coefficients from packet to packet. Finally, for the uniform RDOP problem solved in our tests for speed assessment, we compute the coefficients $\gamma(1,r)$ as described above, and let $\gamma(n,r) = \gamma(1,r)$ for all n and r.

The D&C and DP algorithms were implemented in the C programming language, and run on a 2.4 GHz Intel Core 2 Duo MacBook with 2GB 1067 MHz DDR3 main memory. The input in each case consists of the K primary streams, each of L bytes.

The speed assessment results are shown in Tables I-IV. Each table contains the running time in seconds for one image at one source coding rate. We use the notation gRDOP and uRDOP for general, respectively, uniform RDOP problems. It can be seen that D&C is significantly faster than DP making RDOP feasible in practice. Moreover, for fixed value of the product NL (notice that this value is constant for each table), the running time spent by D&C to solve the uRDOP problem consistently decreases as N increases, fact which supports the claim that the time complexity estimate of $O(NKL \log K)$ is not tight. Furthermore, the D&C algorithm is faster for uRDOP than for gRDOP, as expected, the speed up being accentuated as N increases. On the other hand, the rate of relative speed increase is slow, such that, even at $N = 40$, the the D&C solution for gRDOP remains competitive with the uRDOP counterpart.

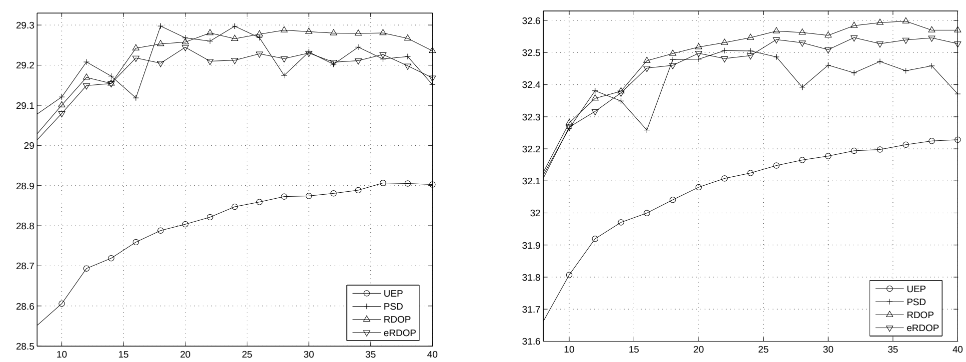
C. Performance Assessment of RDOP

Next we assess the practical impact in performance of using an R-D optimized partitioning via D&C, versus a non-optimized one, for the FMUEP transmission scenario. It is important to point out that although we use the assumptions in Section II-B to solve the RDOP problem, after doing so, the source packets are constructed by running the SPIHT encoder for the set of samples in each group $\mathcal{G}_n$ determined by the algorithm.

We consider two transmission rates $R_t = 0.2$ bpp and $R_t = 0.5$ bpp, and two channel models: EXP (exponential packet loss) and IPL (independent packet loss) with three mean packet loss rates $\mu = 0.05, 0.1, 0.15$. The same two images are tested. For each transmission rate, all even values of N in the range 8 to 40 are considered. The FMUEP redundancy allocation is obtained by using the algorithm described in [12], where the size L' of each channel packet is computed such that $R_t = 8 \cdot NL'/(512 \times 512)$. We compare three partitioning strategies, termed RDOP, eRDOP and PSD. RDOP is the partitioning optimized for minimum expected distortion, i.e., the values L_n and $\gamma(n, r)$, $1 \le r \le L_n$, $1 \le n \le N$, used in the cost function of (4), are computed as explained in the previous subsection. eRDOP (RDOP with Equal coefficients) is optimized for minimum distortion when all N packets are received, assuming that each source packet has the same size as the channel packet (i.e., $L_n = L'$ and $\gamma(n, r) = 1$ for all r and n). RDOP and eRDOP partitionings are obtained using the D&C algorithm. Finally, PSD is obtained by allocating an equal number of consecutive basic streams, according to the same fixed ordering, to each source packet[9], thus there is no need to include a header in each source packet to specify the basic streams in the group. FMUEP with these partitioning strategies are also compared against UEP, which is the classic unequal erasure protection scheme [23] applied to a progressive codestream, without having independently decodable source packets. All the aforementioned strategies are compared by using the expected distortion at the receiver as a performance measure. We point out that the FMUEP decoder needs to know the values $x_1, \cdots, x_N$, which specify the FMUEP packetization array. As usual in the UEP literature, we assume that this information is transmitted via a noiseless channel and it is not included in the total transmission budget.

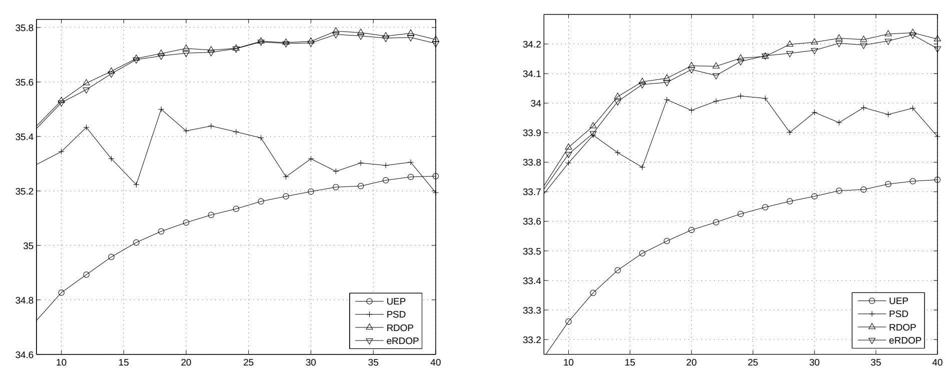
Our tests reveal that RDOP outperforms PSD most of the time and the gap can reach up to 0.5 dB. The only cases when PSD is better are only for the EXP channel at high loss rate (0.15) and small or moderate values of N, but the margin is slim. Moreover, given fixed image, channel model, packet loss rate and transmission rate, the gap between RDOP and PSD tends to increase with N. As expected, the RDOP performance is greater or equal to that of eRDOP with a difference of at most 0.073 dB. This result demonstrates that optimizing the partitioning for minimum expected distortion (RDOP case) rather than for minimum distortion when all packets are received (eRDOP case) can improve the performance.

[9]More specifically, the first $N - j$ source packets are assigned $\lfloor K/N \rfloor$ basic streams each, and the last j source packets are assigned $\lceil K/N \rceil$ basic streams each, where j is the remainder of the division of K by N.



(a) $R_t = 0.2$ bpp (b) $R_t = 0.5$ bpp

Fig. 5. PSNR values for various N, for Lena, EXP channel with $\mu = 0.15$.



(a) $\mu = 0.05$ (b) $\mu = 0.1$

Fig. 6. PSNR values for various N, for Lena, $R_t = 0.5$ bpp, EXP channel with lower packet loss rate.

On the other hand, the improvement is small, fact which suggest that small changes in the coefficients $\gamma(n, r)$ for the RDOP problem do not have a big impact. This observation further supports the use of the RDOP problem with a heuristic choice of the coefficients $\gamma(n, r)$, to boost the performance in transmission schemes where the probability of decoding each symbol cannot be accurately estimated.

To illustrate our observations we plot in Figs. 5 and 6 the performance of the above mentioned partitionings for various values of N, for image Lena under EXP channel for the following settings: Fig. 5a): $R_t = 0.2$ bpp, $\mu = 0.15$; Fig. 5b): $R_t = 0.5$ bpp, $\mu = 0.15$; Fig. 6a): $R_t = 0.5$ bpp, $\mu = 0.05$; Fig. 6b): $R_t = 0.5$, $\mu = 0.1$. For each case the performance is measured by converting the expected distortion to PSNR.

We mention that the impact of eRDOP versus PSD was tested in [12] as well, where the two partitionings were termed OptSD and PSD, respectively. The experimental setting of [12] is similar to ours with only one exception. In [12], the image mean is not subtracted from the value of each pixel, before applying the wavelet transform, as we have done in our experiments. It turns out that subtracting the mean first from the image, enhances the performance of eRDOP since in our experiments eRDOP is always better than or competitive with PSD, while in [12], for exponential channel with loss rate 0.15 the eRDOP results were far worse than PSD and even below UEP level [12].

D. Application of the RDOP Problem with the JPEG2000 Encoder

In our tests we have chosen the SPIHT encoder as the progressive coder in the SPM due to its simplicity. However, our RDOP problem formulation and the proposed D&C algorithm can also be used with the JPEG2000 encoder. Notice that in the case of JPEG2000, assumption A1 (formulated in Section II-B) is a looser approximation than for SPIHT because JPEG2000 encodes the set of samples in group $\mathcal{G}_n$ not just by interleaving the corresponding basic streams B_k, but by additionally inserting side information (SI) at the beginning of each quality layer. To account for the inclusion of SI let us break the SPM module into two submodules. The first submodule partitions the set of basic streams and forms each codestream G_n by interleaving the basic streams from the corresponding group. Then it takes the size L_n prefix to form source packet S_n. The second submodule inserts SI symbols in the codestream S_n in the manner of JPEG2000, to form source packet S'_n of size L'_n. Further, source packets $S'_1, \cdots, S'_N$ are fed to the CCM. Then the idea is to convert the problem of optimizing the SPM into an RDOP problem for the first submodule.

Consider first the objective of minimizing the distortion when all source packets S'_n are decoded, assuming that each L'_n is fixed. Since the SI symbols do not contribute to the decrease of distortion, the cost function still has the form of (4) with all $\gamma(n, r) = 1$. The only difference versus the RDOP problem is the fact that L_n is not fixed since the inserted SI depends on the composition of each group. A simple approach to circumvent this problem is to use a fixed value as an estimate for L_n and solve the RDOP problem to determine the grouping. We speculate that even a crude estimate such as $L_n = L'_n$ should lead to good performance because the amount of SI is small comparing to L_n. This claim is also supported by the good results reported in [12] and in our experiments with eRDOP, where a loose overestimate of the size of source packets was used in the grouping optimization (the size of the source packet was assumed to be equal to the size of a channel packet, which actually contains additional redundancy symbols).

Consider now the objective of minimizing the expected distortion. The expected distortion still has the form of (4). However, each coefficient $\gamma(n, r)$ is no longer fixed for given n and r, but depends on the position r' of the r-th symbol of packet S_n in the source packet S'_n. Clearly, r' equals r plus the number of SI symbols inserted in the current and previous quality layers, thus it depends on the grouping. One approach to circumvent this difficulty is to use a fixed estimate for the amount of SI and for the positions where it is inserted in S_n[10]. Based on this estimate compute r' and set $\gamma(n,r)$ to the probability that the r'-th symbol of source packet S'_n is decoded, value which is fixed for fixed n, r' and RA. Notice that such a choice does not violate condition (3). We hypothesize that such an approach can lead to performance improvement encouraged by the results in the previous subsection where a small change in the coefficients of the RDOP problem did not affect the performance greatly. We expect an improvement to occur even if the amount of SI is loosely approximated to 0.

In conclusion our experiments show that the proposed divide and conquer algorithm for the RDOP problem is very fast in practice. Our tests also demonstrate that optimizing the partitioning by solving the RDOP problem can greatly improve the performance for the FMUEP transmission scheme with a SPIHT progressive coder. Moreover, according to the above discussion, we expect similar gains in performance to be achieved when a JPEG2000 encoder is used in the SPM.

VII. Conclusion

This work addresses the rate-distortion optimal packetization (RDOP) of embedded streams into N independent source packets. The packetization is performed by partitioning a set of K basic embedded streams into N groups, and interleaving the streams within each group to form a source packet. The previous RDOP problem formulated by Wu *et al.* aims at minimizing the distortion when all source packets are decoded. We consider a general formulation which also covers the objective of minimizing the expected distortion for transmission scenarios where erasure/error protection may be further applied. Then we show that the existing dynamic programming algorithm can be adapted to solve the general RDOP problem. Assuming convexity of the rate-distortion curves, the dynamic programming algorithm runs in $O(K^2LN)$ time, where L is the size of each source packet.

The main contribution of this work is a fast divide and conquer globally optimal algorithm, under the same convexity assumption, which reduces the total time consumption to $O(NKL\log K)$. Crucial in obtaining the accelerated solution is our result which reveals a nice monotonicity property of the optimization problem. Finally, simulation results with SPIHT coded images demonstrate that the speed up is significant in practice.

[10]Notice that in FMUEP it is easy to estimate the positions where the SI is included since it makes sense to devise the quality layers according to the layers in the FMUEP packetization, in other words, by letting the m-th quality layer to consist of the source symbols protected by an (N,m) RS code. After optimizing the RA, the positions of these symbols in S'_n are known.

APPENDIX

This appendix presents the proof of Proposition 1. First let us restate this proposition.

Proposition 1. For every $1 \leq n \leq N$, the upper triangular matrix M_n is totally monotone.

Proof. It can be easily seen that in order to prove relation (9), it is sufficient to show that

$$M_n(j,k) + M_n(j',k') \geq M_n(j',k) + M_n(j,k'), \quad j < j' < k < k'.$$

By applying (8) and reducing the like terms, the above relation becomes

$$w_n(j+1,k) + w_n(j'+1,k') \geq w_n(j'+1,k) + w_n(j+1,k'),\ j < j' < k < k'. \tag{12}$$

Let us fix now n and indices j, j', k, k' satisfying $j < j' < k < k'$. In the remainder of this proof we will use the notation L instead of L_n for simplicity. Further, denote by $a_1, a_2, \cdots, a_L$, the symbols in the composite stream $B(j'+1,k)$, labeled in the order they appear in the codestream. Likewise, let $\mu_1, \mu_2, \cdots, \mu_L$, be the symbols in the composite stream $B(j+1,k)$, $\nu_1, \nu_2, \cdots, \nu_L$, the symbols in $B(j'+1,k')$, and $\rho_1, \rho_2, \cdots, \rho_L$, the symbols in $B(j+1,k')$. For any symbol η let us also denote by $\Delta(\eta)$ the decrease in distortion achieved by decoding that symbol. For convenience we will refer to the quantity $\Delta(\eta)$, as the "utility" associated to symbol η. Thus, since in any composite stream the symbols are ordered in non-increasing order of their utilities, the following relations hold

$$\Delta(a_i) \geq \Delta(a_{i+1}), \Delta(\mu_i) \geq \Delta(\mu_{i+1}), \Delta(\nu_i) \geq \Delta(\nu_{i+1}), \Delta(\rho_i) \geq \Delta(\rho_{i+1}), \quad 1 \leq i \leq L-1. \tag{13}$$

Recall that $B(j+1,k)$ is obtained by rate-distortion optimal interleaving of $B(j+1,j')$ and $B(j'+1,k)$. Therefore there must be some $s, 0 \leq s \leq L$, such that $B(j+1,k)$ contains the first s symbols of $B(j'+1,k)$, i.e., $a_1, \cdots, a_s$, and the first $L-s$ symbols of $B(j+1,j')$. For each $1 \leq \ell \leq s$, let us now denote by α_ℓ the position of symbol a_ℓ in $B(j+1,k)$. This means that the symbol μ_{α_ℓ} is identical to a_ℓ. Clearly, we have $\alpha_\ell < \alpha_{\ell+1}$, for all $1 \leq \ell \leq s-1$. Likewise, let $t, 0 \leq t \leq L$, be the number of symbols from $B(j'+1,k)$, included in $B(j'+1,k')$ and let $u, 0 \leq u \leq L$, denote the number of symbols from $B(j'+1,k)$, which appear in $B(j+1,k')$. Furthermore, for each $1 \leq \ell \leq t$, let β_ℓ denote the position of symbol a_ℓ in $B(j'+1,k')$, and for $1 \leq \ell \leq u$, let δ_ℓ be the position of symbol a_ℓ in $B(j+1,k')$. It is easy to verify that $\beta_\ell < \beta_{\ell+1}$, for all $1 \leq \ell \leq t-1$, and $\delta_\ell < \delta_{\ell+1}$, for all $1 \leq \ell \leq u-1$.

To prove the claim we need first to rewrite conveniently each quantity $\Delta(\eta)$. Let us start with the symbols in $B(j'+1,k)$. For each $\ell, 1 \leq \ell \leq L$, one can write

$$\Delta(a_\ell) = \Delta(a_L) + \sum_{m=\ell}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})). \tag{14}$$

With the convention that $\alpha_0 = 0$ and $\alpha_{s+1} = L+1$, we further write for each ℓ and i such that $0 \le \ell \le s$ and $\alpha_\ell + 1 \le i \le \min(L, \alpha_{\ell+1})$,

$$\Delta(\mu_i) = (\Delta(\mu_i) - \Delta(a_{\ell+1})) + \Delta(a_L) + \sum_{m=\ell+1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})). \tag{15}$$

The previous relation holds because $\ell+1 \le L$. To see that the latter equality is valid, notice that, if there exists i such that $\alpha_\ell + 1 \le i \le \min(L, \alpha_{\ell+1})$, then necessarily $\alpha_\ell \le L-1$, which implies $\ell \le L-1$.

Likewise, let $\beta_0 = \delta_0 = 0$ and $\beta_{t+1} = \delta_{u+1} = L+1$ by convention. Then for each ℓ and i such that $0 \le \ell \le t$ and $\beta_\ell + 1 \le i \le \min(L, \beta_{\ell+1})$, rewrite

$$\Delta(\nu_i) = (\Delta(\nu_i) - \Delta(a_{\ell+1})) + \Delta(a_L) + \sum_{m=\ell+1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})). \tag{16}$$

Finally, for each ℓ and i such that $0 \le \ell \le u$ and $\delta_\ell + 1 \le i \le \min(L, \delta_{\ell+1})$, write

$$\Delta(\rho_i) = (\Delta(\rho_i) - \Delta(a_{\ell+1})) + \Delta(a_L) + \sum_{m=\ell+1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})). \tag{17}$$

An important observation, to be used in the proof, is that all the expressions enclosed in parentheses in (14)-(17) are non-negative in virtue of relations (13).

Next we rewrite the terms in (12) using the definition (5) and (14)-(17). For simplicity we will use the notation γ_i instead of $\gamma(n,i)$, for all $1 \le i \le L$, since n is fixed. For convenience we will also denote $\gamma_{L+1} = 0$. Then $w_n(j+1,k)$ can be written as follows

$$\begin{aligned}
w_n(j+1,k) &= \sum_{i=1}^{L} \gamma_i \Delta(\mu_i) \overset{(a)}{=} \sum_{i=1}^{L+1} \gamma_i \Delta(\mu_i) = \sum_{\ell=0}^{s} \sum_{i=\alpha_\ell+1}^{\alpha_{\ell+1}} \gamma_i \cdot \Delta(\mu_i) \\
&\overset{(b)}{=} \sum_{\ell=0}^{s} \sum_{i=\alpha_\ell+1}^{\alpha_{\ell+1}} \gamma_i \cdot \Big((\Delta(\mu_i) - \Delta(a_{\ell+1})) + \Delta(a_L) + \sum_{m=\ell+1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})) \Big) \\
&= \sum_{\ell=0}^{s} \sum_{i=\alpha_\ell+1}^{\alpha_{\ell+1}} \gamma_i \cdot (\Delta(\mu_i) - \Delta(a_{\ell+1})) \\
&+ \sum_{m=1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})) \sum_{\ell=0}^{\min\{m-1,s\}} \sum_{i=\alpha_\ell+1}^{\alpha_{\ell+1}} \gamma_i + \Delta(a_L) \sum_{i=1}^{L} \gamma_i \\
&= \sum_{\ell=0}^{s} \sum_{i=\alpha_\ell+1}^{\alpha_{\ell+1}} \gamma_i \cdot (\Delta(\mu_i) - \Delta(a_{\ell+1})) \\
&+ \underbrace{\sum_{m=1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})) \sum_{i=1}^{\alpha_{\min\{m-1,s\}+1}} \gamma_i}_{T_1} + \Delta(a_L) \sum_{i=1}^{L} \gamma_i,
\end{aligned}$$

where equality (a) follows from $\gamma_{L+1} = 0$, and (b) is based on (15). Similarly, using (16) and (17), the quantities $w_n(j'+1, k')$ and $w_n(j+1, k')$ can be expressed as follows

$$w_n(j'+1, k') = \sum_{\ell=0}^{t} \sum_{i=\beta_\ell+1}^{\beta_{\ell+1}} \gamma_i \cdot (\Delta(\nu_i) - \Delta(a_{\ell+1})) + \underbrace{\sum_{m=1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})) \sum_{i=1}^{\beta_{\min\{m-1,t\}+1}} \gamma_i}_{T_2} + \Delta(a_L) \sum_{i=1}^{L} \gamma_i,$$

$$w_n(j+1, k') = \sum_{\ell=0}^{u} \sum_{i=\delta_\ell+1}^{\delta_{\ell+1}} \gamma_i \cdot (\Delta(\rho_i) - \Delta(a_{\ell+1})) + \underbrace{\sum_{m=1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})) \sum_{i=1}^{\delta_{\min\{m-1,u\}+1}} \gamma_i}_{T_3} + \Delta(a_L) \sum_{i=1}^{L} \gamma_i.$$

Finally, based on (14), we obtain

$$w_n(j'+1, k) = \sum_{i=1}^{L} \gamma_i \Delta(a_i) = \underbrace{\sum_{m=1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1})) \sum_{i=1}^{m} \gamma_i}_{T_4} + \Delta(a_L) \sum_{i=1}^{L} \gamma_i.$$

In order to prove relation (12), we first show that

$$T_1 + T_2 \geq T_3 + T_4. \tag{18}$$

For this notice that

$$T_1 + T_2 = \sum_{m=1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1}))[\sum_{i=1}^{\alpha_{\min\{m,s+1\}}} \gamma_i + \sum_{i=1}^{\beta_{\min\{m,t+1\}}} \gamma_i],$$

$$T_3 + T_4 = \sum_{m=1}^{L-1} (\Delta(a_m) - \Delta(a_{m+1}))[\sum_{i=1}^{\delta_{\min\{m,u+1\}}} \gamma_i + \sum_{i=1}^{m} \gamma_i].$$

Since $\Delta(a_m) - \Delta(a_{m+1}) \geq 0$ for all $1 \leq m \leq L-1$, in order to prove (18), it is sufficient to show that

$$\sum_{i=1}^{\alpha_{\min\{m,s+1\}}} \gamma_i + \sum_{i=1}^{\beta_{\min\{m,t+1\}}} \gamma_i \geq \sum_{i=1}^{\delta_{\min\{m,u+1\}}} \gamma_i + \sum_{i=1}^{m} \gamma_i. \tag{19}$$

A key observation used in the proof is the following. For any $0 \leq \ell \leq u-1$, the symbols in $B(j+1, k')$ appearing between a_ℓ and $a_{\ell+1}$, in other words the symbols $\rho_{\delta_\ell+1}, \cdots, \rho_{\delta_{\ell+1}-1}$, are actually the symbols from the set $\{\mu_{\alpha_\ell+1}, \cdots, \mu_{\alpha_{\ell+1}-1}, \nu_{\beta_\ell+1}, \cdots, \nu_{\beta_{\ell+1}-1}\}$, ordered in non-increasing order of their utilities.

This implies that $\delta_{\ell+1} - \delta_\ell - 1 = \alpha_{\ell+1} - \alpha_\ell - 1 + \beta_{\ell+1} - \beta_\ell - 1$. Summation over all ℓ such that $0 \le \ell \le m-1$, for $1 \le m \le u$, further yields

$$\delta_m = \alpha_m + \beta_m - m, \quad 1 \le m \le u. \tag{20}$$

The definitions of α_m and β_m imply that

$$m \le \alpha_m, \qquad m \le \beta_m, \quad 1 \le m \le u. \tag{21}$$

The above inequalities combined with (20) further lead to

$$\delta_m \ge \alpha_m, \qquad \delta_m \ge \beta_m. \tag{22}$$

To prove (19) we will distinguish between the following two cases: 1) $m \le u$; 2) $m > u$.

Case 1: $m \le u$. Since $u \le \min\{s,t\}$, it follows that $m \le s$ and $m \le t$. Thus, (19) becomes

$$\sum_{i=1}^{\alpha_m} \gamma_i + \sum_{i=1}^{\beta_m} \gamma_i \ge \sum_{i=1}^{\delta_m} \gamma_i + \sum_{i=1}^{m} \gamma_i. \tag{23}$$

We may assume without restricting the generality that $\alpha_m \le \beta_m$ (the other case is symmetric). Then, by using (21) and (22), (23) can be rewritten as relation

$$\sum_{i=m+1}^{\alpha_m} \gamma_i \ge \sum_{i=\beta_m+1}^{\delta_m} \gamma_i. \tag{24}$$

By (20), the number of terms on both sides of the above inequality is the same. Moreover, since $\gamma_i \ge \gamma_{i+1}$, $1 \le i \le L-1$, (according to (3)) and $\alpha_m \le \beta_m$, it follows that any term on the left hand side is larger than or equal to any term on the right hand side. These facts establish the validity of relation (24).

Case 2: $m \ge u+1$. In this case, one has $\delta_{\min\{m,u+1\}} = L+1$. If $m \ge t+1$ then $\beta_{\min\{m,t+1\}} = L+1$, and since $\alpha_{\min\{m,s+1\}} \ge m$ clearly holds, inequality (19) follows. The situation when $m \ge s+1$ can be treated similarly. Therefore, let us consider now the case $m < s+1$ and $m < t+1$. Now (19) becomes

$$\sum_{i=1}^{\alpha_m} \gamma_i + \sum_{i=1}^{\beta_m} \gamma_i \ge \sum_{i=1}^{L} \gamma_i + \sum_{i=1}^{m} \gamma_i, \tag{25}$$

where we have used the fact that $\gamma_{L+1} = 0$. Relations $u+1 \le m < s+1$ and $u+1 \le m < t+1$ imply that $u < s$ and $u < t$, hence symbol a_{u+1} appears in both $B(j+1,k)$ and $B(j'+1,k')$. On the other hand, this symbol does not appear in $B(j+1,k')$, fact which leads to the conclusion that $L - \delta_u \le \alpha_{u+1} - \alpha_u - 1 + \beta_{u+1} - \beta_u - 1$. Combining with $\delta_u = \alpha_u + \beta_u - u$, which holds according to (20), one further obtains that $L \le \alpha_{u+1} - 1 + \beta_{u+1} - 1 - u$. The previous inequality together with

$\alpha_{u+1} \leq \alpha_m + (u+1) - m$[11] and $\beta_{u+1} \leq \beta_m + (u+1) - m$, lead to

$$L \leq \alpha_m + \beta_m - m. \tag{26}$$

By symmetry we may again assume that $\alpha_m \leq \beta_m$. Then (25) can be rewritten as

$$\sum_{i=m+1}^{\alpha_m} \gamma_i \geq \sum_{i=\beta_m+1}^{L} \gamma_i. \tag{27}$$

By (26), the number of terms on the left hand side is larger than or equal to the number of terms on the right hand side. Combining with the monotonicity relations $\gamma_i \geq \gamma_{i+1}$ $1 \leq i \leq L-1$, and with $\alpha_m \leq \beta_m$, inequality (27) follows. Now the proof of (18) is complete.

In order to finalize the proof of relation (12) it remains to show that

$$\sum_{\ell=0}^{s} \sum_{i=\alpha_\ell+1}^{\alpha_{\ell+1}} \gamma_i \cdot (\Delta(\mu_i) - \Delta(a_{\ell+1})) + \sum_{\ell=0}^{t} \sum_{i=\beta_\ell+1}^{\beta_{\ell+1}} \gamma_i \cdot (\Delta(\nu_i) - \Delta(a_{\ell+1})) \geq \sum_{\ell=0}^{u} \sum_{i=\delta_\ell+1}^{\delta_{\ell+1}} \gamma_i \cdot (\Delta(\rho_i) - \Delta(a_{\ell+1})). \tag{28}$$

Let us denote by Z_i the general term in the summation on the right hand side, i.e., $Z_i = \gamma_i \cdot (\Delta(\rho_i) - \Delta(a_{\ell+1}))$, for any $0 \leq \ell \leq u$ and $\delta_\ell + 1 \leq i \leq \delta_{\ell+1}$. Likewise, let $U_i = \gamma_i \cdot (\Delta(\mu_i) - \Delta(a_{\ell+1}))$ for any $1 \leq \ell \leq s$, $\alpha_\ell + 1 \leq i \leq \alpha_{\ell+1}$, and let $V_i = \gamma_i \cdot (\Delta(\nu_i) - \Delta(a_{\ell+1}))$ for any $1 \leq \ell \leq t$, $\beta_\ell + 1 \leq i \leq \beta_{\ell+1}$. The idea of the proof is the following. We will show that each Z_i is either 0 or it has a corresponding term on the left hand side (LHS), which is larger or equal than Z_i. Moreover, this correspondence assigns to each LHS term at most one term on the right hand side. Since all unassigned LHS terms are non-negative, relation (28) follows.

Let us fix some ℓ and i such that, $0 \leq \ell \leq u$ and $\delta_\ell + 1 \leq i \leq \delta_{\ell+1}$. Clearly, if $i = \delta_{\ell+1}$ then one has $\gamma_i = 0$ (when $\ell = u$) or $\Delta(\rho_i) - \Delta(a_{\ell+1}) = 0$ (when $\ell < u$), hence $Z_i = 0$. On the other hand, if $i \neq \delta_{\ell+1}$, then symbol ρ_i must be either a symbol from the set $\{\mu_{\alpha_\ell+1}, \cdots, \mu_{\alpha_{\ell+1}-1}\}$, or from the set $\{\nu_{\beta_\ell+1}, \cdots, \nu_{\beta_{\ell+1}-1}\}$. In the former case, $\rho_i = \mu_{p(i)}$ for some index $p(i)$ such that $\alpha_\ell + 1 \leq p(i) < \alpha_{\ell+1}$, therefore we let $U_{p(i)}$ be the LHS term corresponding to Z_i. It is easy to see that $p(i) \leq i$, thus $\gamma_{p(i)} \geq \gamma_i$ by (3). Combining further with the fact that $\Delta(\rho_i) - \Delta(a_{\ell+1}) \geq 0$, one obtains that $U_{p(i)} \geq Z_i$. Finally, in the latter case, let $q(i)$ be the index such that $\rho_i = \nu_{q(i)}$ and $\beta_\ell + 1 \leq q(i) < \beta_{\ell+1}$. It can be easily verified that $q(i) \leq i$, therefore $\gamma_{q(i)} \geq \gamma_i$. We let $V_{q(i)}$ be the LHS term corresponding to Z_i, and $V_{q(i)} \geq Z_i$ holds. With these observations the proof of (28) is complete.

[11]This relation follows by summing inequalities $\alpha_i \leq \alpha_{i+1} - 1$ for $u + 1 \leq i \leq m - 1$. A similar argument is valid for the next relation.

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 and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

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

[3] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Very low bit-rate embedded video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1374–1385, Dec. 2000.

[4] C. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," *IEEE Trans. Image Proc.*, vol. 6, no. 10, pp. 1436–1442, Oct 1997.

[5] J. Rogers and P. Cosman, "Robust wavelet zerotree image compression with packetization," *IEEE Signal Proc. Letters*, vol. 5, no. 5, pp. 105–107, May 1998.

[6] P. C. Cosman, J. K. Rogers, P. G. Sherwood, and K. Zeger, "Combined forward error control and packetized zerotree wavelet encoding for transmission of images over varying channels," *IEEE Trans. Image Proc.*, vol. 9, no. 6, pp. 982-993, June 2000.

[7] S. Cho and W. Pearlman, "Error resilient compression and transmission of scalable video," *Proc. SPIE*, vol. 4115, San Diego, USA, Jul. 2000, pp. 396–405.

[8] T. Kim, S. Choi, R. E. V. Dyck, and N. K. Bose, "Classified zerotree wavelet image coding and adaptive packetization for low-bit-rate transport," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 9, pp. 1 022 103–4, Sept. 2001.

[9] 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.

[10] N. Thomos, N. Boulgouris, and M. Strintzis, "Optimized transmission of jpeg2000 streams over wireless channels," *IEEE Trans. Image Proc.*, vol. 15, no. 1, pp. 54–67, Jan. 2006.

[11] S.-H. Yang and P.-F. Cheng, "Robust transmission of SPIHT-coded images over packet networks", *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 17, no. 5, pp. 558-566, May 2007.

[12] S. Dumitrescu, G. Rivers, and S. Shirani, "Unequal erasure protection technique for scalable multistreams," *IEEE Trans. Image Proc.*, vol. 19, no. 2, pp. 422–434, Feb. 2010.

[13] W. Jiang and A. Ortega, "Multiple description coding via polyphase transform and selective quantization", *Proc. SPIE: Visual Comm. and Image Proc.*, Jan. 1999.

[14] K. P. Subbalakshmi and S. Somasundaram, "Multiple description image coding framework for EBCOT", *IEEE Intern. Conf. Image Proc.*, Sept. 2002, Rochester, NY.

[15] T. Tillo, M. Grangetto, and G. Olmo, "Multiple description image coding based on Lagrangian rate allocation", *IEEE Trans. Image Proc.*, vol. 16, no. 3, pp. 673–683, Mar. 2007.

[16] E. Baccaglini, T. Tillo, and G. Olmo, "A flexible R-D based multiple description scheme for JPEG 2000", *IEEE Signal Proc. Letters*, vol. 14, no. 3, pp. 197–200, Mar. 2007.

[17] E. Akyol, A. M. Tekalp, and M. R. Civanlar, "A flexible multiple description coding framework for adaptive peer-to-peer video streaming", *IEEE Journal of Selected Topics in Signal Proc.*, vol. 1, no. 2, pp. 231-245, Aug. 2007.

[18] T. Tillo, E. Baccaglini, and G. Olmo, "Multiple descriptions based on multirate coding for JPEG 2000 and H.264/AVC", *IEEE Trans. Image Proc.*, vol. 19, no. 7, pp. 1756–1767, July 2010.

[19] A. Aggarwal, M. Klave, S. Moran, P. Shor, and R. Wilber, "Geometric applications of a matrix-searching algorithm", *Algorithmica*, 2(1987), pp. 195-208.

[20] J. Xu and S. Dumitrescu, "Fast R-D optimal packetization of embedded bitstreams into independent source packets", *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2011, pp. 283–292.

[21] A. E. Mohr, E.A. Riskin, and R. E. Ladner, ”Graceful degradation over packet erasure channels through forward error correction”, *Proc. IEEE Data Compression Conf.*, pp. 92-101, Mar. 1999.

[22] R. Puri and K. Ramchandran, ”Multiple description source coding through forward error correction codes”, *Proc. 33rd Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 342-346, Oct. 1999.

[23] S. Dumitrescu, X. Wu, and Z. Wang, “Globally optimal uneven error-protected packetization of scalable code streams,” *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 230–239, Apr. 2004.

[24] S. Dumitrescu, X. Wu, and Z. Wang, “Efficient Algorithms for Optimal Uneven Protection of Single and Multiple Scalable Code Streams against Packet Erasures,” *IEEE Trans. Multimedia*, vol. 9, no. 7, pp. 1466-1474, Nov. 2007.

[25] J. E. Fowler, ”QccPack: an open source software library for quantization, compression, and coding” in *Applications of Digital Image Processing XXIII*, vol. 4115, A.G. Tescher, Ed., San Diego, CA, Aug. 2000, pp. 294-301.