Decision Trees for Error Concealment in Video Decoding

Song Cen and Pamela C. Cosman, *Senior Member, IEEE*

***Abstract*—When macroblocks are lost in a video decoder such as MPEG-2, the decoder can try to conceal the error by estimating or interpolating the missing area. Many different methods for this type of post-processing concealment have been proposed, operating in the spatial, frequency, or temporal domains, or some hybrid combination of them. In this paper, we show how the use of a decision tree that can adaptively choose among several different error concealment methods can outperform each single method. We also propose two promising new methods for temporal error concealment.**

***Index Terms*—Adaptive error concealment, decision trees, MPEG video.**

I. Introduction

When video signals are compressed and transmitted over unreliable channels, some strategy for error control or concealment must be employed. Possible strategies include forward error correction added at the encoder, post-processing methods employed by the decoder, and interactive requests for repeated data, involving both encoder and decoder. In this paper, we are concerned with post-processing methods employed by the decoder. We consider the single-layer case where coding modes, motion vectors, quantized DCT coefficients, and other information about macroblocks are all sent with the same priority. When errors strike the bitstream, we assume the decoder loses all information about that slice up to the next resynchronization point. In the absence of block interleaving, a horizontal swath of macroblocks is missing, and the decoder's post-processing methods must conceal this from the viewer.

Many post-processing error concealment methods have been proposed (see, e.g., [1], [2], [4], [5], [7]–[10], [12]–[15]). They can be divided into three main approaches: frequency, spatial, and temporal. There are also hybrids of these three main groups, and the methods can be made adaptive.

In *frequency concealment* [9], [12], DCT coefficients of missing blocks are estimated using the corresponding DCT coefficient of neighboring blocks, or using the neighbor's DC values, or other neighborhood features. These methods usually attempt to estimate low frequency coefficients.

Manuscript received July 18, 2000; revised December 19, 2001. This work was supported by the National Science Foundation and by the Center for Wireless Communications at the University of California at San Diego. This work was presented in part at the 1999 IEEE Wireless Communications and Networking Conference and the 1999 Data Compression Conference. The associate editor coordinating the review of this paper and approving it for publication was Dr. Thomas R. Gardos.

The authors are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: pcosman@code.ucsd.edu).

Digital Object Identifier 10.1109/TMM.2003.808825

Spatial concealment: One can interpolate directly in the spatial domain (e.g., [10], [13], [14]). If one had neighboring blocks on all four sides, then each pixel in the missing macroblock (MB) could be reconstructed with bilinear interpolation from the four nearest pixels that are not missing. If MBs are available only above and below the missing MB, then one can do one-dimensional linear interpolation from the nearest pixels from the blocks above and below the missing MB. Other strategies exist, e.g., directional interpolation that seeks to preserve edges [14]. In general, spatial concealment is complex since a computation must be done for each pixel.

In *temporal concealment*, blocks from other frames are used for concealment, either by attempting to reconstruct the motion vector of the lost MB, or by searching for a block that has a good match to the sides and neighborhood of the missing block (see, for example, [1], [2], [8]). If the estimation of the motion vector (MV) is inaccurate, the block obtained will have distracting artifacts at the boundaries with its neighbors. The MV can be estimated using, for example, the average or median of the MVs from MBs above, below, and diagonal. Alternatively, each neighboring macroblock's MV can be considered as a candidate MV for the missing MB, and the corresponding candidate reference blocks are all checked to see which one produces the best match for the boundary pixels.

Hybrid methods: A variety of hybrid algorithms have been proposed which combine more than one of the frequency, spatial, and temporal approaches. For example, in temporal concealment, the referenced block can be improved by spatial smoothing at its edges, to make it conform to the neighbors, at the expense of additional complexity. In [17], a MB is estimated by satisfying a weighted combination of spatial and temporal smoothness constraints.

Adaptive methods: Often, error concealment (EC) involves using a single fixed method for reconstructing any MB which is lost, however, a few adaptive EC methods have been proposed. In [17], the coding mode and block loss patterns are clustered into four groups, and the weighting between the spatial and temporal smoothness constraints depends on the group. A further level of adaptivity appears in [11] and [6]. In [11], temporal concealment is used for most blocks. However, a scene detector (which looks at the mean and variance of the MVs in a frame, as well as at the number of intracoded blocks) attempts to detect scene changes and irregular motion, in which case temporal concealment is likely to do poorly. In that case, a decision is made next on complexity. If there are too many lost blocks, then spatial concealment cannot be used in real time for this frame. If the complexity constraint is satisfied, then a last choice is made between frequency and spatial concealment based on a

TABLE I
OUR SET OF AVAILABLE METHODS FOR ERROR CONCEALMENT

Method Name	Frame Type	How it works
spatial	I,P,B	interpolate linearly from boundary pixels in top/bottom MBs
frequency	I	weighted average of first 9 DCT coefficients of top/bottom MBs
panning	I,P,B	use the camera panning motion vector
top/botMV	P,B	use top MV for top 8×16 sub-MB, use bottom MV for bottom 8×16 sub-MB
averageMV	P,B	use the average motion vectors of top and bottom MBs
useonlyMV	P,B	top or bottom MB is Intra-coded ⇒ use the only MV available
spat+onlyMV	P,B	use only available MV for nearest half, spatial interpolation for rest
copyPmb	I	copy co-sited MB from previous P frame if it's Intra-coded or has MV=0

five-category classification of the overall activity and color requirements of the video. Similar criteria are used in [6]. For P and B frames, if the variance of the MVs in the frame does not exceed a threshold, and also the number of intracoded MBs does not exceed its threshold, then spatial concealment is used, otherwise temporal is used. For I-frames, a choice is made between spatial and frequency concealment.

The research presented in this paper is in a similar spirit of adaptivity, with the added new feature that the encoder can guide the decoder in choosing among EC methods. Of the many methods available, which one is best will depend on the characteristics of the missing block, its neighbors, and the overall frame. We design a decision tree which can examine these characteristics and choose among several EC methods. The tree splits are chosen based on a sequence of data, and thus the trees are tailored for particular sequences, or for an individual group of pictures (GOP). The decision trees must therefore get included as side information with each sequence or with each GOP.

There are other approaches in which enhanced capability for error concealment comes at the expense of added side information. The MPEG-2 standard allows the use of concealment motion vectors (CMVs) to be transmitted for all intracoded MBs [4]; this leads to enhanced EC, but the bits employed for the CMVs detract from the source coding rate. Similarly, the use of more slices per horizontal strip costs more bits but allows faster re-synchronization. The transmission of EC decision trees provides yet an additional tradeoff between the bit-rate for side information and the EC advantage under noisy conditions.

As we will see, the decision tree provides lower distortion in the concealed blocks than does the use of any single fixed concealment method among those tested. We also present two new methods for temporal EC. The paper is organized as follows. In Section II, we present our new EC methods, as well as other ones available to our decision tree. In Section III, we discuss the classification tree design and our experiments. Results and conclusions are described in Section IV.

II. ERROR CONCEALMENT METHODS

We considered the eight different EC methods listed in Table I. The first column lists the name by which we reference the method, the second column lists the types of frames for which it is used, and the last column summarizes how it works. Note that we consider only MBs which have both top and bottom vertical neighbors in a frame, i.e., do not reside in the first or last slice of a frame.

The spatial interpolation method works by linearly interpolating within a vertical column from the two nearest pixels in the adjacent top and bottom MBs. For us, an MB consists of 6 blocks of 8×8 pixels, four from the luminance plane and one from each chrominance plane. In frequency interpolation, the lowest nine DCT coefficients (for each of the six blocks composing the missing MB) are estimated by a weighted average of the corresponding lowest nine DCT coefficients of the blocks above and below. Both spatial and frequency interpolation can be used for any type of frame. However, this frequency interpolation requires the presence of the neighbor's DCT coefficients, thus both top and bottom MBs must be intracoded, which normally happens less than 5% of the time for P frames and 0.5% for B frames. So frequency concealment is not used for P and B frames.

There are five different methods which depend on the presence of other motion vectors in the frame, and so are not immediately applicable to I frames. Two of these are new, and are denoted "panning" and "top/botMV."

Panning: When a camera pans, many MBs in a scene have similar MVs corresponding to the true panning motion. Individual MBs might have different MVs for a variety of reasons (e.g., noise, object motion). If the global panning MV can be estimated, it might constitute a better estimate for EC purposes than the MVs of the neighboring blocks. We estimate the panning MV by putting all nonzero MVs for the current frame into a histogram with 47×47 bins, which represents MVs ranging from $[-23, 23]$ half pixels in both dimensions. The histogram bin with the largest count is assumed to represent the global pan. This was found to give better results compared to just averaging together all nonzero MVs for the frame, since averaging includes objects which may be moving contrary to the global panning direction. This method can be applied to I frames by using the panning parameters estimated from the previous P frame.

Top/botMV: In a P or B frame, if both the top and bottom MBs have MVs associated with them, we can estimate the MV for the missing MB by taking the average of the ones above and below (averageMV method). If the MVs above and below are very different in magnitude or direction from each other, it might not make sense to average them. Instead, we might wish to use the MV for the block above for the top half of the missing MB (8×16 submacroblock for luminance and 4×8 subblock for chrominance), and use the MV for the block below for the bottom half. This is called the top/botMV method. This method performs very well, but it has the disadvantage that since we are not providing one single MV for the missing MB, we cannot consider this error concealer as a front-end to a standard MPEG-2 decoder. If exactly one of the top or bottom MBs is intracoded, then we have only one motion vector to go by. We might want to use this one as the MV for the entire missing MB (useonlyMV) or we might want to use it only for the half MB to which it is closer, using spatial interpolation for the other half (spat + onlyMV).

The last method in Table I was employed only for I frames. If the co-sited MB in the previous P frame was intracoded, or had a motion vector with value zero, then that MB might be useful directly as a replacement for the missing MB. If, however, the co-sited MB had a nonzero motion vector, then likely it is not an accurate reconstruction of the current missing I frame MB. This method is referred to as copyPmb.

III. Experiments

We encoded six sequences, each of 150 frames, with MPEG-2 at a rate of 1.5 Mbits/s. As the size of each frame is 352 × 240 pixels, there are $352/16 * (240/16 - 2) = 286$ MBs per frame that are not in the first or last slices. For all 286 MBs in all frames that are of the same type (i.e., I, P, or B) in each sequence, a classification tree was built at the encoder. It was then transmitted to the receiver. The transmission of the tree was assumed to be error free which can be achieved by duplicating it several times in different packets if it were packed together with the video data, or it can be packed in separate packets and transmitted using a reliable protocol (e.g., TCP). The video data itself is transmitted with some unreliable transport protocol (e.g., RTP or UDP). Upon error when receiving the video data, the decoder employs the decision tree to find the recommended concealment method for every lost macroblock.

The tree classifies every MB (not in the first and last slice of a frame) to a concealment method based on its characteristics. The true class of every MB is defined to be the best concealment method for that MB, that is, the method among those applicable in Table I which provides the minimum MSE. The true class of every MB is obtained at the encoder by reconstructing every MB with all applicable candidate methods, and computing the MSE with the original MB. The true class of every MB is not available at the decoder. Instead, the decoder will classify each lost MB using the tree designed by the encoder and transmitted to the decoder.

The classification parameters are used at the encoder to design the classification tree, as well as at the decoder to classify the MB in order to find its recommended concealment method. They are measurements which describe the spatial, temporal, and frequency domain context of a MB. They must be parameters which are available to the decoder even if the MB in consideration and the slice containing it are lost. For example, they include information on the MVs for the MBs above and below, and on the number of nonzero AC coefficients for the MBs above and below. For missing I, P, and B frame MBs, the parameters contained 18, 19, and 21 measurements, respectively. These include both ordinal and categorical variables. For example, the coding mode of the top MB is a categorical variable; the magnitude of its MV (if it has one) is an ordinal variable. Some of these parameters are listed in Table II.

TABLE II
Examples of Parameters Provided as Inputs to the CART Algorithm. The First Column Provides a Description of What the Parameter Represents, and Whether it is an Ordinal or Categorical Variable. For Categorical Variables, the Number of Categories is Listed. The Second Column Lists the Types of Frame for Which it is Applicable

Parameters related to macroblock position:		
Vertical position of lost MB	O	I,P,B
Horizontal position of lost MB	O	I,P,B
GOP number of the lost MB	O	I,P,B
Frame # (among the same type) of the lost MB	O	P,B
Parameters related to object motion:		
top/bot MBs intra/zeroMV/nonzeroMV mode	C(9)	P
top/bot MBs intra/forw/back/interp mode	C(16)	B
Absolute sum of top/bot MVs	O	P,B
Amplitude of diff. vector (top_MV - bottom_MV)	O	P,B
Angle of diff. vector (top_MV - bottom_MV)	O	P,B
Parameters related to panning motion:		
Percentage of MBs with MV equal 0 in (prev) P frame	O	I,P
$\max(\lvert top\_MV - pan\_MV\rvert, \lvert bot\_MV - pan\_MV\rvert)$	O	P,B
Parameters related to texture:		
|deviation| of luminance values in top/bot blocks	O	I,P,B
Sum of selected DCT coefficients in neighbors	O	I
Avg. position of last non-zero DCT coef of top/bot	O	I
Avg. number of non-zero DCT coefficients of top/bot	O	I

CART[1] algorithm for designing classification and regression trees [3] was used to design the tree at the encoder. It works as follows. Let x be the vector of measurements associated with the missing MB (including those shown in Table II and others that are not shown), and let $C = 1, 2, \ldots, 8$ be the set of eight EC methods listed in Table I. A classifier is a function $d(x)$ which assigns to every vector x a class j from C. A learning sample $\mathcal{L}$ (also called a training sequence) consists of data $(x_1, j_1), (x_2, j_2), \ldots, (x_N, j_N)$ corresponding to all N MBs from the same type of frames in a video sequence; that is, N "lost" macroblocks for which the best concealment method is known. N equals 2860 MBs and 11 440 MBs for the ten I frames and the 40 P frames in each video sequence, respectively.

The root node of the tree contains all the N training cases. To design the classification tree, we consider, for each terminal node of the tree, a standard set of possible splits of the data in that node. In the standard set, each split depends on the value of only a single variable. For each ordinal variable x_m, we include all splitting questions of the form "Is $x_m \leq c$?" For example, "Is the vertical position of the missing $MB < 10$?" If x_m is categorical, taking values in $B = \{b_1, b_2, \ldots, b_L\}$, then we include all questions of the form: "Is $x_m \in S$?" as S ranges over all subsets of B.

There is a finite number of distinct splits. For each variable, we find the split which provides the greatest decrease in node impurity. (We used the Gini index of diversity [3] to measure the purity of a set of data.) We compare all of these, and find the best overall split of the data. A class assignment rule assigns an EC method $j \in \{1, 2, \ldots, 8\}$ to every terminal node. We use the simple plurality rule which assigns to each terminal node the EC method which was best for the largest number of MBs in that node.

[1]CART is a registered trademark of of California Statistical Software, Inc., and is exclusively licensed to Salford Systems.

Once the tree is designed and transmitted to the decoder, whenever a loss occurs, the decoder will measure the input parameters associated with each lost MB, and will follow the splits in the decision tree to find out which EC method to use for that MB. The decoder will reconstruct each lost MB with the designated method.

How large a tree should one transmit as side information? The size of the tree, as measured by the total number of nodes, is directly proportional to the number of bits that will be required as side information to transmit the tree to the decoder. At each stage of tree growth, we are concerned with the size of the tree and its MSE performance. The MSE performance of the tree is measured by supposing that each MB in the sequence is lost, and reconstructing it using the EC method dictated by the tree. The average MSE for all MBs is then computed. If the tree is allowed to grow large enough, eventually the classification will be perfect. The MSE will then be what results from each MB being concealed by its *best* EC method among the set. We call this the "omniscient minimum" MSE, and it could also be obtained by transmitting a few bits explicitly for each MB to tell the decoder which EC method to use for that MB. What we consider the "maximum" MSE is the MSE that results from using a single fixed and best method from Table II. The last three methods are often excluded as candidates for the best fixed method since for most sequences they can be applied to less than 25% of the MBs. Since certain P,B methods cannot be used next to intra-coded MBs, the use of a single fixed method really means employing one method in all the cases to which it is applicable, and using other pre-determined methods in those cases where it is not. The same pre-determined method is also used when the method dictated by the decision tree is not applicable to the lost MB.

Our goal is to see whether much of this difference between the maximum MSE and the omniscient minimum MSE can be efficiently captured by the use of a decision tree, with significantly less overhead than is required by the explicit specification of EC methods for each MB. We are therefore interested in looking at plots of the MSE reduction versus the number of nodes as the tree grows. Trees were developed for several sequences, including mobile calendar, flower garden, bicycle, cact, susi, and tennis, as well as for separate GOPs from these sequences.

IV. RESULTS

We are interested in a loss scenario where occasional isolated slices (a horizontal strip of macroblocks) gets lost. Rather than obtaining results by averaging over random loss patterns (in which case thousands of random loss patterns would be needed, and the comparison between approaches would be obscured by the randomness inherent in the simulation), we chose instead to simulate the loss of each complete horizontal strip in the entire sequence individually, and our MSE results are for reconstructed frames in which *every* slice (except the top and bottom most ones) has been individually lost and concealed. This has the advantage of directly constructing a useful comparison of the decision tree concealment approach against any fixed concealment approach. For this reason, we report results primarily as relative MSE (the MSE of the tree approach compared to that

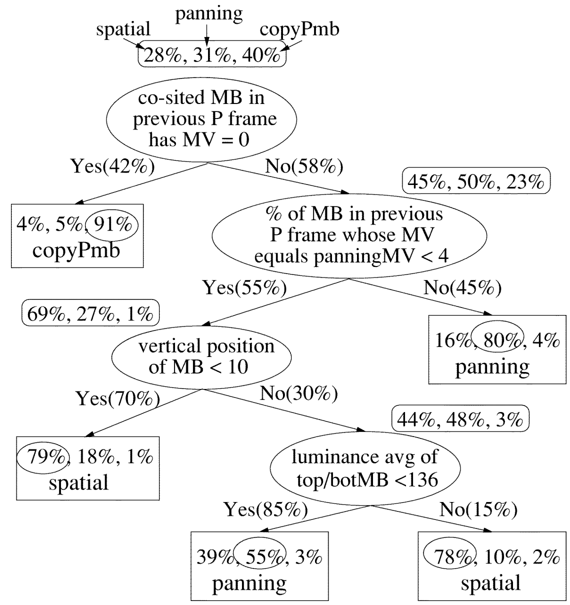

Fig. 1. Five-terminal-node tree grown for table-tennis I pictures. The ovals show the test which is used to split the data in that node. The percentages show the amount of data in each node that has the spatial, panning, or copyPmb concealment method as the best concealment method.

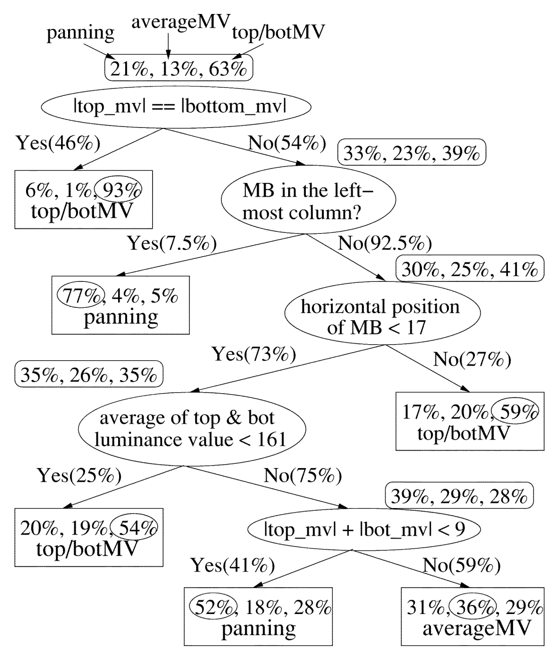

Fig. 2. Six-terminal-node tree grown for mobile-calendar P pictures.

of the best of the fixed concealment methods). Clearly though, by calculating the MSE for a frame based on losing each slice in the frame individually and then concealing it, the MSEs for all the concealment approaches are very much worse than the MSE of the compressed and reconstructed sequence with no slice loss. For this reason, although we also report results as absolute MSE (the MSE of the concealed frames) these values should not be compared to the MSE of compressed frames which underwent no MB loss.

Fig. 1 shows a CART tree with five terminal nodes built for the 10 I frames of the complete table tennis sequence. Fig. 2 shows the six-terminal-node tree grown for the data from the

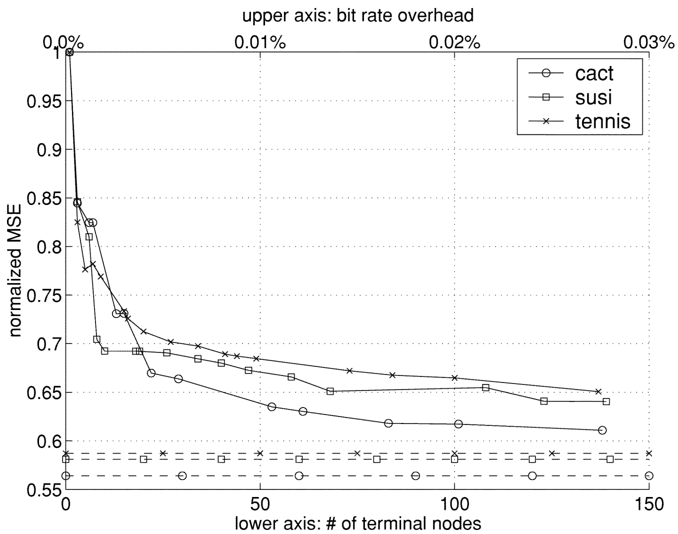

Fig. 3. MSE versus number of tree terminal nodes and bit rate overhead for 10 I frames of cact, susi, and tennis sequences.

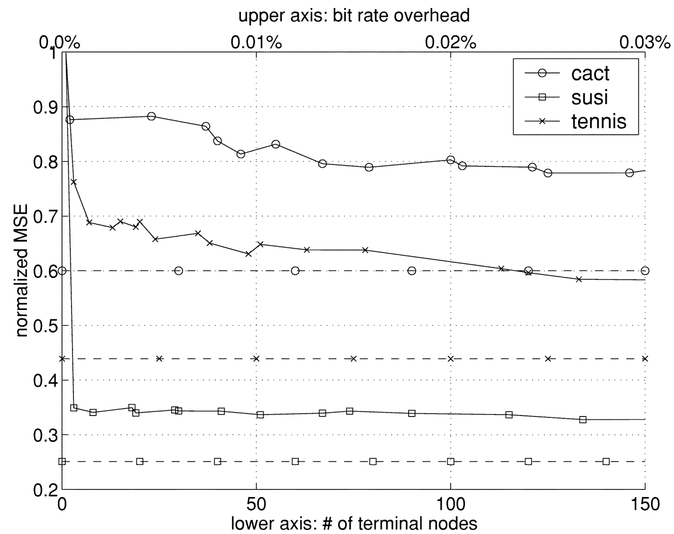

Fig. 4. MSE versus number of tree terminal nodes and bit rate overhead for 40 P frames of cact, susi, and tennis sequences.

mobile-calendar sequence 40 P pictures. At each node of these trees, the oval lists the splitting test which is applied to split the data of that node. Above the oval is listed for each node the percentage of the node data that has the spatial, panning, and copyPmb EC methods as their *best* EC method. For example, for the root node of the tree in Fig. 1, copyPmb wins 40%. This means that, if we take *all* the MBs from *all* the I-frames in the sequence, 40% of the time the best concealment method for those MBs is to use copyPmb. The spatial wins 28% of the time, panning wins 31%, and the frequency concealment method (not listed) makes up the remaining 1%. The test applied to this node is to check whether the co-sited MB in the previous P frame has motion vector equal to zero. The tree branches are labeled with the percentages of the data set that go down each branch. For the terminal nodes, the EC method that has the highest percentage of wins for that node data is selected by the plurality rule as the class for all data in the node. We see that 42% of the I-frame macroblocks *do* have a co-sited MB in the previous P-frame that has a motion vector of zero. For these data, the tree ends. These MBs will be reconstructed with the copyPmb method (copying the co-sited MB from the previous P-frame), and for 91% of them, this will in fact be the best that could be done. Of the I-frame MBs in the root node, 58% *do not* have a co-sited MB in the previous P-frame with $MV = 0$. For these MBs, there are further splits of the data, and either the panning method or spatial concealment ends up being used. The concealment methods dictated by this tree would result in an MSE that is 77% of the best fixed method, which translates to a 1.13 dB improvement in PSNR.

For long sequences, the overhead of transmitting a tree is amortized, and one can consider transmitting large trees. Plots of distortion versus number of terminal nodes and bit rate overhead for ten I frames of three different sequences—cact, susi, and tennis, appear in Fig. 3. For the 40 P frames from each of the same sequences, the plots of distortion reduction versus number of terminal nodes and bit rate overhead are shown in Fig. 4. In the plots, the maximum MSE is normalized to 1, corresponding to the MSE of the best single EC method out of the methods available. In the figure, the dashed horizontal lines show the omniscient minimum MSE, the lowest MSE which the decision tree reaches if it grows large enough. Note that misclassification error always decreases as the size of the tree increases; this does not necessarily mean the MSE also decreases because a larger tree may make fewer classification errors but which are more costly in terms of MSE. However as shown in the figures, MSE usually decreases with increasing tree size as well. The bit rate overhead corresponding to the number of terminal nodes is computed as follows. The total bit count for the video data is $5\ \mathrm{s} * 1.5\ \mathrm{Mbps} = 7.5$ Mbits as the 150 frames are encoded at 1.5 Mbps with 30 frames/s. For a binary tree, the number of internal nodes is always 1 less than the number of terminal nodes. For any node, 1 bit is needed to indicate which type of node it is. Further, for an internal node, another five bits are needed to specify which variable to split on, and seven more bits to specify the splitting threshold; while for a terminal node, only two (for four methods applicable to I frame MBs) or three (for six methods applicable to P/B frame MBs) more bits are needed to specify the concealment method to use.

For the table tennis I pictures, the best fixed method is copyPmb. This corresponds to the max MSE of 1. The omniscient minimum has a relative value of 0.59 and requires a bit rate overhead of 0.08% (as two bits are needed for all 2860 MBs = ten I frames $*$ 286 MBs/frame). As shown in Fig. 3, a tree with 53 terminal nodes (corresponds to a bit rate overhead of 0.01%) achieves a relative MSE of 0.67. The average depth of the 53-terminal-node tree is less than six, so the decoder needs to follow a sequence of 6 binary tests on the average to figure out which concealment method to use. In the same figure, trees with 61 and ten terminal nodes reach relative MSEs of 0.64 and 0.69 for the cact and susi I pictures, respectively (which have omniscient minima of 0.56 and 0.58, respectively). The best fixed method for the cact I frames was panning; and for the susi I frames, the spatial method. For P frames of all three sequences, the best fixed methods are the same, top/botMV.

TABLE III
MSE RESULTS (AVERAGED OVER 4 LUMINANCE AND 2 CHROMINANCE BLOCKS) FOR VARIOUS CONCEALMENT METHODS. ALL DECISION TREES HAVE LESS THAN 120 TERMINAL NODES. (A-MSE: ABSOLUTE MSE; R-MSE: RELATIVE MSE)

Method Name	I frames				P frames			
	susi		tennis		susi		tennis	
	A-mse	R-mse	A-mse	R-mse	A-mse	R-mse	A-mse	R-mse
spatial	146	1.00	964	2.16	148	1.04	927	2.13
frequency	304	2.08	1087	2.43				
panning	388	2.66	747	1.67	300	2.11	765	1.76
top/botMV					44.7	0.31	198	0.46
averageMV					51.9	0.37	241	0.55
copyPmb	231	1.58	165	0.37				
best fixed	146	1.00	447	1.00	142	1.00	435	1.00
no MB loss	3.63	0.025	25.8	0.058	3.70	0.026	20.4	0.047
omnisc. min	85	0.58	262	0.59	35.7	0.25	191	0.44
CART-tree	95	0.65	297	0.66	47.9	0.34	263	0.60

As shown in Fig. 4, the gains for P pictures are larger. The best results are for the susi sequence, where a tiny tree with about five terminal nodes achieves one third of the MSE of the best fixed method. This translates to a PSNR improvement of 4.7 dB. The omniscient minimum is about 25% of the MSE of the best fixed method, so the tiny tree is capturing most of the available gain. However, the omniscient performance comes at a cost of 0.45% bit rate overhead (three bits for all $40 * 286$ MBs), whereas a five-terminal node tree takes only about 0.001% bit rate overhead. The cact and tennis sequences are more typical. Compared to using the best single error concealment method, using a small tree (less than 10 terminal nodes) provides a 10% and 25% reduction in MSE for cact and tennis, respectively.

Some numerical results appear in Table III for the susi and tennis sequences. The MSE for each fixed method (used only on the MBs which are applicable) is listed, as well as the omniscient minimum MSE, and the MSE from using the decision tree. We list the MSE for the "best fix" which is where the best single error concealment method is used for all macroblocks for which it is applicable, and a different method where it is not applicable-this approach provides the normalized value of 1 in Figs. 3 and 4. These MSEs are all given in the column A-mse. Relative MSEs are provided in the column R-mse, where the "best fix" method is normalized to 1. We also list the MSE of the reconstructed sequences *with no MB loss*.

When implemented at the level of individual GOPs, tree success rates varied. In our experiment each GOP was composed of 15 frames, including one I frame, four P frames, and ten B frames. For P frames, in mobile-calendar, trees with about 40 terminal nodes capture 40% to 80% of the available MSE reduction, capturing on average about 70% and providing MSEs typically in the range of 75% to 90% of the best fixed EC method. In the flower garden sequence, often trees with only 20 terminal nodes can capture 75% to 95% of the available MSE reduction, providing MSEs in the range of 50% to 70% of the best fixed EC method. The overhead rate for sending one tree of size 20 terminal nodes for each GOP is 0.04%.

V. CONCLUSIONS

We have presented two new temporal EC methods, one based on estimating global pan parameters, and another based on separating a MB into top and bottom halves for separate use of MVs from above and below. These new EC methods were often the best choices among the fixed methods. The use of a decision tree to choose adaptively among the various methods consistently provided lower distortion than any fixed method alone.

We envision that decision trees could be designed for individual GOPs, or for individual frames; or decision trees could be designed for variable-length groups of data as the previous concealment strategy becomes outdated. The decision tree requires only a very small and adjustable level of overhead that depends on the tree size. The memory and computational requirements for this approach are quite asymmetric. The decoder only has to store a tiny tree; and the main computational complexity involved is using the tree and having more than one error concealment method available. On the other hand, the encoder needs to obtain and store the learning sample. The main computational complexity involved is finding the best concealment method for each MB and building the tree. As for the memory requirement at the encoder, the learning sample derived from a frame is about 20 times smaller than the frame itself, so a learning sample derived from a GOP or multiple GOPs may well be reasonable (consider the memory requirements of motion compensation over a large number of frames as proposed in [16]).

REFERENCES

[1] S. Aign, "A temporal error concealment technique for I-pictures in an MPEG-2 video-decoder," in *Proc. VCIP'98*, Jan. 1998, pp. 405–16.

[2] C. Alexandre and H. V. Thien, "The influence of residual errors on a digital satellite TV encoder," *Signal Process.: Image Commun.*, vol. 11, pp. 105–118, 1997.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

[4] S. Cen and P. Cosman, "Comparison of error concealment strategies for MPEG video," in *Proc. 1999 IEEE Wireless Communications and Networking Conf.*, Sept. 1999, pp. 329–333.

[5] S. Cen, P. Cosman, and F. Azadegan, "Decision trees for error concealment in video decoding," in *Proc. Data Compression Conf.*, Mar. 1999, pp. 384–393.

[6] P. Cuenca, A. Garrido, F. Quiles, L. Orozco-Barbosa, T. Olivares, and M. Lozano, "Dynamic error concealment technique for the transmission of hierarchical encoded MPEG-2 video over ATM networks," in *Proc. 1997 IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing*, vol. 2, Aug. 1997, pp. 912–915.

[7] P. Cuenca, L. Orozco-Barbosa, A. Garrido, F. Quiles, and T. Olivares, "A survey of error concealment schemes for MPEG-2 video communications over ATM networks," in *Proc. 1997 IEEE Can. Conf. on Electrical and Computer Engineering*, 1997, pp. 118–121.

[8] M. Ghanbari and V. Seferidis, "Cell-loss concealment in ATM video coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 238–247, June 1993.

[9] S. Hemami and T. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Trans. Image Processing*, vol. 4, pp. 1023–1027, July 1995.

[10] M.-C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos, "Error concealment algorithms for compressed video," *Signal Process.: Image Commun.*, vol. 14, no. 6–8, pp. 473–92, 1999.

[11] W. Luo and M. El Zarki, "Analysis of error concealment schemes for MPEG-2 video transmission over ATM based networks," in *Proc. VCIP'95*, vol. 2501, Taipei, Taiwan, May 1995, pp. 1358–1368.

[12] J. W. Park, J. W. Kim, and S. U. Lee, "DCT coefficients recovery-based error concealment technique and its application to the MPEG-2 bit stream error," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 845–854, Dec. 1997.

[13] P. Salama, N. B. Shroff, E. J. Coyle, and E. J. Delp, "Error concealment techniques for encoded video streams," in *Proc. IEEE ICIP*, 1995, pp. 405–16.

[14] J.-W. Suh and Y.-S. Ho, "Error concealment based on directional interpolation," *IEEE Trans. Consumer Electron.*, vol. 43, pp. 295–302, Aug. 1997.

[15] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, no. 5, pp. 975–775, May 1998.

[16] T. Wiegand, E. Steinbach, and B. Girod, "Long-term memory prediction using affine motion compensation," in *Proc. ICIP*, vol. 1, Oct. 1999, pp. 51–55.

[17] Q.-F. Zhu, Y. Wang, and L. Shaw, "Coding and cell-loss recovery in DCT-based packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 248–258, June 1993.

Song Cen received the B.S. degree in electrical engineering from the Nanjing University, China, in 1994. He received the M.S. and Ph.D. degrees in electrical engingeering from the University of California at San Diego, La Jolla, in 1998 and 2002, respectively.

He is currently Member of the Technical Staff at PacketVideo, Inc., San Diego. His research interests are in the areas of error control/concealment and transmission protocols for realtime video communications over the Internet and wireless networks.

Pamela C. Cosman (S'88–M'93–SM'00) received the B.S. degree (with honors) in electrical engineering from the California Institute of Technology, Pasadena, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1989 and 1993, respectively.

She was an NSF Postdoctoral Fellow at Stanford University and a Visiting Professor at the University of Minnesota, Minneapolis, during 1993–1995. Since July 1995, she is on the faculty of the Department of Electrical and Computer Engineering at the University of California at San Diego (UCSD), La Jolla, where she is currently an Associate Professor. Her research interests are in the areas of data compression and image processing.

Dr. Cosman is the recipient of the ECE Departmental Graduate Teaching Award (1996), a Career Award from the National Science Foundation (1996–1999), and a Powell Faculty Fellowship (1997–1998). She was an associate editor of the IEEE COMMUNICATIONS LETTERS (1998–2001), and was a guest editor of the June 2000 special issue "Error-resilient image and video coding" of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. She was the Technical Program Chair of the 1998 Information Theory Workshop in San Diego, and is a member of Tau Beta Pi and Sigma Xi. She is an associate editor of the IEEE SIGNAL PROCESSING LETTERS and a senior editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.