I/P FRAME SELECTION USING CLASSIFICATION BASED MODE DECISION

Deepak S. Turaga and Tsuhan Chen

Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
{dturaga, tsuhan}@andrew.cmu.edu

ABSTRACT

There are many mode decisions in the video coding process that are used to optimize the performance in terms of the bit rate, the speed and the quality of the decoded video. We describe a classification based scheme for making mode decisions in the video coding process. We then illustrate the performance of the scheme using the I/P frame selection as an example. The performance of our scheme is measured in terms of both the bit rate as well as the computation complexity, across different kinds of sequences, and the results are very encouraging.

1. INTRODUCTION

Inherent in the video coding process are many mode decisions that improve one or more of the aspects in the speed-quality-bit rate tradeoff. Video standards such as MPEG [1] and H.263 [2] specify the bitstream syntax completely, but allow for optimizations in the encoding process in terms of algorithms and mode decisions used.

The goal of a mode decision is to minimize a cost that may be defined in terms of the speed-quality-bit rate tradeoff and 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 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 cost minimization problem to a more traditional error probability minimization problem and use standard classification techniques like the likelihood ratio test to solve it. More details on this can be found in [3]. In this paper, we use this classification based approach to decide between coding a frame in the intra (I) mode or in the predictive (P) mode.

Some previous work on the adaptive selection of I and P frames while video coding is done in [5] and [6]. Lan et al [5] use motion analysis to determine scene content while Yoneyama et al [6] use macroblock activity information to determine the length of the group of picture (GOP), or the distance between successive I frames. We propose a formal decision scheme, based on classification techniques, to minimize the actual cost of video coding, as against using such heuristics to make the decision.

This paper is organized as follows. Section 2 describes the classification based scheme for making mode decisions. Section 3 illustrates the performance of this scheme for the I and P frame selection. We then conclude with Section 4.

2. CLASSIFICATION BASED MODE DECISION STRATEGY

There are many mode decisions in the encoding process and these may be at different coding levels. For instance some mode decisions are made frame by frame while others are made on a block by block basis. Each of these different modes has a cost associated with it. This cost may be defined in terms of the bits, the time needed or the quality or a combination of some of these. So the mode decision involves choosing the mode that has a smaller cost associated with it.

In principle, to make the optimal decision one can try all the modes and choose the mode that has the lowest cost. However, computing the actual costs before making a decision is very computationally intensive as this involves trying every mode 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 mode, but do not require as much computation to evaluate. We would then like to train a classifier to choose the mode requiring the smaller cost based on the features. For all of the future discussion we limit ourselves to choosing between two modes, however the decision strategy described is not limited to binary mode decisions and may be easily extended to when we have more than two modes to choose from.

In order to train the classifier we need to collect a set of ground truth data. For each coding unit in this data set we need to evaluate the features and the cost for the different modes, which may be collected using an exhaustive strategy. Each coding unit in the training data may thus be labeled as belonging to one of two classes, where the class includes all the coding units for which a particular mode has a smaller cost than the other mode. The goal of the classifier is to correctly partition the feature space so that all coding units are assigned the correct class label. A lot of times it is not possible for the classifier to achieve perfect classification, however a sub-optimal performance is acceptable as it comes with the benefit of reduced computational complexity over the exhaustive strategy.

For each misclassified coding unit instead of incurring the smallest cost, we incur a larger cost and so pay an additional cost corresponding to the difference between the costs for the two modes for that coding unit. Hence each coding unit has this additional cost of misclassification associated with it. We show an example of the feature space with one feature computed per coding unit in Figure 1.

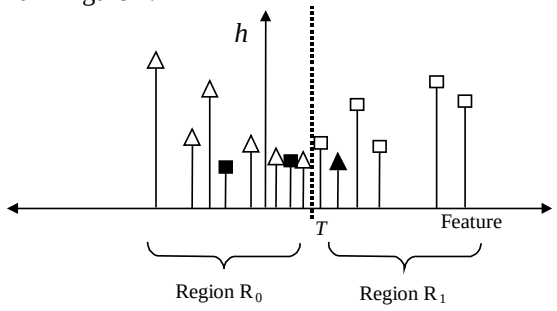

Figure 1. Feature space

In the figure, triangles correspond to coding units belonging to one class while squares are used to represent coding units belonging to the other class. The x-axis corresponds to the value of the feature corresponding to the coding unit while the height *h*, associated with every coding unit corresponds to the additional cost of misclassification for it. In general there may be many features associated with each coding unit and we may group these together as a feature vector. The goal of the classifier is to partition the feature space into two regions, R_0 and R_1, one each corresponding to a different class so that the total cost is minimized or equivalently the total additional cost due to misclassification is minimized. In our figure R_0 corresponds to the triangle class while R_1 corresponds to the square class. For instance, if we partition the space into these two regions using threshold *T*, we see that some coding units from both classes are misclassified. These coding units are represented by dark triangles and squares and the total additional cost incurred is the total of the heights of these dark triangles and squares.

The problem of partitioning the feature space into two regions to minimize a certain cost is reminiscent of standard classification techniques, however one of the significant differences is that each of our coding units has this additional cost of misclassification or height associated with it. To solve the problem using the standard classification techniques we somehow need to convert these coding units with the associated heights to units 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 unit with height *h* with *h* units at that location. Without loss of generality we may assume *h* to be an integer, as we can scale non-integer values appropriately. We illustrate this transformation in Figure 2.

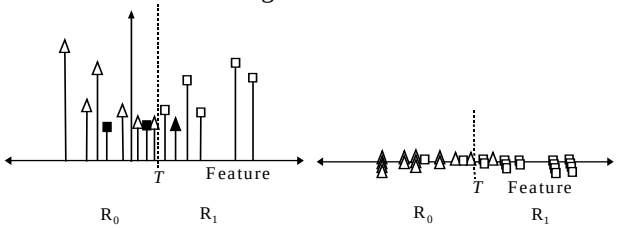

Figure 2. Transformation of feature space

From Figure 2 we can see that each unit in the old feature space is replaced by multiple units at that location, their number being equal to the height associated with the original unit. Now, standard classification techniques may be applied in this new feature space to estimate the probability density functions (pdf) for this new set of feature vectors. We use a mixture of Gaussians to model the pdfs for the two different classes of feature vectors. We show an example of this in Figure 3.

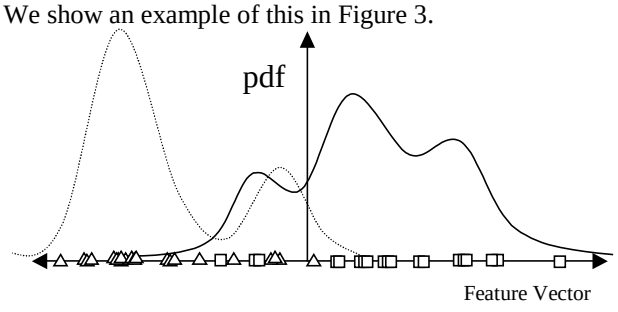

Figure 3. Gaussian mixtures to model the data pdf

The pdf drawn with the dashed line corresponds to feature vectors from the triangle class while the pdf drawn with the solid line represents vectors from the square class.

In [3] we show that the problem of minimizing the total additional cost due to misclassification in the old feature space maps to the problem of minimum error probability classification in the new feature space. Also the decision

boundary obtained in the new feature space is exactly the boundary we want in the old space as none of the training data points are displaced from their original positions.

Minimum probability of error classifiers are well understood in literature [4] and the decision boundary is determined using the likelihood ratio test. 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 as well as the class conditional probability density functions for the feature vectors. Once we have these pdfs, we use the likelihood ratio test on the feature vector corresponding to a coding unit and determine the mode for the coding unit.

3. I/P FRAME SELECTION

Coding standards such as the ISO MPEG series and the ITU H series allow for different kinds of coding modes for frames. A frame may be Intra (I), Predictive (P) or Bidirectionally-predictive (B). An I frame is coded in isolation from other frames using transform coding, quantization and entropy coding. A P frame is predictively coded, while a B frame is predicted bidirectionally. An I frame is often used to efficiently code frames corresponding to scene changes, i.e. frames that are different from preceding frames and cannot be easily predicted. Frames within a scene are similar to preceding frames and hence may be coded predictively as P or B for increased efficiency. Frames between two successive I frames, including the leading I frame, are collectively called a group of pictures (GOP). The work in this paper focuses on video streams with I and P frames only.

Since video sequences have variable scene durations, depending on the content, it is not possible to use a fixed GOP structure to efficiently code the video sequence. This is because the position of I frames in the sequence depends on when scene changes occur. So we need a mechanism to efficiently decide when to code a frame as an I frame and when to code a frame as a P frame.

In practice, video coding standards allow for macroblocks (16×16 regions of the frame) in a P frame to be intra coded if they cannot be predicted efficiently. This means that even if we set all the frame types to be P, there may be many macroblocks in each frame that are intra coded. This macroblock based mode decision may be used to account for scene changes. However coding a P frame is more computationally expensive than coding an I frame. This is because coding P frames uses motion estimation and compensation and also this additional decision for each macroblock in the frame. Hence making the decision at the frame level to code a frame as an I frame is more efficient in terms of computation.

Hence the mode decision that we wish to make using the classification strategy is to choose between coding a frame as an I frame or as a P frame. In order to train the classifier we need to collect a number of training frames and know the number of bits needed to code each frame as an I frame and as a P frame. Hence we disable the macroblock based mode decision and so all macroblocks in an I frame are intra coded and all macroblocks in a P frame are predictively coded. The cost of misclassification is in terms of the additional number of bits needed to code the frame using the wrong mode.

We use two different kinds of video sequences to evaluate the classification scheme. The first was a high motion video sequence made up of advertisements. We call this sequence Ads. This sequence had frequent scene changes, camera zooms and pans and a lot of motion. The second sequence was a news clip and we call it News. This sequence contained a moderate amount of motion and some scene changes. Sample frames from these sequences are shown in Figure 4.

Figure 4. Sample frames from Ads (left) and News

We had five minutes for both Ads and News in QCIF (176×144) format, sampled at 15 Hz for each sequence. We partition these into one-minute clips and call these Ads1 through Ads5 and News1 through News5. We use Ads1, Ads2, News1 and News2 to train the classifier and test the classifier on the remaining six clips.

In order to train the classifier we first code each frame in the training sequences in both the I mode and the P mode and record the number of bits for each mode. Hence we also can identify which frames need to be coded in the I mode and which frames need to be coded in the P mode. Simultaneously we also collect features from these frames representative of the bits needed for either mode. We examine three features, the size of motion vectors (MV) from the previous frame, the frame difference (FD) between the previous frame and the current frame and the high frequency energy (HFE) in the current frame. MV is the sum of the lengths of all motion vectors from the previously coded frame. FD is the sum of absolute values of pixel differences between the current frame and the previous frame. The HFE is obtained by taking the frame, down-sampling it by a factor of 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. Down-sampling includes a pre-processing by a low pass filter and up-sampling includes a post-processing with a low pass filter.

In order to choose between these features we correlate the feature sequences with the optimal decision sequence; a sequence of +1s and –1s with +1 corresponding to a P frame and –1 corresponding to an I frame. Before correlating the feature sequences with this decision sequence, we threshold the feature sequences to convert them to binary sequences. We try different thresholds for each feature sequence and report the best correlation coefficients. These are included in Table 1.

Table 1. Correlation coefficients for thresholded feature sequences

Sequence	MV	FD	HFE
Ads	–0.8558	–0.8223	0.0561
News	–0.9523	–0.7825	0.0516

We can see from the table that MV and FD have larger correlation coefficients with the decision sequence than the HFE. Both of these are negatively correlated as when each of them is high, the decision is biased towards Intra coding, which corresponds to a –1 in our decision sequence. We choose these two features for our classification scheme.

Once we train the pdfs of the feature vectors we then test the performance on the remaining six Ads and News sequences. The results are shown in the following table.

Table 2. Results of classification scheme

Sequ-ence	# Bits Exhaustive	# of I frames	# Bits Classifier	# wrong frames
Ads3	4613173	63	4627298	6
Ads4	12093277	43	12130402	7
Ads5	5987344	64	6082795	11
News3	9897030	7	9897030	0
News4	7216019	7	7216019	0
News5	7066404	6	7066404	0

The exhaustive bits column uses the exhaustive strategy to identify I and P frames and the number of I frames is the number determined using the exhaustive strategy. We can see that for the Ads sequence the scene changes occur quite rapidly, roughly once a second while for the News sequence scene changes occur more slowly. Each of these one minute sequences consists of 900 frames. From the table we can see that the performance of the classifier is 99.11% correct classification for the Ads sequence and 100% correct classification for the News sequence. The total overhead cost in bits paid due to misclassification is less than 2% for the Ads sequences, while there is no overhead for the News sequences. In order to measure the penalty paid when there is a mismatch between training and testing data, we use the classifier trained on the Ads sequences to classify the News sequence and vice versa. We find that the resulting classification performance is 97~99% correct, which means that a single classifier may be used to make this mode decision across different kinds of sequences, with reasonable performance.

If, instead, we label all frames as P and simply let the macroblock mode decision choose intra versus inter for each macroblock, fewer bits are needed than exhaustive bits, but within 0.5%. However, each P frame needs on the average, around 40% more computation time than an I frame due to motion compensation and the additional macroblock level mode decision. We save on this extra computation if we can make the mode decision at the frame level. Finally, we also compare this scheme with a fixed I-P schedule, in particular one I frame followed by thirty P frames, since we have around 190 I frames out of 5400 total P frames for our test sequences. The bits for this fixed I-P schedule are around 8~12% larger than the classifier bits, leading to coding inefficiency. Hence the classification based scheme is an efficient way to determine the type of coding, I or P for a frame.

4. CONCLUSIONS

We introduce a classification based scheme for I and P frame selection for video coding. The classification performance of the scheme is over 99% in terms of the number of frames correctly classified, as compared with an exhaustive strategy to determine the frame type. In terms of bit rate the performance is within 2% of the exhaustive strategy for the Ads sequences and identical to the exhaustive strategy for the News sequences. By making the decision to code a frame as I at the frame level we save around 40% of computation time as against if we label the frame as P and let the macroblock based decision to determine which macroblocks of the frame should be intra coded. We also save around 8~12% in terms of bit rate over using a fixed I-P schedule.

5. ACKNOWLEDGEMENT

The work done in this paper was supported in part by grants from the Institute of Information Industry, Taiwan, and the Texas Instruments University Research Program.

6. 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] Deepak S. Turaga and Tsuhan Chen, "Classification based mode decisions for video over networks," *IEEE Trans. Multimedia*, Special Issue on Multimedia over IP, vol. 3, no. 1, pp. 41-52, March 2001.

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

[5] A. Y. Lan, A. G. Nguyen and J. N. Hwang, "Scene-content-dependent reference frame placement for MPEG video coding," *IEEE Trans. Circuits and Systems Video Technol.*, vol. 9, no. 3, pp. 478-89, 1999.

[6] A. Yoneyama, Y. Nakajima, H. Yanagihara and M. Sugano, "MPEG encoding algorithm with scene adaptive dynamic GOP structure," IEEE Third International Workshop on Multimedia Signal Processing, pp. 297-302, 1999.