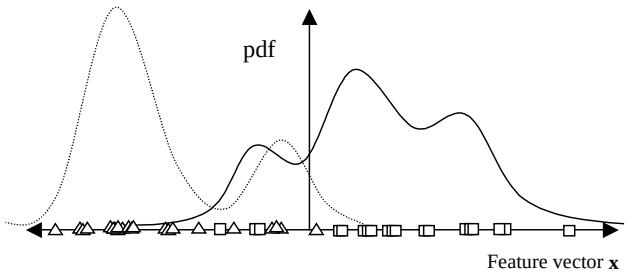
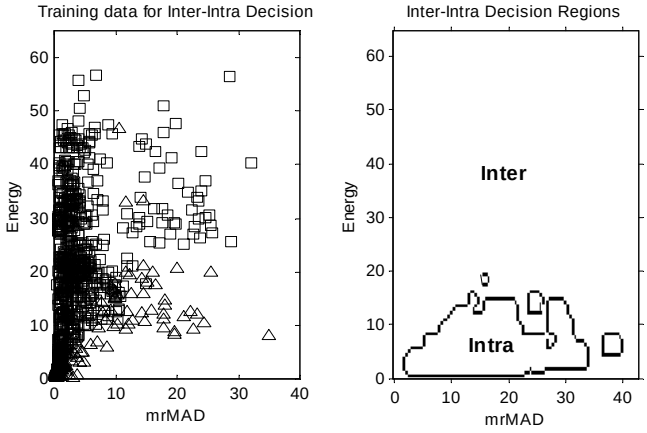
Carnegie Mellon

Electrical & Computer ENGINEERING

Classification Based Mode Decisions for Video over Networks

Deepak S. Turaga and Tsuhan Chen
Advanced Multimedia Processing Lab





Technical Report AMP 00-01
December 2000

Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

A video encoder has to make many mode decisions in order to achieve the goal of a low bit rate, high quality, and fast implementation. We propose a general classification based approach to making such mode decisions accurately and efficiently. We first illustrate the approach using the Intra-Inter coding mode decision. We then focus on the decision to skip or code a frame for rate control of video over networks. Using the classification based approach we show improvement in the rate-distortion sense. We then extend the work to scalable video coding in choosing between scalability modes and examine the performance of our approach over error prone networks, using simulated packet losses.

I. Introduction

A video encoder converts raw video data into a standard specified bitstream, which is then transmitted over the network and reconstructed to video by the decoder. This conversion from video to bitstream is done to achieve compression without sacrificing on the quality of the reconstructed video. Real time video encoding requires fast implementation of the coding process. Hence video encoders have to perform well in terms of the speed-quality-bit rate tradeoff. Inherent in the coding process are many mode decisions that improve one or more of the parameters in this speed-quality-bit rate tradeoff. Video standards such as MPEG [1] and H.263 [2] specify the bitstream syntax completely, but allow for some optimizations in the encoding process in terms of algorithms and mode decisions used. These optimizations are allowed as encoders designed for different applications need to be optimized differently. For instance when the application requires real-time encoding, the speed of the coding becomes critical, while in applications such as off-line coding, the quality and the coding efficiency are more critical than the speed.

In this paper we show that many mode decisions in the coding process can be considered as binary hypothesis testing problems that are well understood in traditional classification theory. The goal of a mode decision is to minimize a cost that may be defined in terms of one or more of the parameters of the speed-quality-bit rate tradeoff. The optimal mode decision is typically data dependent. In theory, for each mode decision, we can try all the possible modes, evaluate the cost corresponding to each mode, and choose the one with the smallest cost. However, such an

---

Contact Author: Prof. Tsuhan Chen, Electrical and Computer Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213. Email: tsuhan@cmu.edu

Accepted for publication in *IEEE Transactions Multimedia*, Special Issue on Multimedia over IP, March 2001.

exhaustive search approach is impractical due to its complexity. An alternative is to identify features that can be easily computed from the video data, and are good indicators of which mode would be optimal. In order to do so, we first collect video data and use exhaustive search to "label" the data with the optimal mode decisions. We then estimate probability density functions for these features under different hypotheses, corresponding to the different options in the mode decision. We then transform this minimization of cost to a more traditional error probability minimization problem and use standard classification techniques like the likelihood ratio test to solve it. We evaluate the performance of this classification approach using two mode decisions that are part of the video coding process. We first use the Inter-Intra mode decision as an example and then focus on the mode decision for skipping or coding a frame for rate control purposes.

As an illustration of our approach we use the Intra-Inter mode decision. We evaluate the performance of the decision proposed in the Test Model Near Term (TMN 10) [3]. Using a new feature, the mean removed mean absolute difference (mrMAD) and our classification based approach we improve this decision. More discussion about the mrMAD can be found in [4].

The focus of this paper is on the mode decision to skip or code a frame for rate control of video that needs to be transmitted over the network. The problem of rate control has been extensively studied in literature. Rate control is essential when we need to constrain the output of a variable bit rate (VBR) encoder to the bit rate provided by the network without sacrificing too much on the quality of the decoded video. This target bit rate that we have to achieve may be constant or variable depending on the nature of the network. Lakshman et. al. [5] classify VBR video into unconstrained, shaped, constrained and feedback depending on the design of buffers for rate control, or algorithms used by the encoder to change the coding parameters using feedback information from the network. Chen and Wong [6] and Choi and Park [7] try to solve the rate control problem using buffer control strategies. As against this approach, Hsu, Ortega and Reibman [8] propose algorithms for rate control using both source rate control as well as channel rate control over asynchronous transfer mode (ATM) networks. Most of the work in these papers tries to control the rate by changing the spatial quality using the quantization step size and consider changing the frame rate for rate control as the last option, when we cannot achieve the target rate. Song, Kim and Kuo [9] propose rate control algorithms for low bit rate unconstrained VBR that allow for a variable target rate provided by the network. The algorithms they propose try to control the spatial and temporal quality simultaneously to improve the perceived quality. They use frame skipping or frame rate changes in order to improve the perceived quality and the primary algorithm to control the bit rate is through the use of the quantization step size. Work by Martins, Ding and Feig [10] tries to achieve rate control by combining the effect of coding and

skipping a frame into a composite cost function. They, however lack granularity in their work, as the decision they make is not on a frame by frame basis. They make a decision for the future using information from coding the current frame. As an example, if after coding a frame they feel that they need to skip 10 frames, they do so without examining any of the following 10 frames. Most of the work above uses models to estimate the relationship between bits needed to code frames and the quantization step size. This is useful to estimate the quantization step size needed to code the frame given a target rate. Some of these models are proposed by Chiang and Zhang [11] and by Ding and Liu [12].

Most of existing rate control algorithms modify the quantization step size to control the rate of VBR encoders. Frame skipping to control the bit rate is used only as a last resort, i.e., when a buffer overflow occurs or when the buffer approaches overflow. Such algorithms do not consider the loss of quality that occurs by this arbitrary skipping of frames. This problem is more severe at low bit rates when it is difficult to achieve the target rate by changing the quantization step size alone and more frames need to be skipped. We propose a scheme for rate control that decides intelligently between changing the quantization step size and skipping the frame to achieve the desired target rate. We look at the suitability of skipping or coding a certain frame in terms of the impact it has on the rate as well as the impact it has on the quality before making a decision. We try to maximize perceived quality while achieving a fixed target rate. We implement classification based strategies to decide between coding and skipping a frame.

We extend our work on rate control to scalable video coding in order to evaluate the performance of our scheme under lossy network conditions. We create an enhancement layer corresponding to the base layer generated using our rate control strategies and examine the reconstructed video quality over different simulated packet loss scenarios. Effectively, this is equivalent to choosing between SNR scalability and temporal scalability adaptively, depending on the video data.

The paper is organized as follows. Section II includes the discussion of the general classification based approach to making mode decisions using the Intra-Inter mode decision as an example. Section III contains the use of this approach towards the rate control problem using an instantaneous mode decision. Section IV includes the description of the mode decision allowing for looking ahead at one future frame to be coded. We then extend this work on rate control to scalable video coding and this is described in Section V. We then conclude with the analysis of our scheme and describe some future work and possible extensions.

II. Classification Based Approach to Mode Decision

In this section we show how to convert a mode decision into a binary hypothesis-testing problem. Let c_o represent the cost for making a decision D_0 and c_1 represent the cost for making decision D_1 for a particular coding unit. These decisions D_0 and D_1 may include mode decisions that may be at different levels of the coding process, i.e., at different coding units. For instance these could be at the macroblock level, with one macroblock being the coding unit, where D_0 could be to decide to code the macroblock using intra coding while D_1 could be to code the macroblock using inter coding. These could also be at the frame level, with one frame being the coding unit, where D_0 and D_1 represent whether to code or skip the frame. The costs c_i could include the number of bits needed to code a block or frame given that we have made a particular decision. The cost could also include the distortion introduced in the decoded video as well as the time needed to encode the video according to the decision. The goal of our strategy is to make the decision that minimizes this cost. So every time we need to make a decision we want to make one that results in the smaller cost. This strategy may be summarized as below.

$$\begin{cases} \text{Choose } D_0 \text{ if } c_0 < c_1 \text{ or } c_0 - c_1 < 0 \\ \text{Choose } D_1 \text{ if } c_1 < c_0 \text{ or } c_0 - c_1 > 0 \end{cases}$$

A. Transforming Mode Decision into Classification Problem

In principle, to make the optimal mode decision one can try all the modes and choose the mode that has the lowest cost. However, computing the actual costs c_i before making a decision is very computationally intensive as this involves trying either decision to determine the cost. In order to reduce computational burden for the decision scheme we would like to identify features that provide a good estimate of the cost for a decision, but do not require as much computation to evaluate. We would, thus like to identify features that let us estimate which of the two following hypotheses H_0 or H_1 is true where

$$H_0 : c_0 - c_1 < 0$$
$$H_1 : c_0 - c_1 > 0$$

For each coding unit we identify *K* features that we group together in a feature vector $\mathbf{x} = [x_0, x_1, \ldots x_{K-1}]^T$. In the optimal scenario we could find features that perfectly represent the cost needed for a decision. However, in most practical applications such features are difficult to find. In most practical applications, the decision strategy thus becomes sub-optimal in terms of

minimizing the cost, however we are willing to settle for this sub-optimality as along with this comes the benefit of small computational requirements.

We want to build a classifier that would take as input the feature vector $\mathbf{x}$ of each coding unit and come up with the probability that H_0 or H_1 is true, which would then enable us to make a decision appropriately. We can come up with such a classifier by training it with sample data. Suppose we collect a set of M coding units with corresponding feature vectors $\mathbf{x}_0, \mathbf{x}_1 \ldots \mathbf{x}_{M-1}$ and associated with each of these feature vectors is a cost difference $d_i = c_{0i} - c_{1i}$ with c_{0i} and c_{1i} being the true costs for decisions D_0 and D_1 respectively for the *i-th* coding unit. Let P of these coding units have $d_i < 0$, i.e., corresponding to when H_0 is true and Q of these coding units have $d_i > 0$, corresponding to when H_1 is true, with $P + Q = M$. For purposes of illustration, we show these two sets of training vectors for a *one-dimensional* feature vector in Figure 1.

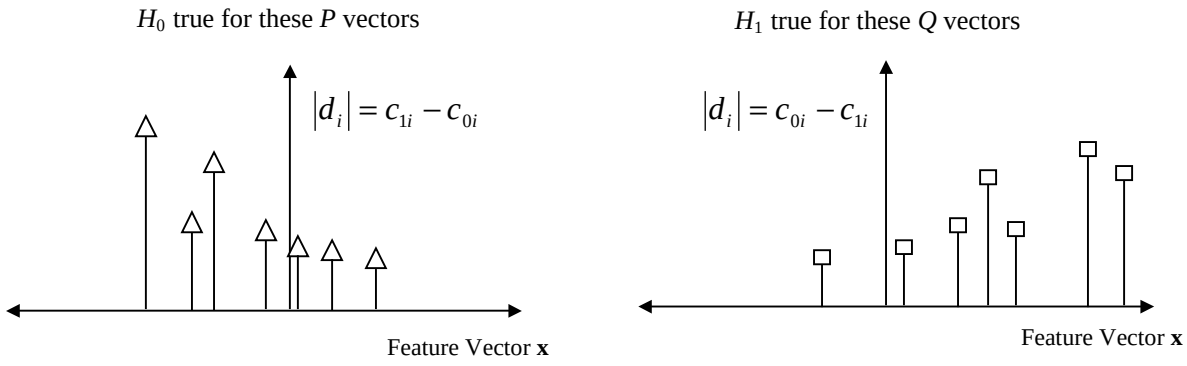


Figure 1. Illustration of feature vectors

In Figure 1, the x-axis for both the cases corresponds to the value of the feature vector $\mathbf{x}$ (we are using a 1-D vector for illustration) for each coding unit while the y-axis corresponds to the magnitude of the cost difference $|d_i|$ between the two mode decisions for that coding unit. Feature Vectors of coding units for which H_0 is true are shown on the left and are represented with triangles while feature vectors of coding units for which H_1 is true are shown on the right and represented with squares.

The magnitude difference $|d_i|$, for any coding unit, corresponds to the additional cost that we need to pay if we make the wrong mode decision for that unit. For instance if we choose

to make a decision D_1, instead of the right decision D_0, for one of the coding units represented by triangles, we incur a cost c_{1i} instead of incurring the smaller cost c_{0i} and so pay an additional cost $|d_i| = c_{1i} - c_{0i}$.

We may put together the two plots from Figure 1 to obtain the entire training feature space and this is shown in Figure 2.

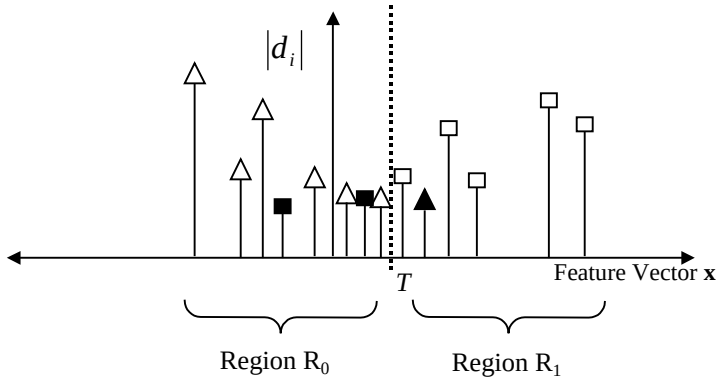


Figure 2. Training feature space

All the feature vectors corresponding to the M coding units are shown on the same plot and we use triangles and squares as before to separate the data into the two classes. We now want to partition this feature space using our classifier so that the total additional cost that we have to pay for misclassification, i.e., making the wrong decision for any coding unit, is as small as possible. For instance we may partition this space into two regions, R_0 and R_1, using the threshold T as shown in Figure 2. We choose to make decision D_0 for all coding units with feature vectors to the left of the threshold, i.e., in R_0, and decision D_1 for all coding units with feature vectors to the right of the threshold, i.e., in R_1. For such a case we pay the additional cost $|d_i|$ for each of the misclassified coding units, i.e., squares in R_0 and triangles in R_1. In Figure 2 we show these as dark triangles and squares, as opposed to the lightly colored ones, for which we make the right decision.

We would like our mode decisions to result in as small a total cost as possible. In order to do this we need to choose R_0 and R_1 so that we minimize the total additional cost from making wrong mode decisions. This kind of a problem of partitioning the feature space in order to minimize the total cost is reminiscent of traditional classification theory, except that in traditional classification theory the vertical axis corresponds to the probability densities of the feature vectors under the different hypotheses, while in our problem this corresponds to the additional

cost we have to pay when we make a wrong decision. Hence, in order to use traditional classification techniques to solve our problem, we need to modify our problem to fit the classification scenario.

Our problem of minimizing the total additional cost may be mathematically written as follows.

$$\min_{R_0,R_1}\left[\sum_{\substack{\mathbf{x}_i\in R_0\\ d_i>0}}|d_i|+\sum_{\substack{\mathbf{x}_i\in R_1\\ d_i<0}}|d_i|\right]$$

These two regions, R_0 and R_1 should together span the entire space, and have no common sub-regions. However, we impose no constraints on the shapes of these regions as long as they satisfy this minimization requirement. These regions may consist of non-contiguous sub-regions and the boundaries between them may be arbitrarily shaped. Hence we formulate the problem of minimization in terms of choosing the regions and not in terms of specifying linear boundaries or thresholds separating them.

Let $N_0=\sum_{d_i<0}|d_i|$ and $N_1=\sum_{d_i>0}|d_i|$. Our minimization problem may be equivalently written as follows.

$$\min_{R_0,R_1}\left[\frac{N_1}{N_0+N_1}\sum_{\substack{\mathbf{x}_i\in R_0\\ d_i>0}}\frac{|d_i|}{N_1}+\frac{N_0}{N_0+N_1}\sum_{\substack{\mathbf{x}_i\in R_1\\ d_i<0}}\frac{|d_i|}{N_0}\right]$$

As mentioned before, we would like to use standard classification techniques to solve the problem and identify these regions R_0 and R_1. Such techniques involve estimating the probability density function (pdf) of the feature vector under the different hypotheses and then using these pdfs to identify the best regions. However, the feature vectors in our problem are different from the feature vectors typically used in classification problems. This is because they not only have a certain value, but also a height (corresponding to the additional cost) associated with them. In the typical classification scenario the vectors do not have this additional associated height. Essentially, we need to convert these feature vectors with the associated heights to vectors that do not have these additional heights, but we need to do this without losing the important information that the heights carry. We may do this transformation by replacing a vector with height $|d_i|$ with $|d_i|$ vectors at that location. In general it is not necessary that all the heights $|d_i|$ are integers, however without loss of generality we can scale them appropriately to make them

integers. We can thus modify our feature space and our modified feature space would look as shown in Figure 3.

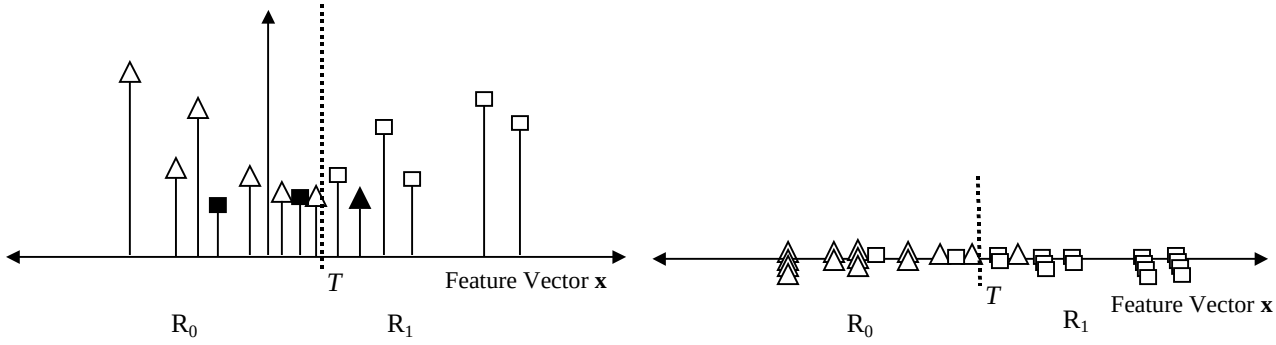


Figure 3. Transformation of feature space from old feature space (left) to new feature space (right)

From Figure 3 we can see that each vector in the old feature space is replaced by multiple vectors at that location, the number of new vectors being equal to the height associated with the original vector. Now, standard classification techniques may be applied in this new feature space to estimate the pdfs for this new set of vectors. We rewrite our minimization problem in this new feature space and then map it to the well-understood minimum probability of error classification problem. We can thus use results from literature to find the regions that minimize our criterion in this new feature space. Because of the way we transform the old feature space to the new feature space, the regions we find in this new feature space are identical to the regions we desire in the original feature space. This is because none of the training data points are displaced from their original positions. The details of this proposed scheme are presented in the following paragraphs.

In this new feature space N_0 is simply the total number of triangles and N_1 is the total number of squares, where N_0 and N_1 are as defined before. We may define two new hypotheses as following.

$$H_0' : \mathbf{x}_i \text{ belongs to the triangle class}$$
$$H_1' : \mathbf{x}_i \text{ belongs to the square class}$$

Hence, in this new feature space $\frac{N_0}{N_0 + N_1} = P(H_0')$, the probability of a feature vector being a triangle and similarly $\frac{N_1}{N_0 + N_1} = P(H_1')$.

Also $\frac{|d_i|}{N_0} = P(\mathbf{x} = \mathbf{x}_i \mid H_0')$ when $d_i < 0$, because $|d_i|$ is the number of triangles at $\mathbf{x}_i$ and N_0 is the total number of triangles. Similarly, $\frac{|d_i|}{N_1} = P(\mathbf{x} = \mathbf{x}_i \mid H_1')$ when $d_i > 0$. Hence our minimization problem may be rewritten in this new domain as follows

$$\min_{R_0, R_1} \left[P(H_1') \sum_{\mathbf{x}_i \in R_0} P(\mathbf{x} = \mathbf{x}_i \mid H_1') + P(H_0') \sum_{\mathbf{x}_i \in R_1} P(\mathbf{x} = \mathbf{x}_i \mid H_0') \right]$$

In practice, instead of using these discrete probabilities, we model the data using a continuous pdf consisting of a mixture of Gaussians. This is because in order to classify a feature vector we need to have the probability of occurrence of that vector. However we do not have these probabilities for new input vectors that are not present in the training data set. By modeling the feature vector pdf using a mixture of Gaussians we ensure that any feature vector may be classified. These Gaussian mixtures are trained on the modified feature vectors using the Expectation Maximization (EM) algorithm, more details on which may be obtained from [14]. An example of trained Gaussian mixtures in the modified feature space is shown in Figure 4.

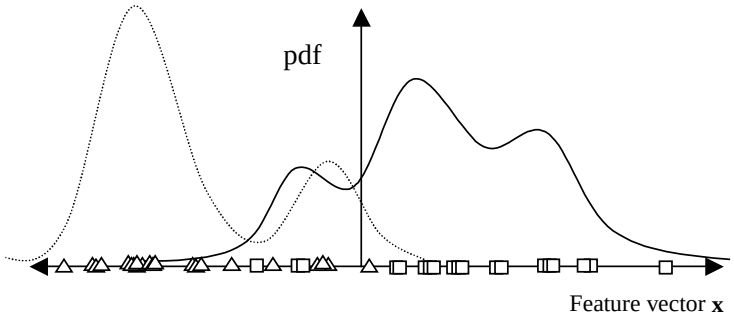


Figure 4. Modeling the data by a mixture of Gaussians

The density function drawn with the dotted line corresponds to the data from the triangle class while the density function drawn with the solid line corresponds to the data from the square class.

Using our continuous pdfs we may rewrite our minimization problem as follows.

$$\min_{R_0, R_1} \left[P(H_1') \int_{\mathbf{x} \in R_0} p(\mathbf{x} \mid H_1') \, \mathbf{dx} + P(H_0') \int_{\mathbf{x} \in R_1} p(\mathbf{x} \mid H_0') \, \mathbf{dx} \right]$$

The functions $p(\cdot)$ correspond to the continuous pdf comprising of a mixture of Gaussians. This kind of a minimization problem is equivalent to a minimum probability of error classification scheme in this new feature space and it is well known from literature [13] that the

regions may be determined using the likelihood ratio test. The likelihood ratio test may be written as

$$\frac{p(\mathbf{x} \mid H_1')}{p(\mathbf{x} \mid H_0')} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P(H_0')}{P(H_1')}$$

Hence, in order to classify a feature vector $\mathbf{x}$, we look at the likelihood ratio for it and if this exceeds the threshold obtained from training, we believe H_1 is true and make decision D_1 otherwise we believe H_0 is true and make decision D_0. In summary, this likelihood ration test defines the decision regions in this new space. As we mentioned before, the regions that we determine in this new feature space are identically the regions that we desire in our original feature space. So by transforming our feature space and mapping the problem to a well-understood minimization problem we can obtain the solution we desire.

The entire classification scheme may be summarized as follows. Given the training data and the cost differences, we first transform the feature space to the new feature space and then estimate the apriori probabilities, $P(H_0')$ and $P(H_1')$, as well as the class conditional probability density functions, $p(\mathbf{x} \mid H_0')$ and $p(\mathbf{x} \mid H_1')$, using the EM algorithm to train the Gaussian mixture. Once we have these pdfs, we use the likelihood ratio test for a new input feature vector corresponding to a coding unit and determine which of the two hypotheses is more likely to be true and using this result we make decision D_0 or D_1 for that coding unit.

B. An Example: Intra versus Inter Mode Decision

This decision is made for every macroblock (a 16×16 region in a frame) in the video sequence. Intra coding involves coding using transform coding followed by quantization and entropy coding. As opposed to this, Inter coding involves building a prediction for the current macroblock using data from the previous frame using motion estimation and compensation and coding the residue using transform coding, quantization and entropy coding. For most macroblocks, Inter coding is more efficient in terms of compression, however this is not always true. When there is a scene change or when we have a high motion sequence the prediction for the macroblock from the previous frame is likely to be poor and in such cases it may be more efficient to use Intra coding as opposed to Inter coding. Hence the encoder has to decide between these for every macroblock. The decision that requires fewer bits is preferred.

As converting to bits is computationally expensive, encoders use features as estimates for the bits. Typically the energy (with DC value removed) in the block is used as an estimate of the bits needed for intra coding, while the MAD is used as an estimate for the bits needed for inter

coding. For example, the mode decision as recommended by the TMN-10 of the H.263 standard is

$$\begin{cases} Intra\ Coding\ if\ MAD - E_x > T \\ Inter\ Coding\ \ otherwise \end{cases}$$

For a 16×16 block $E_x = \frac{1}{256}\sum_{i=1}^{16}\sum_{j=1}^{16}\left|x_{i,j} - m_x\right|$ is the energy, m_x is the mean or the DC value of the block and *T* is an empirically found threshold, specified in the TMN as 500.

We also test another feature, the mean removed MAD (mrMAD) as a feature to estimate the bits needed for inter coding. This feature is very similar to the MAD, except that instead of taking the absolute pixel difference and summing them up we first remove the means from the blocks and then sum up the absolute pixel difference.

In order to collect training data we perform the exhaustive test for a variety of sequences and generate a sequence of values for the features and a sequence of the optimal decisions. These exhaustive tests involve actually computing bits needed for Intra and Inter coding and making the right decision, i.e. the decision resulting in fewer bits. After collecting the sequence of right decisions and the values of the features we correlate these with the decision sequence. From experiments we see that the energy is very well correlated with the bits needed for intra coding (correlation coefficient of 0.87). The MAD and the mrMAD are features representative of the bits needed for Inter coding. They have correlation coefficients of 0.90 and 0.94 respectively with bits for inter coding, independent of sequence. In order to test the suitability of using these features, we correlate these with the optimal decision sequence (one determined using the exhaustive test). The decision sequence is viewed as a sequence of +1s and −1s with +1 corresponding to Intra and −1 corresponding to Inter. More details on our computation of correlation coefficients are included in the Appendix. The correlation coefficients for each feature are presented in Table 1.

Table 1. Correlation coefficients of features with decision sequence

Feature	Correlation coefficient with decision sequence
Energy	0.3221
MAD	0.4741
mrMAD	0.5111

From the table we see that the mrMAD is better correlated with the decision sequence than the MAD, hence we choose this as a feature. Although the MAD has a higher correlation with the decision sequence than the energy, it is highly correlated with the mrMAD. As the energy is representative of the bits needed for intra coding, we use this feature, instead of the MAD, along with the mrMAD for our classification scheme.

We collect feature vectors from the Foreman, Coastguard and Silent sequences. Snapshots from these sequences are shown in Figure 5.

Figure 5. Snapshots from Foreman (left) Coastguard (center) and Silent (right)

The sequences are in 176×144 (QCIF) format at a frame rate of 30 Hz. We collected 79200 feature vectors of which we used 5000 to train our classifier pdfs. The number of training vectors is small as compared to the test set, but using a larger training set does not improve the performance of our classifier significantly. Using the training data, we train the Gaussian mixtures using to obtain decision regions as shown in Figure 6.

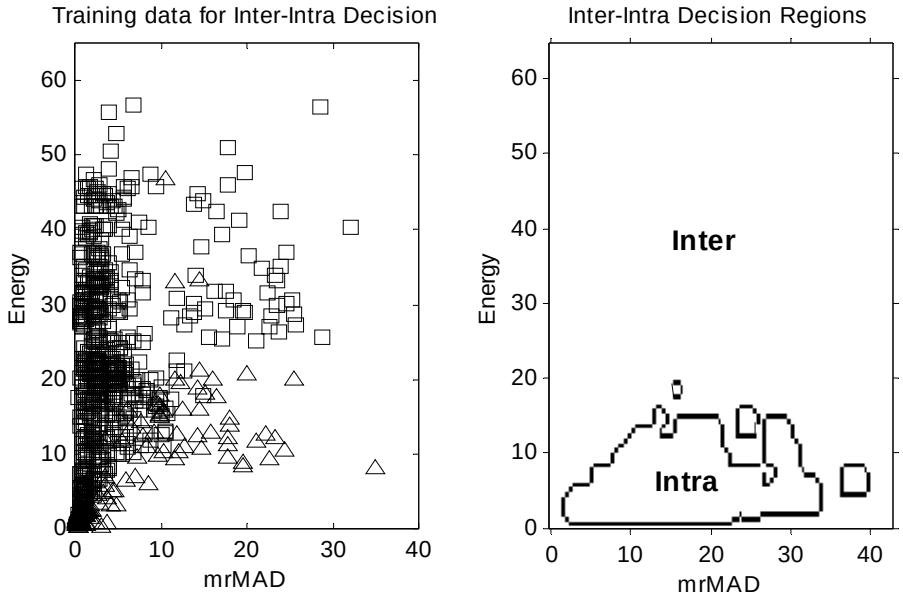


Figure 6. Inter-Intra training data (left) and decision regions (right)

The plot in Figure 6 shows 1000 training feature vectors on the left with triangles corresponding to vectors that belong to the Intra class and squares corresponding to Inter class. On the right we also show the decision regions that we obtain after training Gaussian mixtures on this training data, with the Intra decision region enclosed inside the black boundary. We can see that the decision regions are representative of the training data and may consist of disjoint sub-regions, as we have for the Intra case. From the training data we can see that a linear decision

boundary is not suitable in this case. Thus, imposing no constraint on the shape of the decision boundary aids our classifier in finding a better decision boundary. We achieved 98.2% correct classification using our minimum probability of error classifier. The classification result shows a variation of less than 1% across these different sequences. The classification scheme proposed in the TMN achieves around 92% correct classification, because it uses a linear decision boundary. Hence we can improve the mode decision using this scheme. We then implemented this mode decision in the H.263 framework and observed that the corresponding savings in total bit rate over the TMN decision (including residue bits, motion vector bits and overhead bits) were around 4.5~4.8% for different video sequences.

III. Mode Decision for Skipping or Coding Frames: Instantaneous case

The goal of this mode decision is to decide between skipping and coding a frame in order to maximize the perceived quality while achieving a target bit rate. In order to collect training data for our classifier, we first implement an exhaustive search based mode decision. We compute the effect of skipping as well as coding a frame on the quality q and the bit rate r before making a decision. In order to measure the perceived quality we use a spatio-temporal quality metric introduced by Wolf and Pinson [15]. In order to control the rate of coded frames, we change the quantization step size using the inverse quadratic model to relate the bit rate with the quantization step size. This model was proposed in [11] and it relates the rate (r) to the quantization step size (Q) using $r = \frac{a}{Q} + \frac{b}{Q^2}$, where a and b are constants that may be estimated using training data. We found that such a simple model cannot capture the variation of bit rate with quantization step size across many different scenarios, so we train separate models, i.e., parameters a and b, for low motion, medium motion and high motion sequences. More details about the quality metric can be found in the appendix.

Our cost is defined as a combination of the quality and rate, defined as $q + \lambda r$, where the factor λ is adjusted depending on application and we discuss this in more detail later. We compare this cost for the skipping or coding and choose the one that requires the smaller cost. Simultaneously we collect the cost difference and also evaluate some features that we use for training our classifier. Using the method described in Section II, we train the density functions for our features and implement our classification based scheme.

We first describe the exhaustive search based mode decision, after which we describe the features that we choose for the classifier and finally we include the results of our implementation. In all of the following discussion, the video sequence is represented by a sequence of frames $\ldots X(n-1), X(n), X(n+1)\ldots$ with n representing the time index. Since we are using lossy compression techniques the sequence of decoded frames may be represented as $\ldots \hat{X}(n-1), \hat{X}(n), \hat{X}(n+1)\ldots$. These may not be identical to the original video sequence. The previous decoded frame is used as reference to code the current frame and when a frame is skipped it is replaced by the previous decoded frame.

The steps for the exhaustive search based mode decision are as follows. We first compute the rate and quality when we skip a frame. We replicate the previous decoded frame to simulate skipping the current frame. We then estimate the quality q_1 of this sequence of two frames $\left\{\hat{X}(n-1), \hat{X}(n) = \hat{X}(n-1)\right\}$. The bit rate r_1 is estimated by averaging the bits needed to code the past ten frames, setting the bits for the current frame to zero and multiplying by the frame rate. We estimate the bit rate using a ten frame window as this smooths out the fluctuation due to large or small number of bits to code the current frame.

We then estimate quality and bit rate for coding the frame. We determine the bits available to code the current frame using history information and the target rate. We then estimate the quantization step size needed to code this frame using the inverse quadratic model. Using the previous decoded frame as reference and the computed quantization step size, we code the current frame and reconstruct it. We compute the quality q_2 of this two frames sequence $\left\{\hat{X}(n-1), \hat{X}(n)\right\}$ and the bit rate r_2 as before.

We then compare $q_1 + \lambda r_1$ with $q_2 + \lambda r_2$ and decide which of the two is better. The factor λ can be specified by the user in terms of the relative importance of either the rate or the quality. In our tests we place a greater emphasis on the quality of the sequence. This is done by adjusting λ so that λ times the target rate is 0.5 that is comparable to the range of the quality [-1, 0]. This is acceptable as we already use the quadratic model to try to control the bit rate. This decision strategy may be represented pictorially as in Figure 7.

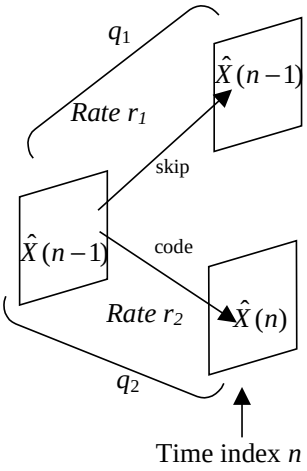


Figure 7. Pictorial representation of Exhaustive scheme

This exhaustive scheme is very similar to the Viterbi decoding scheme [16] with no look-ahead allowed, as at every instant in time *n*, costs for both the paths (skipping the frame or coding it) are compared and the best path is chosen, with the other being discarded. In fact this rate control strategy can be extended to allow for look ahead and this is described in Section IV. There we also include a greater discussion relating our scheme to the Viterbi decoding scheme.

During this exhaustive mode decision we also evaluate some features. We start with a large set of features and look at the correlation of these features with the decision sequence of the exhaustive search based mode decision to identify the features we use for our classifier. Some of the initial features that we identified are described in the following.

1) Size of motion vectors. This is computed as the sum of the square length of all the motion vectors in the frame.
2) MAD or SAD as a measure of quality of motion compensation. This is obtained as the sum of the MAD across all the macroblocks of the frame.
3) Measure of high frequency energy (HFE) in frame. This is obtained by taking a frame, down-sampling it by a factor 2 horizontally and vertically, then up-sampling it back to the original size and finding the energy in the difference between this and the original frame. This process may be viewed as in Figure 8.

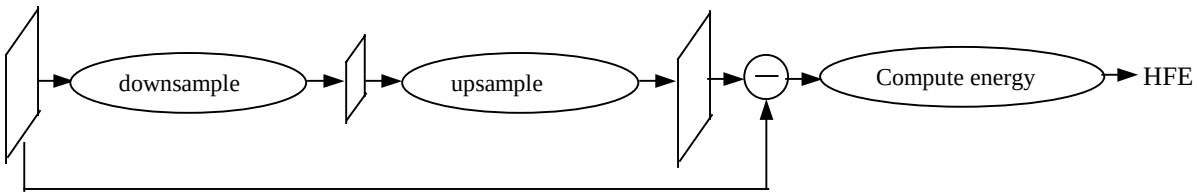


Figure 8. Computation of HFE

In the above figure down-sampling includes a pre-processing by a low pass filter and up-sampling includes a post-processing with a low pass filter.

4) Bits available to code current frame. This may be obtained from rate history.
5) Quantization step size used for current frame.
6) Energy in the frame difference between the current frame and previous frame.

We collected these features across different sequences and across different target rates using our exhaustive search based mode decision. We then correlate these features with the decision sequence that is viewed as a sequence of +1s and −1s with +1 corresponding to skipping a frame and −1 corresponding to coding it. More details of our computation of these correlation coefficients are included in the Appendix. The correlation coefficient for each of the features is included in Table 2.

Table 2. Correlation coefficients for features with decision sequence

Sequence	Target (kbps)	Size of MV	SAD	HFE	Available Bits	Quant. Step Size	Frame difference
Foreman	150	–0.5281	0.0643	–0.4741	0.0324	–0.1551	–0.4612
	300	–0.3342	–0.0200	–0.2761	0.0113	0.0412	–0.2129
	450	–0.4441	–0.0810	–0.3821	–0.251	0.1203	–0.1660
	600	–0.4765	–0.2060	–0.4109	–0.1185	0.355	–0.3132
Coastguard	150	–0.3318	–0.0578	–0.5202	–0.0993	0.2612	–0.0745
	300	–0.4001	–0.0339	–0.3942	–0.4613	0.5244	–0.0299
	450	–0.4114	–0.1320	–0.431	–0.4112	0.4197	–0.0299
	600	–0.4759	–0.1203	–0.4527	–0.4176	0.4003	–0.0868
Silent	150	–0.3914	–0.0218	–0.3452	–0.2147	0.2881	–0.057
	300	–0.3873	–0.1008	–0.3704	–0.2321	0.3092	–0.213
	450	–0.3901	–0.2137	–0.3883	–0.3001	0.3111	–0.2247
	600	–0.3874	–0.0789	–0.3872	–0.1492	0.2427	–0.2019

As can be seen from the table, most features are negatively correlated with the decision sequence while the quantization step size is positively correlated. This is as expected because small motion vectors, small SAD, small HFE and a small frame difference all imply that the current frame can be very well predicted by the previous frame, thereby meaning that they are

similar. This means that we can skip the frame, as we would then replace it with the previous frame. This biases the decision towards not coding the frame or a +1. So a smaller value of each of these features corresponds to a large value in the decision sequence, hence a negative correlation coefficient. A small number of available bits means that the quality of coding the frame will be poor, hence this also tends to bias the decision towards skipping the frame, thereby leading to a negative correlation. On the other hand a small quantization step size indicates that the quality of coding the frame will be good, thereby biasing the decision towards coding the frame and hence leading to a positive correlation coefficient.

Among these features we can see that the size of motion vectors and the HFE have the largest correlation coefficient values across most sequences and most rates. They are also relatively uncorrelated with a correlation coefficient 0.43. Hence, we choose these as representative features for out test. The motion vectors are expensive to compute but we can replace them with motion vectors from the previous frame, as the correlation between them is quite large at 0.87. The HFE for any frame of the sequence needs to be evaluated only once as this can be stored and looked up every time we code the sequence irrespective of the target rates. We train the density functions, as described in Section II, for these selected features and build our classification based mode decisions. We evaluate these mode decisions over three different sequences, Foreman, Coastguard and Silent. All these sequences are CIF at a frame rate 30 Hz. Of these sequences, Foreman is a high motion sequence, Coastguard is a medium motion sequence and Silent is a low motion sequence.

The results are evaluated using rate distortion curves. Four different target rates (150, 300, 450 and 600 kbps) were chosen for the test. The distortion is measured using the quality metric that we described earlier. We also compare the rate control using these mode decisions with one that we call No Skip rate control. This scheme tries to control the rate by only changing the quantization step size and skips a frame only when there are no bits available to code it. These results are shown in the following figures.

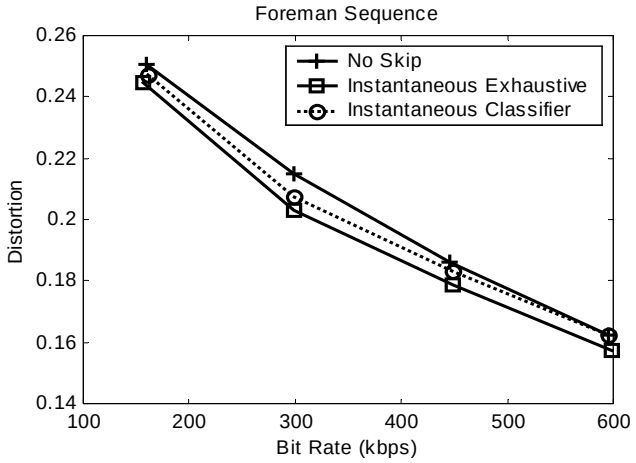


Figure 9. Rate Control results for Foreman sequence

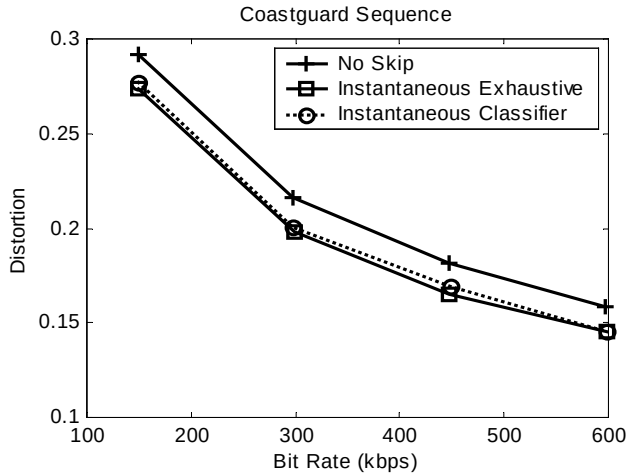


Figure 10. Rate control results for Coastguard sequence

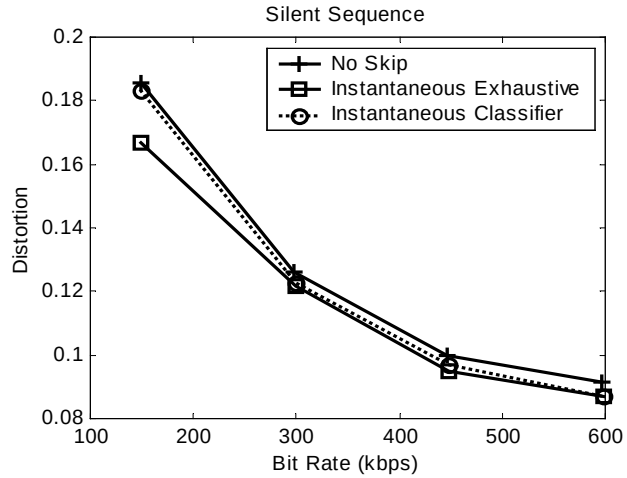


Figure 11. Rate Control results for Silent Sequence

We can see from these curves that rate control using both the exhaustive as well as the classification based mode decision perform better than the No Skip rate control as they provide smaller distortion at the same target rate. The performance of the classification based mode decision is close to the exhaustive search based decision. The error probability for the classifier for Foreman is 0.171, for Coastguard is 0.127 and for Silent is 0.131. This leads to the classifier curve for Foreman not being as close to the exhaustive mode decision curve as it is for the other sequences. We can also see that the average percentage improvement in quality across all the rates over the No Skip rate control is smallest for the Foreman sequence. This is because Foreman is a high motion sequence, hence skipping a frame is worse in quality than coding a frame a majority of the time, thereby leading to fewer frames being skipped. In terms of computation requirements the exhaustive mode decision uses roughly 3.5 times the computation as the classification based mode decision. The encoder with the classification based mode decision has a computation complexity within 5% of an encoder with no rate control strategy.

IV. Mode Decision for Skipping or Coding Frames: Look-Ahead Case

We extend the mode decision in the previous section by allowing a one-step look-ahead before making a decision. As in the previous section we first describe the exhaustive approach followed by a description of the features that we select and the results for our experiments.

The steps in the exhaustive approach using look-ahead are as follows. We first skip the current frame and replicate the previous decoded frame. The quality and the rate for this set of two frames $\left\{\hat{X}(n-1), \hat{X}(n) = \hat{X}(n-1)\right\}$ are computed as described before. Using this current reconstructed frame as reference, the future frame is both coded and skipped. Quality and rate for the two frame sequences $\left\{\hat{X}(n) = \hat{X}(n-1), \hat{X}(n+1) = \hat{X}(n-1)\right\}$, when we skip the future frame as well and $\left\{\hat{X}(n) = \hat{X}(n-1), \hat{X}_1(n+1)\right\}$, when we code the future frame using this skipped frame as reference, are computed.

We then code the current frame and quality and rate for $\left\{\hat{X}(n-1), \hat{X}(n)\right\}$ are computed. Then using this coded frame as reference the next frame is both skipped and coded. Quality and rate for $\left\{\hat{X}(n), \hat{X}(n+1) = \hat{X}(n)\right\}$, when we skip the future frame and $\left\{\hat{X}(n), \hat{X}(n+1)\right\}$, when we code the future frame, are also computed. This process may be represented pictorially as in Figure 12.

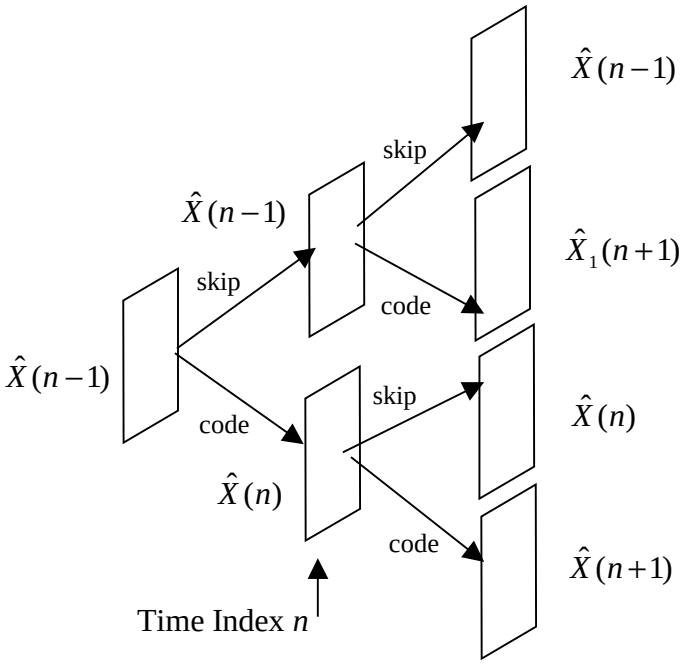


Figure 12. All frames generated for look-ahead exhaustive decision

The decision on coding or skipping the current frame is made after looking at the total cost ($q_i + \lambda r_i$) for each of the four paths (skip, code), (skip, skip), (code, skip) and (code, code). The path that provides the best cost is identified and the decision for the current frame is made appropriately. This strategy is very similar to the Viterbi decoding scheme with a one step look-ahead. The similarity is shown in Figure 13.

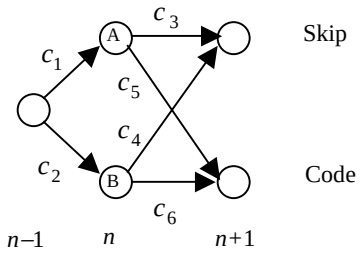


Figure 13. Similarity with Viterbi Decoding Scheme

We look ahead one step while trying to make a decision at the current time instant *n*, i.e. decide between nodes A and B. The costs c_i in the figure corresponding to $q_i + \lambda r_i$ are compared i.e. $c_1 + c_3$, $c_1 + c_5$, $c_2 + c_4$ and $c_2 + c_6$ are compared to get the best cost. The node through which this path passes is chosen as the decision for the current frame while the other node is discarded.

For the look-ahead classifier, the feature set that we start with is the same as in the previous section. Of these features we find that the size of motion vectors, the HFE and the quantization step size are the most representative features, i.e. they have the largest correlation coefficients with the decision sequence and have relatively low correlation between themselves. Of these features, identifying motion vectors requires a large amount of computation, so we can replace the motion vector size with the motion vectors from the previous frame. However, it is not good to approximate the future frame motion vectors with those from the previous frame as the correlation between the motion vectors that are two frames apart is not so large. Hence we decide against using motion vector size as a feature for this decision strategy. As against this, the quantization step size for the current frame and those for future frames can be estimated using the model we have, so we prefer to use this. We choose four features for our classifier, a) Quantization step size for current frame.

b) Estimate of quantization step size for future frame, if we skip the current frame.

c) Estimate of quantization step size for future frame, if we code the current frame.

d) HFE for the current frame.

All these features are easy to compute and as stated before, the HFE for any frame needs to be evaluated only once for any sequence.

These look-ahead mode decisions are applied to the Foreman, Coastguard and Silent sequence and the results are compared with the No Skip rate control. These results are shown in the following figures.

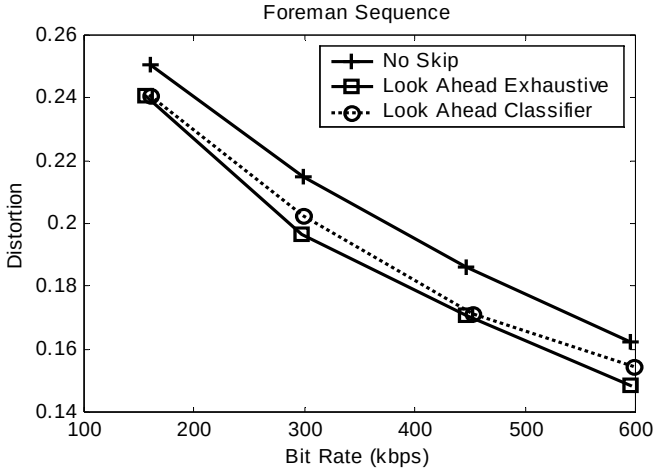


Figure 14. Rate control results for Foreman sequence

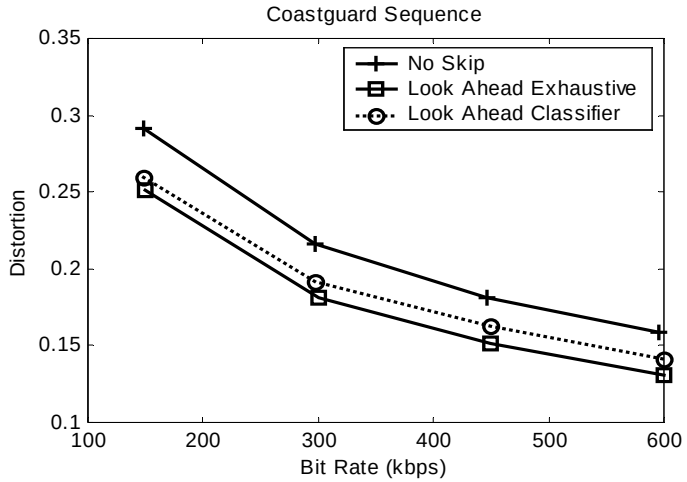


Figure 15. Rate control results for Coastguard sequence

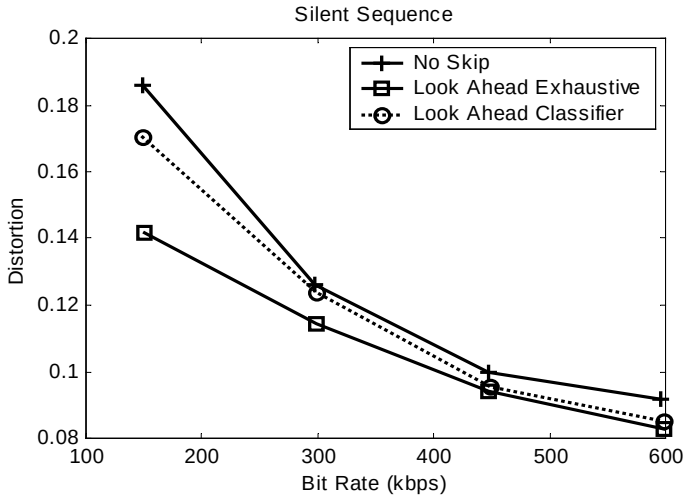


Figure 16. Rate control results for Silent sequence

As before, we can see that rate control using both the exhaustive as well as the classification based mode decision perform better than the No Look rate control. The performance of the look-ahead mode decisions is better than the instantaneous mode decisions. The error probability for the classifier for Foreman is 0.168, for Coastguard is 0.202 and for Silent is 0.206 and hence the performance of the classifier is not as good as the exhaustive mode decision. As pointed out earlier the average percentage improvement in quality across all the rates over the No Skip rate control is smallest for the Foreman sequence as it is a high motion sequence. In terms of computation requirements the look-ahead exhaustive mode decision uses roughly 8x times the computation as the classification based mode decision. As before, the encoder with the classification based mode decision has a computation complexity within 5% of an encoder with no rate control. The number of steps that we are allowed to look-ahead can be

increased for a greater improvement in performance, as the classification based mode decision does not require a significant increase the computation requirements.

Through our experiments we have shown that we can use classification based strategies to intelligently decide between skipping a frame and coding it to achieve a better rate control strategy than just changing the quantization step size to control rate and use frame skipping only when we have no bits available to code the current frame. All of this discussion was for rate control at the frame level, so we use one quantization step size for the entire frame, however our classification based schemes can easily be extended to macroblock layer rate control, when we can decide to skip or code a macroblock intelligently.

V. Extension to Scalable Coding

Scalable bitstreams are used by video coding schemes to improve the error resilience over lossy networks. The bitstream is partitioned into multiple layers and consists of a base layer and one or more enhancement layers. The base layer is usually assigned the highest priority and error protection and possesses enough information for the decoder to reconstruct the video sequence at a lower resolution, frame rate or quality. The enhancement layers consist of residue information between the base layer and the actual video sequence thereby allowing for reconstruction of the video at a higher resolution, frame rate or quality. There are three different scalabilities supported in the H.263 and MPEG-2 standards, which are the Spatial, Temporal and SNR scalabilities. In the spatial scalability, video at a lower resolution forms part of the base layer. In temporal scalability, the base layer consists of the video sequence coded at a lower frame rate, while in SNR scalability the base layer consists of the video sequence coded at a high quantization step size. In this paper we consider two of these scalability modes, temporal and SNR that are relevant to the skip or code mode decision mentioned earlier. We focus our discussion to the use of one enhancement layer. The work described here can be extended to using multiple enhancement layers.

Temporal scalability is achieved by skipping frames while coding the base layer, to obtain a lower frame rate video. Frames that are skipped are predicted (may be forward, backward or bi-directionally predicted) from the current and previous coded frames and the residue and motion vectors are included in the enhancement layer. SNR scalability is achieved by coding frames at a higher quantization step size at the base layer and then coding the residue between these frames and the actual video at a lower quantization step size to form the enhancement layer.

Using the mode decisions described in Sections III and IV, we code a video sequence by sometimes using a high quantization step size and sometimes skipping frames. Hence, this coded video may be viewed as a base layer generated using a encoder that switches between SNR and temporal scalabilities, as detailed later. We can therefore extend our work from the previous section to investigate the error resilience and performance of our strategy when implemented over lossy networks. To do this, we first generate an enhancement layer corresponding to our base layer. This process is highlighted in Figure 17 and Figure 18.

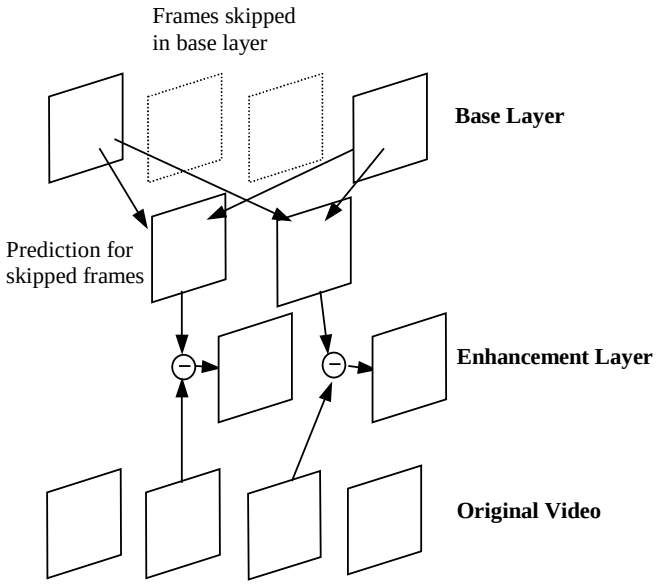


Figure 17. Enhancement layer generation for skipped frames

From the figure we can see that when we skip a frame in the base layer, we build a prediction for the frame, that may be forward, backward or bi-directionally predicted from the preceding and following coded frames and the residue between this prediction and the original video is included as part of the enhancement layer. This is equivalent to temporal scalability.

The process of generating the enhancement layer when we code a frame in the base layer is highlighted in the following figure.

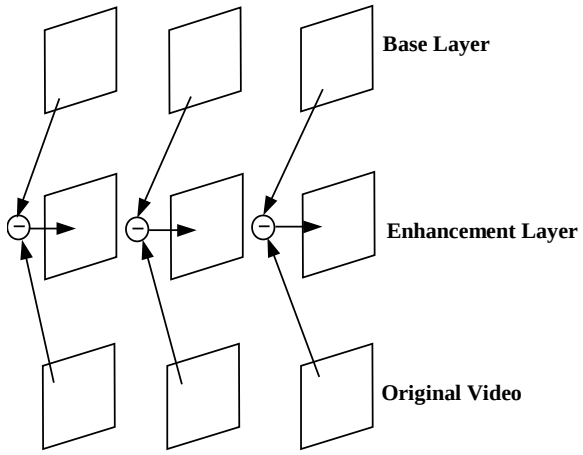


Figure 18. Enhancement layer generation for coded frames

Coded frames are subtracted from the original video and the residue for each of these frames is included as part of the enhancement layer, which is coded at a lower quantization step size. This is equivalent to SNR scalability.

Our coding scheme switches between these two modes and we call it adaptive SNR/temporal (AST) scalable coding. Once we have built the enhancement layer corresponding to our base layer, we code it at the same target rate as the base layer. In order to achieve the target rate, we change only the quantization step size and do not allow for skipping of frames. We then simulate lossy network conditions and then reconstruct the video sequence by combining the two layers. The lossy conditions are simulated by throwing away some of the base layer macroblocks and some of the enhancement layer macroblocks and then combining the layers. We examine different error rates and their impact on the performance. Some error concealment is used at the decoder side to improve the quality of the decoded video. When a base layer macroblock is corrupted it is replaced by the corresponding macroblock from the previous frame, while when an enhancement layer macroblock is corrupted, it is thrown away. We also generate an enhancement layer for the No Skip rate control scheme (identical to SNR scalability) and code and combine the layers as before. The resulting rate distortion curves are plotted in Figure 19.

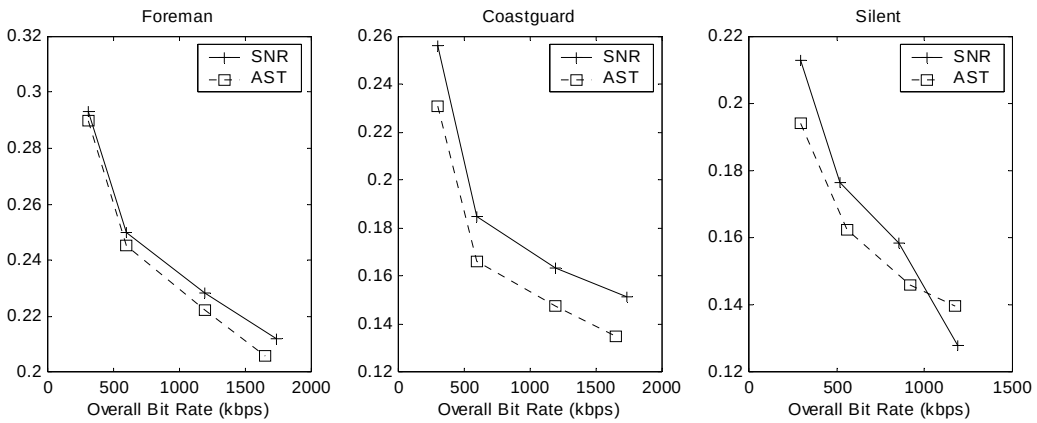


Figure 19. Performance under lossy network conditions

The curves plotted in the figure are with 5% loss in the base layer and 10% loss in the enhancement layer. As can be seen from the plots, the performance of the AST is better than using just SNR scalability across different target rates for all the three sequences, with only one exception (Silent sequence at 1200 kbps, 600 kbps for base layer and 600 kbps for enhancement layer). Sample frames from the Foreman sequence to highlight the improvement in distortion are shown in Figure 20.

Figure 20. Sample frames from Foreman sequence with 5% base loss and 10% enhancement loss SNR Scalability (left) and AST Scalability (right)

The SNR scalability frame has a PSNR of 27.7 dB while the AST scalability frame has a PSNR of 29.09 dB as compared to the original frame.

We also investigate the effect of varying the enhancement layer and base layer losses at a fixed target rate of 300 kbps for base layer and 300 kbps for enhancement layer. We compare the performance with SNR scalability and the resulting improvements are shown in Figure 21.

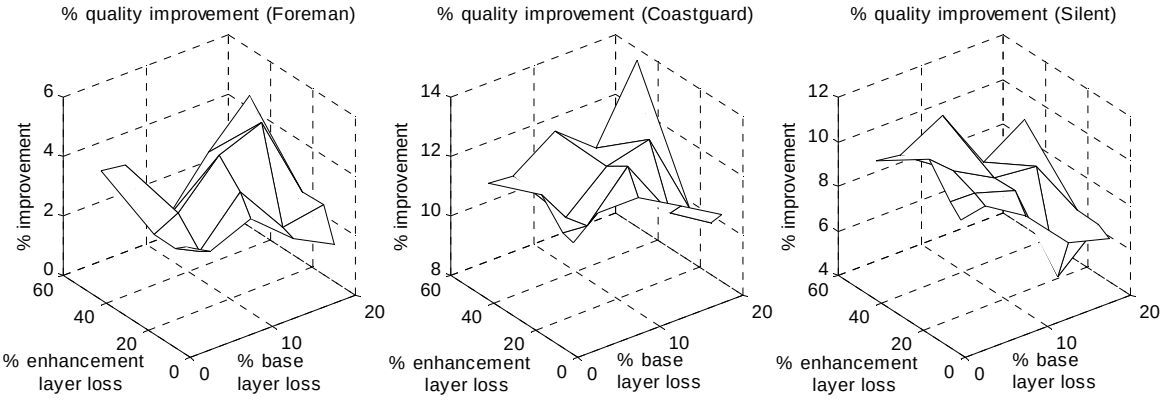


Figure 21. Improvement in quality of AST over SNR scalability for different error rates

All of the above plots are generated for a target rate of 300 kbps for the base layer and the same for the enhancement layer. We can see that for all the different error rates the performance in terms of quality is better when we generate the base layer using the classification based mode decision as opposed to just changing the quantization step size. We can also see that the percentage improvement in quality is higher for Silent and Coastguard sequences as opposed to the Foreman sequence. This may be explained by a combination of facts. When we skip a frame, the enhancement layer carries greater amount of information for high motion sequences than it would for a low motion sequence. Also we have a larger amount of losses in the enhancement layer as compared to the base layer. So we lose more information for higher motion sequences when we skip frames in the base layer. Hence we have a smaller improvement for the Foreman sequence. The case with 0% loss in the base layer and 100% loss in the enhancement layer degenerates to the rate control problem that we focused on in Sections III and IV.

VI. Conclusion

The main contribution of this paper is the classification based approach to mode decisions in the video encoding process. We successfully convert the problem of minimization of a certain cost function into a standard minimization of classification error problem and use traditional pattern classification techniques to solve it. We use this approach to improve the performance of the Inter-Intra mode decision and reduce the bitstream size by 4.5~4.8% over the mode decision as provided in TMN 10. We then use this approach to the rate control problem and show an improvement in performance in the rate-distortion sense over using the no skip approach both for the instantaneous as well as the look-ahead mode decision. The improvement in quality for the instantaneous decision is 4~12% while for the look-ahead decision it is 7~18% over the no skip rate control. We also extend this work to the scalable video coding and show that with the

adaptive SNR/temporal (AST) scalability we improve the performance in terms of quality under error prone conditions by 5~15% over using SNR scalability only.

Acknowledgement

We would like to thank the anonymous reviewers for their feedback and valuable comments that helped improve the manuscript. We are also grateful to Prof. Kumar for his valuable and insightful comments.

References

[1] Motion Pictures Experts Group, "Overview of the MPEG-4 Standard", ISO/IEC JTC1/SC29/WG11 N2459, 1998.

[2] *Video Coding for Low Bit rate Communication*, ITU-T Recommendation H.263 Version 2, Jan. 1998.

[3] Video Encoder Test Model, Near-Term, Version 10 (TMN10) Draft 1, Apr. 1998.

[4] D. Turaga and T. Chen, "Estimation and Mode Decision for Spatially Correlated Motion Sequences", submitted to *IEEE Trans. Circuits Syst. for Video Technol.*.

[5] T. V. Lakshman, A. Ortega and A. R. Reibman, "VBR video: Tradeoffs and potentials," *Proc. IEEE*, vol. 86, no. 5, pp. 952-73, January 1998.

[6] C. –T. Chen and A. Wong, "A self-governing rate buffer control strategy for pseudoconstant bit rate video coding," *IEEE Trans. Image Processing*, vol. 2, pp. 50-59, January 1993.

[7] J. Choi and D. Park, " A stable feedback control of the buffer state using the controlled multiplier method," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 546-58, September 1994.

[8] C. Y. Hsu, A. Ortega and A. R. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," *IEEE J. Selected Areas in Commun.*, vol. 15, no. 6, August 1996.

[9] H. Song, J. Kim and C. –C. Jay Kuo, "Real-time encoding frame rate control for H.263+ video over the internet," *Signal Processing: Image Communication*, no. 15, September 1999.

[10] F. C. Martins, W. Ding and E. Feig, "Joint control of spatial quantization and temporal sampling for very low bit rate video," IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996.

[11] T. Chiang and Y. –Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no.1, p. 246-50, September 1997.

[12] W. Ding and B. Liu, “Rate control of MPEG video coding and recording by rate-quantization modeling,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 12-20, February 1996.

[13] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley and Sons, New York, NY, 1973.

[14] G. McLachlan and T. Krishnan, “The EM algorithm and extensions,” Wiley Interscience, New York, NY, 1996.

[15] S. Wolf and M. H. Pinson, “Spatial-Temporal Distortion Metrics for In-Service Quality Monitoring of Any Digital Video System,” SPIE International Symposium on Voice, Video, and Data Communications, Boston, MA, September 11-22, 1999.

[16] G. D. Forney, Jr., “The Viterbi Algorithm,” *Proc. IEEE*, vol. 61, pp. 268-78, March 1973.

Appendix A. Spatio-Temporal Quality Metric

A quality metric should measure spatial as well as temporal quality. The PSNR provides a poor measure of temporal quality, hence we use the metric proposed by Wolf and Pinson [15]. To evaluate this metric, first the luminance components of the input and output video streams are processed using horizontal and vertical edge enhancement filters. These processed streams are partitioned into spatio-temporal (S-T) regions in which features that quantify spatial activity as a function of angular orientation are extracted. These are then clipped to emulate perceptibility thresholds. Distortions due to gains and losses in feature values are calculated using functional relationships between the input and output feature values that emulate visual masking. These distortions are then collapsed over space and time. The choice of edge enhancement filters and the perceptibility thresholds are optimized based on their correlation with perceptual distortions. The block diagram of this process is shown in Figure 22.

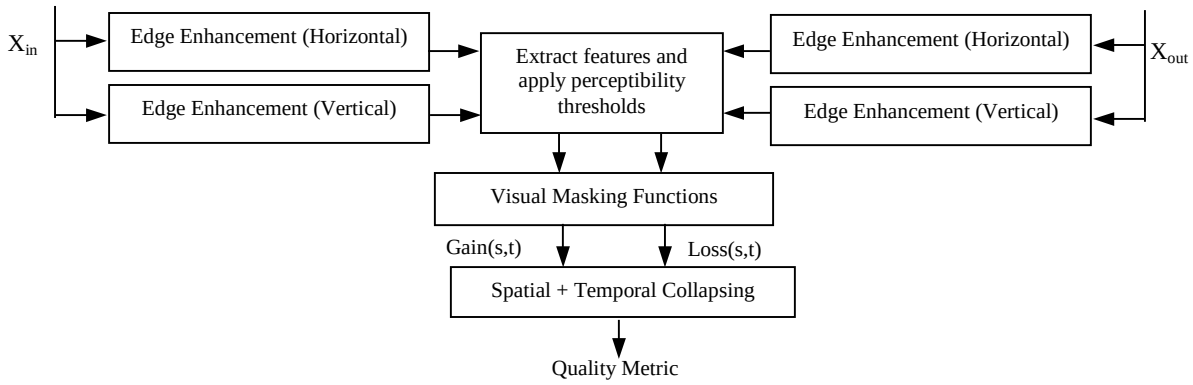


Figure 22. Computation of spatio-temporal metric

The quality metric computed as in Figure 22 lies in the range [-1,0] with 0 corresponding to perceptually lossless video. We use the negative of this feature as a measure of distortion, with 0 corresponding to no visible distortion and 1 corresponding to a large distortion.

Appendix B. Correlation coefficient between decision sequence and feature sequence

In order to compute the suitability of features to use in our classifier, we compute the correlation coefficient between the features and the optimal decision sequence. The optimal decision sequence is found using the exhaustive schemes and we view it as a binary sequence of +1s and –1s, corresponding to the two mode decisions. Before we correlate the feature sequence with the decision sequence, we threshold the feature sequence to convert it also to a binary sequence of +1s and –1s. This is done so that we get a better estimate of correlation between the feature sequence and the decision sequence. If we do not convert the feature sequence to a binary sequence, even if the feature is perfectly representative of the decision sequence, i.e. it is high when we have decide Intra (+1) and low when we decide Inter (–1) we do not get a correlation coefficient 1. We highlight this with an example in Figure 23.

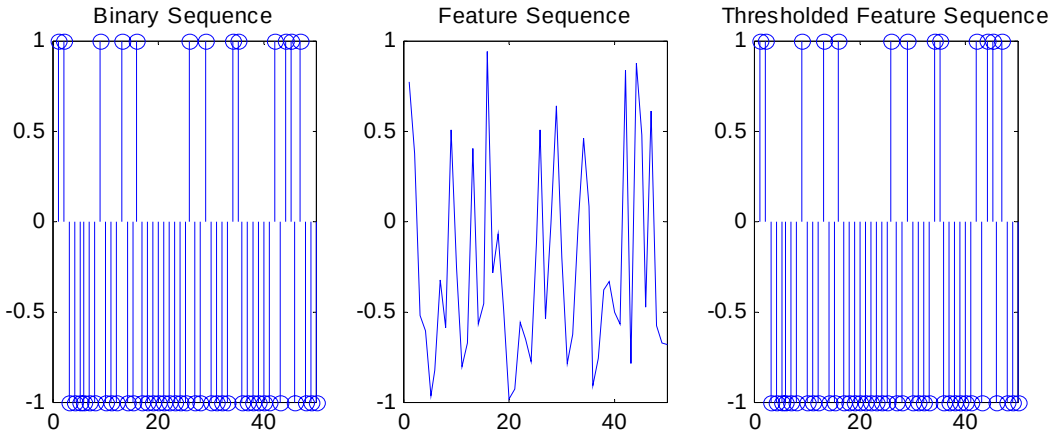


Figure 23. Correlation between decision sequence and feature sequence

In this figure each of the three sequences has 50 samples. The feature sequence may be used to predict the decision sequence very precisely, because we can see that when the feature has a high value, the decision sequence has value +1 and when the feature value is low, the decision sequence has value –1. However, when we compute the correlation coefficient between these two sequences, the resulting value is only 0.44. As against this, if we convert the feature sequence to a binary sequence, using a threshold of 0, before computing the correlation coefficient, the resulting value is 1 as desired. We try multiple thresholds to convert every feature sequence into a binary sequence before correlating with the decision sequence and report the best correlation coefficient obtained.