Title

Error Concealment for Dual Frame Video Coding with Uneven Quality

Permalink

https://escholarship.org/uc/item/2jj8h8hp

Authors

Chellappa, V
Cosman, P C
Voelker, G M

Publication Date

2005-03-01

Peer reviewed

Error Concealment for Dual Frame Video Coding with Uneven Quality

Vijay Chellappa, Pamela C. Cosman and Geoffrey M. Voelker
University of California, San Diego, vchellap@ucsd.edu,pcosman@ucsd.edu

Abstract

When losses occur in a transmission of compressed video, the decoder can attempt to conceal the loss by using spatial or temporal methods to estimate the missing macroblocks. We consider a multi-frame error concealment approach which exploits the uneven quality in the two reference frames to provide good concealment candidates. A binary decision tree is used to decide among various error concealment choices. The uneven quality of the reference frames provides an advantage for error concealment.

1 Introduction

Video transmission over error-prone wireless channels requires some form of error handling mechanism. In standard video coders, errors caused by the communication channel can propagate across frames due to temporal prediction, which uses the previous reference frame for coding. There are many different types of error handling mechanisms, including forward error correction, retransmission, resynchronization codes, and error concealment. By error concealment (EC), we mean post-processing EC methods, those methods where the decoder, recognizing that an uncorrectable error has occurred, seeks to hide or minimize the glitch from the viewer so that a more visually pleasing rendition of the decoded video can be obtained.

In this paper, we propose a dual frame EC algorithm which chooses between a short-term (ST) reference frame (the previous frame) and a long-term (LT) reference frame (from the distant past). We use a binary decision tree algorithm called Classification and Regression Trees (CART) to make the concealment choice. This paper is organized as follows. Background on EC and on CART is provided in Section 2. We describe our methodology and results in Sections 3 and 4. We conclude in Section 5.

2 Background

In this section we describe previous work in error concealment, and provide background on CART decision trees and how they apply to the problem of EC.

2.1 Prior work on error concealment

In this paper, we are concerned with the set of post-processing methods that can be employed by the decoder. When errors strike the bitstream, we assume the decoder loses a slice up to the next resynchronization point. In the absence of block interleaving, this slice corresponds to missing a horizontal swath of macroblocks (MBs). The decoder's post-processing methods seek to conceal this loss from the viewer. Many post-processing EC methods have been proposed (see [1] for a review article). They can be divided into three main approaches: frequency, spatial, and temporal.

In *frequency concealment,* DCT coefficients of the missing blocks are estimated using either the corresponding DCT coefficient of neighboring blocks, or using the neighbor's DC values, or other neighborhood features. In *spatial concealment,* one interpolates directly in the spatial domain, using, for example, bilinear interpolation (if neighboring blocks on all 4 sides are available) or one-dimensional linear interpolation (if only MBs above and below are available), or directional interpolation (to preserve edges). 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, [2, 3, 4]). 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 neighboring received MBs. Many video decoders today conceal errors using the simplest possible temporal concealment approach: using the co-located MB in the previous frame to conceal a lost MB in the current frame.

Hybrid algorithms combine more than one of the frequency, spatial, and temporal approaches. For example, in temporal concealment, the referenced block can be further improved by spatial smoothing at its edges to make it conform to the neighbors, at the expense of additional complexity. Often, EC involves using a single fixed method for reconstructing any MB which is lost. However, a few *adaptive EC methods* have been proposed. In [5], 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 [6] and [7]. In [8], a CART decision tree is used to decide upon a particular EC approach among several EC algorithms.

Multiple-reference frame concealment: Multiple-reference frame coding has been included in the new H.264 standard. A small number of papers have studied how multiple reference frames might be used to improve EC. In [9], two reference frames are examined for candidate concealment blocks. Either boundary matching conditions are used to select one concealment block, or else candidate concealment blocks from different reference frames are averaged together to produce the final concealment block (called a *multihypothesis* block). In [10], candidate concealment blocks from different reference frames are adaptively weighted to minimize a boundary match criterion, or one candidate concealment block is held constant while another one varies until a cost converges. In [11], a lost block is first classified into foreground or background based on neighboring pixels. If it is in the background, then temporal replacement with the co-located block in the previous frame is used. If it is in the foreground, then candidate concealment blocks in each of the multiple reference frames are found and either selected or averaged.

2.2 Classification and Regression Trees

The CART^{TM} algorithm for designing classification and regression trees [12] is applied to error concealment as follows. Let x be a vector of measurements associated with a missing MB. For example, x includes information on whether the MBs above and below are better concealed by the ST or LT reference frame. Let C be a set of classes, where each class represents a possible EC method: $C = 1, 2, ...J$. The classifier assigns to every vector x a class from C.

A learning sample or training sequence $\mathcal{L}$ consists of data $(x_1, j_1), (x_2, j_2), \ldots, (x_N, j_N)$ on N cases where the class is known, that is, N MBs for which the best EC method is known. To form the training sequence, we can take each MB in the sequence, assume it is lost, reconstruct it using each of the EC methods, and see which one yields the smallest distortion. The measurement vector can include both ordinal and categorical variables. The root node of the 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. We measure the impurity of a set of data using the Gini index of diversity [12].

During the design of 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 ordered variable x_m, we include all splitting questions of the form "Is $x_m \leq c$?" 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 since the learning sample contains only N distinct points. For each single variable, we find the split which provides the greatest decrease in node impurity. We compare all of these, and find the best overall split of the data. A class assignment rule assigns a class $j \in \{1, 2, \ldots, J\}$ to every terminal node t. We use the plurality rule which assigns the most popular class for each terminal node. There are three standard ways of estimating the true misclassification rate of a classifier: cross-validation, test sample, and the resubstitution estimate. As discussed below, we used 10-fold cross-validation to determine the size of the final tree, but we used test samples to estimate the misclassification.

3 Methodology

Dual frame motion compensation is depicted in Figure 1, and works as follows. While encoding frame n, the encoder and decoder both maintain two reference frames in memory. The ST reference is frame $n - 1$. The LT reference can be selected in a number of ways; we used *jump updating* in which the LT reference frame varies from as recent as frame $n - 2$ to as old as frame $n - N - 1$. When encoding frame n, if the LT reference frame is $n - N - 1$, then, when the encoder moves on to encoding frame $n + 1$, the ST reference frame slides forward by one to frame n, and the LT reference frame jumps forward by N to frame $n - 1$. The LT reference frame then remains static for N frames, and then jumps forward again. We refer to N as the updating parameter. This approach was first adopted in [13] and was also used in [14, 15].

In this paper, as in [15], every Nth frame is coded with additional bit rate at the expense of other regular frames. This high-quality frame is then buffered and used as the LT reference frame for the subsequent N frames. The amount of extra quality to be given to the high quality frames is subject to further research. If there is too large a difference in quality, the end user will notice an annoying pulsing of the quality even though the overall average distortion of the sequence may be lower. In our work, the rate allocation was heuristic. Pulsing of quality was not perceptible.

In dual frame motion compensation, each macroblock can be encoded in one of three coding modes: intra coding, inter coding using the ST buffer (inter-ST-coding), and inter coding using the LT buffer (inter-LT-coding). In [14], the choice among these three was made using an optimal distortion estimation. In the current work, we choose between inter/intra coding using a similar method to ([16], p. 178). For the two inter coding modes, we choose the one with lower distortion.

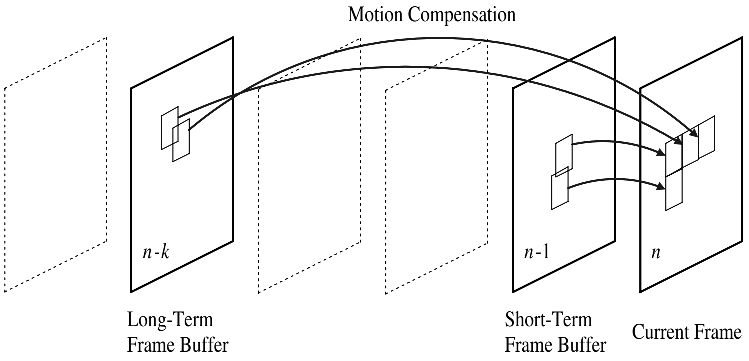

Figure 1: Dual Frame Buffer Motion Compensation.

We simulated the dual buffer coder by modifying a standard MPEG-4 encoder. The frame rate for all sequences is 30 frames per second. The bit rates for the sequences are 70-75 kbps.

CART predictor variables: To form the predictor variables for CART, we first examine whether the six neighboring MBs are motion compensated using the ST or LT frame. For each neighbor, a value of zero is assigned for ST compensation, and a value of one for LT compensation. The input variable OPTBITS is the sum of these 6 values. If the neighboring MBs are not lost, it is possible to calculate, for each of them, whether it would have been better concealed (had it been lost) using an ST MB (motion compensated or co-located) or using an LT MB. As above, we assign the value zero if the ST MB conceals better, otherwise the value is one. A CART variable called CONCEALBITS is the sum of these values. The distortion between each neighbor MB and its previous co-located MB is computed; the sum of these distortions is another CART input, which we call DIST0. Similar distortions are computed for the case of ST median MB, LT co-located MB, and LT median MB. The respective CART input parameters are DIST1, DIST2 and DIST3. So far, all the input variables depend only on information about the neighbors of the lost MB, and so the information is available at the decoder (assuming the neighbors themselves arrive intact). One last input variable, OMNIBIT, is different, in that it depends on the lost MB itself. OMNIBIT=1 if LT prediction was used by the current MB, or it is set to zero if ST prediction was used. For this particular variable to be used, we must assume that the information (single bit) corresponding to the reference frame choice is prioritized more highly than the rest of the data, and is received losslessly even when all other information about the MB is lost. This is not an unreasonable assumption, since it is less than 0.3% of the bit rate even at the low rates we used, and would be a much smaller percentage of the overall rate if higher overall rates were used. We have constructed the CART tree both with and without using this particular input variable.

Error Concealment modes: Our decoder is equipped with four possible temporal EC methods: it can conceal using either a co-located or motion compensated MB from either the ST or LT reference frame. The co-located MBs of the ST and LT frames are the MBs with MV=0. The median of MVs is computed using the MVs of the three MBs above and the three MBs below the missing one. The medians of the x and y components are computed separately. Of the six neighbors, only those that point to the same reference frame are used; e.g., if we are computing the ST median MV, then only those neighbors that have MVs pointing to the ST frame are used. If none of the neighbors points to the ST frame, then the ST median concealment is taken to be identical to the ST co-located concealment. The same rules apply to the LT median concealment.

The MBs on the edge of the frame do not have 6 neighbors; these MBs, if lost, are concealed by using the ST or LT co-located block. Where OMNIBIT is available, the reference block specified by it is used to conceal a lost MB on the image edge. Where OMNIBIT is not assumed to be losslessly preserved, the ST co-located MB is used.

4 Results

We carried out simulations for the Carphone, News, Container, Akiyo, Claire, Football, and Tennis sequences. The first four sequences are QCIF with 300 frames each. Football and Tennis are 352 × 240 with 125 and 112 frames, respectively. For each video sequence, we constructed a binary classification tree using all the other sequences as the training data, and reserved the sequence of interest as the test data.

For each of the six sequences to be used for training, we formed a training sequence by considering each slice in the sequence as being lost. We computed the 7-dimensional input vector (OPTBITS, CONCEALBITS, OMNIBIT, DIST0, DIST1, DIST2, and DIST3) for each MB in the lost slice. We concealed that MB using each of the 4 concealment methods, and determined which one produced the lowest mean squared error (MSE). That method was considered the class associated with that training vector. We used 10-fold cross-validation within the training set to determine the size of the final decision tree. We repeated the same procedure in constructing the tree without using the OMNIBIT.

To obtain the results in Table 1, each MB in the test sequence is individually assumed to be lost, and the concealment MB for it is found using the concealment method named at the top of the column. The average Mean Squared Error is reported, averaged over the entire sequence in each case. The first column is the name of the test sequence. The second column is the average concealment MSE when the ST median approach was used for EC for all MBs. The ST median was chosen because that was the single best EC method among the 4 approaches. The third column gives the value of the MSE when an omniscient decoder knows the exact concealment mode to use.

The next two columns give results for the CART decision tree concealment. The column labeled CART show the case where OMNIBIT was used, and the column labeled CART1 did not use OMNIBIT. So far, we can observe that the two variants of the decision tree both produce higher MSE than the omniscient decoder (as expected) but they also both produce lower MSE than the ST median concealment. CART constructed using OMNIBIT provides superior performance over ST median, ranging from as low as 1% to as high as 57%. Without using OMNIBIT, CART still gives a good performance improvement from no gain to 46% gain. When OMNIBIT is not used, the concealment is fully standard compliant, and the results are nearly as good as when CART uses OMNIBIT.

For comparison, we also simulated the multihypothesis EC algorithm proposed in [10] using optimal weight coefficients (which is shown in Table 1 as the opt weights column). We found that, in our pulsed quality dual reference frame scenario, the algorithm did not give good results. Upon further investigation, the basis block which is used in the algorithm is insufficient to judge the concealment candidates from the long-term frame. We found that there was almost always a bias towards choosing the ST frame candidate using this method.

The column labeled "With OMNIBIT" shows results for when the decision tree is not used, but rather the concealment choice between ST median and LT median is dictated directly by OMNIBIT (i.e., if the ST frame was used for motion compensation, then the ST median is used for concealment). We introduce another parameter OMNIBIT2, whose value is zero if ST median provides better concealment than LT median, and whose value is one otherwise. The last column gives the result when the concealment choice between ST median and LT median is dictated directly by OMNIBIT2, assumed transmitted losslessly. Using OMNIBIT or OMNIBIT2 directly to dictate the concealment choice in general performs less well than using the CART tree. Sending OMNIBIT2 improves performance marginally over just using OMNIBIT at the expense of added complexity and rate. Unlike OMNIBIT, which states which frame was used for motion compensation, and is therefore part of the compressed stream, OMNIBIT2 states which frame is better for median concealment, and therefore requires the encoder to compute the best concealment method for each MB, and to transmit an additional bit.

In one final variant (not shown), we also simulated the case when the LT frame is not a high-quality frame. We found that even here, CART performs better than the ST median of motion vector EC.

In Table 2, the first 4 columns present the fraction of time the omniscient decoder chooses each of the 4 concealment modes when operating in a pulsed quality mode. Most of the time, the omniscient decoder chooses between the ST median and the LT median. We observe that the LT frames are very useful for concealment: across the sequences, the omniscient decoder uses the LT median 22–45% of the time. Further, co-located blocks are used by the omniscient decoder only 7–15% of the time, so they see substantially less use than the median MV concealment.

In the last 4 columns of numbers, the table presents the fraction of time the omniscient decoder chooses each of the 4 concealment modes when operating in a regular quality mode. In the regular quality mode, every Nth frame is still used as an LT reference frame

Sequence	ST Med	Omni.	CART	CART1	opt. weights	With OMNIBIT	With OMNIBIT2
News	208	80	121	149	232	199	190
Container	132	35	59	71	51	96	93
Claire	46	22	41	43	62	35	32
Akiyo	70	22	30	38	67	45	45
Football	1412	842	1178	1218	1542	1274	1244
Tennis	602	338	516	549	738	578	567
Carphone	211	119	208	211	240	210	209

Table 1: Mean Squared Error for different concealment approaches

in a dual frame coder, but the Nth frame is not allocated more than its share of bit rate, so it has the same quality, on the average, as any other frame. We see that the omniscient concealer chooses the LT frame for concealment substantially less often when the LT frame is just a regular frame. In earlier work [14, 15], we concluded that pulsing the quality of frames for use as long term reference frames can have a beneficial effect on lowering the overall average distortion of the sequence. Table 2 shows that pulsing the quality of long term reference frames can also provide an advantage for error concealment.

	Pulsed Quality Coder				Regular Quality Coder			
Sequence	ST med	LT med	LT CL	ST CL	ST med	LT med	LT CL	ST CL
News	54	39	2.5	4.5	69.6	21.2	2.9	6.3
Container	55.7	33.6	3.9	6.7	80.1	8	4	7.9
Claire	60	26.58	5.25	8.1	75.8	12.3	3.1	8.8
Akiyo	50.4	45.4	1.87	2.22	56.2	36.7	4.1	3
Football	65	22.5	1.3	11.2	76.9	11.1	0.8	11.2
Tennis	57.4	30.5	3.7	8.4	69	19.3	2.1	9.6
Carphone	60.5	25.1	4.48	9.9	67.3	14.8	5.4	12.5

Table 2: Percentage of time that each concealment mode (ST median, ST co-located (CL), LT median, and LT co-located (CL)) provides the best concealment.

In our scenarios so far, the CART tree is not transmitted to the decoder since the learning data used to construct the tree is independent of the transmitted data. Hence a stand-alone decoder implementation with a preloaded tree is feasible. Another option is to customize the CART tree for each video sequence. This can potentially improve performance at the cost of needing to send the CART tree as side information. To explore this tradeoff, we used each test sequence as its own training sequence to construct the CART trees. Table 3 shows the video quality results using this approach. The columns labeled “ST Med” and “CART” are the same as in Table 1. Under “Customized CART tree”, the column labeled MSE gives the MSE of the customized CART tree, and the column labeled “% imprv.” shows the percentage improvement that the customized tree gives over the tree created

Sequence	ST Med	CART	Customized CART tree			Omniscient concealment		
	MSE	MSE	MSE	% imprv.	Bits	MSE	bits	% bits
News	208	121	104	14	666	80	43659	1.67
Container	132	59	54	8	510	35	49599	2.1
Claire	46	41	26	37	120	22	51885	3.5
Akiyo	70	30	30	0	751	22	36884	2.1
Football	1412	1178	1170	-1	1686	842	76888	0.8
Tennis	602	516	509	1	970	338	73260	1.1
Carphone	211	208	176	15	530	119	57113	2.1

Table 3: Customized CART tree MSE shows the MSE for a CART tree constructed from transmitted sequence. The percentage improvement over the generic CART tree is also shown, as well as the number of bits required for transmitting the customized CART tree. Under omniscient concealment, the MSE, number of bits, and percentage of bits are shown for achieving omniscient concealment.

from a disjoint training set. The improvement from customizing the tree is generally small.

The cost of using a customized CART tree is that the tree must be transmitted in addition to the data. The next column in Table 3 shows the number of bits required for the customized CART tree for each sequence. The number of bits required for each transmitted tree is negligibly small and would not impose much overhead on transmission, but the gain from using a customized tree over a generic tree is also small, so customizing a concealment decision tree does not appear to be useful. We note that, for real-time transmission, sending a customized tree of this type would in any case not be feasible, since hundreds of frames are not available in advance.

Rather than sending the CART tree as side information, we could instead send information to the decoder informing it specifically which of the four EC methods to use. This approach would provide the video quality of the omniscient decoder. However, it would also require extra overhead. The last two columns of Table 3 show the number and percentage of bits (using Huffman coding) required to inform the decoder of the best EC method. Across the sequences, this approach would require 1–3.5% of the transmission bandwidth to transmit this side information.

Finally, we simulate the CART EC algorithm using actual packet loss ratios of 5% and 20%. Recall that, in the experiments above, we averaged across all MBs assuming slices of each frame are lost in turn. Since each slice is individually lost and concealed, the results above do not have any case where the concealed block also has its neighbors lost, or where the neighbor is motion compensated using a block that was concealed.

In simulating with actual packet losses, there can be adjacent lost slices. When concealing a lost MB, the decoder avoids using a replacement block that was itself based on a lost block. For example, if the ST replacement MB is corrupted, we use the LT median MB as the candidate. Table 4 shows the results of this experiment. CART provides better performance than ST median of MV for both loss rates. The difference, though, is smaller for lower loss rates. For the CART tree with 20% loss, not using the OMNIBIT slightly degrades performance 0.1–0.2 db. For 5% loss, the CART results with and without OMNIBIT show negligible difference.

Sequence	ST Med 5% loss	ST Med 20% loss	CART with OMNIBIT 20% loss	CART with OMNIBIT 5% loss	CART without OMNIBIT 20% loss
News	29.8	28.7	29.9	29.4	29.2
Container	30.65	29.5	30.6	29.9	29.82
Akiyo	35.45	35	35.52	35.1	35.1
Claire	36.1	35	36.07	35.58	35.46
Football	23.7	22	23.86	22.87	22.71
Tennis	26.02	24.7	26.08	25.1	24.96
Carphone	28.7	27.7	28.8	27.9	27.9

Table 4: PSNR values for 5% and 20% packet loss ratios. The first two columns show the performance for ST median MV error concealment, for the two different loss rates. The next two columns show the performance for CART using OMNIBIT, for the two different loss rates. The last column is for CART concealment without using OMNIBIT.

5 Conclusions

We have presented a method for concealing lost MBs in a video stream when a dual frame buffer codec with uneven quality is used to code the stream. The CART-based concealment method performed better than just concealing using the short-term median MV block. The CART tree allows the decoder to choose between using the long-term and short-term frame for concealment, and also to choose between using the co-located and median motion compensated blocks for concealment.

One significant conclusion is that the decision tree approach gives a performance advantage over not using it, even when the tree is not customized for the particular sequence, but rather is constructed from unrelated sequences. This is in contrast to the work reported in [8] where a CART decision tree (for a single frame codec) was used to choose among spatial, temporal, and frequency concealment, and it was found that performance gains could be had only by a customized tree. Because it required a customized tree, that system required side information and could not be standard compliant. Unlike that work, our approach requires no side information, and (when used without OMNIBIT) is fully standard compliant.

A second significant conclusion is that using a pulsed quality coder gives an advantage for error concealment. If the pulsed quality is too uneven, the effect may be irritating to the viewer. When used with restraint, it was shown in [14, 15] that the pulsed quality provides an advantage in reducing the overall distortion of the dual frame coded sequence, while the pulsing itself is not perceptible. Since the high quality LT frame is attractive also for the concealer, pulsing the quality provides a concealment advantage to the decoder as well, which is equipped to take advantage of the high quality reference frames.

References

[1] Y. Wang and Q. F. Zhu, "Error control and concealment for video communications: A review," *Proc. IEEE,* 86(5):, May 1998.

[2] M. Ghanbari and V. Seferides, "Cell-loss concealment in ATM video codecs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, June 1993, pp. 238-247.

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

[4] S. Aign, "Error Concealment for MPEG-2 Video,", *Signal Recovery Techniques for Image and Video Compression and Transmission, Kluwer Academic Publishers,* 1998, pp. 235-268.

[5] Q.-F. Zhu, Y. Wang, and L. Shaw, "Coding and Cell-Loss Recovery in DCT-based Packet Video," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 3, no. 3, June 1993, pp. 248-258.

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

[7] 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," *Proc. 1997 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing,* vol. 2, Aug. 1997, pp. 912-915.

[8] S. Cen and P. Cosman, "Decision Trees for Error Concealment in Video Decoding," *IEEE Trans. on Multimedia*, vol 5, issue 1, Mar. 2003, pp 1-7.

[9] M.E. Al-Mualla, C.N. Canagarajah and D.R. Bull, "Multiple-reference Temporal Error Concealment," *Proc. ISCAS*, vol. 5, 2001, pp. 149-152.

[10] Y.O. Park, C.-S. Kim and S.-U. Lee, "Multi-hypothesis Error Concealment Algorithm for H.26L Video," *International Conference on Image Processing*, 2003, pp. 465-468.

[11] B. Jung, B. Jeon, M.-D. Kim, B. Suh, and S.-I. Choi, "Selective temporal error concealment algorithm for H.264/AVC," in *Proc. IEEE International Conference on Image Processing*, Oct. 2004.

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

[13] T. Fukuhara, K. Asai and T. Murakamai, "Very Low Bit-Rate Video Coding with Block Partitioning and Adaptive Selection of Two Time-Differential Frame Memories," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 212-220, Feb. 1997.

[14] A. Leontaris and P. Cosman, "Video compression with intra/inter mode switching and a dual frame buffer," *IEEE Data Compression Conference 2003*, pp. 63-72, 2003.

[15] V. Chellappa, P. Cosman and G. Voelker, "Dual Frame Motion Compensation with uneven quality assignment," *IEEE Data Compression Conference 2004.*

[16] K.R. Rao and J.J. Hwang, "Techniques and Standards for Image, Video and Audio Coding," *Prentice Hall.*