Macroblock Mode Decision In Video Encoding As a Classification Problem

A Project Report
Submitted in partial fulfilment of the requirements for the Degree of
Master of Engineering
in
Computer Science and Engineering

by

Joshi Urvang Bharadwaj

Computer Science and Automation
Indian Institute of Science
BANGALORE – 560 012
JUNE 2009
TO

My Grandfather
and
My Parents
Acknowledgements

Firstly, I would like to sincerely thank my project guide Dr Chiranjib Bhattacharyya and my co-guide Dr K R Ramakrishnan for providing me the required guidance throughout my project-work.

I would also like to thank Dr Hari Kalva (Associate Professor, Computer Science & Engineering, Florida Atlantic University), and his student Rashid Jilani for helping me with the data generation for my experiments.

I want to thank all my lab-mates and friends who have knowingly or unknowingly extended their support for my project.

Finally, I would like to thank my parents who have always been a great source of inspiration and support for me.
Abstract

Latest video standards such as H.264/AVC are extremely efficient in terms of coding efficiency at the expense of higher computational complexity. On the other hand, older codecs such as MPEG-2 and MPEG-1 are still widely in use. Fast video transcoding methods are desirable to facilitate the gradual transition to the newer codecs. Also, various devices like HDTV, computers and mobile devices use various formats and resolutions of videos. Such co-existence of video standards requires efficient conversion algorithms. However, video encoding and transcoding incur intensive computation cost, which has only increased in newer codecs such as H.264. Hence, video encoders and transcoders have to manage the trade-off between quality and complexity. Optimal coding decisions are usually too expensive to be practical. Sub-optimal techniques are typically used, but they are far from satisfactory in terms of the conversion speed and video quality. Notably, the most computation expensive phase in this conversion process is the macroblock-mode decision during inter-frame prediction and intra-frame prediction. These decisions also directly reflect on resulting video quality. We propose to approximate these mode-decision calculations using a chance-constraint based classifier. We implement our method for intra-frame prediction in H.264/AVC encoding and show that we can achieve approximately 20% reduction in encoding time as compared to reference software implementation with average loss in quality of merely 0.05 dB in PSNR.
List of Tables

3.1 Video Sequences Performing Poorly .................................................. 12
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Training decision trees</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Low complexity encoder using decision trees</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>MPEG-2 to H.264 Trascoding Decision Tree</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>Training chance constrained classifier for video coding</td>
<td>13</td>
</tr>
<tr>
<td>3.2</td>
<td>Using chance constrained classifier in video coding</td>
<td>14</td>
</tr>
<tr>
<td>3.3</td>
<td>Binary tree used in Intra MB coding</td>
<td>15</td>
</tr>
<tr>
<td>3.4</td>
<td>Geometrical Interpretation of SOC Constraint</td>
<td>21</td>
</tr>
<tr>
<td>4.1</td>
<td>(a)(c) RD performance of the proposed H.264 Intra frame encoder with Chance Constrained classifier; (b)(d)Time complexity reduction.</td>
<td>24</td>
</tr>
<tr>
<td>4.2</td>
<td>(a)(c) RD performance of the H.264 Intra frame encoder with Decision Tree classifier; (b)(d)Time complexity reduction.</td>
<td>25</td>
</tr>
</tbody>
</table>
Keywords

H.264/AVC, video coding, mode decision, classification, chance constrained programming
Chapter 1

Introduction

1.1 Motivation

Recent developments in video coding standards have resulted in the generation of new codecs which achieve high compression efficiency. But this compression is achieved at the cost of a high computational complexity associated with it. This is the major reason why the increased compression efficiency cannot be obtained in all application domains, especially resource constrained devices. The same limitations apply to transcoding as well. Hence, it is desirable to be able to improve the speed of the encoding/transcoding process.

Moreover, due to the rapid rise in internet and mobile video demands, today video content providers face the challenge of providing their content in a myriad of codecs and resolutions. This process is made complex by the difference in capabilities of the target devices. As a result, without the use of transcoding, a content provider would have to store each and every video in multiple formats. Considering, the huge growth in the user-generated and other videos on the internet, this would be utterly infeasible. Instead, the best idea for content providers would be to store all videos in a single format with best quality and transcode it to other formats on the fly as and when required. But this requires that the transcoding algorithms are fast enough to perform the process on-demand.

1.2 Scope For Improvement

What coding standards specify is the decoding process and the bitstream syntax of the compressed video. But the encoding process or how to produce the compressed video complying to the standard is not standardized. This leaves a scope for innovation in the development of the encoding algorithm. One of the key issues in developing such an algorithm is the trade-off between computational complexity
Chapter 1. Introduction

of encoding and resulting video quality. Using complex encoding options may lead to highly efficient compression but at the cost of higher coding complexity. On the other hand, using simple encoding options may lead to speedup in compression time, but the quality will be adversely affected. Hence, design of algorithms to reduce the computational complexity without sacrificing the quality too much has recently become an active area of research.

Coding complexity can be significantly reduced by using classification techniques to replace expensive calculations involved in macroblock mode-decision by a classifier. Since coding complexity is directly related to power consumption, this technique also results in reduction in power consumption on resource-constrained devices such as mobile phones.

1.3 Problem Definition

Video Encoding is the method of compressing a raw video stream into a compressed video using a standard encoder. Video transcoding, on the other hand, is the process of decoding and recoding digital video from one format to another. For instance, converting from MPEG-2 to MPEG-4 is a transcoding process.

Among the current video codecs, H.264/AVC is the latest and most complex encoding standard. Hence, we focus on encoding/transcoding to H.264/AVC. Notably, the macroblock(MB) mode & prediction mode decision calculation is the most computationally expensive phase of encoding/transcoding to H.264. For each MB of each frame of video, an optimal block-size and prediction direction is calculated during this phase. There are various block sizes ranging from 16x16 to 4x4 as well as various prediction directions like DC, Horizontal, Vertical etc are available in H.264. Typically, these mode-decisions are performed by brute-force approach, i.e., for each block size & each prediction direction, the Rate-Distortion (RD) cost is calculated and the particular mode which incurs the minimum RD cost is selected for encoding/transcoding.

We want to approximate these mode-decisions using various MB features, so as to eliminate the need for brute-force calculations. A good approximation will ensure speedy encoding to H.264/AVC without sacrificing the resulting video quality and amount of compression. We achieve this approximation by learning a classifier.

We formally define the problem as follows:
Given video features for each macroblock, whose calculation must not be computationally expensive, predict the macroblock coding mode as given by a reference software.

1.4 Organization of Report

The rest of the report is organized as follows: In chapter 2, we review the concepts related to Video Coding & H.264 standard and review the related work. In chapter 3, we present the shortcomings of the state-of-the-art method and use these ideas to derive innovative Chance constraints based formulation for solving this problem, which is the main contribution of our work. Chapter 5 outlines the experimental results obtained using this novel approach.
Chapter 2

Background & Related Work

2.1 Background

In this section, an overview of related video coding fundamentals is presented.

2.1.1 Video Coding

Redundancy Reduction

Video Compression is performed by exploiting the inherent redundancies present in the video signals. The various video codecs are based on the three fundamental redundancy reduction principles:

1. Spacial Redundancy Reduction: To reduce the spacial redundancy among the pixels within frames by employing intra-coding and transform coding.

2. Temporal Redundancy Reduction: To remove similarities between successive pictures, by coding their differences instead of the frames themselves. This is called inter-coding. A motion-vector may also be used to signify a moving object.

3. Entropy Coding: To reduce the redundancy between the compressed data symbols, using variable length coding techniques.

Macroblock

During the compression process, each frame is divided in blocks of a fixed size (16x16 for example) or one of the possible sizes (based on the codec standard), which is called a Macroblock. Each macroblock is encoded separately with the aim of removing the above-mentioned redundancies. Based on the type of redundancies within a macroblock, the encoder may chooses to encode it using Inter-coding, or intra-coding or not to code it at all (if the region is non-changing).
Luma & Chroma Components

The intensity of a video signal is represented by two components:

1. Luminance(Y): Luma for short, represents the brightness of the picture
2. Chrominance(CbCr): Chroma for short, represents the colour information of the picture

2.1.2 H.264 Standard

A major goal while designing the H.264 (or MPEG-4 AVC) codec was to create a standard capable of providing good video quality at substantially lower bit rates than previous standards. We overview the features introduced to achieve this goal which are related to our problem here.

Inter-Frame Coding

As in previous standards starting from H.261, H.264 also uses block-based motion compensation. A major difference though, is the support for a number of block sizes. It supports block sizes ranging from 16x16 down to 4x4 with many options between the two. The luminance component of each Macroblock (MB) can be split up in 4 ways: 16x16, 16x8, 8x16 or 8x8. Each of the subdivided region of the MB called a partition. If 8x8 size is chosen, each of the four 8x8 MB can be further divided in 4 ways: 8x8, 8x4, 4x8 and 4x4 known as sub-MB partitions. Each chroma component is partitioned in the same way as the luma component, except that the horizontal and vertical resolutions are halved.

A separate motion-vector is required for each partition or subpartition. Each motion-vector must be coded and transmitted. In addition, the choice of partition(s) must be encoded in the compressed bit stream. Choosing a large partition size would mean that small number of bits are required to code the motion-vectors and the type of partition(s). On the other hand, choosing a small partition size may give lower energy residual after motion compensation but requires more number of bits to encode the motion-vectors and the choice of partition(s). Hence, in general, large partition size is appropriate for homogeneous areas of a frame and small partition size is appropriate for areas with high detail.

Intra-Frame Coding

H.264 also supports Intra-frame coding within all frames. An MB can make use of either 4x4, 8x8 or 16x16 block sizes for intra-frame prediction. There are nine prediction directions each for 4x4 and 8x8 block size and four prediction directions for 16x16 block size.
2.1.3 Choice of Macroblock Mode

Though the support for multiple block sizes and prediction modes significantly improves the compression efficiency, it adds a huge computation cost to choose the appropriate mode for each MB. Typically, the encoder tries all possible MB modes and chooses the one which minimizes the Rate-Distortion (RD). Hence, novel approaches are needed to perform this operation more efficiently without sacrificing compression efficiency.

2.2 Related Work

A lot of work has gone into the problem of improving the mode-decision speed by using heuristic-based approaches. Most of these approaches are based on limiting the number of prediction modes to be evaluated. In one approach, reported in [1], only the most probable prediction modes are evaluated resulting in reduced complexity. Prediction complexity reduction approach proposed in [2], computes an edge histogram to locate a spatial edge selects the prediction mode based on the edge angle.

Recently, Hari Kalva et al. have proposed a new approach for solving this problem using machine learning to approximate the mode-decision process by a classifier. Their work is presented in [3], [4], [5], [6], [7]. This approach achieves a much better speedup compared to the previous methods with negligible loss in compression efficiency. We summarize their approach here.

2.2.1 Basic Approach

The overall approach is summarized in figures 2.2.1 & 2.2.1. The basic idea is to determine the coding decisions such as the MB size and prediction mode decisions that are computationally expensive
Chapter 2. Background & Related Work

Figure 2.2: Low complexity encoder using decision trees

using easily computable features derived from input video. The input video is compressed in case of transcoding problem and is uncompressed (raw) in case of encoding problems. Machine learning is used to build a tree using such features. The coding decisions as given by the reference encoder software are used as the class labels for training. Thus, once the tree is built, the coding mode decisions which were previously performed by cost-based methods by evaluating all possible options are replaced with a decision tree lookup. In essence, decision tree lookup in software can be implemented by simple if-else statements, which require negligible processing.

The key issues to be considered are:

1. Selecting the training set

2. Selecting proper attributes/features

3. Choosing a classifier

4. Performance evaluation criteria

Selecting The Training Set

The training set has to be large enough to build a proper decision tree. At the same time, it cannot be too large; because that will result not only in overfitting but also in a large-size decision tree. Now, most of the video encoding mode decisions are based on similarity of the block being encoded to those that are previously coded. For example, in Intra-prediction, the mode is determined by the similarity of pixels in current block to the neighbouring blocks in current frame; while in Inter-prediction, the mode is decided by similarity between current block and corresponding block in previously coded frames. Thus
a set of frames having various amounts of similarity values are enough to create a good training set. As the encoding is typically performed on a block-by-block basis, the training set should have a sufficient number of macroblocks to make a good classification model. Such a training set is selected empirically.

The following standard video sequences were used for training and testing: akiyo, coastguard, container, flower, foreman, hall monitor, mobile, mother-daughter, silent, stefan and table-tennis.

**Feature Selection**

The features that are of interest in determining the coding modes are the similarity between current and previously coded block as well as the self-similarity of the current block. This reasoning of self-similarity comes from the fact that the coding mode depends upon whether the MB is homogeneous or of high detail. Simple metrics such as mean and variance of a macroblock can be used to characterize the MB coding mode. But as we have a large number of coding modes in H.264, we have to use more sophisticated features in addition to these for proper classification. For a specific problem, the features are selected based on experimentation and performance.

Moreover, use of more complex features improves the performance, but increases the computational complexity, and vice versa. Hence, we have to get a good trade-off between performance and speed.

**Classifier Selection**

For our problem, computational complexity reduction is the main goal. Additionally, the class labels, i.e. the coding mode selected by a reference encoder is available. Hence, *supervised learning* is more appropriate.

Decision tree (C4.5) is selected as the classifier, which not only performs well on large class of problems, but has some inherent advantages for our problem. When we convert a decision tree to code, it is essentially a set of rules in the form of *if-else statements*. Such statements take negligible computation time, which exactly is our aim.

**Performance Evaluation Criteria**

The performance of video coding and transcoding is evaluated using an R-D curve (Rate-Distortion curve) that shows the measure of distortion at a give bitrate. R-D curve only shows the coding efficiency but does not provide the information about computational complexity. The computational complexity is represented by the percentage of reduction in encoding/transcoding as compared to a reference method. The aim is to achieve R-D performance as close to reference method as possible, with significant reduction
in encoding/transcoding time.

Next we see how this approach is used in a transcoding and an encoding problem.

2.2.2 MPEG-2 to H.264 Inter-frame Transcoding

Training Set

It was found that the sequences that contain regions varying from homogeneous to high detail serve as good training sets. By empirical experiments, training based on flower garden sequence was found to give good classification performance over large number of other video sequences.

Classification

H.264 supports these inter-frame block sizes: 16x16, 16x8, 8x16, and 8x8. Further an 8x8 partition can be subdivided into 8x8, 4x8, 8x4 or 4x4 size sub-partitions. We perform this classification using a hierarchical decision tree consisting of 3 decision trees as shown in figure 3. Each node of this hierarchical tree is a decision tree in itself. For example, node1 classifies the MB mode into one of skip, Intra, Inter 16x16 and Inter 8x8; node2 further classifies 16x16 into one of 16x16, 16x8 and 8x16 etc.

Feature Set

For each macroblock, following features are used in various decision trees: (i) 16 means and 16 variances of the 4x4 motion estimation residual sub-blocks. (ii) the MB coding mode in MPEG-2 (iii) the CBPC in MPEG-2 The class label, the one we are trying to understand, is the H.264 coding mode, which
can be either of skip, Intra, Inter 16x16 or Inter 8x8 with possible further divisions as seen in previous classification section.

2.2.3 H.264 Intra-Coding

Training Set

After extensive experimentation, training set was selected by verifying the results from training one sequence by other sequences. Flower and mobile were found to be the best sequences to provide good generalization performance. In Intra-MB coding in H.264, we have 9 prediction modes for 4x4 blocks and 4 modes for 16x16 blocks. As the number of possible 4x4 intra MB modes are more than possible 16x16 modes, more frames to create training set for 16x16 modes are used.

Classification

The baseline profile of H.264 supports Intra MBs to be coded in 16x16 or 4x4 sizes. Further, 4x4 size has 9 prediction modes and 16x16 profile has 4 prediction modes. Thus we have a total of 13 classes. This classification problem is solved by using a hierarchical decision tree as in previous transcoding problem (the figure, being complex, is not included due to space constraints). Each node is a decision tree performing a binary classification. The easiest decision, that is, the decision of using a 16x16 mode Vs a 4x4 mode is performed at the topmost node of the tree. Then, the left subtree evaluates the Intra 16x16 prediction modes, while right subtree evaluates 4x4 modes. Thus, total four decision trees are hierarchically used to predict one of the four Intra 16x16 modes and nine trees are used to predict one of the nine Intra 4x4 modes.

Feature Set

The set of features used for each decision tree within the hierarchical tree varies. Top level decision, which is primarily based on the homogeneity of the MB, is performed only using the mean of the MB and variance of 16 4x4 sub-blocks of the MB. For lower levels, more complex features have to be used.

The following are some of the features used: (i) 16 means and 16 variances of the 4x4 blocks of an MB, (ii) the variance of top/bottom rows and right/left columns of an MB, (iii) the difference of means of the bottom row of top MB and bottom row of current MB, (iv) the difference of means of right column of left MB and right column of current MB.

The class label is the corresponding H.264 coding mode as determined by the standard reference software JM 14.2.
Chapter 3

Chance-Constrained Programming Approach

3.1 Observations on State-of-the-art Method

3.1.1 Observations

We constructed decision trees using state-of-the-art approach presented in chapter (2) for all nodes of the hierarchical decision tree for H.264 Intra-mode decision problem depicted in figure (3.3) according to the parameters and specifications specified by the authors. Using these decision trees, we tested the training set cross-validation accuracy as well as the test-set accuracy using various test video sequences for each tree node, to serve as the base for comparison to our approach. It was observed that some of the video sequences resulted in very poor classification accuracy for certain nodes of the hierarchical tree. Table 1 shows these results specifying on which sequences which node of the tree performed poorly. In the table, 'N' denotes the node of the decision tree, which are numbered from 1 to 13.

3.1.2 Notes

As can be clearly seen, there is significant scope for improvement in the classification accuracy. This is especially true for some of the nodes, which are in fact at lower levels of the hierarchy, where classification is more challenging.
<table>
<thead>
<tr>
<th>N</th>
<th>Training Acc</th>
<th>Test Seq.</th>
<th>Test Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>98.3</td>
<td>akiyo_cif</td>
<td>78.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>table_cif</td>
<td>67.83</td>
</tr>
<tr>
<td>2</td>
<td>76.67</td>
<td>akiyo_cif</td>
<td>62.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>flower_qcif</td>
<td>59.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mobile_720x480</td>
<td>62.42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>silent_cif</td>
<td>61.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stefan_cif</td>
<td>47.13</td>
</tr>
<tr>
<td>3</td>
<td>76.21</td>
<td>mobile_720x480</td>
<td>58.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>silent_qcif</td>
<td>29.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>table_qcif</td>
<td>57.96</td>
</tr>
<tr>
<td>4</td>
<td>78.34</td>
<td>akiyo_qcif</td>
<td>55.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hall_monitor_cif</td>
<td>59.42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>silent_qcif</td>
<td>56.5</td>
</tr>
<tr>
<td>6</td>
<td>71.18</td>
<td>coastguard_cif</td>
<td>54.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>table_cif</td>
<td>54.36</td>
</tr>
<tr>
<td>7</td>
<td>72.19</td>
<td>akiyo_qcif</td>
<td>55.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>foreman_qcif</td>
<td>55.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>silent_cif</td>
<td>55.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stefan_qcif</td>
<td>52.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>table_qcif</td>
<td>43.13</td>
</tr>
<tr>
<td>8</td>
<td>73.33</td>
<td>akiyo_qcif</td>
<td>51.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td>coastguard_cif</td>
<td>25.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>flower_720x480</td>
<td>48.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mother_slaughter_cif</td>
<td>48.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stefan_cif</td>
<td>22.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stefan_qcif</td>
<td>34.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stefan_qcif</td>
<td>33.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>table_qcif</td>
<td>50</td>
</tr>
<tr>
<td>9</td>
<td>62.31</td>
<td>foreman_qcif</td>
<td>50.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hall_monitor_cif</td>
<td>50.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mother_slaughter_cif</td>
<td>47.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>silent_qcif</td>
<td>47.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>table_qcif</td>
<td>48.22</td>
</tr>
<tr>
<td>10</td>
<td>64.76</td>
<td>akiyo_qcif</td>
<td>53.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>foreman_cif</td>
<td>55.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>table_cif</td>
<td>54.21</td>
</tr>
<tr>
<td>11</td>
<td>56.05</td>
<td>coastguard_cif</td>
<td>35.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>container_qcif</td>
<td>43.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>foreman_qcif</td>
<td>40.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hall_monitor_cif</td>
<td>33.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hall_monitor_qcif</td>
<td>28.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>silent_qcif</td>
<td>47.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stefan_qcif</td>
<td>43.93</td>
</tr>
<tr>
<td>12</td>
<td>74.2</td>
<td>container_qcif</td>
<td>44.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mother_slaughter_qcif</td>
<td>55.6</td>
</tr>
<tr>
<td>13</td>
<td>73.18</td>
<td>hall_monitor_qcif</td>
<td>55.23</td>
</tr>
</tbody>
</table>
3.2 Chance-Constrained Programming Method

3.2.1 Overview

Figure (3.1) and (3.2) show what we want to achieve in a nutshell. For H.264 video coding, macroblock mode decision step has to be performed for each macroblock of each frame in a video. In standard reference encoder JM14.2, this is achieved by first calculating the Rate-Distortion cost of each possible MB mode-prediction mode pair and then selecting the one achieving least RD cost. We want to replace this brute-force calculation by a classifier.

Figure (3.1) depicts the procedure for training the classifier. For each MB of uncompressed training video, various mean-variance pairs of intensity values are extracted. Also, the optimal MB mode & prediction mode for that MB are obtained using standard reference encoder. Using mean-variance pairs as features and optimal MB mode-prediction modes as class labels, the chance-constrained programming-based classifier is trained.

Figure (3.2) depicts how this classifier can be used for low-complexity H.264/AVC encoding. For given
uncompressed video, the MB features are first extracted. Using these features, the classifier predicts the MB mode & prediction mode for the MB.

Intra MBs in H.264 are coded as Intra 16x16, Intra 4x4 for the baseline profile. Intra 16x16 has 4 prediction modes and Intra 4x4 has 9 prediction modes. Baseline profile encoders typically evaluate both Intra 16x16 and Intra 4x4 modes and the associated prediction modes before making MB mode decisions. In the proposed approach, we separate the Intra MB mode and Intra prediction mode decisions. Intra MB mode is determined as Intra 16x16 or Intra 4x4 without evaluating any prediction modes. The appropriate prediction modes for the MB mode are then determined. Since the MB mode is determined first, our approach right away eliminates the computation of any prediction modes for the MB mode that is not selected. If the MB mode is determined to be Intra 16x16, there is no need to evaluate any prediction modes for the 4x4 sub-blocks. This is achieved using a hierarchical tree as shown in figure (3.3) where each node of the tree is a binary classifier.

### 3.2.2 Feature Selection

The features that are of significant value when taking Intra-coding mode decision in H.264 are those which capture the similarity between current Macroblock (MB) and neighboring blocks as well as the ones which capture the self-similarity of the MB. The importance of self-similarity comes from the fact that coding mode depends upon whether the MB is homogeneous or contains highly detailed information. For instance, to minimize Rate-Distortion, optimum coding mode size of a homogeneous MB would be
Figure 3.3: Binary tree used in Intra MB coding

large and vice versa. Metrics such as mean and variance of intensity values of an MB provide a good measure of self-similarity. We can also partition each 16x16 MB into 16 sub-macroblocks (SMBs) of size 4x4 and then we can take the mean and variance of intensity values within each SMB; which will signify how different are the intensity values and homogeneity in various parts of the MB. The similarity with neighboring MBs can be captured using the difference in intensity values between current and neighboring MBs. These differences can again be in terms of mean and variance of pixel wise differences of intensities.

3.3 Mathematical formulation

In view of the above discussion the problem of predicting the optimal mode from a video is essentially that of computing $f$ such that

$$y = f(X) \ X \sim (\mu, \Sigma)$$
where \( \mathbb{E}(X) = \mu \) and \( \mathbb{E}(X - \mu)(X - \mu)^\top = \Sigma \)

g denotes the mode, and \( X \) denotes vector of random variables with each random variable representing an attribute.

Note that \( X \sim (\mu, \Sigma) \) denotes that \( X \) can have any multivariate distribution but the mean should be \( \mu \) and covariance should be \( \Sigma \). We propose to model the distribution in \( X \) by chance constraints, which when relaxed via Chebychev Cantelli inequality leads to a Second Order Cone program (SOCP). This approach was used previously to classifiers which are robust to uncertainty in data and to train large scale classifiers [8]. For a set of references on application of Chance constraints to uncertain data please see [8].

This is an instance of multi-category problem as \( y \) can take any one of finite number of modes. Following [3] we reduce the multi-category problem to a series of binary decision problems. The final computation can be understood as that of traversing a hierarchical tree, see Fig. 3.3, where at each node a binary decision is taken.

Now, we study the problem of designing linear classifiers for each node of the tree. For a given node in the tree let \( y \in \{1, -1\} \) be the output on an observation \( X \). If \( X \) was deterministic then \( y = \text{sign}(w^\top X + b) \) could be a useful decision function. For a properly chosen \( \{w, b\} \) it maybe desired that

\[
y_i(w^\top X_i + b) \geq 1 \quad i = 1, \ldots, N
\]

where \( N \) is the number of training examples.

However in the context of our problem, \( X \) is stochastic, and hence such constraints are absolutely impossible to satisfy. As a way out, we can alternately require that the probability of the event \( y_i(w^\top X_i + b) \geq 1 \) should be high thus motivating constraints of the form

\[
P(y_i(w^\top X_i + b \geq 1)) \geq \eta, \quad X_i \sim (\mu_i, \Sigma_i)i = 1, \ldots, N \tag{3.1}
\]

where \( \eta \) is a user-defined parameter, which lower-bounds the probability of correct prediction.

These kind of constraints are called Chance constraints.

Using the chance constraints, the problem of learning the classifier \( \{w, b\} \) cab be posed as the following
optimization problem:

\[
\min_{w,b,\xi} \sum_{j=1}^{N} \xi_j \quad \text{s.t.} \\
P(y_j(w^\top X_j + b \geq 1)) \geq \eta \quad j = 1, \ldots, N \\
\xi_j \geq 0, \quad j = 1, \ldots, N \\
\|w\|_2 \leq W
\] (3.2)

where, the constraint \(\|w\| \leq W\) is required from a generalization perspective and the slack variables \(\xi_i\) are introduced to keep the problem feasible if \(\eta\) is set close to 1.

Modelling Chance constraints for an arbitrary distribution, as is the case here, is difficult and is an active area of research in optimization theory and machine learning. An interesting tractable alternative for such kind of problems can be provided by the Chebychev-cantell inequality. We will use the following theorem proved in [8].

**Theorem 3.3.1** Let \(w, X, b, \mu, \Sigma\) be as defined previously. The constraint

\[P(y(w^\top X + b) \geq 1) \geq \eta, X \sim (\mu, \Sigma)\]

can be enforced by satisfying the inequality

\[y(w^\top \mu + b) \geq 1 + \kappa\|\Sigma^{\frac{1}{2}}w\|_2\]

where \(\Sigma\) is assumed to be an invertible positive definite matrix and \(\kappa = \sqrt{\frac{\eta}{1-\eta}}\).

**Proof** The proof is similar to that in [8] and follows from a direct application of multivariate generalization of Chebychev-Cantelli inequality.

By using Chebychev-Cantelli inequality,

\[P(y(w^\top X + b) \geq 1) \leq \frac{(w^\top \mu + b - 1)^2_+}{(w^\top \mu + b - 1)^2_+ + w^\top \Sigma w}\] (3.3)

where \((x)_+ = \max(x, 0)\)

Now considering the fact that we have assumed covariance matrix \(\Sigma\) to be diagonal, we can write
Chapter 3. Chance-Constrained Programming Approach

\[ \Sigma = \Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}} \]

Hence, from equation (3.3),

\[ P(y(w^T X + b) \geq 1) \leq \frac{(w^T \mu + b - 1)^2}{(w^T \mu + b - 1)^2 + \|\Sigma^{\frac{1}{2}} w\|_2^2} \]

Now using the constraint \( P(y(w^T X + b) \geq 1) \geq \eta \) we have

\[ \eta \leq \frac{(w^T \mu + b - 1)^2}{(w^T \mu + b - 1)^2 + \|\Sigma^{\frac{1}{2}} w\|_2^2} \]

Hence,

\[ y(w^T \mu + b) \geq 1 + \kappa \|\Sigma^{\frac{1}{2}} w\|_2 \] (3.4)

where \( \kappa = \sqrt{\frac{\eta}{1-\eta}} \).

Using above theorem to replace the probabilistic constraints in (3.2), one can motivate the following optimization problem for computing \( \{w, b\} \).

\[
\begin{align*}
\text{min}_{w, b, \xi} & \sum_{j=1}^{N} \xi_j \\
\text{s.t.} & \\
y_j(w^T \mu_j + b) & \geq 1 - \xi_j + \kappa \|\Sigma_j^{1/2} w\|, \quad j = 1, \ldots, N \\
\xi_j & \geq 0, \quad j = 1, \ldots, N \\
\|w\|_2 & \leq W
\end{align*}
\]

(3.5)

where \( \kappa = \sqrt{\frac{\eta}{1-\eta}} \).

In our problem, we noticed that the variances of individual attributes differ significantly. We thus considered a scaled random vector \( Z_j = \Sigma^{-1/2}_j X_j \) leading to the fact that \( E(Z_j) = \Sigma^{-1/2}_j \mu_j \) and \( \text{cov}(Z_j) = I \); where \( I \) denotes the identity matrix.

Now, using \( Z_j \) in place of \( X_j \) in (3.5) (note that this is possible because, we have considered \( X_j \) to be
any random vector when deriving (3.5), we get

$$
\min_{w,b,\xi} \sum_{j=1}^{k} \xi_j \quad \text{s.t.}
$$

$$
y_j(w^T \Sigma^{-\frac{1}{2}} \mu_j + b) \geq 1 - \xi_j + \kappa ||I^{1/2}w||_2, \\
\xi_j \geq 0, \quad j = 1, \ldots, N \\
||w||_2 \leq W 
$$

(3.6)

Hence, we obtain the following equivalent formulation:

$$
\min_{w,b,\xi} \sum_{j=1}^{k} \xi_j \quad \text{s.t.}
$$

$$
y_j(w^T \mu_{j,\text{scaled}} + b) \geq 1 - \xi_j + \kappa W; \\
\xi_j \geq 0, \quad j = 1, \ldots, N \\
||w||_2 \leq W 
$$

(3.7)

where $\mu_{j,\text{scaled}} = \Sigma^{-\frac{1}{2}} \mu_j$

Please note that this is an instance of an SOCP problem which could be solved efficiently.

### 3.4 Geometric Interpretation

Geometric interpretation of (3.7) turns out to be classifying most of the spheres - with center being the mean value and radius $\kappa$ - correctly, as opposed to classifying points in the usual case. This can be seen as follows:

Let the set of points lying in sphere $B$ with center $c$ and radius $r$ be denoted by

$$B(c,r) = \{x| (x-c)^T(x-c) \leq r^2 \}.$$

Now, the problem of classifying points in $B(\mu, \kappa)$ correctly (including the case of outliers) is:

$$w^T x + b \geq 1 - \xi, \quad \forall x \in B(\mu, \kappa)$$

(3.8)

The constraints in (3.8), which imply that the whole sphere should be on the positive halfspace of
the hyperplane $w^T x - b = 1 - \xi$, can be replaced by a single constraint:

$$w^T x_0 + b \geq 1 - \xi$$

(3.9)

where $x_0 = \arg \min_{x \in B(\mu, \kappa)} (w^T x - b)$

which specifies that the point nearest to the hyperplane $w^T x - b = 1 - \xi$ should be on the positive halfspace.

Now, $x_0$ can be found as follows: Drop a perpendicular to the hyperplane from center of the sphere. The point at which the perpendicular intersects with the sphere is $x_0$. Using this geometry and noting that the sphere has center $\mu$ and radius $\kappa$, we get $x_0 = \mu - \kappa \frac{w}{||w||}$

Now, $x_0 \in B(\mu, \kappa)$. Hence, from (3.8),

$$w^T x_0 + b \geq 1 - \xi$$

(3.10)

Putting value of $x_0$ in (3.10), we get

$$w^T \mu + b \geq 1 - \xi + \kappa ||w||, \quad j = 1, \ldots, k$$

(3.11)

which is similar in form to the Second Order Cone (SOC) constraint in (3.7) considering that we have considered positive half-space here ($y = 1$).

This is geometrically illustrated in Figure (3.4). All dotted spheres have positive, while solid spheres have negative labels. Note that except the solid sphere intersecting the hyperplane, all spheres are classified correctly in this case.

### 3.5 Prediction and Complexity

It is important that procedure for inferring the correct mode should be extremely fast. We use the following decision rule for an instance $X$:

$$y = \text{sign}(w^T \mu_{scaled} + b)$$

(3.12)
where $E(X) = \mu$, $E(X - \mu)(X - \mu)^\top = \Sigma$ and $\mu_{\text{scaled}} = \Sigma^{-\frac{1}{2}} \mu$

We have also assumed that the attributes are un-correlated implying that $\Sigma$ is a diagonal positive definite matrix. As $\Sigma$ is a $d \times d$ matrix, the computation of $\mu_{\text{scaled}}$ is $O(d)$. Again at each node, the computation is $O(d)$ given that $\mu_{\text{scaled}}$ is precomputed. Thus the inference time complexity is $O(hd)$ where $h$ is height of the tree in Fig. 3.3. Note that this is competitive with previously suggested decision trees\[3\] whose complexity is $O(ht)$ where $t$ is the number of rules at each node.

### 3.6 Implementation

Our mode-decision problem is a multiclass problem consisting of 13 possible class labels: 4 prediction modes for block size of 16x16 and 9 prediction modes for block size of 4x4. As in the approach used in [4], we solve this problem using a set of 12 binary sub-classifiers arranged as a hierarchical tree as shown in figure (3.3). At each node we solve the formulation in (3.7) using an open source SOCP solver Sedumi\(^1\) to obtain model parameters $w$ and $b$. Note that in our experimentation we have used $W = 1000$, $\eta = 0.8$, obtained by tuning on the dataset. For testing, firstly the required features of the incoming macroblock are computed. Then, the mean values are scaled using the equation $\mu_{\text{scaled}}^j = \Sigma^{-1/2} \mu^j$. Note that this computation is done only once per macroblock and then reused at all levels of the tree. Note also that, as $\Sigma$ is a diagonal matrix, computation of $\Sigma^{-1/2}$ as well as matrix multiplication are very low cost operations. After this initial computation, each sub-classifier takes a decision based

\(^1\)Available at http://sedumi.ie.lehigh.edu/
on \( \text{sign}(w^T \mu_{scaled}^j + b) \) where \( \mu_{scaled}^j \) consists of only those mean values which are relevant to sub-classification at this level of tree. Note that one major task in our problem is the speed-up achieved in encoding as compared to standard reference encoder (JM 14.2). This is clearly achieved here, as all operations required during testing are low-cost operations as can be seen in the last paragraph. The experimental results support this claim as well. Another major task is the RD performance in terms of bitrate penalty. This highly depends upon the performance of the classifier. Experimental results suggest that RD performance of the approach is on par with the machine learning approach, that is, the bitrate penalty of the video encoded using this technique is in acceptable range.
Chapter 4

Experimental Results

In this section, we present the various experiments performed and results obtained.

We implemented the hierarchical tree shown in figure (3.3) in order to evaluate the performance of the chance constraint based classifier. Test sequences are encoded with all Intra frames.

For each node in the hierarchical tree, a chance-constraint based classifier was trained by implementing the mathematical formulation presented in section 3.3 as a matlab code. At each node, the set of features and training video sequences were experimentally chosen to get best empirical results across a number of test video sequences. We used standard video sequences such as akiyo, coastguard, container, flower, foreman, hall, mobile, mother & daughter, silent stefan and tennis at CIF and QCIF resolution for training and testing the classifier models. The QP values taken were 20, 24, 28, 32 and 36.

These classifier models and video feature extraction code were implemented in reference H.264 encoder JM14.2 so as to replace the existing brute-force mode-decision calculations.

Finally, the calculation of measures such as Rate-Distortion cost, time required to encode, PSNR (Peak Signal-to-Noise Ratio) were implemented within the encoder code to evaluate the performance. Note that, the time required to calculate performance measures was not included so as to get accurate encoding time.

All the experiments were performed on a machine with Intel 64-bit Quad-core processor and 4GB RAM.

Fig. 4.1 (a)-(c), shows the RD performance with the proposed approach and Fig. 4.2 (a)-(c), shows the
RD performance with the decision tree based approach in [9]. The process of mode prediction is complex and introduces a small loss in PSNR as shown in Fig. 4.1 (a)-(c). The maximum PSNR loss suffered has been less than 1 dB in both cases. The reduction in encoding time of the proposed method is about 1/5 of the reference encoding time shown in Fig. 4.1 (b)-(d). Fig. 4.2 (b)-(d) shows the time complexity measurement of the H.264 Intra MB coder developed by decision tree based approach in [9] for the same sequences used in Fig. 4.1. From both of these figures, it is clear that the performance of both of these classifiers is almost similar. However, exact results suggest that the proposed approach results in slightly better RD performance as compared to decision tree based approach in [9], but time-complexity reduction in slightly better (about 20%) in their approach than proposed approach (about 18%). This means that with almost the same time-complexity, proposed approach provides slightly better video quality.

Figure 4.1: (a)(c) RD performance of the proposed H.264 Intra frame encoder with Chance Constrained classifier; (b)(d) Time complexity reduction.
Figure 4.2: (a)(c) RD performance of the H.264 Intra frame encoder with Decision Tree classifier; (b)(d) Time complexity reduction.
Chapter 5

Conclusion

We have considered a practical problem of making the encoding/transcoding of videos more time-efficient without sacrificing on the quality. Noting that features useful in this application are essentially random variates with specified mean and co-variance values, we proposed an innovative chance-constraint based classification approach to the macroblock mode prediction problem. This approach is naturally suited for the problem, which is borne out by its superior empirical performance. Implementation of this technique into the standard JM14.2 encoder shows that it achieves a time-complexity reduction of about 20% as compared to the reference encoder JM14.2; while the RD performance is very close to the reference encoder.

Although, we have experimented on the Intra-frame prediction in H.264, we believe that the proposed method has the potential to reduce the time-complexity in Inter-frame encoding in H.264 as well as in other video standards, too.
Bibliography


