Title

Comparison of error concealment strategies for MPEG video

Permalink

https://escholarship.org/uc/item/70h1m0v5

Authors

Cen, S
Cosman, P C

Publication Date

1999

Comparison of error concealment strategies for MPEG video

Song Cen and Pamela Cosman
Department of Electrical and Computer Engineering, University of California at San Diego
9500 Gilman Drive, San Diego, California 92093
{scen,pcosman}@code.ucsd.edu

Abstract: When macroblocks are lost in an MPEG decoder, the decoder can try to conceal the error by estimating the missing area. Many different methods for this type of concealment have been proposed. In previous work, we showed how the use of a decision tree adaptively choosing among several different error concealment methods can outperform each single method. In this paper, we improve the decision tree approach, and compare it against the use of concealment motion vectors.

1 Introduction

When video signals are compressed and transmitted over unreliable channels, some strategy for error control or concealment must be employed. We consider the single-layer case where coding modes, motion vectors, quantized DCT coefficients, and other information about macroblocks (MBs) 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, and a horizontal swath of macroblocks is missing. The decoder seeks to conceal this from the viewer. A variety of alternatives exists: spatial, frequency, and temporal concealment (see [3] for a review). Which method is best depends 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 error concealment (EC) methods. The decision tree provides lower distortion in the concealed blocks than does the use of any single fixed concealment method among those tested. The decision tree is compared against the use of concealment motion vectors for I frames.

2 Error concealment methods

The post-processing error concealment methods with which we are concerned can be divided into three main groups: spatial, frequency, and temporal error concealment. We considered the eight different EC methods listed in Table 1. The first column lists the name by which we reference each method; the second column lists the types of frames for which it is used, and the last column summarizes how it works.

Spatial concealment (SC): One can interpolate directly in the spatial domain. If one had neighboring blocks on all 4 sides, then each pixel in the missing MB could be reconstructed, for example, by using bilinear interpolation from the four nearest pixels. The spatial interpolation method we used assumes that MBs are available only above and below the missing MB, and so it linearly interpolates within a vertical column from the two nearest pixels in the adjacent top and bottom MBs. In general, spatial concealment methods are the most complex, since a computation must be done for each pixel.

Frequency concealment (FC): In frequency concealment, some low-order DCT coefficients of the missing blocks are estimated using either the corresponding DCT coefficient of neighboring blocks, or the neighbor's DC values. These methods cannot be used to estimate high-frequency coefficients. In our FC, the lowest 9 DCT coefficients (for each of the 6 blocks composing the missing MB as in MPEG-2 main profile) are estimated by a weighted average of the corresponding lowest 9 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 intra coded, which normally happens less than 5% of the time for P frames and 0.5% for B frames. So FC is not used as a concealment method for P and B frames.

Method	type	How it works
spatial	I,P,B	interpolate from pixels in top/bot MBs
frequency	I	average 9 DCT coefs of top/bot MBs
panning	I,P,B	use the camera panning motion vector
top/botMV	P,B	top/bottom MVs for top/bottom halves
averageMV	P,B	average MVs of top/bottom MBs
useonlyMV	P,B	top or bot MB I-coded ⇒ use one MV
spat+onlyMV	P,B	half MV and half spatial interpolation
copyPmb	I	copy co-sited MB from prev. P frame

Table 1: Our available methods for error concealment

Temporal concealment (TC): One can attempt to reconstruct the motion vector (MV) of the lost MB, and use the referenced block for concealment. We considered five different temporal concealment methods which depend on the presence of other motion vectors in the frame, and so are not immediately ap-

0-7803-5668-3/99/$10.00 © 1999 IEEE.

plicable to I frames. In the "panning" method, we assume the camera is panning, and estimate the global panning MV by looking for the peak in a histogram of MVs after the zero MVs have been excluded. This MV is then used for temporal concealment. If the camera is actually not panning, then this MV would likely correspond to the motion of the largest object, and could be called a peak-motion-vector, rather than a panning vector. The method applies to I frames by using the panning parameters estimated from the previous P frame.

In a P or B frame, if both the top and bottom MBs have MVs associated with them, we can estimate the missing MV by averaging 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, and use the MV for the block below for the bottom half (top/botMV method). This method performs well, but has the disadvantage that since we do not provide one single MV for the missing MB, we cannot consider the error concealer as a front-end to a standard MPEG-2 decoder. If exactly one of the top or bottom MBs is intra-coded, 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 1 (copyPmb) was employed only for I frames. If the co-sited MB in the previous P frame was intra-coded, or had zero motion vector, then that MB might be useful directly as a replacement for the missing MB. If, however, the co-sited MB had a non-zero motion vector, then likely it is not an accurate reconstruction of the current missing I frame MB.

3 Decision trees

Of the 8 methods listed in Table 1, none is consistently best for all MBs. In our previous work [2], the EC method is chosen by a decision tree which looks at the context of the missing MB. The tree is designed at the encoder using the CART^{TM} algorithm [1], and is transmitted to the decoder. The approach is as follows. Let x be a vector of measurements associated with a missing MB. It can include both ordinal and categorical variables. The measurements are of four types: position of the lost macroblock (e.g., vertical and horizontal position of lost MB, frame number), object motion (e.g., magnitude of the MVs above and below, difference in amplitude and in angle between the MVs above and below, type of MBs above and below – intra, forward, or backward coded), panning motion (e.g., magnitude of panning vector, difference between panning vector and MVs above and below, percentage of MBs with MV equal to zero), and texture/intensity of the neighbors (e.g., sum and difference of intensities in the MBs above and below, standard deviation of the intensities in the MBs above and below, # of DCT coded blocks in the MBs above and below, etc.).

For each slice that is not the first or last slice of a frame, we considered the loss of that slice, and reconstructed each MB in the slice with each of the candidate methods which could possibly be used for it. For each MB, the method with the lowest reconstructed MSE (over both luminance and chrominance blocks) was considered the "winner" and became the classification associated with that MB. A training set $\mathcal{L}$ consists of data (x_1, j_1), (x_2, j_2), ..., (x_N, j_N) on N cases where the class is known, that is, N macroblocks for which the winning EC method has been determined. For each sequence, the training set was formed with the measurements taken for each MB and the associated class.

The vector of measurements attempts to describe the spatial and temporal context of a missing MB. In our previous work, for missing I, P, and B frame MBs, there were 29, 62, and 101 input parameters, respectively. Many of these parameters were found not to be useful for predicting best error concealment method, and so the parameter set was pruned down to 18, 19, and 21. The root node of the CART tree contains all the N training cases; a mix of best EC methods is present for the data in this root node. The goal of CART is to successively subdivide the training set using binary splits in such a way that the data associated with the terminal nodes of the tree do not have a mix of best EC methods; rather each node should be as "pure" as possible. A simple rule is to assign the most popular class for each terminal node; this is called the plurality rule.

At each stage of growth, we are concerned with the size of the tree and its MSE performance. 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. The MSE performance of the tree is measured by choosing, for each MB in the sequence, the EC method dictated by the tree, and reconstructing the missing MB. 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 not then be equal to the MSE of the noiseless channel case (no MB loss), but will be the MSE that 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 couple of 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 in Table 1. 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 where 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 CART 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 using a decision tree, with 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 overhead bit rate as the tree grows. Trees were developed for several sequences as well as for separate groups of pictures (GOPs) from these sequences.

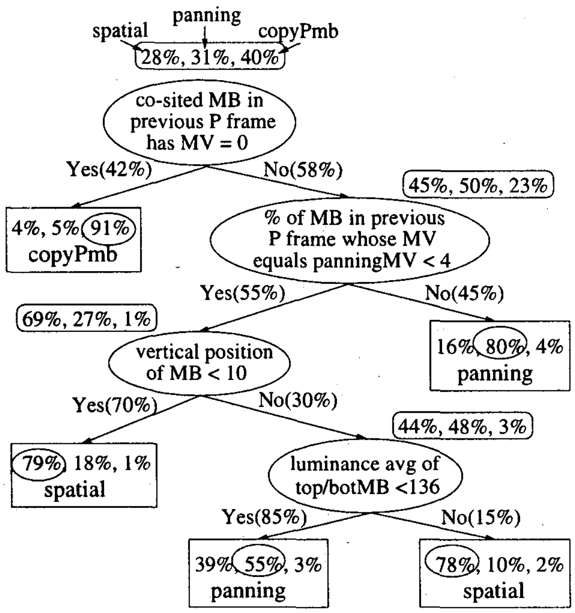


Figure 1: 5-terminal-node tree grown for table-tennis I pictures, achieves relative MSE 0.77 compared to an omniscient minimum MSE of 0.59

Figure 1 shows a CART tree with 5 terminal nodes built for the I frames of the complete table tennis sequence (150 frames). At each node of the tree, 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, the spatial wins 28% of the time, panning wins 31%, and copyPmb wins 40%. The remaining methods make up the remaining 1% of the time. 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.

3.1 Results of Decision Trees

Five sequences were encoded by MPEG-2 at a rate of 1.5 Mbits/sec. For each one, error concealment decision trees of different sizes were designed. At the decoder, we assumed that each slice was separately lost, and we reconstructed each macroblock in the slice using the error concealment method dictated by the decision tree. Thus a series of tree-size/distortion pairs were obtained.

Plots of distortion versus number of terminal nodes for I frames of 3 different sequences – cact, susi, and tennis, appear in Figure 2. 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. The 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.

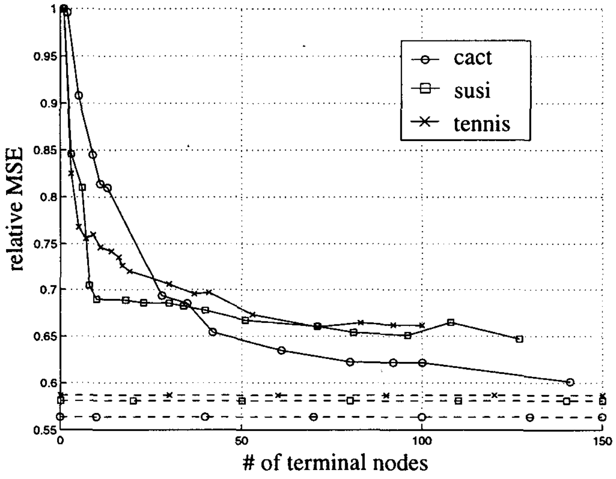


Figure 2: MSE vs. number of tree terminal nodes for I frames of cact, susi, and tennis sequences

For the table tennis I pictures, the best fixed method is copyPmb (corresponding to max MSE of 1), and the omniscient minimum has a relative value of 0.59 and requires a bit rate overhead of 0.08%. As shown in Figure 2, a tree with 53 terminal nodes (105 total nodes, corresponds to a bit rate overhead of 0.01% compared to 0.08% for explicit specification of EC methods), achieves a relative MSE of 0.67. The average depth of the 53-terminal-node tree is less than 6, so the decoder needs to follow a sequence of only 6 binary tests on the average in order to obtain the concealment method. In the same figure, trees with 61 and 10 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.

4 Concealment motion vectors

In our work thus far, the decision trees are tailored for particular sequences, or for an individual GOP. The decision trees therefore need to be included as side information with each sequence or GOP. This side information can be included in the "user data" which the standard allows. There are other approaches in which enhanced capability for error concealment comes at the expense of having to use side information. The MPEG-2 standard allows the use of concealment motion vectors (CMVs) to be transmitted for all intra-coded MBs. A CMV is found in a manner identical to a regular motion vector, that is, it points to the best match block in the reference frame, as located by some search strategy and some matching criterion. However, unlike regular motions vectors, a CMV is not used by the decoder unless an intra-coded MB is lost. The CMVs can be included with the MB data itself. In this case, if the MB data is lost, the CMV is lost too, and cannot be used for concealing the loss. However, if the MBs above and below have not been lost, the CMVs for those MBs can be used in reconstructing the lost MB. In our work, we assume that the CMVs are not sent with their corresponding MBs, but rather, they are sent in "user data" in the same way that the decision tree would be sent. Thus, we can assume that when an intra-coded MB is lost, its own, and most correct, CMV is available, and the decoder does not have to resort to using the CMVs of the neighbors. Whether in the MB data or the user data, the bits employed for the CMVs are the same, and they detract in a small way from the source coding rate.

We wanted to see whether CMVs or decision trees provide better error concealment for the bit rate that they require. The decision trees require an adjustable amount of overhead: bigger trees consume more overhead bits but provide better error concealment. For comparison, we wanted to consider CMVs with an adjustable amount of overhead rate as well. The concealment motion vector field can be sub-sampled, so that CMVs are not sent for all MBs. In this case, the bit rate for side information will be less, and the error concealment will deteriorate as well, since some MBs will have to interpolate the neighboring CMVs instead of having their own CMV to use. We considered the use of all CMVs, as well as 5 different sub-sampling patterns for the CMVs, as shown in Figure 3.

4.1 Results of Concealment Motion Vectors

For the CMVs, we calculated the bit rate associated with each quantity of motion vectors by Huffman coding their horizontal and vertical components. We did not encode them differentially because for the sub-sampled CMV fields, the difference between motion vectors can be substantial. For comparison, the overheads associated with decision trees are converted to bit rate. A tree with N terminal nodes contains $N-1$ internal nodes; any node requires 1 bit to specify whether it is terminal or internal. Terminal nodes require 2 bits (for I pictures) or 3 bits (for P/B pictures) to specify the EC method, and internal nodes require 14 bits to specify the decision test (5 bits for the splitting variable, and 8 bits for the splitting threshold).

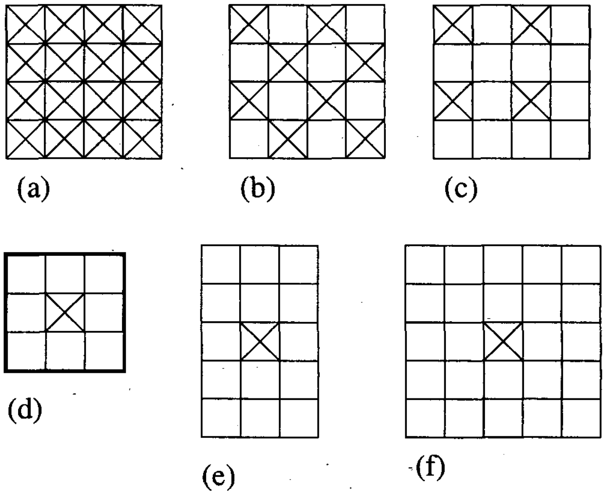


Figure 3: Sub-sampling patterns for concealment motion vector field: (a) all CMVs transmitted, (b) 1/2, (c) 1/4, (d) 1/9, (e) 1/15, (f) 1/25 of CMVs transmitted

Plots of distortion versus rate for 3 different sequences appear in Figure 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; and the bit rate is given as a percent overhead relative to the entire encoded sequence. The dashed horizontal line shows the omniscient minimum MSE, the solid line shows the behavior of the decision tree and the dotted line illustrates CMV performance.

For the flower garden I pictures, the CMVs outperform the decision tree, even at the very low sample rate of $1/25$. This is because the flower garden is basically a camera panning scene with very smooth and regular motion, where temporal concealment will give much better reconstruction than other methods, if the correct MV can be found. Because of the different distances from the camera, the (larger) top half and the (smaller) bottom half of the scene have different panning speeds, while within each portion, motions of different MBs are almost the same. For the decision tree, the only temporal method available in an I frame is panning, which uses the common MV from the larger (upper) half of the frame. This MV could be quite different from the MVs in the other half. On the other hand, even a very sparse sub-sampling of the MV, one in every 5 by 5 MB, gives much better estimation of the correct MV, which in turn gives a much better reconstruction than using the global panning MV (which is not correct everywhere) or other methods.

The susi sequence is a moving talking head scene. There are many global irregular motions, including rotation, zoom in and zoom out. Motion compensation is generally not very efficient; the spatial interpolation available to the decision tree usually performs better. In this case, the reconstruction guided by the decision tree outperforms the concealment MV.

The table tennis sequence is a mix of local and global motion, including zooming out and panning, and it also incorporate a scene change. Because of the characteristics of the different objects in the scene, spatial interpolation does as well as temporal concealment with correct MVs in some portion of the picture; but temporal methods with correct MVs perform much better than other methods in other parts of the picture. Because of the complicated motions, good estimation can be gotten only with a dense MV field; one in every 2 by 2 MB is barely enough. Thus, the decision tree beats sparsely subsampled concealment MV, but loses as the density of concealment MVs increases.

5 Conclusion

We have presented an improved decision tree which chooses among various concealment methods, consistently providing lower distortion than any of the fixed methods alone. Compared with our earlier work [2], the decision tree here uses many fewer input parameters, and is therefore less complex, but has some new and more useful parameters (including information about the brightness levels of neighboring MBs) and so it performs better. 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 small and adjustable level of overhead that depends on the tree size. Similarly, the use of a sub-sampled field of concealment motion vectors provides a flexible trade-off between the bit-rate for side information and the error concealment advantage under noisy conditions. The CMVs can provide better or worse results compared to the decision tree, depending of the characteristics of the video.

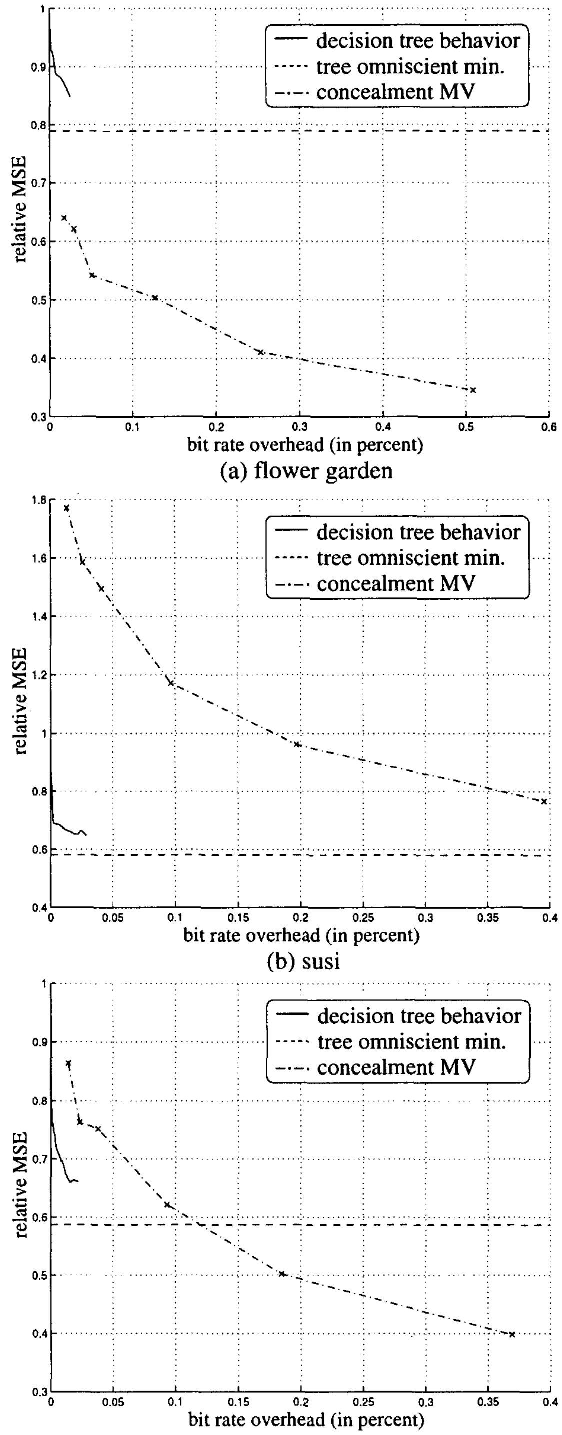
Acknowledgment: This work was supported by the National Science Foundation and by the Center for Wireless Communications at UCSD.

References

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

[2] S. Cen, P. Cosman, and F. Azadegan. Decision trees for error concealment in video decoding. 1999 IEEE Data Compression Conference, Snowbird, Utah, March 29-31, 1999, pp. 384-393.

[3] Y. Wang and Q.-F. Zhu. Error control and concealment for video communication: A review. *Proc. IEEE,* 86(5):975–775, May 1998.



(a) flower garden

(b) susi

(c) table tennis

Figure 4: MSE vs. bit rate for concealment motion vectors and for the CART decision trees for (a) flower garden, (b) susi, and (c) table tennis sequence.