Region-of-Interest Data Compression with Prioritized Buffer Management

Sam Dolinar, Gilbert Chinn,
Jonathan Harel, Aaron Kiely,
Matt Klimesh, Roberto Manduchi,
Shervin Shambayati, Melanie Vida
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California
e-mail: { {sam,harel,aaron,
klimesh,shervin,melanie}@shannon,
gil@zorba, manduchi@helios }.jpl.nasa.gov

Antonio Ortega, Sang-Yong Lee,
Phoom Sagetong, Hua Xie
Deptartment of Electrical Engineering
Signal and Image Processing Institute
Integrated Media Systems Center
University of Southern California
Los Angeles, California
e-mail: { ortega@sipi, sagetong@scf-fs,
{sangyong,huaxie}@biron }.usc.edu

Abstract — **We describe an integrated system for intelligent compression and transmission of copious data acquired by spaceborne instruments. At its core, our system contains a modification of a progressive image compression algorithm, ICER, that will be used on the Mars Exploration Rovers (to be launched in 2003). The ICER algorithm applies a wavelet decomposition and prioritizes the compressed bit layers from the wavelet subbands so as to (attempt to) progressively transmit the layer that gives the largest improvement in image quality per transmitted bit. Our modified version accepts additional input priorities that reflect the relative importance of various "regions of interest" in the source data, and arranges its output packets to reflect both the input regional priorities and the wavelet bit layer priorities.**

The output of the data compression module is supervised by an intelligent buffer manager that shuffles the prioritized packets from many different source images and tries to select packets for transmission that will maximize the total science value received on the ground. Just as importantly, it attempts to discard only the least valuable packets when the buffer overflows, which is inevitable if the average data transmission rate is lower than the average data collection rate.

After briefly describing our current system, the paper analyzes in more detail various algorithms for optimization of the buffer control algorithm, for bit allocation across regions of the source data, and for efficiently compressing data features useful for establishing a prioritization feedback loop between spacecraft and ground.

I. Description of the Overall System

We are developing integrated data compression and buffer management algorithms to maximize the science value of data returned from spacecraft instruments. Typically, imagers and remote sensors have the capability to collect far more data than can be transmitted to earth, and it is important to maximize the science value of the data returned. Onboard science processing algorithms that recognize scientifically relevant features in the collected data can be used to drive progressive data compression algorithms such as wavelet-based image compression. During progressive compression, the science data is parsed into hierarchical data segments that yield continual but diminishing improvement of fidelity with each segment. The compression schemes for science-directed progressive compression produce data segments specially tailored to "regions of interest" (ROIs) specified by science processing modules. The prioritized buffer manager tries to ensure that the highest priority data segments are transmitted first, and the least valuable data segments are discarded.

Our approach is to adapt existing progressive compression algorithms for amenability with identified ROIs, and to develop buffer strategies for prioritizing, storing, and delivering the most valuable compressed segments, and eventually reconstituting the original data. We attempt to incorporate ROI considerations across many images or different data types. These algorithms are subject to practical limits on the onboard computers speed, memory, and storage. We attempt to measure the gain in science return versus required processing speed, memory, and storage of the onboard computer.

II. The Current Algorithms

The current system takes the raw data together with a data prioritization map and produces packets of compressed data using a modified version of the ICER wavelet-based compression software.[1] Then the prioritized buffer manager decides which packets to admit, discard, or transmit, depending on their priorities and given constraints on the buffer's input and output rates.

A. Data prioritization

A data prioritization map is an assignment of a priority number to each pixel of an image. In our implementation, a priority number is an integer in the range 0–5, with higher numbers indicating higher priority. A difference of some number b between two priority numbers indicates that the higher priority pixel should be reconstructed to roughly b more bits of precision than the lower priority pixel.

The priority map is generated by identifying and classifying features of the source image that are of interest to the data's scientist-users. How to best produce such a map is a very complex question, and one best left to the scientists for each intended application. Our system does include some rudimentary general-purpose classification algorithms based on colors and textures, but the main purpose of this system is to provide scientists with powerful tools for intelligently compressing and transmitting their source data no matter how the priorities have been identified.

B. ROI compression algorithm

We have modified the ICER software to accept the additional input of a priority map indicating the relative importance of regions of

This work was funded by the ESTO Technology Program and performed at the Jet Propulsion Laboratory, California Institute of Technology, and at the Signal and Image Processing Institute, Integrated Media Systems Center, University of Southern California.

[1] Developed at JPL by A. Kiely and M. Klimesh.

the image. The compression takes the priority map into account and produces as output a sequence of packets paired with priority values. The priority values serve as a means of comparing the importance of packets, even those from separate images (assuming of course that the corresponding priority maps are also comparable).

As in the basic ICER, the input image is transformed using a wavelet decomposition. A priority map is computed for the wavelet-transformed image such that the spatial correspondence that existed between the original image and priority map is maintained between the transformed image and transformed priority map. Specifically, the priority corresponding to a transformed pixel is equal to the maximum priority of the original image pixels to which the transformed pixel corresponds.

Pixel priorities are incorporated into the compression in a manner similar to that in [2]. Each pixel of the transformed image is left-shifted (multiplied by two) a number of times equal to the corresponding value in the transformed priority map. This modified transformed image is now compressed progressively one bit layer at a time, by a procedure similar to that used in the basic ICER. The packets produced are assigned priority values based on the bit layers for which they contain data. The output packets form a progressively coded "chain," such that truncation of the chain at any point leaves a subset of packets that can be used to reconstruct the original image with a certain amount of distortion, which decreases monotonically with the number of packets retained.

Each transformed pixel has a dynamic range which depends on the dynamic range of the original image and on the subband to which the pixel belongs. The process of modifying the transformed image by left-shifting pixels changes the bit positions for which the pixels are necessarily zero, and in addition these positions now can vary within a subband. Thus, to maintain compression efficiency, ROI ICER contains provisions for compressing only those bits which are allowed to be nonzero.

The decompressor requires as input the same priority map supplied to the compressor. After decoding the (approximation to) the modified transformed image, the pixels of this image are right-shifted by the values in the transformed priority map, thereby undoing the opposite process which occurred during compression.

The input images are color RGB images. Before compression, the images are transformed into the YCrCb domain, producing three new sub-images, one for each component, all referred to the same priority map. These images are compressed separately and produce separate chains of output packets with priority values. The priority values are suitable for comparing the priorities of packets from different components.

C. Buffer management algorithms

Our baseline buffer manager uses a simple form of double-valued prioritization. It determines all admissions and discards from the buffer according to the packet priorities computed by ROI ICER. However, the order of transmissions from the buffer is based on a simple first-in, first-out (FIFO) prioritization among all packets that survive the ROI-prioritized admission/discard process.

Using a FIFO transmission priority insures that all transmissions will consist of truncated chains of packets as produced by ROI ICER. This eliminates the need to unshuffle the packets received on the ground, because the successive (truncated) packet chains can be used to reconstruct the source images in the same order in which they were acquired, but to different levels of distortion depending on how many packets from each chain survived the prioritized admission/discard process.

The buffer management algorithm described here is the same with respect to buffer admissions and discards as the "on-line algorithm with buffer sorting" described in Section III.C, but that algorithm also does full sorting by packet priorities to determine transmissions. Thus far, we have not noticed any large penalty for substituting the much simpler FIFO transmission protocol as long as we use the ROI-coded packet priorities to determine admissions and discards. But this tradeoff is a subject for further research.

III. Research on Future Algorithms

In parallel with developing our baseline system, we are also conducting research into various algorithmic approaches whereby the system performance might be improved or better understood.

A. Feature compression and classification for prioritization feedback

Priority should be given to information that has high science value, and in particular to information that is novel. This novelty should be determined with respect not only to a priori notions of what is to be expected, but also with respect to what has recently been transmitted. In other words, we would like to keep track of information already captured and give preference to transmitting data that has not been seen before. This requires keeping a database to represent information that has already been received. However, it is unrealistic to keep such a system entirely on board due to finite onboard memory and storage capacity.

We assume that to prioritize the data the onboard device first extracts features from the image regions. An unassisted onboard prioritization module would then assign priorities based on these features, perhaps comparing them to a limited history of previously transmitted features remembered in the onboard storehouse. But if these features are compressed and transmitted to the ground, they can be compared with the essentially unlimited ground database to determine whether the corresponding image information should be given high priority for transmission. If the distance between the transmitted features and the best match in the database is large , the corresponding image (or image section) should be given high priority. By doing this we establish a prioritization feedback loop between spacecraft and ground. The ground database is used to classify unknown features. Moreover this database is continuously updated to incorporate new data as it is received. For example, when a new feature has been found, the corresponding feature set will be inserted in the database as a new entry if it is classified as an "outlier."

Several criteria should be kept in mind to design such a compression and classification system. The improved prioritization scheme comes with a cost. Feature information must be transmitted to the ground database, and it is important to compress this information efficiently. Efficiency of a compression scheme is measured both by its rate versus classification performance, and by its conservation of onboard processing power, i.e., its coding complexity.

Several scenarios for a system to compress feature information were considered in [6], where a quantizer is applied before classification; we call this approach System A. In this paper we present a hybrid compression-classification system (System B) where the classifier is broken into two stages, one applied before, and the other after the quantizer. Figure 1 shows a block diagram of such a system. By applying a pre-classifier before compressing the data,

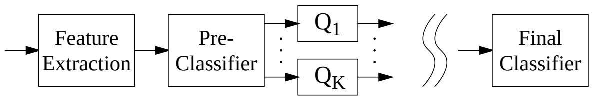

Figure 1: A hybrid compression-classification system

the original sample space $\mathcal{S}$ is partitioned into a set of subspaces $\{S_i, i = 1, \ldots, K\}$. Then we are able to exploit the local statistics of the data in each subspace by designing separate quantizers $\{Q_i, i = 1, \ldots, K\}$ for them. Thus, the hybrid System B should achieve improved compression efficiency relative to System A, at a cost of slightly more onboard complexity to perform the pre-classification.

In this paper we consider a Decision Tree Classifier (DTC) [7]. In a DTC, the sample space is partitioned hierarchically and organized in a tree structure $\mathcal{T}$ which is used to answer the query in $\log N$ time, where N is the number of samples in the database. A decision tree classifier can be easily broken into a pre-classifier T_0, which is kept on board, and a set of final classifiers $\{T_i, i = 1, 2, \ldots, K\}$, kept on the ground, as shown in Figure 2. Any partitioning of the original

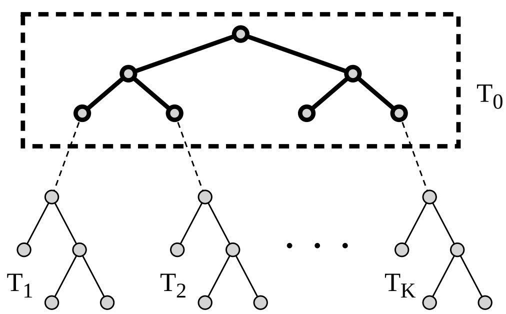

Figure 2: Partitioning the decision tree classifier

decision tree $\mathcal{T}$ is completely determined as soon as we have T_0. Then $\{T_i, i = 1, \ldots, K\}$ are just the subtrees rooted at the leaf nodes $\{v_i, i = 1, \ldots, K\}$ of T_0.

For any given pre-classifier subtree T_0, the cost for the system is primarily due to two terms:

$$F(Q, \mathcal{T}, T_0) = C(T_0) + \lambda \sum_{i=1}^{K} R(T_0, Q_i, T_i) P(T_i) \tag{1}$$

where $C(T_0)$ is the onboard computational cost for traversing subtree T_0, $R(T_0, Q_i, T_i)$ is the misclassification risk of a query following the branch $T_0 \rightarrow Q_i \rightarrow T_i$, and $P(T_i)$ is the associated probability. The parameter λ controls the tradeoff between the onboard classification cost and the reduction in misclassification. The optimization problem is then, given a training set $\mathcal{L}$, find $T_0^\star$ such that:

$$F(Q, \mathcal{T}, T_0^\star) = \min_{T_0} F(Q, \mathcal{T}, T_0) \ \text{ such that } \ C(T_0) \leq C_{\text{onboard}} \tag{2}$$

with C_{onboard} the maximum computational complexity allowed for doing such a pre-classification on board.

To find the optimal pre-classifier subtree $T_0^\star$, optimal in the sense of minimizing the expected cost to compress and classify an unknown feature, requires an exhaustive search. In this paper we use a heuristic method, the pruning method of Chou et al [8], to determine the pre-classifier subtree T_0. We start by building a binary tree $\mathcal{T}$ based on a labeled training set $\mathcal{L} = \{X, Y\}$ generated from the Brodatz texture album [9]. X is the wavelet texture feature vector [10] and Y is the texture label associated with it. The tree is built in a top-down manner [7] using the K-means [8] algorithm until each leaf node contains only one class label. The dissimilarity between two feature vectors is measured by an L_1-norm distance function. Then we prune $\mathcal{T}$ with respect to the Marginal Return cost function [8], until the constraint $C(T_0) \leq C_{\text{onboard}}$ is satisfied. $C(T_0)$ is measured as the expected depth of the tree T_0, and D is measured as the expected distortion:

$$D = \mathrm{E}\left[d(x_i, c_i)\right] \tag{3}$$

where c_i is the centroid of samples lying within leaf node v_i, and $d(\cdot)$ is the L_1-norm distance function. By performing the pruning, we expect to make the distortion of the pre-classifier T_0 as low as possible under the constraint of the onboard complexity budget.

We compare Systems A and B in terms of classification performance and overall complexity at different bit rates. Uniform scalar quantizers are used for this experiment. For System A, each feature vector is uniformly quantized and then classified using the tree classifier $\mathcal{T}$. For System B, the original tree $\mathcal{T}$ is partitioned into an onboard pre-classifier T_0 and four final classifiers $\{T_i, i = 1, 2, 3, 4\}$. The input feature vector is pre-classified to decide to which of the subspaces $\{S_i, i = 1, 2, 3, 4\}$ it belongs, and then it is quantized using a corresponding uniform quantizer $\{Q_i, i = 1, 2, 3, 4\}$. At the receiver, the quantized feature vector is sent to the final classifier T_i, which outputs a class label. Figure 3 compares the probability of correct classification at different bit rates for systems A and B. We see

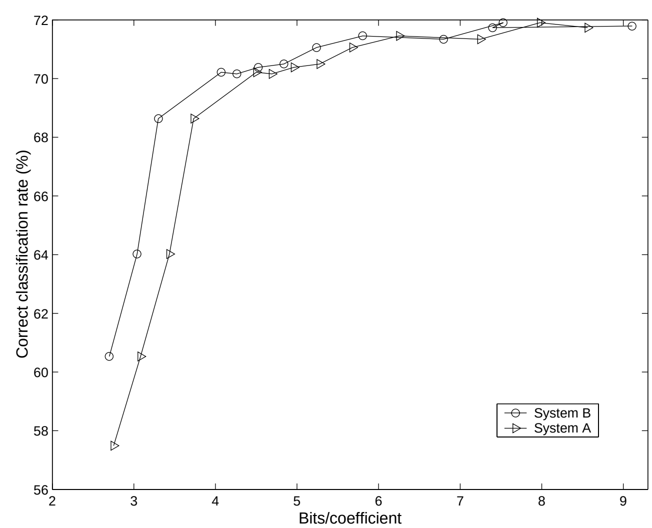

Figure 3: Probability of correct classification for Systems A and B

that System B achieves better performance due to the fact that the individual quantizers Q_i can exploit the local statistics of the subspaces S_i. Note that because the texture labels are given to the test images beforehand, it is possible that using the tree classifier will result in a wrong label even when unquantized features are used.

B. Analytical Model-Based Bit Allocation for Optimization of Region of Interest Coding

In this section, we address the problem of allocating bits to different regions in an image coded with a progressive wavelet coder, such as ICER, SPIHT (Set Partitioning in Hierarchical Trees) [1] or JPEG 2000, in order to achieve Region of Interest (ROI) coding objectives. In most progressive wavelet coders, each wavelet coefficient is successively refined, bitplane by bitplane, starting with the most significant bit. When the refinement process starts, most of the information to be transmitted is zero, e.g., in a b-bit representation, if most of the wavelet coefficients have magnitudes smaller than 2^{b-1}, then the most significant bitplane consists mainly of zeros. Various techniques have been proposed that enable efficient representation of these zeros. The basic idea is that very little information about a given wavelet coefficient is sent until it becomes "significant," i.e., until a non-zero bit is reached in the bitplane-by-bitplane progression. Generally speaking, a progressive coder transmits information about large-magnitude coefficients before it transmits information corresponding to smaller coefficients.

This intuition leads to a very simple technique to provide ROI coding [2, 3]. The goal in ROI coding is for the region(s) of interest to be transmitted with higher quality than other areas in the image. Since large coefficients are sent first, it is enough to divide the wavelet coefficients in areas outside the ROI by a factor greater than one, so that they are transmitted later (on average) in the resulting bitstream. We call this dividing factor the *priority scaling factor* (PSF) ρ. At the decoder the reconstructed coefficients are then multiplied by the corresponding PSF before inverting the wavelet transform. Since all the coefficients (after scaling) have been refined to a particular bitplane it follows that those coefficients to which a PSF $\rho > 1$ has been applied will be more coarsely quantized, i.e., their binary representation will be "shifted" with respect to coefficients with PSF $\rho = 1$. As a result, more bits per pixel are used (on average) for the ROI than for the rest of the image.

This approach, proposed in [2, 3], is a simple and effective way to achieve the goal of providing a different bit allocation for each region in an image. Prior work on bit allocation for ROI coding was based on heuristic techniques or required that rate-distortion characteristics be measured at each of the potential operating points. For example, a design based on empirical data could start by measuring overall image rate-distortion data at a number of different PSF values, and then proceed to selecting the optimum or the most appropriate PSF for the application.

The main contribution of our work is to provide a model-based bit allocation technique to determine the different PSFs to be used in an image, given a distortion criterion based on the relative importance of each region. Our goal is to choose the PSFs such that a total rate budget R is met and a criterion based on the relative distortions of the regions is optimized. We take SPIHT as an example for which our analysis is valid. Other progressive wavelet coders can be similarly modeled.

Without loss of generality we consider only two regions, and divide the wavelet coefficients outside the ROI by a value of PSF to be determined from the model. The region of interest is designated as region 1, and the rest of the image is called region 2. In Figure 4, we show wavelet coefficients $\{x_1(i), i = 1, \ldots, N_1\}$, for region 1, and $\{x_2(i), i = 1, \ldots, N_2\}$, for region 2, sorted in descending order of magnitude. Since the coefficients in region 2 are divided while the coefficients in region 1 are left unchanged, this means that the

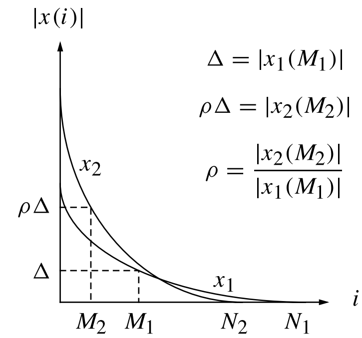

Figure 4: Sorted wavelet coefficients in two regions.

progressive bitplane refinement operates differently in each region. If the final coefficient sent at the final refinement has magnitude Δ, then coefficients larger than Δ in region 1 are significant, while in region 2 the only significant coefficients are those with magnitude larger than $\rho\Delta$. Equivalently, it is as if different quantization bins were used for region 1 and region 2, namely Δ and $\rho\Delta$, respectively.

i. Rate-Distortion Models

Our approach is an extension of Mallat's work [4], which provides a model for rate and distortion in a progressive wavelet coder. In this model, the average distortion $D(M)$of an image is represented as

$$D(M) = \left(1 + \frac{1}{12}\right) \frac{1}{N} \sum_{i=M+1}^{N} |x(i)|^2 \qquad (4)$$

where N is the total number of wavelet-transformed coefficients, M is the number of significant coefficients, and $\{x(i), i = 1, \ldots, N\}$ are the wavelet coefficients sorted in monotonically decreasing order by amplitude. The number of significant coefficients is directly related to the bit rate R according to

$$M = \left(\frac{N}{5.8}\right) R \qquad (5)$$

The coefficients $\frac{1}{12}$ and 5.8 in (4) and (5) are fixed constants for arbitrary images for the SPIHT codec used in our experiments.[2]

For ROI coding, we can write separate distortion equations for each of the two regions:

$$D_1(M_1) = \left(1 + \frac{1}{12}\right) \frac{1}{N_1} \sum_{i=M_1+1}^{N_1} |x_1(i)|^2 \qquad (6)$$

$$D_2(M_2) = \left(1 + \frac{1}{12}\right) \frac{1}{N_2} \sum_{i=M_2+1}^{N_2} |x_2(i)|^2 \qquad (7)$$

The overall rate R is still given by (5) in terms of the total number of significant coefficients transmitted, $M = M_1 + M_2$.

ii. Optimization of Bit Allocation for ROI Coding

We can reduce the distortion in region 1 by increasing M_1. But for a given overall rate R this increases the distortion in region 2 because M_2 must be reduced to keep $M = M_1 + M_2$ constant. There are many possible distortion criteria that can be used to encode an

[2] in this work we used version 6.05 of SPIHT

image with ROIs. For example, one could decide on a minimum acceptable quality for region 2, determine the requisite M_2 from (7), and then assign the remaining significant coefficients to region 1, $M_1 = M - M_2$.

We consider another case where the goal is to optimize a weighted distortion metric, which assigns a higher weight to the distortion in region 1. In other words, we seek to minimize $w_1 D_1(M_1)+D_2(M_2)$, $w_1 > 1$, subject to $M_1 + M_2 \leq M$. Lagrangian optimization techniques can be used [5] in this case to determine the optimum M_1, M_2 from (6) and (7).

iii. Determination of the Optimum Priority Scaling Factor

After an optimum pair M_1, M_2 are determined according to the desired distortion criterion, it is a simple matter to locate the smallest coefficients declared to be significant in each region, namely $x_1(M_1)$ and $x_2(M_2)$. Then from the earlier discussion it follows that the desired PSF is

$$\rho = \frac{|x_2(M_2)|}{|x_1(M_1)|}. \tag{8}$$

This is illustrated in Figure 4.

iv. Experimental Results

For our experimental results, we applied the weighted distortion criterion to the standard gray-level lena image of size 512×512, with a rectangular ROI of size 16×16 in the middle of the image. The whole image, after dividing the wavelet coefficients outside the ROI by the PSF, is coded by SPIHT at rate 0.5 bps. Figure 5 compares the results achieved using our model-based PSF with those derived from an exhaustive search of admissible PSFs. The figure shows mean

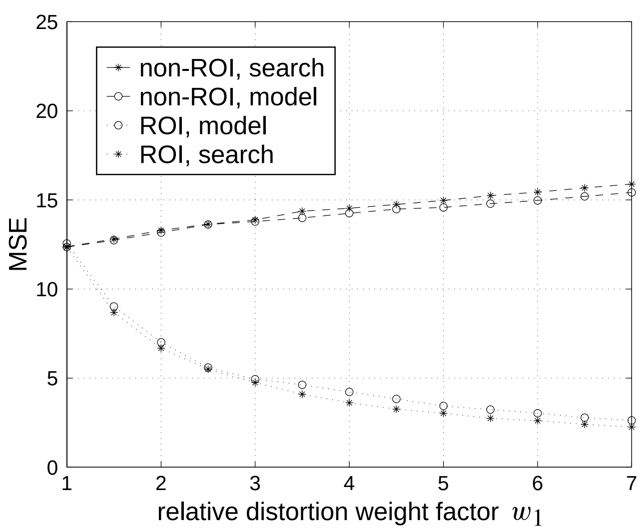

Figure 5: Performance of model-based bit allocation for a weighted distortion criterion.

squared error (MSE) distortion, for both the ROI and the remainder of the image outside the ROI, versus the relative distortion weight factor w_1. It compares results obtained by selecting the PSF ρ based on our model to those obtained by exhaustive search for the value of $\rho \in \{1, 1.1, 1.2, \ldots, 400\}$ that provides the best weighted distortion. We see from the figure that the MSEs achieved by our model-based PSF very closely approximate those achieved by the PSF obtained by exhaustive search, both inside and outside the ROI.

C. Rate control of prioritized data with a buffer constraint

We assume that each image is progressively coded to produce a chain of packets that can be truncated after an arbitrary number of packets to yield reconstructions of varying fidelity. For a given image the packets are sequenced according to their order of importance in the progressive stream, i.e., a coarser layer is always sent before a finer layer. To each packet there can be attached a distortion value, which is the the "provisional" distortion that would be achieved in reconstructing the image if its chain of progressively coded packets were terminated without sending the given packet.

The ROI-coded images are stored in a finite buffer and then transmitted through a constant bit-rate (CBR) channel. The central question is how to assign bits to each of the images given the constant channel rate and the limited buffer size. To solve this problem, we first need to select a criterion for our allocation. Since the objective is to transmit as much high priority data as possible, we aim to minimize the highest level of priority of data that cannot be transmitted. This is equivalent to the minimax (MMAX) distortion criterion [11] that measures the distortion of the coded image that has higher distortion than all other coded images in the image sequence.

In our problem, the constraints are a constant transmission rate C and buffer size B. We assume one image is coded every T seconds and is immediately moved to the transmission buffer after encoding. Then the "off-line" optimal solution based on the MMAX criterion (OOM) can be formulated as follows.

$$\text{Minimize}: \max_{\text{i}}(D_i), \tag{9}$$

subject to the constraint

$$B_i \leq B \ \text{ for all } i, \tag{10}$$

where D_i is the distortion of the i^{th} image, and B_i is the buffer occupancy after coded packets from the i^{th} image are moved into the buffer, i.e., $B_i = \max\left(0, B_{i-1} + R_i - C \times T\right)$, where R_i is the selected bit allocation for the i^{th} image and $C \times T$ is the number of bits transmitted during the interval since the previous image. For a given image i, the rate-distortion pair (R_i, D_i) is determined by selecting the number of packets n_i to transmit, i.e., the point at which to truncate the corresponding packet chain. The off-line optimal algorithm can be described as follows:

Algorithm OOM (*Off-line Optimal bit allocation in a CBR channel with buffer constraints under the MMAX distortion criterion*):
Step 0: *Initialize buffer occupancy and quantize all images with the coarsest quantization available.*
Step 1: *Find the image i that has maximum distortion D_i, and increment the number of retained packets n_i for this image. This decreases the distortion D_i of the i^{th} image, and increases its bit allocation R_i.*
Step 2: *Simulate the buffer behavior by placing all the images in the buffer, in the order in which they are generated and with their current bit allocation.*
Step 3: *If the buffer has not overflowed then go to Step 1. Otherwise, undo the last adjustment in Step 1 and STOP.*

After application of the OOM algorithm, the image with maximum distortion is the one for which it was not possible to increase the number of packets without producing buffer overflow. Fig. 6 depicts an example of OOM. The slope of the diagonal lines is the transmission rate per image arrival, $C \times T$, and the distance between

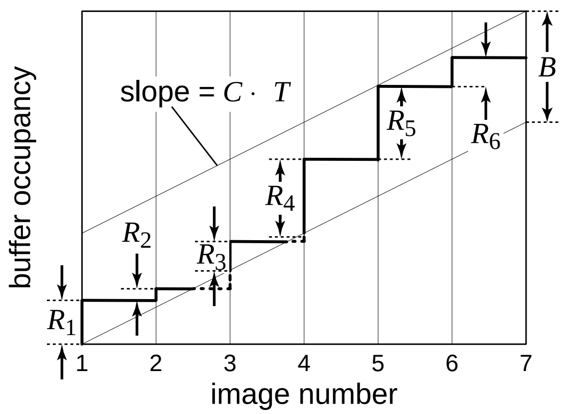

Figure 6: Buffer occupancy in CBR transmission for off-line optimal bit allocation under the MMAX criterion.

the two diagonal lines is the size of the transmission buffer, B. The buffer occupancy B_i at the time of processing the i^{th} image is indicated by the vertical distance from the lower diagonal line to the solid staircase line. The height of an individual riser of this staircase above the lower diagonal is the bit allocation R_i for the i^{th} image. Thick lines and dashed lines indicate the amount of buffer fullness and buffer underflow, respectively.

The OOM algorithm can only be implemented if the rate-distortion characteristics of all images are known in advance. In a real situation, decisions must be made on current images before the statistics of future images are known. Therefore we would like to establish whether there exist good “on-line” methods that achieve performance close to that of the off-line optimal algorithm.

In this paper, we show that an on-line method with buffer sorting under the MMAX criterion (OSM), achieves the same performance as OOM, as long as the granularities of buffer admissions and transmissions are the same. To show this, we add a granularity constraint that B, R_i and $C \times T$ are all integer numbers of packets, where a packet is the basic data unit produced by the progressive coder and is also the basic unit of sorting. Packets from a new image always come into the buffer immediately after transmission of another packet is finished (except when the buffer was empty).

The on-line method with buffer sorting can be described as follows. The buffer admits packets sequentially in real time as each image’s chain of packets is coded. All packets are accepted until the buffer starts to overflow. When overflow occurs, the packet with the lowest priority in the entire buffer is always the one discarded. Conversely, the packet with the highest priority is always the next packet transmitted. The highest priority packet is the one whose provisional distortion value is highest, and the lowest priority packet is the one whose provisional distortion value is lowest.

First we show that OSM is not better than OOM. If OSM were to outperform OOM, then at least one more packet should be sent for the image that has maximum distortion. But the image with highest distortion under OOM is the one that causes a transition to the buffer-full state; otherwise, its distortion could have been reduced by retaining one more packet. For the example in Fig. 6, this is the 5^{th} image. If it were possible to send more packets under OSM to reduce the maximum distortion, a packet from the maximum-distortion image would have to be moved to a different time interval since the buffer would overflow if any packet were added during its own time interval. But a packet cannot be moved to previous intervals since the system is causal, and it cannot be moved to later intervals since the buffer is already full and cannot hold any more packets for later admission. Therefore, OSM cannot be better than OOM.

Next, we show that OOM can be achieved by OSM. This follows because OSM always transmits the highest priority packets, always keeps the buffer as full as possible, and only discards packets with lowest priority. Any packet for image j that is dropped under OSM must have been discarded during the time interval for some image $i \geq j$. At the time of its discard, that packet must have had lower priority than any other packet retained in the buffer, i.e., the buffer at time i was completely full of higher priority packets. Under OOM all of these higher priority packets would have been admitted ahead of the given discarded packet, and the buffer-full state would have been reached at time i, thus disallowing further admittance of lower priority packets. Thus, no packet discarded by OSM could ever have been admitted to the buffer by OOM. This shows that we can achieve the off-line optimal solution under the MMAX criterion by using the on-line method with full buffer sorting.

The preceding analysis ignores the fact that there is a need to transmit header information under OSM to tell the decoder the image index of each sorted packet. However, for reasonably sized buffer packets the price of this overhead is small.

REFERENCES

[1] A. Said and W. A. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, no. 4, pp. 243-250,* June 1996.

[2] E. Atsumi and N. Farvardin, “Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees,” *Proc. IEEE International Conference on Image Processing (ICIP 98, pp. 87-91),* Oct. 1998.

[3] D. Nister and C. Christopoulos, “Lossless region of interest with a naturally progressive still image coding algorithm,” *Proc. IEEE International Conference on Image Processing (ICIP 98, pp. 856-860)*, Oct 1998.

[4] S. Mallat and F. Falzon, “Analysis of low bit rate image transform coding,” *IEEE Trans. on Signal Processing, vol. 46, no. 4, pp. 1027-1042*, April 1998.

[5] Y. Shoham and A. Gersho, “Efficient bit allocation for an arbitrary set of quantizers,” *IEEE Trans. Acoust. Speech Signal Process., ASSP-36(9), pp. 1445-1453*, Jan. 1988.

[6] H. Xie and A. Ortega, “Feature representation and compression for content-based image retrieval,” *VCIP*, vol. 4310, pp. 111–122, Jan. 2001.

[7] S. Rasoul and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 21, pp. 660–674, May. 1991.

[8] P. A. Chou, T. Lookabough, and R. M. Gray, “Optimal pruning with applications to tree-structured source coding and modeling,” *IEEE Trans. on Info. Theory*, vol. IT-35, pp. 299–315, March. 1989.

[9] P. Brodatz, “Textures: A photographic album for artists & designers,” Dover, New York, 1966.

[10] T. Chang and C. C. Jay Kuo, “Texture analysis and classification with tree-structured wavelet transform,” *IEEE Trans. on Image processing*, vol. 2, pp. 429–441, Oct. 1993.

[11] G. Schuster, G. Melnikov, and A. Katsaggelos, “A review of the minimum maximum criterion for optimal bit allocation among dependent quantizers,” *IEEE Trans. on Multimedia*, vol. 1, pp. 3–17, Mar. 1999.