PREDICTIVE CODING TOOLS IN MULTI-VIEW VIDEO COMPRESSION

by

Jae Hoon Kim

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

December 2008

Dedication

To my family

Acknowledgements

First of all, I would like to thank Prof. Antonio Ortega for his advice, guidance, support and patience throughout the years I have been pursuing my Ph.D degree at the University of Southern California. I am also grateful to Prof. C.-C. Jay Kuo and Prof. Ulrich Neumann for their advices and comments in my defense. I would like to extend my gratitude to Prof. Alexander A. Sawchuk, Prof. Ramakant Nevatia and Prof. Karen Liu for serving on my Qualification Exam Committee.

I would like to thank Yeping Su, Peng Ying, Purvin Pandit, Dong Tian and Cristina Gomila for their advices and supports that I had during my invaluable internships with Thomson.

I would like to thank all the colleagues in Compression group for their friendships and useful discussions over life and research. I would like to thank Joaquin Lopez and Po-Lin Lai for enjoyable discussions and collaborations.

I thank all my friends for the moments I shared with them, which gave me rest, passion and energy to overcome all difficulties in my life and research. I will not forget the moments and coffees that I had with Wonseok Baek, Hyukjune Chung and In Suk Chong. And my special thanks to Young Gyun Koh for his lifetime friendship.

My family, I can not express my thank enough to them. Without their belief, support, endurance and love, this work could not have been even started, let alone finished. And Jung Yeun, my wife, thank you and I love you.

Table of Contents

List of Tables

List of Figures

Abstract

Multi-view video sequences consist of a set of monoscopic video sequences captured at the same time by cameras at different locations and angles. These sequences contain 3-D information that can be used to deliver new 3-D multimedia services. Due to the amount of data, it is important to efficiently compress these multi-view sequences to deliver more accurate 3-D information.

Since the captured frames by adjacent cameras have similar contents, cross-view redundancy can be exploited for disparity compensation. Typically both temporal and cross-view correlations are exploited in multi-view video coding (MVC), so that a frame can use as a reference the previous frame in time in the same view and/or a frame at the same time from an adjacent view, thus leading to a 2-D dependency problem. The disparity of an object depends primarily on its depth in the scene, which can lead to lack of smoothness in the disparity field. These complex disparity fields are further corrupted by the brightness variations between views captured by different cameras. We propose several solutions to solve these problems in block based predictive coding in MVC.

Firstly, the 2-D dependency problem is addressed in Chapter 2. We use the monotonicity property and the correlation between anchor and non-anchor quantizers to reduce the complexity in data collection of an optimization based on the Viterbi algorithm. The proposed bit allocation achieves 0.5 dB coding gains as compared to MVC with fixed QP.

In Chapter 3, we propose an illumination compensation (IC) model to compensate local illumination mismatches. With about 64% additional complexity for IC, 0.3-0.8 dB gains are achieved in cross-view prediction. IC techniques are extended to compensate illumination mismatches both in temporal and cross-view prediction.

In Chapter 4, we seek to enable compensation based on arbitrarily-shaped regions, while preserving an essentially block-based compensation architecture. To do so, we propose tools for implicit block-segmentation and predictor selection. Given two candidate block predictors, segmentation is applied to the difference of predictors. Then a weighted sum of predictors in each segment is selected for prediction. Simulation results show 0.1-0.4 dB gains as compared to the standard quad tree approach in H.264/AVC.

Chapter 1

Introduction

1.1 Multi-view Video

Since early 20th century when the first generation of television came into being, many novel technologies have been introduced (e.g., color, new types of displays, etc.). However, the main framework of video service has not been changed significantly in that the frames captured by the camera are edited to generate a single sequence that is delivered and displayed on a 2-dimensional (2-D) screen. With this monoscopic video system, 2-D scenes are regenerated and shown to users for the fixed viewpoint provided by the camera at each time instant.

The visual information of an object can be defined by the intensity and the location. The object intensities are represented by three color channels - R, G and B. The locations of the objects are defined by 3-D information - horizontal and vertical location and depth. In the captured 2-D frame where 3-D location of an object is projected, only horizontal and vertical information is delivered to users with three color channels, while depth information is not delivered explicitly.

In the real world, the depth of an object can be estimated by various depth cues. For example when moving head, objects that are closer will move farther across the

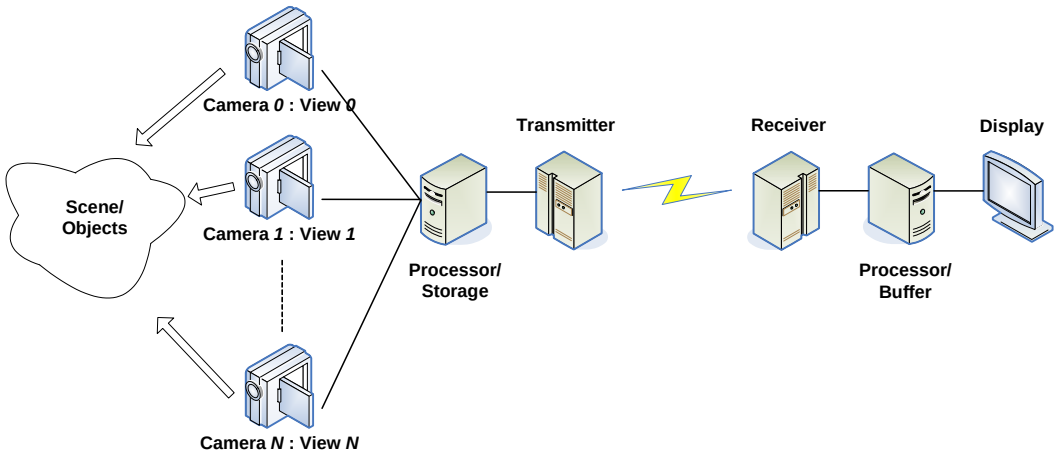


Fig. 1.1: End to end multi-view system

field of view and different scenes will be observed according to the displacement of head. This is called motion parallax, a monocular depth cue. Also the occlusions or the exposed areas can give information about which objects are closer. When both eyes are open and the head is not moving, each eye will see different images of the same scene. Stereo images from this binocular parallax are used to measure depth. Binocular parallax is the most important cue and depth information can be obtained even if all other depth cues are removed.

In the conventional monoscopic video system, only limited cues are available for depth estimation (e.g., from the occluded or disclosed regions of scenes). However, in order to deliver complete information to users and enable 3-D multimedia services, depth information needs to be transmitted or estimated accurately. Multi-view video systems are used for simultaneously capturing the scenes or objects with multiple cameras from different view points. In multi-view video, different perspectives by cameras for the same scenes or objects provide binocular parallax, from which depth information can be extracted, thus enabling 3-D multimedia services.

An end to end multi-view video system is depicted in Fig. 1.1. Multi-view sequences captured by an array of cameras are stored, processed and transmitted.

Received sequences are processed and displayed on 2-D or 3-D display devices. After sequences are reconstructed at the receiver, intermediate views can be interpolated to provide smooth transition and improved quality of display. Sequences can be displayed on conventional 2-D displays with view switching capability [16], or specially designed 3-D display devices can be used [28] for better 3-D perception.

In the sequence acquisition step, the number of views decides the range of 3-D scene and the quality of service e.g., the more cameras are used, the more accurate depth information will be, thus enabling improvements in the quality of interpolated views. However, the amount of data for the captured sequences also increases proportionally to the number of views. For example, transmitting uncompressed multi-view sequences with 8 views, 1280×720 resolution and 24 bits per pixel at 30 frames/sec requires 5.3 Gbps. Because of the increased amount of data in multi-view system, efficient coding of multi-view sequences is essential for the widespread use of services.

1.2 Applications of Multi-view Video System

There have been active research efforts on applications of multi-view video, especially for 3-D TV and free-viewpoint video. In [28], a 3-D TV prototype system is proposed which uses an array of 16 cameras, clusters of network-connected PCs and a multi-projector 3-D display. Two types of display, rear-projection screen and front-projection screen are implemented according to the location of projectors. Although blur is prominent on both types of display due to the crosstalk between subpixels of different projectors and light diffusion, the display reflects user's viewpoint and shows different images. In [10], a video-plus-depth data representation is proposed as a flexible solution to diverse 3-D display technologies. A depth map is

created from frames captured using stereo cameras or multiple monocular cameras and streams including N video sequences and a depth sequence are used to render M views. Depth image based rendering (DIBR) is proposed as a solution to 3-D reproduction.

For free-view point video, in [20] view generation methods are explained using ray-space approach [11]. For coding of multi-view video sequences, a group of GOP (GoGOP) structure is proposed in order to enable low delay random access, which is an extension of the group of picture (GOP) structure in standard video coding. In [16], a color segmentation-based stereo algorithm is used to generate high quality photo-consistent correspondences across views captured by high-resolution cameras. Scene depth is recovered by disparities and matting information is used at object boundaries to compensate depth discontinuities. A real time rendering system is described, which interactively synthesizes intermediate views.

In [31], panoramic video capturing, stitching and display is proposed to provide users individual control of viewing direction. A camera array is used to capture sequences and captured scenes are stitched, then displayed on head-mounted display with orientation tracker so that different scene can be displayed according to user's orientation. This system assumes that the user has fixed location but his/her viewing direction can rotate so that scenes around user can be viewed in 360 degree. This approach is different to multi-view system in that the viewing direction of user rotates at fixed location.

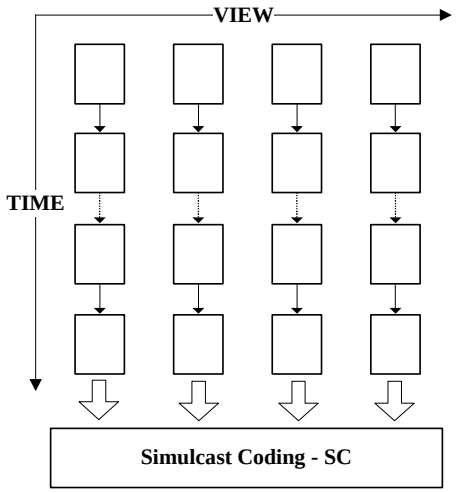
With the recognition that a multi-view video coding is a key technology for a wide variety of applications including 3-D TV, free viewpoint TV and surveillance, various topics related to multi-view video are covered in [12]. In [39], an overview of 3-D TV and free viewpoint video is given with related standardization activities in MPEG.

1.3 Contributions of the Research

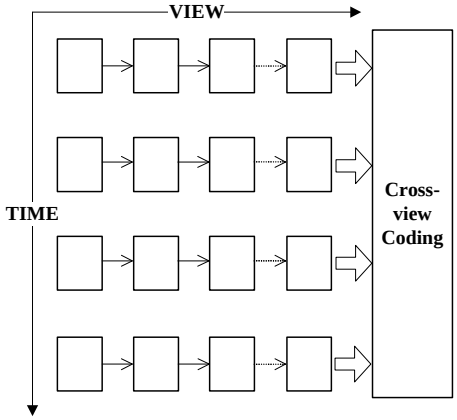
Design of multi-view video system involves multiple disciplines such as video coding, optics, computer vision, computer graphics, stereoscopic displays, multi-projector displays, virtual reality and psychology, in order to enable services that can bridge the gap between 2-D and full 3-D experience e.g., holographic display. In this work, we focus on providing efficient compression methods for multi-view video coding.

A straightforward approach for compression of multi-view video sequences would be to apply standard video coding techniques to each view independently. This simulcast (SC) approach allows temporal redundancy to be exploited using block-based motion compensation techniques as shown in Fig. 1.2(a). Since the captured frames by adjacent cameras have objects in common, cross-view redundancy could also be exploited in the form of disparity compensation as shown in Fig. 1.2(b). To achieve high coding gains, multi-view video coding exploits both temporal and cross-view redundancies. In Fig. 1.2(c), a multi-view video coding structure using both temporal and cross-view correlation is depicted. To facilitate random access, anchor frames are inserted at predefined time intervals. These anchor frames are encoded using only cross-view prediction.

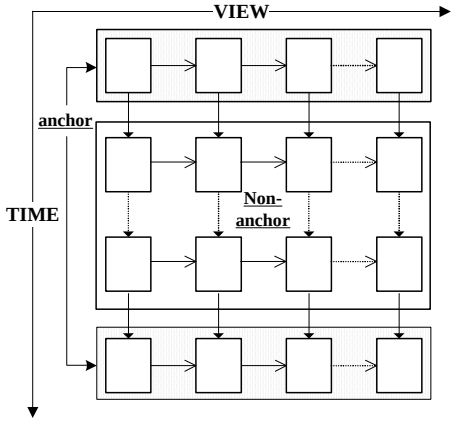
In block based predictive coding, the block most correlated to the current block is searched for in the previously encoded frame. Therefore the gains in coding efficiency mainly come from finding highly correlated blocks leading to residuals with high degrees of energy compaction. Applying a block based predictive coding to both temporal and cross-view prediction in MVC, the following problems are observed.



(a) Simulcast coding



(b) Cross-view coding



(c) An example of temporal and cross-view prediction in MVC

Fig. 1.2: Multi-view video coding structure

To avoid drift between encoder and decoder, the same frame should be used for prediction at encoder and reconstruction at decoder. Therefore reconstructed frames are used for the prediction of current frames, so that encoding of current frame depends on the quality of reconstructed frames. This dependency problem was addressed in [34] in monoscopic video coding. This one dimensional (time) dependency problem is expanded into two dimensional (time and view) problem in multi-view systems.

In cross-view prediction of multi-view video, imperfectly calibrated cameras give different brightness and focus to sequences at different views. Even if cameras are perfectly calibrated, differences in camera positions and orientations lead to differences in how certain objects appear in different views. The accuracy of disparity search is corrupted by these mismatches between views, which can lead to irregular disparity fields and degrade cross-view coding efficiency.

In block based motion/disparity compensation, block sizes used for compensation can be chosen to achieve a good trade-off between signaling overhead and prediction accuracy. However current quad-tree based motion compensation leads to motion boundaries that are not necessarily aligned with arbitrary object boundaries, which limits the accuracy of block-based compensation, even when small block sizes are chosen. Therefore, moving objects in motion compensation and objects at different depths in disparity compensation result in significant distortion in places where the object boundary is not aligned with the rectangular grid that can be represented by quad-tree.

The main contribution of this research is to provide new predictive coding tools to solve the problems described above and improve overall coding efficiency in multi-view video coding. In the proposed bit allocation scheme, we use the

monotonicity property and the correlation between anchor and non-anchor quantizers to reduce the complexity in data collection of the Viterbi algorithm, which was proposed in [34] for solving a dependency problem in standard video coding. To improve the accuracy of disparity search under brightness variation between views captured by different cameras, a local illumination compensation technique is proposed. Implicit block segmentation algorithm is proposed to find a match corresponding to arbitrary object boundaries while preserving a block based compensation architecture.

1.4 Organization of Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, we consider the bit allocation problem in MVC. A dependent coding technique using trellis expansion and the Viterbi algorithm (VA) is proposed, which takes into account dependencies across time and views. We note that, typically, optimal quantizer choices have the following properties: i) quantization choices tend to be similar for frames that are consecutive (in time or in view), ii) better quantization tends to be used for frames closer to the root of the dependency tree. We propose a search algorithm to speed up the optimization of quantization choices. Our results indicate 0.5 dB coding gains can be achieved by an appropriate selection of bit allocation across frames.

In Chapter 3, we propose a block-based illumination compensation (IC) technique for cross-view prediction in MVC. Models for illumination (brightness) mismatches across views are proposed and new coding tools are developed from the models. In IC, disparity field and illumination changes are jointly computed as part of the disparity estimation search. IC can be adaptively applied by taking

into account the rate-distortion characteristics of each block. By compensating the effect of mismatches, we improve the quality of references obtained via disparity search, which leads to coding gains of up to 0.8 dB.

In Chapter 4, we propose an implicit block based segmentation method to improve quality by using multiple predictors for each block. Given two candidate block predictors, segmentation is applied to the difference of predictors and the optimal predictor is selected in each segment. Implicit block segmentation is implemented in H.264/AVC as an additional inter block mode and achieves 0.1-0.4 dB gains as compared to the results obtained with only a hierarchical quad-tree.

In Chapter 5, conclusions and future work are discussed.

Chapter 2

Dependent Bit Allocation in Multi-view Video Coding

2.1 Preliminaries

To achieve coding gains in multi-view video coding (MVC), both temporal and cross-view correlation can be exploited using block-based predictive coding. Any such block-based predictive coding technique leads to dependencies, as quantization choices for one frame affect the achievable rate-distortion points for those frames that depend on it [34]. In Fig. 2.1, an MVC coding structure is shown with temporal and view indeces. Note that different types of coding dependencies arise depending on the coding scheme being used. In the simulcast case of Fig. 2.2(a), each view is coded independently, so only temporal dependency (1-D) within each view can be observed, similar to the monoscopic video case. Instead, Figs. 2.2(b) and 2.2(c) represent cases where the set of anchor frames are encoded in IPPP or IBBP modes. This introduces additional dependencies across views (2-D). For example, when encoding frame $V2T2$, reconstructed frame $V2T1$ is used as a reference, and in turn $V2T1$ uses frame $V1T1$ as a reference (see Fig. 2.1).

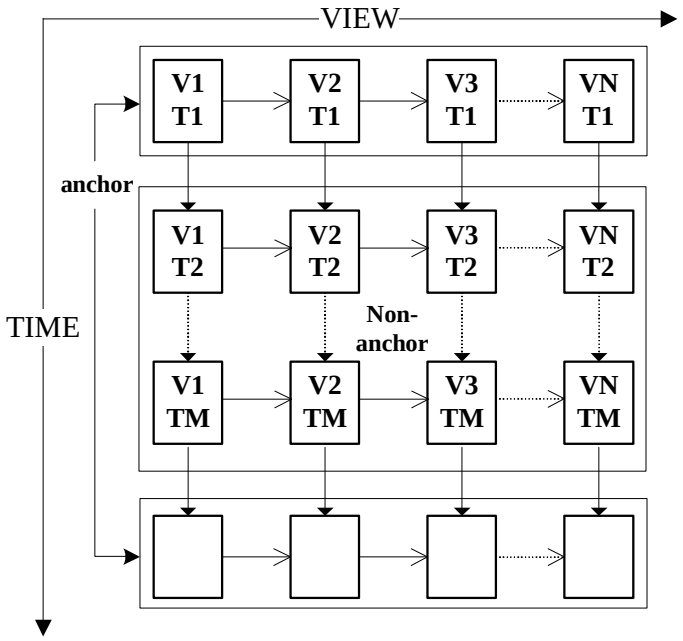
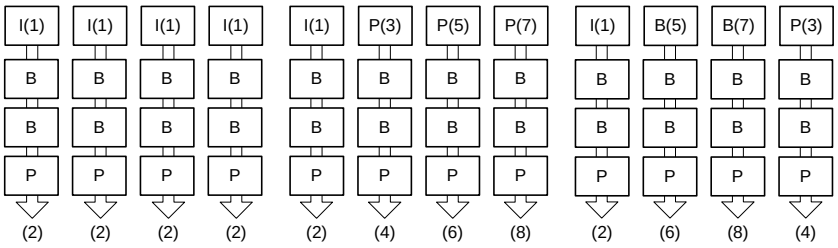


Fig. 2.1: Diagram for Multiview Video Coding. N is the number of views and M is the anchor frame interval. Anchor is encoded only by cross-view prediction.



(a) Simulcast only (b) IPPP in anchor (c) IBBP in anchor

Fig. 2.2: MVC examples where the number of views is 4. The number in parenthesis is the order of encoding in the trellis expansion.

While the problem of dependent bit allocation has been considered in several contexts, including standard video [24, 34, 36, 41] and stereo image coding [45], its potential impact in multiview video coding has not been considered yet. In this chapter, we extend previously proposed frame-wise dependent bit allocation techniques [34] (using a trellis representation and the Viterbi algorithm) to a multiview video coding scenario where cross-view prediction is used. This leads to a complex 2-D dependency problem, where the total number of video frames and candidate quantization choices involved can be very large. Moreover, a suboptimal choice of quantizer for a given frame may affect many other frames (if the frame in question is close to the root of the dependency tree). This suggests that a proper quantizer allocation may be more important in an MVC environment than for standard video. Indeed this was initially motivated by the observation that in an H.264/AVC encoder, which we modified for MVC, coding results were very sensitive to bit allocation [8].

In order to reduce the complexity of searching for the optimal solution in our MVC environment, we make use of the monotonicity property observed in [34]. To further reduce complexity, we show that the number of solutions to be searched can be reduced by considering only candidate solutions such that anchor and non-anchor frames are allocated similar quantizers.

2.2 2-D Dependent Bit Allocation

In what follows, distortion (D) is measured as frame-wise mean square error (MSE). The quantization parameter, q, rate, R, distortion, D, and Lagrangian cost J of the anchor frame in View i, are represented as q_i, R_i, D_i, and J_i, respectively. We denote $\bar{q}_i$ the quantization choice for the non-anchor frames in View i. $\bar{R}_i$, $\bar{D}_i$,

and $\bar{J}_i$ will denote the total rate, distortion and Lagrangian cost for all non-anchor frames in View i.[1] In our notation $q < q'$ means that quantizer q is finer, i.e., better quality, than q'.

A solution to the dependent bit allocation problem was proposed based on a trellis expansion and the VA in prior work [34]. Our problem, which includes dependency across views, can be seen as an extension of this 1-D problem. A constrained 2-D dependent coding problem can then be formulated as follows (for the 2-view case):

$$\begin{aligned} &\min_{q_1,q_2,\bar{q}_1,\bar{q}_2} [D_1(q_1) + \bar{D}_1(q_1,\bar{q}_1) + D_2(q_1,q_2) + \bar{D}_2(q_1,q_2,\bar{q}_2)] \\ &\text{such that} \\ &R_1(q_1) + \bar{R}_1(q_1,\bar{q}_1) + R_2(q_1,q_2) + \bar{R}_2(q_1,q_2,\bar{q}_2) \leq R_{budget}. \end{aligned} \tag{2.1}$$

Note that because of the dependency on the previously coded frames, some of the R and D values include multiple q's. For example, because non-anchor frames refer to the anchor frames as a reference, the values of $\bar{R}_1$ and $\bar{D}_1$ depend on q_1 and $\bar{q}_1$. Also the anchor frame in View 2 refers to the anchor frame in View 1, the values of R_2 and D_2 depend on q_1 and q_2. This problem can be solved by considering an unconstrained problem with Lagrange multiplier $\lambda \geq 0$ and cost $J = D + \lambda R$ [37]:

$$\min_{q_1,q_2,\bar{q}_1,\bar{q}_2} [J_1(q_1) + \bar{J}_1(q_1,\bar{q}_1) + J_2(q_1,q_2) + \bar{J}_2(q_1,q_2,\bar{q}_2)], \tag{2.2}$$

[1] First, we begin assuming the same quantizer is used for all non-anchor frames in a view. Thus, the quantizer selection for anchor and non-anchor is a sub-optimal solution for frame level bit allocation. In Section 2.2.3, a search algorithm for non-anchor frames is proposed.

where

$$J_1(q_1) = D_1(q_1) + \lambda R_1(q_1) \tag{2.3a}$$

$$\bar{J}_1(q_1, \bar{q}_1) = \bar{D}_1(q_1, \bar{q}_1) + \lambda \bar{R}_1(q_1, \bar{q}_1) \tag{2.3b}$$

$$J_2(q_1, q_2) = D_2(q_1, q_2) + \lambda R_2(q_1, q_2) \tag{2.3c}$$

$$\bar{J}_2(q_1, q_2, \bar{q}_2) = \bar{D}_2(q_1, q_2, \bar{q}_2) + \lambda \bar{R}_2(q_1, q_2, \bar{q}_2) \tag{2.3d}$$

In a system with N views, assume that our bit allocation requires evaluating, on average, n_a coding choices for each anchor frame, and n_b for each set of non-anchor frames in a view. The main complexity in the bit allocation comes from encoding/decoding step to determine R-D values. Because non-anchor frames in a view are not further referred by frames in other views, the maximum dependency would be $n_b n_a^N$ to encode non-anchor frames in View N. Thus, the bit allocation complexity will be $O(n_b n_a^N)$. We achieve a reduction in complexity based on two methods. First, as in [34], we exploit the monotonicity property of dependent coding to helps us reduce n_a. Second, we choose the non-anchor frame quantizers to be coarser than the quantizers chosen for the corresponding anchor frame, i.e., $\bar{q}_i \geq q_i$ for View i, so that fewer quantization choices for the non-anchor frames need to be evaluated (smaller n_b).

2.2.1 Monotonicity

The monotonicity property observed in [34] for a temporal dependency scenario states that, for two dependent frames (the second frame is motion/disparity predicted from the first one), we usually have:

$$J_2(q_1, q_2) \leq J_2(q_1', q_2) \text{ for } q_1 \leq q_1', \tag{2.4}$$

i.e., for a given quantizer, q_2, applied to the predicted frame, finer quantization of the predictor tends to lead to better R-D characteristics for the predicted frame. This property usually holds when the frames in (2.4) are anchor frames. Similar properties can also be observed for the dependency within a view

$$\bar{J}_1(q_1, \bar{q}_1) \leq \bar{J}_1(q_1', \bar{q}_1) \text{ for } q_1 \leq q_1', \tag{2.5}$$

as well as when various levels of dependencies, across both views and time, are present, so that, for example:

$$\bar{J}_2(q_1, q_2, \bar{q}_2) \leq \bar{J}_2(q_1', q_2, \bar{q}_2) \text{ for } q_1 \leq q_1' \tag{2.6a}$$

$$\bar{J}_2(q_1, q_2, \bar{q}_2) \leq \bar{J}_2(q_1, q_2', \bar{q}_2) \text{ for } q_2 \leq q_2' \tag{2.6b}$$

From these monotonicity properties, the following lemma can be derived.

Lemma 1: If

$$J_1(q_1) + \bar{J}_1(q_1, \bar{q}_1) + J_2(q_1, q_2) < J_1(q_1') + \bar{J}_1(q_1', \bar{q}_1) + J_2(q_1', q_2) \text{ for } q_1 < q_1' \tag{2.7}$$

then q_1' is not in the optimal path set and can be pruned out.

Proof: Similar to the proof in [34], we prove the lemma by contradiction. Assume that q_1' for any $q_1 < q_1'$ is part of the optimal path. Let the optimal anchor

frame quantizer sequence be $(q_1', \bar{q}_1, q_2, \bar{q}_2, ..., q_N, \bar{q}_N)$. However, for $q_1 < q_1'$, by the monotonocicy from (2.4),

$$J_3(q_1, q_2, q_3) < J_3(q_1', q_2, q_3) \tag{2.8}$$

$$\cdots$$

$$J_N(q_1, q_2, ..., q_N) < J_N(q_1', q_2, ..., q_N) \tag{2.9}$$

and by the monotonicity from (2.5) and (2.6),

$$\bar{J}_2(q_1, q_2, \bar{q}_2) < \bar{J}_2(q_1', q_2, \bar{q}_2) \tag{2.10}$$

$$\bar{J}_3(q_1, q_2, q_3, \bar{q}_3) < \bar{J}_3(q_1', q_2, q_3, \bar{q}_3) \tag{2.11}$$

$$\cdots$$

$$\bar{J}_N(q_1, q_2, ..., q_N, \bar{q}_N) < \bar{J}_N(q_1', q_2, ..., q_N, \bar{q}_N) \tag{2.12}$$

Summing up (2.7), (2.8), ..., (2.12), we get the contradiction that the Lagrangian cost with $(q_1, \bar{q}_1, q_2, \bar{q}_2, ..., q_N, \bar{q}_N)$ is smaller than the one with $(q_1', \bar{q}_1, q_2, \bar{q}_2, ..., q_N, \bar{q}_N)$ thus, q_1' is not in the optimal path. $\square$

Lemma 1 above and the Lemma 2 in [34] are used in the pruning steps in our proposed algorithm. This algorithm is based on an IPPP anchor frame coding scheme as shown in Fig. 2.2(b). For the trellis expansion in anchor and non-anchor frames, refer to Figs. 2.3 and 2.4. In the following algorithm, $\mathbf{q}_1^i = \{q_1, q_2, ..., q_i\}$ is an anchor frame quantizer allocation for views 1 through i. $J_i(\mathbf{q}_1^{i-1}, q_i)$ is the Lagrangian cost of the anchor frame in View i for a surviving anchor frame quantizer allocation $\mathbf{q}_1^{i-1}$ and anchor frame quantizer q_i in View i. $\bar{J}_i(\mathbf{q}_1^i, \bar{q}_i)$ is the Lagrangian cost of the non-anchor frame in View i for anchor frame quantizer allocation $\mathbf{q}_1^i$ and non-anchor frame quantizer $\bar{q}_i$ in View i. $J(\mathbf{q}_1^i, \bar{\mathbf{q}}_1^i)$ is the total cost with quantizer

allocations $\mathbf{q}_1^i$ and $\bar{\mathbf{q}}_1^i$ for views 1 through i.

Algorithm 1:

1. For View $i > 1$, generate the Lagrangian cost of the anchor frame: $J_i(\mathbf{q}_1^{i-1}, q_i)$, for all surviving quantizer allocations $\mathbf{q}_1^{i-1}$, and for all choices of q_i. The anchor frame of View 1 is coded independently for all possible quantizer allocations q_1 with the Lagrangian cost $J_1(q_1)$.

2. Compute $J(\mathbf{q}_1^{i-1}, \bar{\mathbf{q}}_1^{i-1}) + J_i(\mathbf{q}_1^{i-1}, q_i)$ and use pruning condition of *Lemma 1* and *Lemma 2* in [34] to eliminate suboptimal paths up to View i.

3. For View i, generate the non-anchor frame cost: $\bar{J}_i(\mathbf{q}_1^i, \bar{q}_i)$ for $\bar{q}_i$ for all surviving allocations $\mathbf{q}_1^{i-1}$, and all surviving anchor frame quantizers q_i.

4. Find minimum non-anchor frame cost $\bar{J}_i(\mathbf{q}_1^i, \bar{q}_i)$ for each $\mathbf{q}_1^i$.

5. For View $i > 1$, compute total cost $J(\mathbf{q}_1^i, \bar{\mathbf{q}}_1^i) = J(\mathbf{q}_1^{i-1}, \bar{\mathbf{q}}_1^{i-1}) + J_i(\mathbf{q}_1^{i-1}, q_i) + \bar{J}_i(\mathbf{q}_1^i, \bar{q}_i)$ for each anchor frame quantizer q_i. For View 1, total cost is $J(q_1, \bar{q}_1) = J_1(q_1) + \bar{J}_1(q_1, \bar{q}_1)$ for each anchor frame quantizer q_1.

6. With every surviving path, $q_1, q_2, ..., q_i$, proceed to View $i+1$ and go to Step 1.

Note that for each anchor frame quantizer in each surviving allocation $\mathbf{q}_1^i$, there is a corresponding non-anchor frame quantizer with minimum cost, which is shown as a thick line in Fig. 2.4.

The above algorithm can be easily modified for either IBBP or IBP coding of anchor frames. An additional step required to search for a solution under IBP coding of anchor frames would be to populate branches between I and P1 with costs $J_{B1}(q_I, q_{P1}, q_{B1})$ and $\bar{J}_{B1}(q_I, q_{P1}, q_{B1}, \bar{q}_{B1})$.

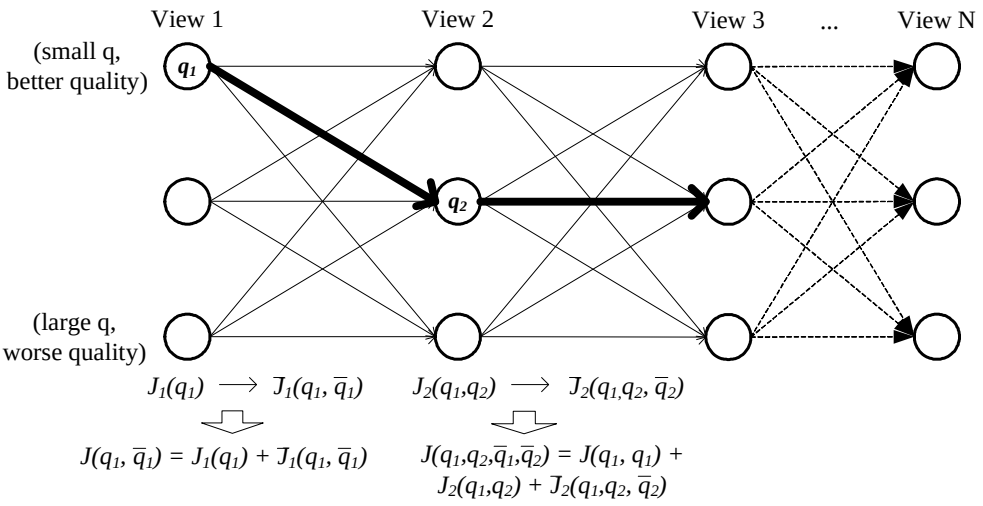


Fig. 2.3: Trellis expansion in anchor frame. The thick line shows one of anchor frame quantizer allocations.

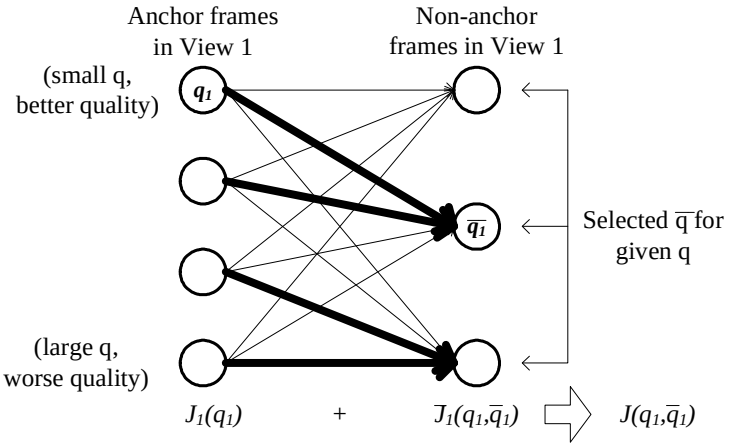


Fig. 2.4: Trellis expansion in View 1. For each anchor frame quantizer q a non-anchor frame quantizer $\bar{q}$ with minimum cost can be chosen (thick line) and the total cost for each quantizer allocation can be calculated.

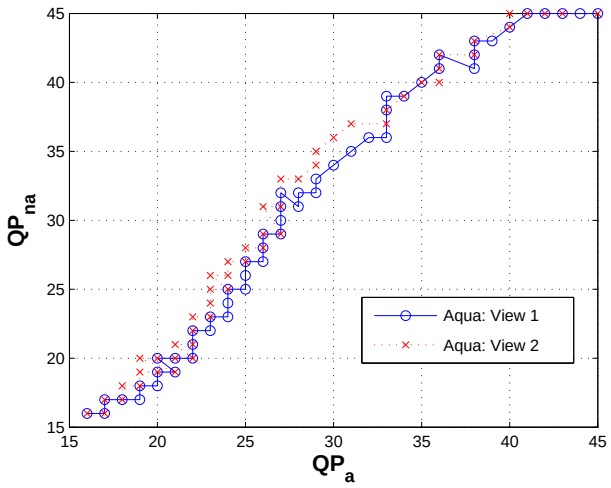


Fig. 2.5: Relationship between $q(QP_a)$ and $\bar{q}(QP_{na})$ for Aquarium sequence.

2.2.2 Reduced Search Range

Even though complexity is reduced by taking advantage of the monotonicity property, further reductions are achievable by considering the relationship between the anchor and non-anchor quantizer chosen in an optimal solution. According to our experiments, optimal bit allocations are such that there exists a strong correlation between q and $\bar{q}$. This is shown in Fig. 2.5, where we plot, for different values of λ, the pair of quantization values for anchor and non-anchor frames that minimize the Lagrangian cost for the given λ. The exact slope in Fig. 2.5 depends in general on the number of non-anchor frames and how the anchor frame is encoded.

In what follows we provide an analysis that supports the type of relationship between quantizers that we observe in optimal solutions. Let Q_1 and Q_2 be the quantization choices made for the anchor frame in a view and the non-anchor

frames in the same view, respectively, where smaller Q means finer quantization. The Lagrangian cost J for that view is then

$$J = D_1(Q_1) + D_2(Q_1, Q_2) + \lambda(R_1(Q_1) + R_2(Q_1, Q_2)). \tag{2.13}$$

In order to better understand the properties of the optimal solution we take derivatives of J with respect to Q_1 and Q_2, and set them to zero:

$$\frac{\partial J}{\partial Q_1} = \frac{\partial D_1}{\partial Q_1} + \frac{\partial D_2}{\partial Q_1} + \lambda(\frac{\partial R_1}{\partial Q_1} + \frac{\partial R_2}{\partial Q_1}) = 0 \tag{2.14}$$

$$\frac{\partial J}{\partial Q_2} = 0 \Leftrightarrow \lambda = -\frac{\partial D_2}{\partial Q_2}/\frac{\partial R_2}{\partial Q_2} = -\frac{d_2}{r_2}, \tag{2.15}$$

where we define $d_i = \partial D_i/\partial Q_i$ and $r_i = \partial R_i/\partial Q_i$. Then, from (2.14) and (2.15),

$$d_1 r_2 - d_2 r_1 = d_2 \frac{\partial R_2}{\partial Q_1} - \frac{\partial D_2}{\partial Q_1} r_2 \tag{2.16}$$

$$\frac{d_1}{r_1} - \frac{d_2}{r_2} = -\frac{1}{r_1}(\frac{\partial D_2}{\partial Q_1} + \lambda \frac{\partial R_2}{\partial Q_1}) \tag{2.17}$$

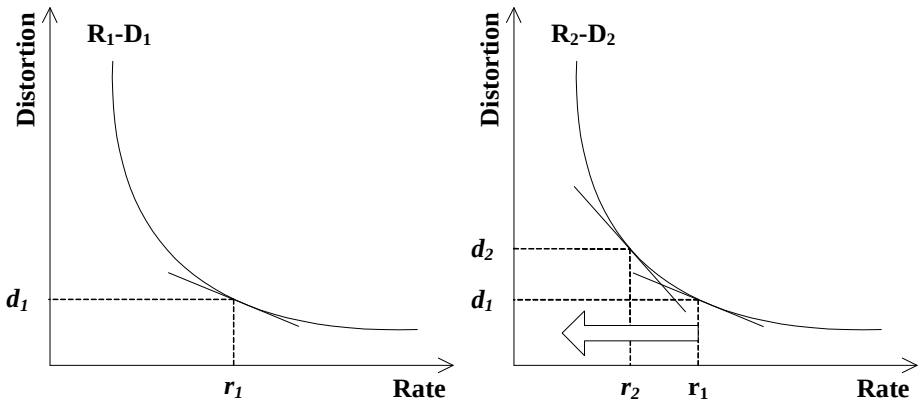
Note that, by the monotonicity property, if Q_1 increases while Q_2 remains constant then both D_2 and R_2 will tend to increase. Thus, $\frac{\partial D_2}{\partial Q_1} \geq 0$ and $\frac{\partial R_2}{\partial Q_1} \geq 0$. Because $d_i \geq 0$ and $r_i \leq 0$, from (2.17)

$$\frac{d_1}{r_1} \geq \frac{d_2}{r_2} \tag{2.18}$$

so that we can say that

$$|\frac{\Delta D_1}{\Delta R_1}| \leq |\frac{\Delta D_2}{\Delta R_2}|. \tag{2.19}$$

In words, at optimality, the slope of operating point in the R_1-D_1 characteristic is smaller than the slope of the operating in the R_2-D_2 characteristics. Note that to



(a) R_1-D_1 (b) R_2-D_2

Fig. 2.6: Example of R-D curves for reduced search range

derive (2.18) we only had to make one assumption, namely that the monotonicity property holds.

Given that the slope $|\frac{\Delta D}{\Delta R}|$ of a convex R-D decreases as Q decreases (i.e., as the coding quality improves), we can conclude that if R_1-D_1 and R_2-D_2 have similar shape, then from (2.18), at optimality $Q_2 > Q_1$. In our case of interest, Q_2 is the quantizer used to encode several non-anchor frames. In this case $|\frac{\Delta D_2}{\Delta R_2}|$ would be the slope of the aggregate R-D characteristic. While the absolute values of R_2 and D_2 are likely to be larger than those for R_1 and D_1 at a given Q, the shapes of the curves and corresponding slopes can still be assumed to be similar. This approximation agrees well with our observed experimental behavior and provides a tool for complexity reductions.

For example, in Fig. 2.6, it is assumed that the R_1-D_1 and R_2-D_2 curves are similar. To have $|\frac{\Delta D_1}{\Delta R_1}| \leq |\frac{\Delta D_2}{\Delta R_2}|$ as shown in (2.19), the operating point should be in the left of r_1 (i.e., r_2) in Fig. 2.6(b). Because rate decreases and distortion increases as Q increases, Q for r_2 (Q_2) should be larger than Q for r_1 (Q_1), i.e., $Q_1 \leq Q_2$.

Based on this observation, Step 3 in *Algorithm 1* can be modified as

3. For View i, generate the non-anchor frame cost: $\bar{J}_i(\mathbf{q}_1^i, \bar{q}_i)$ for $\bar{q}_i$ for all surviving allocations $\mathbf{q}_1^{i-1}$, and all surviving anchor frame quantizers q_i, *such that* $\bar{q}_i \geq q_i$.

2.2.3 Search Algorithm for Non-anchor Frames

Up to now, for simplicity, we have assumed that the same quantizer is used for all non-anchor frames.

We now propose a non-anchor frame quantizer search algorithm, which operates for a given anchor frame quantizer and, to reduce complexity, uses the following property (based on the discussion of the previous section): a frame close to root of the dependency tree has more influence on cost and therefore a better quantizer should be applied to it. Thus we begin the search with the frame which is close to the root. In the following algorithm, $\bar{\mathbf{q}} = \{Q_2, Q_3, ..., Q_M\}$, is the vector of quantizers allocated to the non-anchor frames in a given view.

Algorithm 2: Dependent coding in each view

1. Given λ and the QP of anchor frame q_0, initialize $\bar{\mathbf{q}} = \{Q_2, Q_3, ..., Q_M\} = \{q_0, q_0, ..., q_0\}$.

2. For frames $i = 2, 3, ..., M$,
   find $\alpha_i = \frac{\partial J}{\partial Q_i} = (\sum_{j=i}^{M} \frac{\partial D_j}{\partial Q_i}) + \lambda(\sum_{j=i}^{M} \frac{\partial R_j}{\partial Q_i})$.

3. - If $\alpha_i < 0$, $Q_i = Q_i + 1$. Increase Q_j which is less than Q_i for $j = \{i + 1, ..., M\}$.
   - If $\bar{J}$ decreases, update $\bar{\mathbf{q}}$. Proceed to the next frame.

4. Repeat step 2 - 3 until there is no update in Q_i

In this algorithm α_i is calculated for the current $\bar{\mathbf{q}}$. Then in order to make α_i closer to 0, we increase Q_i by 1 if $\alpha_i < 0$. Then, using the property motivated in the previous section, we also increase Q_j such that $Q_j < Q_i$ for $j > i$.

2.3 Simulation Results

Using the H.264/AVC reference codec, we encoded the Aquarium multiview sequences from Tanimoto Lab shown in Fig. 2.7 using three different coding schemes, i.e., SC in 2.2(a), MVC in 2.2(b) with fixed QP and optimized QP using proposed *Algorithm 1*. In the experiment all non-anchor frames in a view were assigned the same quantizer. Two different coding conditions are used: First, all possible block sizes can be used and intra coding is enabled (C1). Second, only 8x8 block size is used and intra coding is disabled except for I frame (C2). The first 7 frames of Views 1, 2, and 3 are used in the experiment. As can be seen in Fig. 2.8, the proposed algorithm provides a gain of 0.5 dB as compared to MVC with fixed QP. In trellis expansion, six quantizers are selected as candidates for anchor and only three quantizers are selected for non-anchor frames using correlation between anchor and non-anchor quantizer.

Note that in C2, intra coding is disabled except I frame thus, dependencies between frames are higher than C1. In C2, the proposed algorithm achieves higher coding gains (e.g., up to 1 dB compared to MVC) than in C1.

2.4 Conclusions

In this chapter, 2-D bit allocation scheme was proposed. Complexity of data generation in trellis expansion is significant due to the increased dimensionality in

Fig. 2.7: Aqua sequence

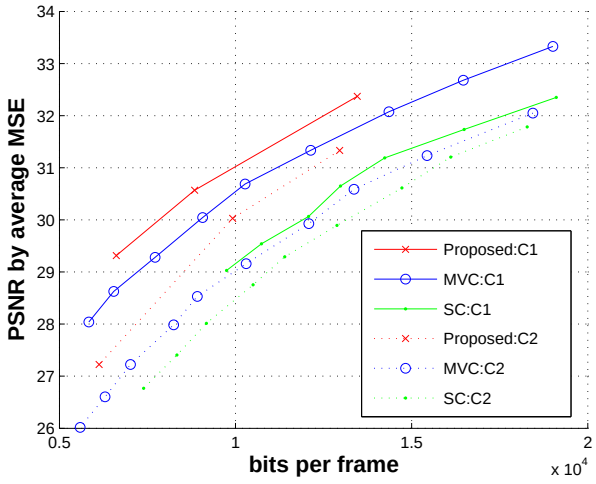


Fig. 2.8: SC vs. MVC with fixed QP vs. MVC by proposed Algorithm 1. Average number of bits for 21 frames is used and PSNR is calculated as $10log_{10}(255^2/(average\ MSE\ for\ 21\ frames))$. λ is 200, 500, and 900 with C1 and 300, 700, and 2500 with C2 in proposed algorithm.

MVC. We extend the monotonicity property from [34] and use it to prune sub-optimal quantizers. Complexity can be reduced further using the fact that optimal solutions tend to show correlation between quantizers of anchor and non-anchor frames. Proposed algorithm with reduced complexity achieves 0.5 dB gains as compared to MVC.

Chapter 3

Illumination Compensation In Multi-View Video Coding

3.1 Preliminaries

In Chapter 2, we addressed the dependency problem in multi-view video coding (MVC) and proposed a quantizer search method. This optimization is performed at the frame level according to the multi-view sequence structure and coding scheme. In this chapter and Chapter 4, we move down to block level in a frame and propose methods to improve the quality of estimation for the original block in order to improve coding efficiency in MVC.

In block based predictive coding, a frame is divided into blocks first, then for each block, the most correlated match (*predictor*[1]) is searched in reference frames, and residual error between the original and the best match is encoded and transmitted with signaling information (i.e., the motion vector). These block based

[1] Here by predictor, we mean the selected estimate of current block after motion/disparity search in the references. Note that a prediction is also used to select the center of motion/disparity search. This is obtained from motion/disparity vectors of neighboring blocks using spatial correlation. We will refer to this as motion/disparity vector predictor in this work.

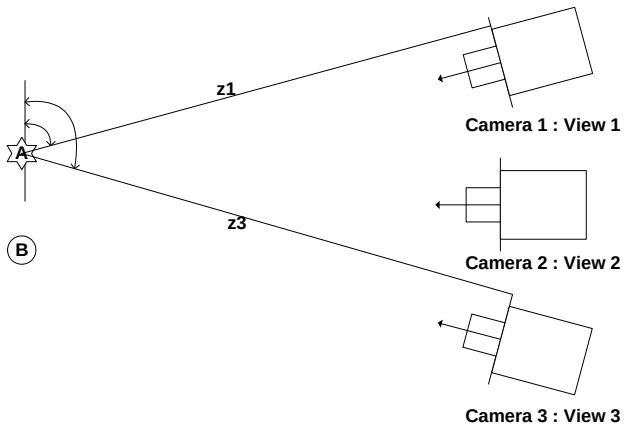
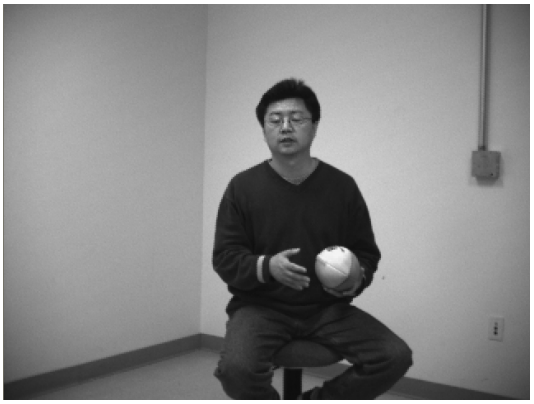


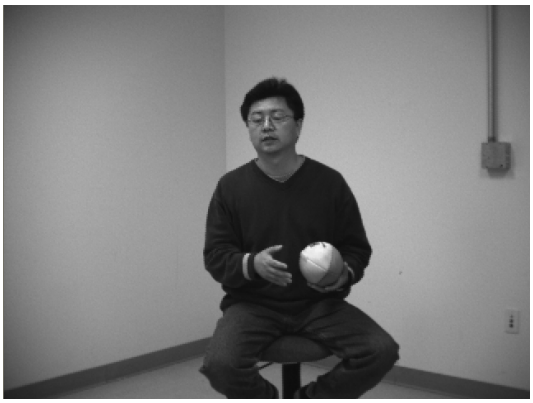
Fig. 3.1: Camera arrangement that causes local mismatches

approaches exploiting the correlation between frames are applied for disparity estimation and compensation in cross-view prediction, e.g., [3]. While the motion in temporal prediction is caused by the displacement of the objects, the disparity in cross-view prediction and the depths of objects in the scene comes from the displacement and orientation of the cameras. Generally, disparity in cross-view prediction is known to be more difficult to compensate than motion because of the irregularity of the disparity field [5] and the severe occlusion effects that are caused by different object depths. In contrast, in temporal prediction, most of the background is static and only moving objects need to be motion compensated. Furthermore, frames from different views are prone to suffer from mismatches other than disparity. We now consider other mismatch cases.

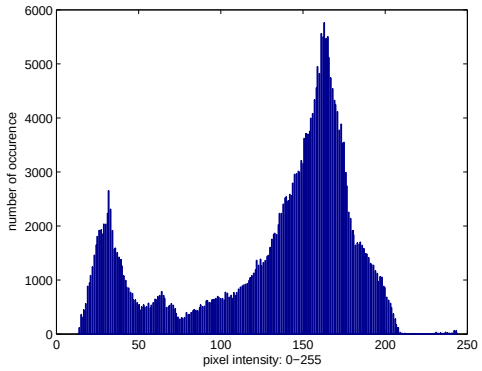
Firstly, in a generic multi-view video capturing system, we can not assume that a perfect calibration is achieved among different cameras because there are too many variables to be adjusted including intrinsic camera parameters. These heterogenous cameras can cause global (frame-wise) mismatches among different views, which can manifest themselves in both luminance and chrominance channels.



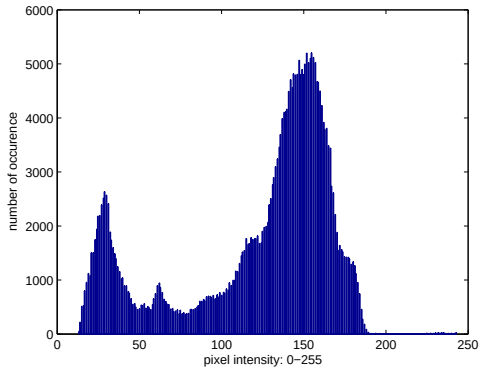
(a) ST sequence first frame of view 3



(b) ST sequence first frame of view 4



(c) Histogram of frame in (a); the mean grayscale value is 131.



(d) Histogram of frame in (b); the mean grayscale value is 122.

Fig. 3.2: Illumination mismatches in ST sequence

For example, frames in one view may appear brighter and/or out of focus as compared to frames from the other view, due to mis-calibration.

Secondly, even if camera calibration is perfect, objects may appear differently in each view due to camera locations and orientations. Consider the camera arrangement in Fig. 3.1, object A is projected to camera 1 and camera 3 at different angle, and therefore it causes different reflection effects with respect to the cameras. For this example, different portions of a video frame can undergo different illumination changes with respect to the corresponding areas in frames from the other views. Fig. 3.2 demonstrates illumination mismatches between two views

from the *ST* sequence. In Fig. 3.2(a) and 3.2(b), severe illumination mismatches can be observed in the background, which correspond to the different maximum pixel intensities in Fig. 3.2(c) and 3.2(d). However, the minimum pixel intensities of different views are similar (e.g., the person's clothes). From Fig. 3.2(c) and 3.2(d), two histograms show similar shape with local variations, which are caused by global and local illumination mismatches. Average pixel intensities of View 3 and View 4 are 131 and 122 respectively.

In addition to illumination mismatches in cross-view frames, focus may change from one view to another view [21]. In Fig. 3.1, Object A is at a greater scene-depth (z_1) in View 1 than in View 3 (z_3). Even if all cameras are perfectly calibrated with the same focus at scene depth z_1, Object A appears focused in View 1 while it is de-focused (blurred) in View 3. On the other hand, Object B will become sharpened in View 3 as compared to in View 1.

All these factors lead to discrepancies among video sequences in different views. The efficiency of cross-view disparity compensation could deteriorate in the presence of these mismatches. In this work, we focus on techniques for illumination compensation in order to improve coding efficiency in the presence of illumination mismatches between views.

3.2 Related Work

Various approaches have been proposed for monoscopic video coding to address illumination changes in temporal prediction. In [19], illumination is compensated in two steps. First, illumination mismatch is compensated globally using a decimated image (that contains the DC coefficients of all blocks). Then block-wise compensation is applied. In both steps, multiplicative and additive terms are

used. This two step compensation is applied only to frames classified as having large illumination mismatches, which does not occur as frequently in monoscopic temporal prediction, as compared to cross-view prediction in MVC. Note also that local compensation is not fully integrated into the search step and that an efficient coding for mismatch parameters is not provided. In [32], an illumination component and a reflectance component are both compensated using scale factors that are quantized and Huffman coded. This illumination model is useful for contrast adjustment but cannot model severe mismatches in MVC properly. In [15], a brightness variation is modeled by two parameters for the multiplier field and offset term, respectively. These parameters are used globally for whole frames. To reduce the impact of local brightness variation, a set of parameters is collected and a pair is chosen based on the relative frequency of all parameter pairs. Illumination compensation is disactivated for those blocks for which the selected parameters are not efficient. This global approach cannot adapt to some large luminance variations in MVC, which are dependent on relative positions of camera and objects. Recently, in [23], illumination mismatches are compensated using scale and offset parameters, which is similar to the approach proposed in this work. Mismatch parameters are computed as part of the motion search and are differentially coded and selectively activated. However, this approach mainly targets the illumination compensation in video sequences where luminance changes progressively or due to abrupt changes in lighting, e.g., a flash, which can be compensated by a global model (e.g., weighted prediction). In cross-view frames, illumination mismatches are caused by heterogeneous cameras and different depths and perspectives, which leads to both local and global mismatches.

Weighted prediction (WP) methods have been proposed and adopted in H.264/AVC [6]. Multiplicative weighting factors and additive offsets are applied to the

motion compensated prediction. According to whether two parameters are coded for each reference picture index or are derived based on the relative picture order count, these techniques are categorized as explicit and implicit, respectively. This global approach provides significant bitrate reduction in coding fades in monoscopic video. However, for multi-view video where severe local variations are present, WP does not provide efficient compensation.

Next, block-based illumination compensation (IC) techniques [17, 25, 26] are presented. These were originally addressed in [25]. In [26], vector quantization of two parameters was proposed. In [17], we proposed to use only an additive term considering the trade-off between the computational complexity and the coding efficiency. We start by defining an illumination model, and derive a coding scheme that efficiently compensates for illumination changes across views.

3.3 Illumination Compensation Model

Block-wise disparity search aims to find the block in the reference frame that best matches a block in the current frame, leading to minimum residual error after prediction. Under severe illumination mismatch conditions, coding efficiency will suffer because i) residual energy for the best match candidate will generally be higher, and ii) true disparity is less likely to be found, leading to a more irregular disparity field and likely increases to the rate needed for disparity field encoding.

As described previously, illumination mismatches can be local in nature. Thus, we adopt a local IC model to compensate both global and local luminance variation in a frame. The IC parameters are estimated as part of the disparity vector search and these parameters are differentially encoded for transmission to the decoder, in order to exploit the spatial correlation in illumination mismatch. Finally, a

decision is made to activate IC on a block per block basis using a rate distortion criterion.

When considering pixels corresponding to a given object but captured by different cameras, observed illumination mismatches need not be the same for all pixels, and will depend in general on the continuous plenoptic and radiance functions [4]. However, since the goal is to transmit explicit illumination mismatch information to the decoder, block-wise IC models are adopted, with the optimal block size decided based on R-D cost. As an initial step we evaluate a simple block-wise affine model, with an *additive offset* term C and a *multiplicative scale* factor, S, leading to a mismatch model $\Psi = \{S, C\}$ as proposed in [15].

For the original block signal to be encoded ($\bar{\mathbf{x}}$), the i^{th} predictor candidate ($\bar{\mathbf{p}}^i$) in the reference frames can be decomposed into the sum of its mean μ_{p^i} and a zero mean signal, $\bar{\mathbf{p}}_0^i$: $\bar{\mathbf{p}}^i(x,y) = \mu_{p^i} + \bar{\mathbf{p}}_0^i(x,y)$, where (x,y) is the pixel location within the block. Then the illumination compensated predictor $\hat{\mathbf{p}}^i(x,y)$ with IC model Ψ^i is:

$$\hat{\mathbf{p}}^i(x,y) = [\mu_{p^i} + C^i] + S^i \cdot \bar{\mathbf{p}}_0^i(x,y). \tag{3.1}$$

This formulation allows us to separate the effect of each parameter, so that DC and AC mismatches are compensated separately. Furthermore, by applying a multiplicative compensation to the mean removed prediction in (3.1) we avoid the propagation of quantization error from scale to offset [26].

As shown in Fig. 3.3, for the original block signal, we look for the best matching predictor within the search range in the reference frame using a modified matching metric that incorporates an IC model between the original block and a predictor candidate. This new metric, sum of absolute differences after compensation

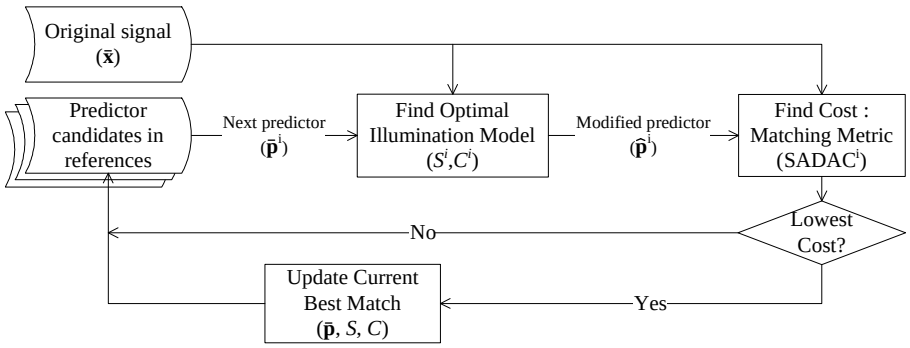


Fig. 3.3: Modified Search Loop for the current block

(SADAC), essentially computes the SAD between the original block and the predictor to which IC has been applied. Thus, for each predictor candidate, optimal IC parameters have to be computed. While SADAC is used for the search with IC, similarly to how SAD is used in H.264/AVC, a quadratic metric, namely, sum of squared differences after compensation (SSDAC) is used to find IC parameters.[2] For the original signal $\bar{\mathbf{x}}$ and illumination compensated i^{th} predictor candidate $\hat{\mathbf{p}}^i$, the SSDAC is defined as

$$SSDAC^i \equiv \sum_{\forall(x,y)} |\bar{\mathbf{x}}(x,y) - \hat{\mathbf{p}}^i(x,y)|^2. \tag{3.2}$$

Replacing $\hat{\mathbf{p}}^i$ using (3.1) and separating the mean from $\bar{\mathbf{x}}$, we have

$$SSDAC^i = \sum_{\forall(x,y)} |[\mu_x - \mu_{p^i} - C^i] + [\bar{\mathbf{x}}_0(x,y) - S^i \cdot \bar{\mathbf{p}}_0^i(x,y)]|^2 \tag{3.3}$$

[2] To reduce the computational complexity in searching step, SADAC is used instead of SSDAC, which is only used to find the optimal IC parameters. However for normal and Laplace distribution models of residual error, the same search results will be obtained with the two metrics under the conditions as discussed in Appendix A

Then the optimal IC parameter $\Psi^i = \arg\min_{\{S^i,C^i\}}\{SSDAC^i\}$ can be obtained by setting to zero the gradient of (3.3):

$$S^i = \frac{\sigma^2_{\bar{\mathbf{x}}\bar{\mathbf{p}}^i}}{\sigma^2_{\bar{\mathbf{p}}^i\bar{\mathbf{p}}^i}}, \tag{3.4}$$

$$C^i = \mu_x - \mu_{p^i}, \tag{3.5}$$

where

$$\sigma_{AB}{}^2 = \frac{1}{N}\sum_{\forall(x,y)}[A(x,y)-\mu_A][B(x,y)-\mu_B], \tag{3.6}$$

with $A, B \in \{\bar{\mathbf{x}}, \bar{\mathbf{p}}^i\}$ and N is the number of pixels in the block.

This solution shows that the additive parameter directly removes the offset mismatch and the multiplicative parameter compensates zero-mean variations according to block statistics. If the mean removed current and reference blocks are not highly correlated to each other, this scale factor will be small and thus only additive offset compensation will affect the reference block.

Among all candidates within the search range, the predictor $\bar{\mathbf{p}}$ minimizing SADAC with IC parameters is selected as the best match and the minimum SSDAC is given as follows,

$$\widehat{SSDAC} = N\cdot(\sigma^2_{\bar{\mathbf{x}}\bar{\mathbf{x}}} - \frac{\sigma^4_{\bar{\mathbf{x}}\bar{\mathbf{p}}}}{\sigma^2_{\bar{\mathbf{p}}\bar{\mathbf{p}}}}) = N\cdot\sigma^2_{\bar{\mathbf{x}}\bar{\mathbf{x}}}\cdot(1-\rho^2), \tag{3.7}$$

where ρ is the correlation coefficient between the original block signal $\bar{\mathbf{x}}$ and the predictor $\bar{\mathbf{p}}$.

3.4 Illumination Mismatch Parameter Coding

Using both scale and offset parameters leads to more flexibility in compensating for illumination mismatches but may not be efficient for coding, given the overhead required to represent both IC parameters. In our observation the scale parameter is also sensitive to quantization noise because of its multiplicative nature given that:

$$\widetilde{SSDAC} = \widehat{SSDAC} + N\Delta C^2 + N\Delta S^2\sigma^2_{\bar{\mathbf{p}}\bar{\mathbf{p}}} \tag{3.8}$$

where $\widetilde{SSDAC}$ is the SSDAC after quantization of IC parameters, $\widehat{SSDAC}$ is the minimum SSDAC in (3.7), N is the number of pixels in the block and ΔC and ΔS is the quantization noise of offset and scale parameter, respectively. The quantization noise of scale parameter is multiplied by the variance of the predictor, $\sigma^2_{\bar{\mathbf{p}}\bar{\mathbf{p}}}$, thus, even small quantization errors in the scale parameter can lead to fairly large differences in the compensated reference block. Taking this into account, as well as the complexity involved in calculating this parameter within the disparity search step, in the rest of the work we use only the offset parameter for IC.

To encode the offset parameter we exploit the correlations between illumination compensation parameters in neighboring blocks. As a predictor of the IC parameter of a block, we use the IC parameter of the block to its left; this allows prediction to be performed in a causal manner. If the left block was not encoded using IC, the block above is used instead as a predictor. If IC is disabled for both of these blocks then no prediction is used to encode the IC parameter for the current block (equivalently, the predictor is set to zero).

The prediction residue is quantized and then encoded. We use a simple uniform quantizer, which offers good performance and low complexity. This quantized differential offset is encoded using the context adaptive binary arithmetic coder

Tab. 3.1: Unary binarization and assigned probability for index of quantized differential offset

Absolute value (*val*)	Bin 1	Bin 2	Bin 3	Bin 4 ...
0	0			
1	1	0		
2	1	1	0	
3	1	1	1	0
...	...	...	...	...
Assigned probability.	P1	P2	P3	P4

(CABAC) [27], which consists of (*i*) binarization, (*ii*) context modeling and (*iii*) binary arithmetic coding. We first separate the absolute value *(val)* and the sign of these quantized differential offsets. Then, the absolute values of quantized offsets are binarized by selecting a unary representation as in Tab. 3.1. These representations of symbols (IC parameters) reduce the alphabet size of symbol and enable context modeling on a sub-symbol level [27].

The differential offset parameters are prediction residues which tend to be small and exhibit a symmetric distribution around zero, with very limited spatial correlation. Therefore, different probability models are used for the different binary symbol positions of *val* as shown in Tab. 3.1. The number of different probability models for binary symbols in *val* is chosen to be four experimentally. Bits corresponding to *val* greater than 3 use the same probability model. All probability models are initialized with equal symbol probability and updated according to the binary symbols to be coded. Arithmetic coding is also used for the sign, with a probability model initialized with equal symbol probability.

Clearly, different blocks suffer from different levels of illumination mismatch, so that potential R-D benefits of using IC differ from block to block. Thus we allow the encoder to decide whether or not the IC parameters are used on a block by

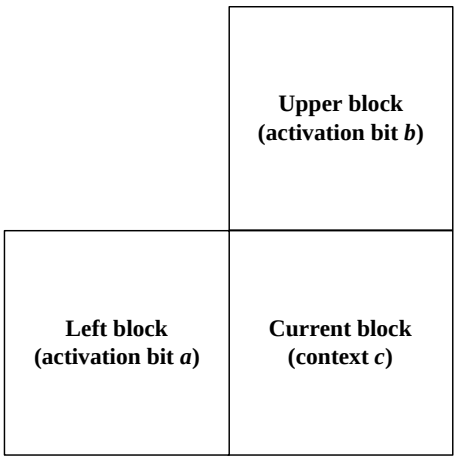


Fig. 3.4: Context of current block $c = a + b$, where $a, b \in \{0, 1\}$.

block basis. This is achieved by computing the R-D values associated to coding each block with and without IC, and then letting the Lagrangian optimization tools in the H.264/AVC codec make an R-D optimal decision. There is an added overhead needed to indicate for each block whether IC is used but this is more efficient overall than sending IC parameters for all blocks. This IC activation bit is also entropy-encoded using CABAC. The context is defined based on the activation choices made for the left and upper blocks. If IC is enabled or disabled in *both* these blocks, it is highly probable that the same choice will be made for the current block. However if only one of these two neighboring blocks uses IC, the probability of the current block using IC should be close 1/2. Based on this observation, three contexts are assigned and initialized for activation switch, which is similar to the context setup for the Skip flag or the transform size in H.264/AVC. Fig. 3.4 demonstrates how the context of current block is defined by IC activation bits from left and upper block. Tab. 3.2 shows the initialization of the context for IC activation bit.

Tab. 3.2: Initialization of the context for IC activation bit with different most probable symbol (MPS)

Context c	MPS	Probability
0	0	1
1	0	$\frac{1}{2}$
2	1	1

Tab. 3.3: Number of addition/subtraction for SAD and SADAC. N is the number of pixels in a macroblock and S is the number of search points.

SAD (Original)	SADAC (IC enabled)	
$\sum_{\forall(x,y)} \lvert\bar{\mathbf{x}}(x,y) - \bar{\mathbf{p}}^i(x,y)\rvert \rightarrow 2N$	$\mu_{\bar{\mathbf{x}}} \rightarrow N$	$\mu_{\bar{\mathbf{p}}^i} \rightarrow N$ (*or 0 in Fast IC mode*)
		$C^i = \mu_{\bar{\mathbf{x}}} - \mu_{\bar{\mathbf{p}}^i} \rightarrow 1$
		$\sum_N \lvert\bar{\mathbf{x}}(x,y) - \bar{\mathbf{p}}^i(x,y) - C^i\rvert \rightarrow 3N$
For S search points $\rightarrow 2NS$	N	For S search points $\rightarrow 4NS + S$
$2NS$	$N + S + 4NS \approx 4NS$	

3.5 Complexity of IC

The impact of IC on encoding complexity is mostly due to changes in the disparity estimation metric computation (other changes to the encoder such as encoding of IC parameter and R-D based IC activation, have a negligible effect on overall complexity). Thus, in what follows, additional complexity for IC is analyzed in terms of the number of addition/subtraction operations in the SAD calculation.

As can be seen in Tab. 3.3, for N pixels in a macroblock and S search points, in each block mode, IC requires $4NS$ calculations for SADAC, while $2NS$ are required in SAD. For SAD, the differences of current and reference pixels (N) are calculated first. After the absolute value operation, N absolute differences are added to compute SAD, which require a total of $2N$ operations. Similarly, for SADAC a total of $4N$ operations are required, including $\mu_{\bar{\mathbf{x}}}$, $\mu_{\bar{\mathbf{p}}^i}$ and C^i calculations. For the mean

Tab. 3.4: Complexity for SAD Calculation in Different Block Modes. N is the number of pixels in a macroblock and S is the number of search points. Note that in Fast IC mode, $\mu_{\bar{\mathbf{p}}^i}$ in 8×8 block is saved to be used in the larger block mode so that $4NS$ complexity is required in 8×8 block mode and $3NS$ in 8×16, 16×8, 16×16 block modes.

Block Modes	Original	IC	Fast IC
4×4	$2NS$	-	-
4×8	$2NS$	-	-
8×4	$2NS$	-	-
8×8	$2NS$	$4NS$	$4NS$
8×16	$2NS$	$4NS$	$3NS$
16×8	$2NS$	$4NS$	$3NS$
16×16	$2NS$	$4NS$	$3NS$
TOTAL	$14NS$	$16NS$	$13NS$

calculation, we need to sum N pixels, which requires N additions. Throughout the analysis, shift operation for mean calculation and absolute value operation are not counted. Assuming the center of search for different block modes does not deviate significantly, $\mu_{\bar{\mathbf{p}}^i}$ in small blocks can be reused in larger blocks avoiding redundant calculations. For example, by storing $\mu_{\bar{\mathbf{p}}^i}$ in 8×8 blocks, the calculation of $\mu_{\bar{\mathbf{p}}^i}$ in the larger block (e.g., 16×16, 16×8 and 8×16) can be simplified as the sum of $\mu_{\bar{\mathbf{p}}^i}$ in 8×8 blocks (e.g., 4, 2 and 2, respectively) when a predictor candidate comes from the same location. Thus, the SADAC complexity can be lowered from $4NS$ to $3NS$ (*Fast IC mode*).

Considering different block modes supported in H.264/AVC, complexity for SAD calculation is summarized in Tab. 3.4. For IC, both SAD and SADAC need to be calculated for IC activation thus, the total complexity for IC would be the sum of $14NS+16NS$ (or $13NS$ for fast IC mode). Therefore total complexity with IC is about 2.1 (or 1.9 for fast IC mode) times to H.264/AVC without IC. However, this complexity can be reduced further noting that the same search range is used

Tab. 3.5: Complexity when SAD and SADAC are calculated at the same time. N is the number of pixels in a macroblock and S is the number of search points.

Block Modes	'SAD+SADAC'	'SAD+SADAC' in Fast IC mode
4×4	$2NS$	$2NS$
4×8	$2NS$	$2NS$
8×4	$2NS$	$2NS$
8×8	$5NS$	$5NS$
8×16	$5NS$	$4NS$
16×8	$5NS$	$4NS$
16×16	$5NS$	$4NS$
TOTAL	$26NS$	$23NS$

for SAD and SADAC with IC. In the calculation of SADAC in Tab. 3.3, $B_C - B_R^i$ can be used to calculate SAD, so that SAD and SADAC are calculated at the same time for the same search point, which requires only N operations for SAD instead of $2N$. This leads to a total complexity with IC in fast mode that would be about 1.64 times that of H.264/AVC without IC, as can be seen in Tab. 3.5.

When the fast mode decision algorithm is used, all block sizes are not tested. For example, in [46], 16×16, 8×8 and 4×4 block modes are examined first and their R-D costs are used to decide which block modes are tested further. The complexity of IC in this case can be evaluated by adding the complexities of those block modes that are tested (obtained from Tab. 3.5).

3.6 Simulation Results

The three sequences used in our experiments, *Ballroom*, *Race1* and *Rena*, have different characteristics [1]. All test sequences are 640(w)x480(h) with 8 views that are captured by an array of 8 cameras located horizontally (1-D) with viewing

(a) Ballroom: 8 cameras with 20cm spacing

(b) Race1: 8 cameras with 20cm spacing

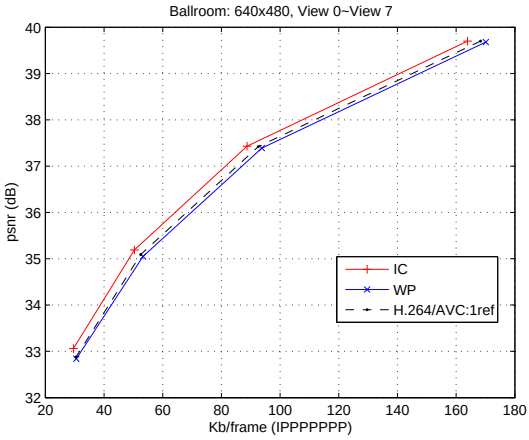
(c) Rena: 8 cameras with 5cm spacing

Fig. 3.5: MVC sequences: 1D/parallel. Sequences are captured by an array of cameras located horizontally with viewing directions in parallel.

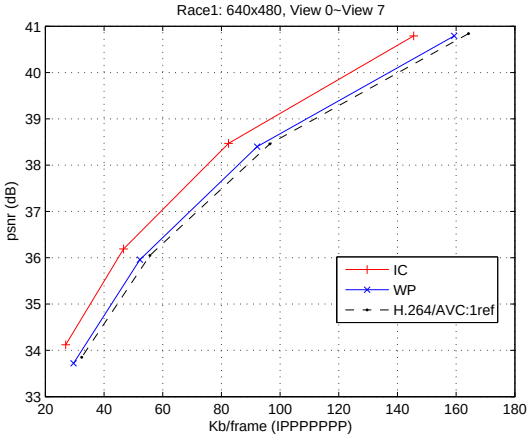
directions in parallel. In Fig. 3.5, sample frames of test sequences are shown. *Ballroom* has the most complicated background and fast moving objects. Objects are located at multiple depths and the distance from the camera to the front objects is small so the disparity of front objects is large. In *Race1*, a mounted and fixed camera array is used to follow racing carts so that there is global motion. Significant luminance and focus changes between views are observed due to imperfect camera calibration and illumination changes are also observed in time because of global motion by camera. In *Rena*, a gymnast moves fast in front of curtains. Distance between cameras is smaller than in the other sequences and luminance and focus changes between views are observed clearly.

Our proposed IC technique is combined with standard H.264/AVC [14] coding tools. IC is enabled only for 16×16, 16×8, 8×16 and 8×8 blocks. While the encoder could be given the option to select whether to use IC on smaller blocks, we observed that this choice was rarely made and thus, for complexity reasons, we choose 8×8 to be the smallest block size. Also IC can be applied in Skip/Direct mode so that model parameters are predicted from neighboring blocks using spatial correlation [22].

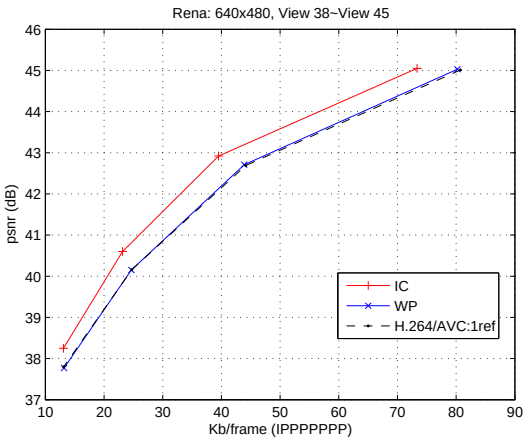
Using the reference codec JM-10.2 [14] as a starting point, we encode frames in cross-view direction only, i.e., we take a sequence of frames captured at the same time from different cameras and feed this to the encoder as if it were a temporal sequence. 8 frames at time stamp 0 are concatenated with 8 frames at time stamp 10. These 16 frames are concatenated again with 8 frames at time stamp 20. By repeating this procedure, we generate a sample sequence with 40 frames from time stamps 0, 10, 20, 30 and 40. By setting the intra period to the number of views, sample sequences are encoded with cross-view prediction.



(a) Ballroom



(b) Race1



(c) Rena

Fig. 3.6: Cross-view coding with IC, at time stamps 0, 10, 20, 30, 40

Tab. 3.6: Percentage of non intra selection in cross-view prediction (% in H.264 → % in H.264+IC). Note that more significant PSNR increases can be observed for those sequences where the increase in number of inter code blocks is greater.

Sequence	QP24	QP28	QP32	QP36
Ballroom	68.8 → 75.2	72.3 → 79.9	73.3 → 81.8	77.2 → 85.6
Race1	53.1 → 71.1	53.4 → 71.2	53.6 → 72.6	54.6 → 73.9
Rena	53.0 → 66.9	54.0 → 70.3	56.0 → 72.3	62.5 → 72.8

We performed simulations with full search, range equal to ±64 pixels, quarter-pixel precision, 1 reference frame, and tested four different QP values (24, 28, 32, 36) to obtain different rate points in Fig. 3.6. It can be seen that for *Race1* and *Rena* there is significant improvement by using IC (0.8 dB) as compared to the results by H.264/AVC because of severe illumination mismatch across views. Instead, *Ballroom* showed small improvement (0.2 dB). *Ballroom* is the most difficult sequence to encode because of its complicated background and irregular disparity field, due to large variances in object depths. Also illumination mismatch is not significant compared to the other sequences. It can be seen that WP does not provide significant coding gains because it cannot compensate severe local mismatches in cross-view prediction. From Tab. 3.6, we can see that the number of blocks in Inter and Skip mode increases once IC is used, which means that disparity search finds more correct matches after compensation. Note that IC gains can be observed even at low bit rates because the selection of IC in each block is optimized based on R-D criteria.

In MVC, multiple references from different time and views are available. For example, if the current frame is at View 2 and time stamp 1 (V2T1) as shown in Fig. 3.7, 4 references are available for current B slice - (V2T0),(V2T2),(V1T1) and (V3T1). In [40], in addition to the two reference lists (L0 and L1) in H.264/AVC,

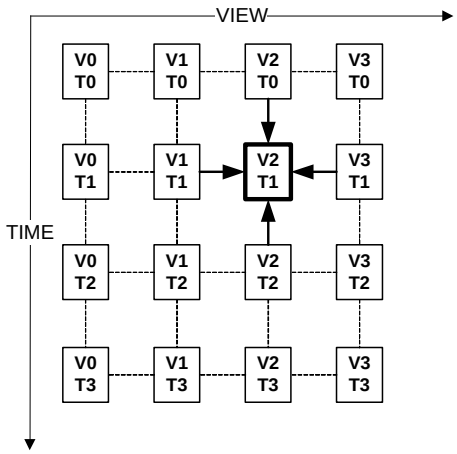


Fig. 3.7: Example of multiple references from different time stamps and views

two view reference lists (VL0 and VL1) are proposed to enable both temporal and cross-view prediction. In [42], a prediction structure using hierarchical B pictures proposed in [30] is adopted as a reference encoder for multi-view video coding. The size of decoded picture buffer (DPB) is increased to store additional reference frames from the other views. The coding structure can be specified using the configuration file of H.264/AVC. An example of this prediction structure is shown in Fig. 3.8 with 8 views and GOP length 8. IBPBPBPP is used for cross-view prediction in anchor frames and hierarchical B is used in temporal prediction. In the even numbered views of non-anchor frames, only temporal prediction is used and in the odd numbered views of non-anchor frames, both temporal and cross-view predictions are used. Note that all B frames in Fig. 3.8 are encoded as B-store frames, i.e., they can be used as references.

Although IC techniques primarily aimed at compensating illumination mismatches in cross-view prediction, they can easily be used to compensate illumination mismatches in temporal prediction, which happens in moving objects and abrupt scene changes. With the prediction structure described in Fig. 3.8, IC is implemented to be applied in both temporal and cross-view prediction [18].

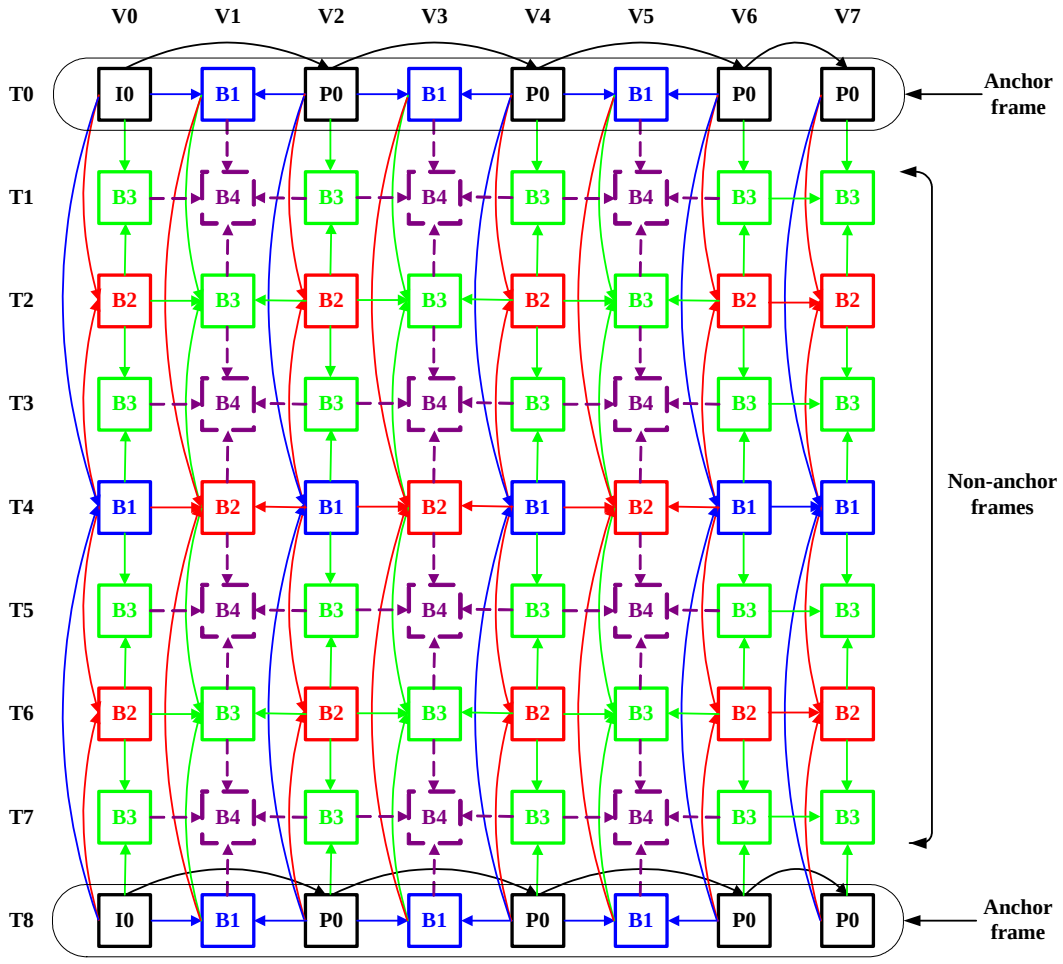


Fig. 3.8: Prediction structure for multi-view video coding with 8 views and GOP length 8

Tab. 3.7: Temporal partitioning of test data sets

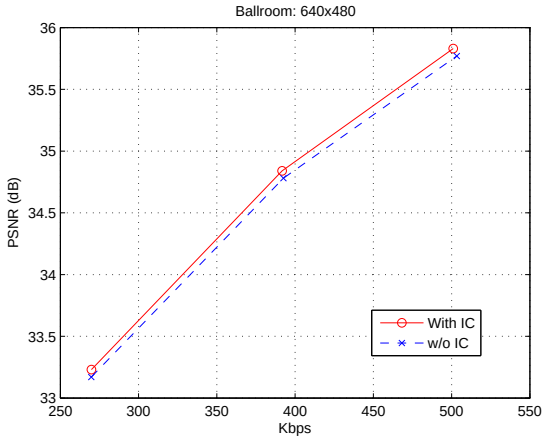
Data set	Temporal Partitioning
Ballroom	250 frames = 20 × GOP_12 + GOP_9
Race1	532 frames = 35 × GOP_15 + GOP_6
Rena	300 frames = 19 × GOP_15 + GOP_14

Fig. 3.9 provides coding results for the parameters (GOP length and total number of frames) of Tab. 3.7. For *Ballroom*, *Race1* and *Rena*, IC achieves 0.1-0.5 dB gains. Overall gains from using IC (as compared to using the same temporal/cross view prediction but no IC) are lower relative to the case where only cross-view prediction is used (Fig. 3.6) because illumination mismatches between frames in time are not as severe as across views and most static background can be efficiently encoded by Skip/Direct mode in temporal prediction. Complete simulation results of proposed IC in MVC for various multi-view test sequences can be found in [18].

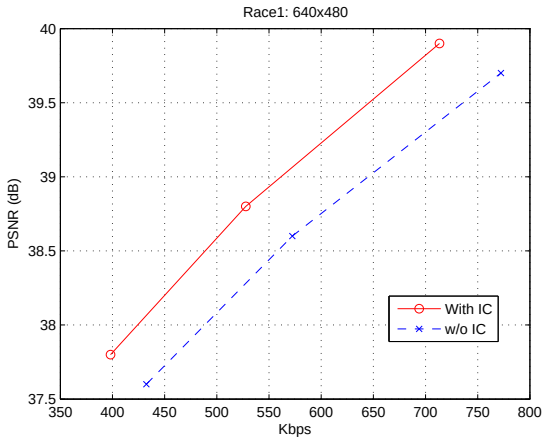
Fig. 3.10 demonstrates coding results of IC and WP in MVC. In this comparison, 73, 76 and 31 frames/view (rather than the complete sequences as in Tab. 3.7 to lower encoding complexity) are encoded for *Ballroom*, *Race1* and *Rena*, respectively. IC achieves higher coding efficiency as compared to WP. In particular for *Race1*, IC achieves a 0.5 dB gain over WP. More detailed comparisons of IC with WP in MVC for various multi-view test sequences can be found in [22].

3.6.1 Combined Solution with ARF

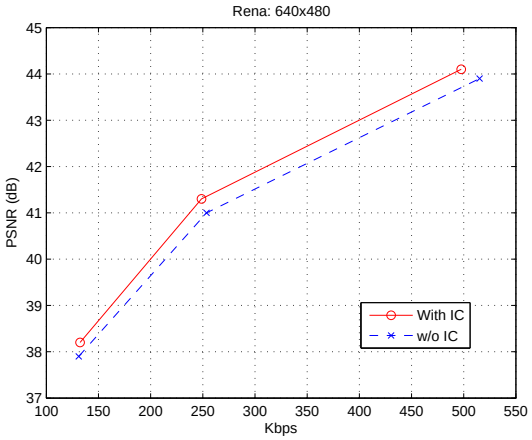
In [21], adaptive reference filtering (ARF) is proposed to compensate focus mismatch in cross-view prediction. To compensate both illumination and focus mismatches in cross-view prediction, IC and ARF techniques are combined [17]. Mean-removed-search (MRS) is adopted to remove redundancies in combined system so



(a) Ballroom

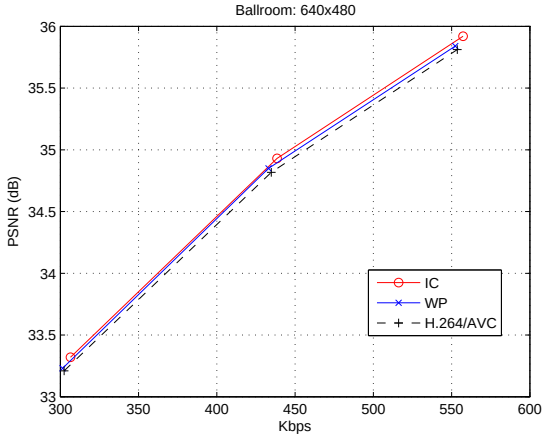


(b) Race1

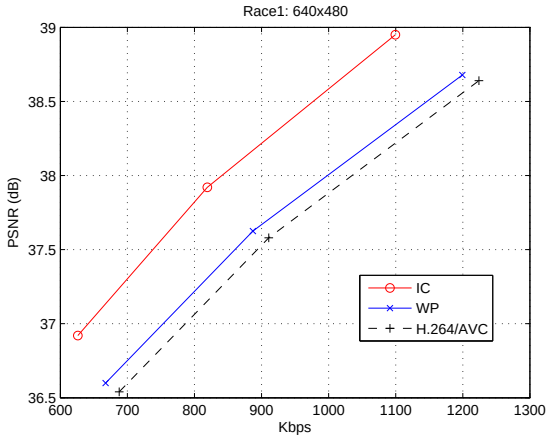


(c) Rena

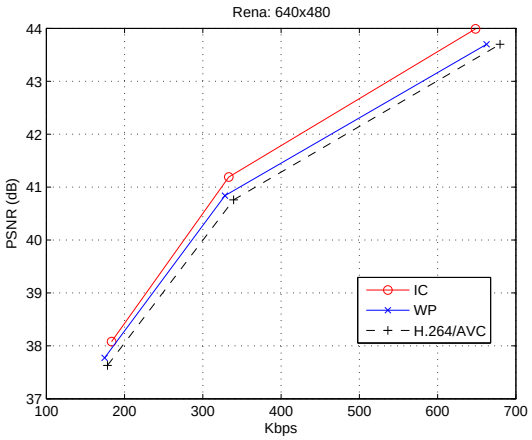
Fig. 3.9: Multi-view coding with IBPBPBPP cross-view, hierarchical B temporal [2]



(a) Ballroom (73 frames/view)



(b) Race1 (76 frames/view)



(c) Rena (31 frames/view)

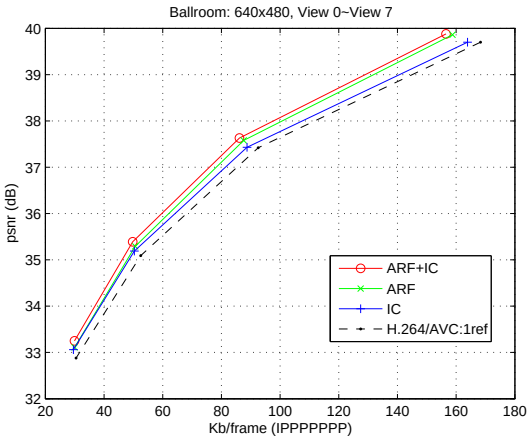
Fig. 3.10: Comparison of IC with WP in MVC with IBPBPBPP cross-view, hierarchical B temporal [2]

that DC and AC compensation is performed by IC and ARF respectively. From the matches by MRS (first search), ARF filter coefficients are calculated and additional reference frames are generated by filtering the original reference frame. Finally, IC is applied for the disparity search (second search) with the original reference frame and reference frames generated by ARF. Since the different filtered references created by ARF come from the same original reference frame, the disparity fields obtained from the first (MRS) and second (IC) search should not be very different. Complexity reduction can be achieved by taking the disparity field obtained from MRS search as predictor for the second search with a much reduced search range.

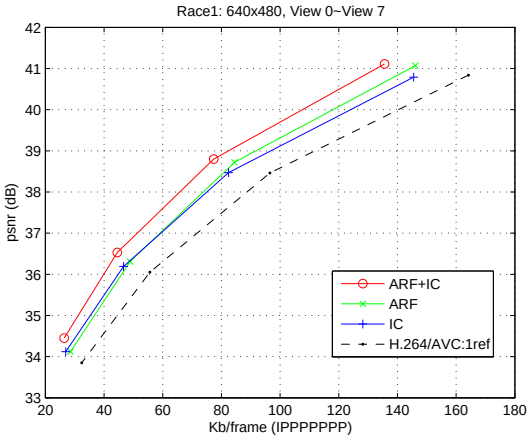
Under the same coding conditions used for Fig. 3.6, we encode frames in cross-view prediction using the combined system. The simulation results are shown in Fig. 3.11. For *Ballroom*, block-wise IC alone provides very limited gain so that the combined system also barely outperforms the ARF coding. On the other hand, ARF and IC each achieve 0.5∼0.8 dB gain for *Race1* and *Rena*. The combined system produces an additional 0.5 dB gain over either IC only or ARF only. The overall coding gain, as compared to using H.264/AVC with 1 reference for cross-view coding, is about 0.5 dB for *Ballroom*, about 1.3 dB for *Race1* and about 1 dB for *Rena.*

3.7 Conclusions

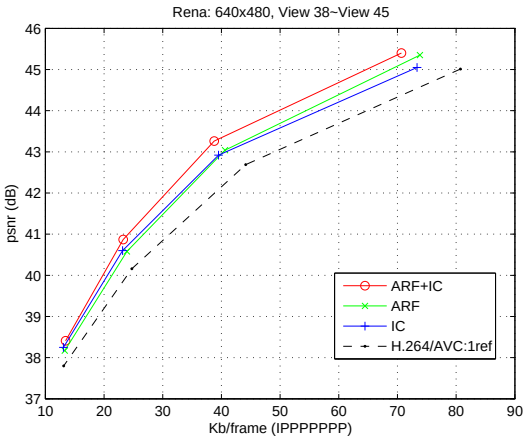
To compensate localized illumination mismatches across different views in multi-view systems, block-wise illumination compensation techniques are proposed. IC coding tools are developed from the corresponding mismatch models and show significant gains over standard H.264/AVC in cross-view prediction. The proposed



(a) Ballroom



(b) Race1



(c) Rena

Fig. 3.11: Cross-view coding with H.264/AVC, IC, ARF and ARF+IC at time stamps 0, 10, 20, 30, 40

techniques are applied to a general multi-view video coding system where both temporal and cross-view prediction are used and to more general prediction structures. Simulation results show that, when performing predictive coding across different views in multi-view systems and in general multi-view video coding, our proposed methods provide higher coding efficiency than other advanced coding tools. Joint coding benefit and complexity of the combined system are discussed and an improved coding algorithm is presented.

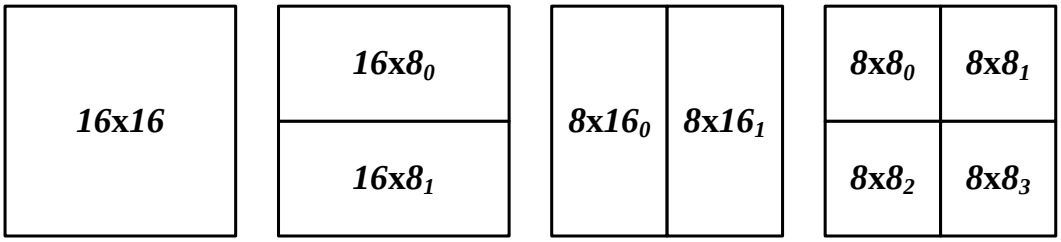
Chapter 4

Implicit Block Segmentation

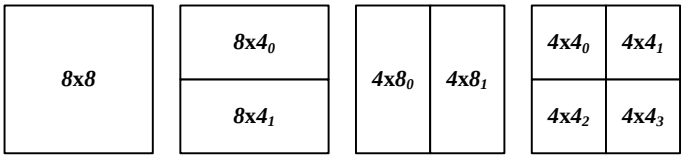
4.1 Preliminaries

In Chapter 3, a predictor from each reference is compensated using the IC model in order to minimize the residual error with respect to the original block signal. This additional IC model was introduced because the true match is corrupted by brightness variations in the multi-view system. In this chapter, assuming that references are not corrupted, we propose a technique to improve predictor quality for the original macroblock.

Exploiting inter-frame correlation via motion estimation is key in achieving high video compression efficiency. Block-based motion estimation and compensation provides a good balance between prediction accuracy and rate overhead. Clearly, blocks of pixels are not guaranteed to have uniform displacement across frames. For video sequences this is the case if an object boundary exists in a block and pixels which belong to different objects move in different ways. In stereo or multi-view sequences this is the case if an object boundary formed by objects in different depths exists in a block and pixels which belong to different objects are occluded or uncovered due to disparity.



(a) Block mode - $INTER16 \times 16$, $INTER16 \times 8$, $INTER8 \times 16$ and $INTER8 \times 8$



(b) Sub-block mode for $INTER8 \times 8$

Fig. 4.1: Inter block modes in H.264/AVC. Each 8×8 sub-block in 4.1(a) can be split into different sub-block sizes as in 4.1(b).

Numerous approaches have been proposed to provide more accurate motion compensation by providing different prediction for different regions in a macroblock. Examples include techniques used in the H.264/AVC video coding standards [43] or the hierarchical quad-tree (QT) approach [38]. In these methods a macroblock is split into smaller blocks and the best match for each block is searched. As the number of blocks in a macroblock increases, overhead increases while distortion between the original and the match decreases. Therefore, there is an optimal point in terms of rate-distortion behavior so that the best block mode can be decided based on Lagrangian techniques. Fig. 4.1 depicts different block modes available in H.264/AVC. The R-D costs from all candidate block modes are computed in inter frame prediction and the block mode with minimum R-D cost is chosen. For 8×8 block mode in Fig. 4.1(a), a block can be further split into 4 sub-blocks as shown in Fig. 4.1(b). To increase the quality of matching achievable by square or rectangular block shapes available in QT, a geometry based approach (GEO) is proposed in [9, 13]. A block is split into two smaller regions called

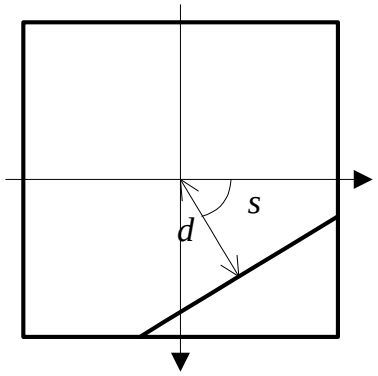


Fig. 4.2: A straight line in GEO is defined by slope s and distance d from center in 16×16 macroblock.

wedges by a line described by a slope (s) and a translation parameter (d) as shown in Fig. 4.2. These parameters and matching wedges are jointly estimated for each candidate within the motion search. Although GEO captures object boundaries better than QT, it is still limited in that the boundary has to be a straight line. Furthermore, the search for the best slope and translation parameters combined with motion search increases the complexity significantly.

In [33], an object based motion segmentation method is proposed to solve the occlusion problem. To estimate different motions in a block, motion vectors from neighboring blocks are copied after block segmentation. To avoid transmitting segmentation information, previously encoded frames at $(t-1)$ and $(t-2)$ are used to estimate segmentation for the current frame at (t). However, since only motion vectors in neighboring blocks are used to estimate motion, the accuracy of this estimation may suffer.

In this chapter, we present a framework for implicit block segmentation to improve prediction quality. Implicit block segmentation is obtained based on the predictors from previously encoded frames as in [33]. However, segmentation is applied to the difference of two predictors, rather than directly to the predictor

itself. Also, unlike in [33], motion vectors are explicitly transmitted to signal the location of chosen predictors and the encoder searches for the best combination of predictors. We use 16×16 macroblocks, which are assumed to be small relative to typical objects in the scene, so that in many cases at most two objects[1] move with different displacements at the boundaries [33]. Although distortion can be reduced as the number of predictors increases, the overhead required for motion/disparity vectors and for identifying the selected predictor for each segment also increases with the number of predictors. While the number of predictors can be optimally chosen based on R-D cost (as is done in the hierarchical quad-tree case), in this work for simplicity we choose the maximum number of predictors to be two.

4.2 Implicit Block Segmentation

4.2.1 Motivation from Block Motion Compensation

Fig. 4.3 shows an example of block motion estimation between current and reference frame. In the current block, we have two objects which are separated by a smooth boundary. Let us assume that the correct matches of each object can be found as a base predictor ($\bar{\mathbf{p}}_0$) and an enhancement predictor ($\bar{\mathbf{p}}_1$) as shown in the reference frame. For the current macroblock signal $\bar{\mathbf{x}}$,[2] QT and GEO find best predictors by selecting the best match for regions as defined in those algorithms (i.e., constrained to be rectangular regions or to have a straight line boundary). Therefore, although the correct matches for each object are given as $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$, neither QT nor GEO finds a correct match without significant prediction error in

[1] Thus, the initial number of segments, N_c in K-means clustering algorithm in 4.2.2 is set to 2.

[2] The vector notations $\bar{\mathbf{x}}$, $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ are used to represent block signals. For pixel data or random variables, the terms x, p_0 and p_1 are used.

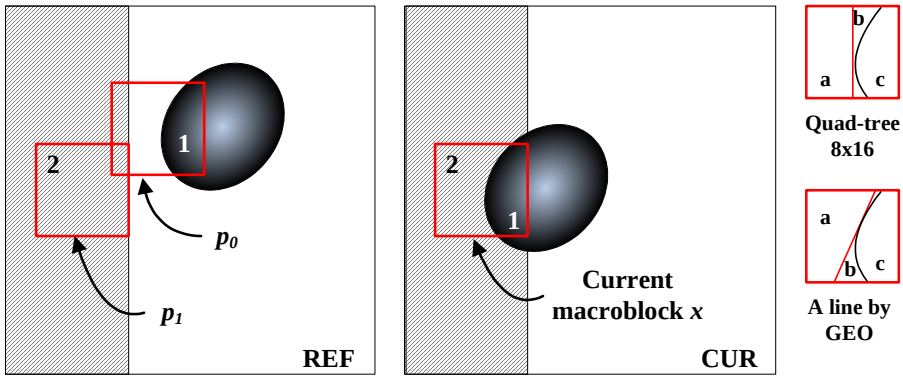


Fig. 4.3: Example of block motion compensation. The best match of current macroblock $\bar{\mathbf{x}}$ can be found in two locations for different objects. However, in region b of matches by QT and GEO, significant prediction error exists.

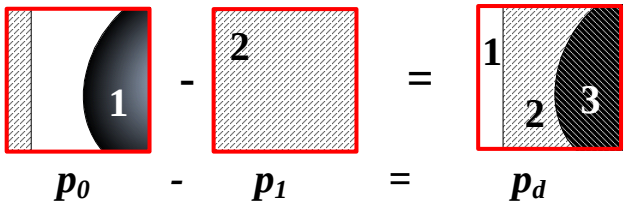


Fig. 4.4: Definition of predictor difference $\bar{\mathbf{p}}_d$. Pattern of predictors are from Fig. 4.3.

some regions (e.g., those labeled b in Fig. 4.3) because object boundaries are not necessarily well described by a straight line.

Following the same example, in Fig. 4.4 we depict the difference between the two predictors, $\bar{\mathbf{p}}_d = \bar{\mathbf{p}}_0 - \bar{\mathbf{p}}_1$. In region 1 of $\bar{\mathbf{p}}_d$, the absolute difference of pixel values is small because $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ come from the same object and both $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ will estimate the original block with small error. Therefore, the difference in residual error when using the two predictors (i.e., $|\bar{\mathbf{x}} - \bar{\mathbf{p}}_0| - |\bar{\mathbf{x}} - \bar{\mathbf{p}}_1|$) will tend to be small, which means either predictor would be a good estimate of the original signal. In regions 2 and 3 of $\bar{\mathbf{p}}_d$, $\bar{\mathbf{p}}_1$ and $\bar{\mathbf{p}}_0$ provide the best match, respectively. Thus, the absolute difference between the two predictors will tend to be large, and

we similarly would expect that the differences in residual error after prediction will be large.

For each region several scenarios are possible. In the area where $|\bar{\mathbf{p}}_d|$ is small, because the two predictors are similar we have that either i) both predictors provide a good match or ii) the residual error is large with respect to both predictor and choosing one of the predictors over the other will not lead to significant improvements. Instead, in areas where $|\bar{\mathbf{p}}_d|$ is large, either i) only one of the two predictors provides a good match, or ii) a combination of both predictors may lead to a better matching performance. Clearly, choosing the "right" predictor among the two available choices is more important for regions where $|\bar{\mathbf{p}}_d|$ is large; it is in these regions where signaling a predictor choice can lead to a more significant gain in prediction performance.

We propose implicit block segmentation (IBS), where each macroblock is segmented first, based on these observations. For each segment, weights are chosen so that the prediction generated by the weighted sum of predictors minimizes residual error. An estimate of the original macroblock is obtained by combining predictions for each segment. Next, a block based segmentation method is proposed.

4.2.2 Block Based Segmentation

Assume two predictors are available for a given macroblock (i.e., two 16×16 blocks from neighboring frames). These two predictors have been chosen by the encoder and their positions will be signaled to the decoder. The optimal segmentation for the purpose of prediction would be such that each pixel in the original macroblock is assigned to whichever predictor, $\bar{\mathbf{p}}_0$ or $\bar{\mathbf{p}}_1$, provides the best approximation.

However this cannot be done implicitly (without sending side information) since the decision depends on the original block itself.

In [33], MAP estimation of block segmentation is proposed based on a Markov random field (MRF) model. This Bayesian image segmentation method provides optimized segmentation results for given probability models. If the original block signal to be encoded ($\overline{\mathbf{x}}$), base predictor ($\bar{\mathbf{p}}_0$) and enhancement predictor ($\bar{\mathbf{p}}_1$) are given, MAP segmentation ($\hat{\mathbf{s}}$) can be found as

$$\hat{\mathbf{s}} = \arg\min_{\bar{\mathbf{s}}} P(\bar{\mathbf{s}}|\overline{\mathbf{x}}, \bar{\mathbf{p}}_0, \bar{\mathbf{p}}_1) \tag{4.1}$$

However, it is difficult to find the segmentation minimizing (4.1) with reasonable computational complexity considering that $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ have to be jointly searched. Thus, this approach is not applied in this work. When the depth information of a frame is available, object boundaries can be extracted by segmenting the depth map. Because occlusions and uncovered regions are caused by moving objects in different depths, better matches can be found for the segments using the depth map. However in our work, we assume that auxiliary information, such as depth maps, is not available.

Based on our previous observations about the expected gains depending on the differences between predictors, we apply segmentation to the block of predictor differences, $\bar{\mathbf{p}}_d$. Due to the noisy characteristics of predictor differences, edge based segmentation methods do not detect simple boundaries efficiently in 16×16 macroblocks. In this work, K-means clustering [29] is used as a basic segmentation algorithm. To take the spatial information of pixels into account with the pixel value of predictor difference, 3-D K-means clustering algorithm can be used taking horizontal (x), vertical (y) location and predictor difference (p_d) as three inputs.

Because of different ranges for (x, y) and p_d, p_d needs to be scaled before K-means clustering. However, the segmentation results are quite sensitive to this scaling factor and an accurate scaling factor is hard to find because the range of p_d changes depending on the disparities between base and enhancement predictors. Therefore, instead of 3-D K-means clustering, 1-D K-means clustering followed by two step post-processing is adopted. The input to the K-means clustering is the pixel value of predictor difference $\bar{\mathbf{p}}_d$ in 16×16 macroblock. N_c centroids are initialized uniformly spaced between maximum and minimum value of $\bar{\mathbf{p}}_d$. The maximum number of iterations N_{it} is set to 20. According to the minimum distance to N_c centroids, pixels are classified into the N_c segments. After 1-D K-means clustering, disconnected pixels exist within each segment because spatial connectivity is not considered in 1-D K-means clustering. A two step post-processing is applied to take spatial information into account. First, using connected component labeling [7], disconnected pixels assigned to the same segment are classified into different segments. Second, to prevent the occurrence of segments due to noise, if the number of pixels in a segment is smaller than a threshold, N_{th}, it is merged into the neighboring segment that has the minimum segment-mean difference with current segment. Fig. 4.5 depicts this post-processing. Note that the number of segments depends on the disparities between base and enhancement predictors. In this work, N_c and N_{th} are set to be 2 and 10, experimentally.

4.2.3 Weighted Sum of Predictors

For each segment k in $\bar{\mathbf{p}}_d$, the optimal predictor $\hat{\mathbf{x}}^k$ can be calculated as a weighted sum of base and enhancement predictors when the original $\bar{\mathbf{x}}$ is known. If scalar

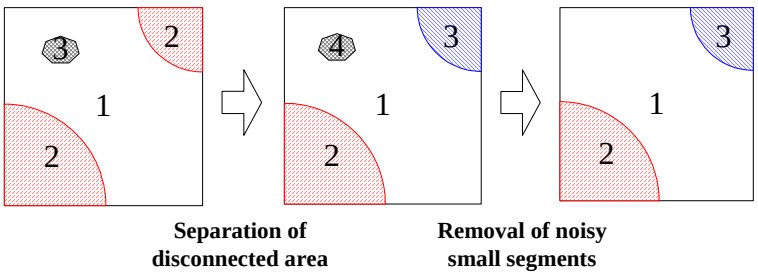


Fig. 4.5: Example of two step post-processing after 1-D K-means clustering. First, disconnected segment 2 is classified as different segment increasing the number of segment N from 3 to 4. Second, segment 4 is merged into segment 1 decreasing N to 3 again.

weights α_0^k and α_1^k are applied to all pixels in segment k of $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$, the sum of squared difference (SSD) for the segment k is

$$SSD^k = ||\bar{\mathbf{x}}^k - \hat{\mathbf{x}}^k||^2 = ||\bar{\mathbf{x}}^k - (\alpha_0^k \bar{\mathbf{p}}_0^k + \alpha_1^k \bar{\mathbf{p}}_1^k)||^2, \tag{4.2}$$

where $\bar{\mathbf{p}}_0^k$ and $\bar{\mathbf{p}}_1^k$ specifies the pixels of $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ belonging to segment k. By setting to zero the gradient of (4.2) with $\alpha_0^k + \alpha_1^k = 1$, optimal weights can be found as

$$\begin{aligned} \alpha_0^k &= \frac{-(\bar{\mathbf{p}}_1^k - \bar{\mathbf{x}}^k) \cdot \bar{\mathbf{p}}_d^k}{||\bar{\mathbf{p}}_d^k||^2} \\ \alpha_1^k &= \frac{(\bar{\mathbf{p}}_0^k - \bar{\mathbf{x}}^k) \cdot \bar{\mathbf{p}}_d^k}{||\bar{\mathbf{p}}_d^k||^2}. \end{aligned} \tag{4.3}$$

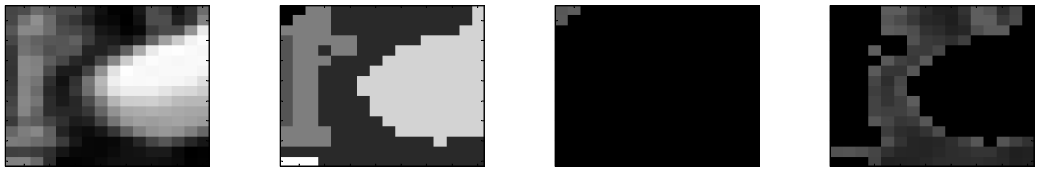
Because the optimal α_0^k is calculated using information from the block to be encoded, the chosen value has to be signaled. For 16×16 blocks, this signaling overhead may not be justified given the overall reductions in residual error. Also the complexity to find the optimal weight is significant due to the multiplications and the divisions in calculation, which increases as the predictors $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ are jointly searched during the motion searching step. Therefore instead of finding

optimal weight for each block and signaling them, we define W, a set of weights most frequently chosen and find the one with minimum distortion in each segment. In this work, W is chosen to be $\{(1,0),(0,1),(\frac{1}{2},\frac{1}{2})\}$, which corresponds to predictors $\{\bar{\mathbf{p}}_0, \bar{\mathbf{p}}_1, \frac{1}{2}(\bar{\mathbf{p}}_0 + \bar{\mathbf{p}}_1)\}$ respectively. The additional weight $(\frac{1}{2},\frac{1}{2})$ has been selected as the one that is most frequently chosen, as shown in Appendix B. Thus a weight index with only three values $\{0,1,2\}$ has to be signaled. Note that it is easy to extend this framework by including additional weights in W. With binary arithmetic coding or variable length coding of weight indices, a given weight will be chosen only if it leads to gains in an R-D sense.

In summary, prediction for the block to be encoded is achieved by signaling the two predictors, $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$, and the weights to be used for each segment, w_k. The segmentation itself is generated by encoder and decoder in the same manner from the decoded predictors, so that there is no need for side information to be sent.

4.2.4 Joint Search of Base and Enhancement Predictors

Since prediction is performed by combining two predictors using proposed IBS technique, there is no guarantee that one can obtain the best matching pair of predictors by searching for each predictor individually using standard residual energy metrics based on the whole 16×16 block. In theory one would have to search for *pairs* of predictors, i.e., for each base predictor candidate, it would be necessary to search all candidate enhancement predictors and choose the best one by computing the prediction residue *after segmentation and combined base/enhancement prediction*. If the number of locations in search window is denoted N_S, this pair-wise search would have N_S^2 pairs of candidates when all candidates in search range are tested for base predictor. For example, for 32×32 full search window, N_S is 1024



(a) Original macroblock (b) All segments (c) Segment-0 (d) Segment-1

(e) Segment-2 (f) Segment-3 (g) Segment-4 (h) Segment-5

Fig. 4.6: After segmentation of the original macroblock from *MERL_ballroom* sequence, the best matches for the segments are added to the set of base predictor candidates.

thus, $N_S^2 = 1048576$. As an alternative solution to individual search or pair-wise search, we start by obtaining a set of base predictor candidates. First, the original macroblock is segmented as shown in Fig. 4.6 and the best matches for the segments are collected as good base predictor candidates by SAD distortion measure. Then, for each base predictor candidate in the set, we perform the joint search for enhancement predictor. For the example of Fig. 4.6, a total of 6 pairs of base and enhancement predictors will be found.

Fig. 4.7 illustrates the IBS search loop of the enhancement predictor $\bar{\mathbf{p}}_1$ for given base predictor $\bar{\mathbf{p}}_0$. To decide the best pair of base and enhancement predictors, three decisions should be made. First, *for each segment*, the best weight index should be decided. Second, *for each base predictor*, the best complementary enhancement predictor should be chosen. Third, *for the given macroblock*, the best pair of base and enhancement predictor should be decided for IBS. These decisions are made based on three different error metrics explained next.

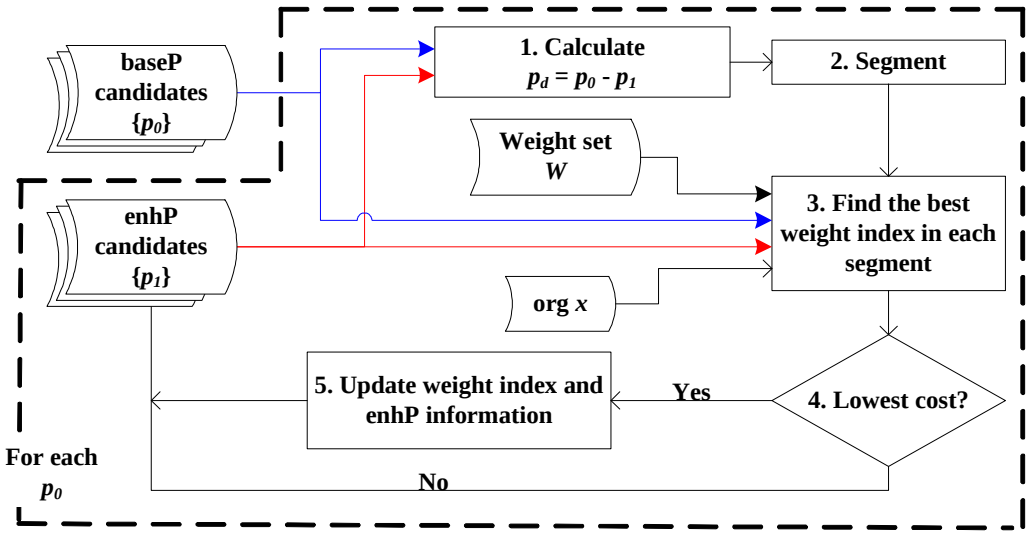


Fig. 4.7: Search loop of enhancement predictor for given base predictor

4.2.5 Three Error Metrics in Joint Search

In our proposed approach a predictor for the original block is generated by combining the best prediction in each segment from $\bar{\mathbf{p}}_0$, $\bar{\mathbf{p}}_1$ and $\bar{\mathbf{p}}_a$, where $\bar{\mathbf{p}}_a$ is defined as an average of $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$. Therefore, the first error metric is used to decide which weight index or predictor is used in each segment. In the comparison of three predictors, SSD can be used as a distortion measure, but due to the multiplication complexity in SSD, SAD is adopted instead. In Appendix C, it is shown that there is no penalty for SAD if the residuals are normally distributed. If the residuals follow a Laplace distribution, the residual distortion by SAD can increase up to 11% when $\frac{\sigma_0^2}{\sigma_1^2} \in (\frac{1}{3}, 0.382)$ or $(2.618, 3)$. However, the probability, $P\left(\frac{\sigma_0^2}{\sigma_1^2} \in (\frac{1}{3}, 0.382) \text{ or } (2.618, 3)\right)$ is relatively low (for example, less than 7% from coding results of *Foreman* with QP 24) thus, on average this penalty is negligible. SAD for k^{th} segment is defined as

$$SAD_k = \min_{\bar{\mathbf{p}}_j} \sum_{i \in SEG_k} |\bar{\mathbf{x}}(i) - \bar{\mathbf{p}}_j(i)| \tag{4.4}$$

and the associated weight index is

$$w_k = \arg\min_j \sum_{i \in SEG_k} |\bar{\mathbf{x}}(i) - \bar{\mathbf{p}}_j(i)| \tag{4.5}$$

where $\bar{\mathbf{x}}(i)$ denotes the original signal at the pixel location i and SEG_k denotes the k^{th} segment. The total distortion for the macroblock would be the sum of SAD_k over all the segments, $\sum_{k=1}^{N_{seg}} SAD_k$.

Now, the total SAD for a given enhancement predictor candidate, $\sum_{k=1}^{N_{seg}} SAD_k$, is calculated for every enhancement predictor candidate within search range. These values should be compared to decide what is the best complementary pair for the given base predictor candidate. Because the location of base and enhancement predictor is signaled, the motion vector cost of enhancement predictor candidate needs to be added to J_{enh}, the total cost by the enhancement predictor candidate. Because the enhancement predictor candidates are compared for the given base predictor candidate, the motion vector cost of the base predictor is not added to J_{enh}. Also different enhancement predictors would lead to a different segmentation and as the number of segments increases, total distortion decreases and the signaling cost of weight indices increases. Thus, in order to consider enhancement predictor selection based on a rate and distortion trade-off, we define a new cost metric as:

$$J_{enh} = \sum_{k=1}^{N_{seg}} SAD_k + \sqrt{\lambda} N_{seg} \lceil \log_2 N_w \rceil + \sqrt{\lambda} C_{mv}(\bar{\mathbf{p}}_1) \tag{4.6}$$

where N_{seg} is the number of segments, N_w is the number of weight indices and $C_{mv}(\cdot)$ is the signaling cost of motion vector. In (4.6), $N_{seg} \lceil \log_2 N_w \rceil + C_{mv}(\bar{\mathbf{p}}_1)$ corresponds to the signaling bits for weight index and motion vector. Considering

$\sum_{k=1}^{N_{seg}} SAD_k$ is the SAD distortion measure, not SSD, $\sqrt{\lambda}$ is used as a scaling factor instead of λ [44]. In the implementation of IBS, C_{mv} and λ follow the definition in H.264/AVC reference codec.

If we pick the best enhancement predictor for the given base predictor using J_{enh}, for M base predictor candidates, equal numbers of matching enhancement predictors will be found. Finally, R-D cost of M base and enhancement predictor pairs are calculated to decide the best pair for IBS.

4.2.6 IBS algorithm in H.264/AVC

We summarize the IBS algorithm when it is implemented as an additional block mode ($INTER16 \times 16\_IBS$) in the H.264/AVC.

IBS Algorithm:

1. Collect base predictor candidates
   (a) Apply 1-D K-means clustering to the original macroblock followed by two step post-processing and find segments
   (b) During motion/disparity search for $INTER16 \times 16$, find a match for each segment of the original macroblock from *Step-(1a)* and form W, a set of base predictor candidates
2. ($INTER16 \times 16\_IBS$ block mode) For each base predictor candidate ($\bar{\mathbf{p}}_0$) from W, the complementary enhancement predictor is searched within search window. Each enhancement predictor candidate in search window is denoted as $\bar{\mathbf{p}}_1$.
   (a) Calculate predictor difference, $\bar{\mathbf{p}}_d = \bar{\mathbf{p}}_0 - \bar{\mathbf{p}}_1$

(b) Apply 1-D K-means clustering to $\bar{\mathbf{p}}_d$ followed by two step post-processing and find segments

(c) For each segment of $\bar{\mathbf{p}}_d$ from *Step-(2b)*, find the weight index minimizing SAD_k as shown in (4.4) and (4.5) and generate new prediction for the original macroblock

(d) Calculate J_{enh} in (4.6). If J_{enh} is the minimum, save $\bar{\mathbf{p}}_1$ as the best enhancement predictor to $\bar{\mathbf{p}}_0$.

(e) Repeat *Step-(2a)* - *Step-(2d)* until there is no more enhancement predictor candidate in search window

3. Calculate R-D costs of the pairs found in *Step-(2)* and find the pair with minimum R-D cost

4. Compare R-D cost with other QT block modes and choose the one with minimum R-D cost as the best block mode (R-D mode decision)

4.3 Complexity of IBS

The impact of IBS on encoding complexity is mostly due to joint search of base and enhancement predictor, where for each pair of base and enhancement predictor candidates, segmentation based on the predictor difference is applied and in each segment, the weight with minimum distortion is selected (other changes to encoder such as encoding of weight index and R-D based IBS mode decision have a negligible effect on overall complexity). Therefore, in what follows, the complexity of motion/disparity estimation in IBS is analyzed in terms of arithmetic operations, e.g., addition and multiplication. Tab. 4.1 explains the symbols used in this analysis. Assuming that n-bit integers are used to represent pixel values,

Tab. 4.1: Definition of symbols in complexity analysis. The integers in parenthesis besides N_x are the values used in the simulation.

Variable	Meaning	Complexity	Meaning
$N_p(256)$	# of pixels in a macroblock	$C_+ : O(n)$	addition or substraction
$N_{it}(20)$	# of iteration (K-means)	$C_\times : O(m)$	integer multiplication or division
$N_c(2)$	# of centroids (K-means)	$C_{\|\|} : O(1)$	absolute operation
$N_w(3)$	# of weights	$C_s : O(1)$	shift operation

addition/substraction can be done in $O(n)$ and multiplication/division can be done in $O(n^2)$ for the worst case. Depending on the algorithm used for multiplication, the complexity of multiplication/division can be different, and thus the complexity of multiplication is denoted as $O(m)$ as in Tab. 4.1. Because absolute or shift operations are applied to the whole number, $C_{||}$ and C_s is equal to $O(1)$.

We start by analyzing the complexity of segmentation by 1-D K-means clustering algorithm. Tab. 4.2 summarizes the complexity for each step in the K-means clustering algorithm. To find predictor difference $\bar{\mathbf{p}}_d = \bar{\mathbf{p}}_0 - \bar{\mathbf{p}}_1$, N_p subtractions are required. Then, 1-D K-means clustering is applied to $\bar{\mathbf{p}}_d$ with the maximum number of iterations, N_{it}. At each iteration, pixels are classified into the bins according to the distance to the centroids. Let c_k and $d_k(i)$ denote the k^{th} centroid and the distance of pixel i to c_k then, $d_k(i) = |\bar{\mathbf{p}}_d(i) - c_k|$. This distance should be calculated for all the pixels in the macroblock with respect to all centroids, thus the complexity would be $N_c N_p(C_+ + C_{||})$. After pixel classification, the centroids (c_k) are updated based on the pixels in the same bins (BIN_k) as $c_k = \frac{\sum_{i \in BIN_k} \bar{\mathbf{p}}_d(i)}{\sum_{i \in BIN_k} 1}$ with $N_p C_+ + N_c C_\times$ complexity.

Secondly, in each segment the best weight is chosen by comparing the distortions for all weight configurations. With SAD distortion measure, Tab. 4.3

Tab. 4.2: Complexity analysis of K-means clustering

Predictor difference	1-D K-means clustering	
	Pixel classification	Centroid update
$\bar{\mathbf{p}}_d(i) = \bar{\mathbf{p}}_0(i) - \bar{\mathbf{p}}_1(i)$ $\Rightarrow N_p C_+$	$D_k(i) = \lvert\bar{\mathbf{p}}_d(i) - c_k\rvert$ $\Rightarrow N_c N_p (C_+ + C_{\lvert\rvert})$	$c_k = \frac{\sum_{i \in BIN_k} \bar{\mathbf{p}}_d(i)}{\sum_{i \in BIN_k} 1}$ $\Rightarrow N_p C_+ + N_c C_\times$
$N_p C_+$	$N_{it}(N_c N_p (C_+ + C_{\lvert\rvert}) + N_p C_+ + N_c C_\times)$	
TOTAL: $N_p(1 + N_{it} N_c + N_{it}) C_+ + N_{it} N_c C_\times + N_p N_{it} N_c C_{\lvert\rvert}$		

Tab. 4.3: Complexity analysis of weight index decision for each $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ pair

New predictor generation	Weight index selection (SAD)
$\bar{\mathbf{p}}_j(i) = (\alpha_j \bar{\mathbf{p}}_0(i) + (1 - \alpha_j)\bar{\mathbf{p}}_1(i)) >> r$ for N_p pixels $\Rightarrow N_p(2C_\times + C_+ + C_s)$	$w_k =$ $\arg\min_j \sum_{i \in SEG_k} \lvert\bar{\mathbf{x}}(i) - \bar{\mathbf{p}}_j(i)\rvert$
for $j \geq 2 \Rightarrow (N_w - 2) N_p (2C_\times + C_+ + C_s)$	$\Rightarrow N_w N_p (2C_+ + C_{\lvert\rvert})$
TOTAL: $N_p(3N_w - 2)C_+ + 2N_p(N_w - 2)C_\times + N_p(N_w - 2)C_s + N_p N_w C_{\lvert\rvert}$	
for $N_w = 3$ with the weight $(\frac{1}{2}, \frac{1}{2}) \Rightarrow 7N_p C_+ + N_p C_s + 3N_p C_{\lvert\rvert} \sim 7N_p C_+$	

summarizes the complexity of the weight index decision for each $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ pair. Because $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ are given, the number of additional predictor is $N_w - 2$, which are generated as a weighted sum of $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$. It is assumed that multiplication by a weight with floating point precision can be replaced with the multiplication by the integer weight α_j and shift operation by r. To find the weight index, SAD's for N_w weights are calculated in each segment, which corresponds $N_w N_p (2C_+ + C_{||})$. If the only $\bar{\mathbf{p}}_a$ is the average of $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_1$ computed with the weights $(\frac{1}{2}, \frac{1}{2})$, $N_w = 3$ and multiplication can be skipped in the calculation of $\bar{\mathbf{p}}_a$ so that the complexity for predictor generation would be $N_p(C_+ + C_s)$.

Using the definitions in Tab. 4.1, the complexity for K-means clustering algorithm is approximated as $O(n)N_p(1 + N_{it}N_c + N_{it}) + O(m)(N_{it}N_c) = O(61nN_p) + O(40m)$ and the complexity for the weight index decision is approximated as $O(7nN_p)$.

Tab. 4.4: Comparison of IBS and GEO complexity. M is the number of base predictor candidates.

	IBS	GEO
Complexity	$M\,(O(68nN_p) + O(40n^2))$ $\sim O(68nN_pM)$	$O(4024nN_p)$ or $O(154nN_p)$
for $n = 8$, $M = 10$ and $N_p = 256$	$O(5440N_p) + O(25600)$ $\sim O(5440N_p)$	$O(32192N_p)$ or $O(1232N_p)$

In GEO [13], 2012 or 77 (fast mode) wedge partitions are compared to find the slope and the displacement for 16×16 macroblock. If SAD is used as a distortion measure, this corresponds to $2012N_p(2C_+ + C_{||})$ ($\sim O(4024nN_p)$) or $77N_p(2C_+ + C_{||})$ ($\sim O(154nN_p)$) in fast mode. In Tab. 4.4, the complexities of IBS and GEO are compared when the number of base predictor candidates is $M = 10$. The complexity of IBS is in between that of the original and that of fast mode of GEO.

4.4 Simulation Results

4.4.1 Implementation within an H.264/AVC Architecture

Implicit block segmentation is implemented in the H.264/AVC reference codec - *JSVM 8.4*. Current inter block modes are extended inserting $INTER16 \times 16\_IBS$ between $INTER16 \times 16$ and $INTER16 \times 8$. The R-D optimization tool in H.264/AVC is applied to choose the best mode for each macroblock.

To find base predictor candidates, the original macroblock is segmented first. If N_{org} segments are obtained after post-processing, N_{org} best matches for the segment are found during $INTER16 \times 16$ motion search as reliable base predictor candidates. Because the matches from $INTER16 \times 16$, $INTER16 \times 8$, $INTER8 \times$

Tab. 4.5: Percentage of times that different motion vector predictors (mvp) are selected for enhancement predictor in the current macroblock. Data is collected by encoding 15 frames of *Foreman* sequence with QP 24 (IPPP).

mvp selected	Percentage (%)
mvp of $INTER16 \times 16$	51.4
mv of baseP in the same MB	20.4
mv of baseP from left MB	5.7
mv of enhP from left MB	10.1
mv of baseP from upper MB	3.2
mv of enhP from upper MB	9.1

16 and $INTER8 \times 8$ motion search can be good candidates, those are added and $M = N_{org}+9$ would be the maximum number of base predictor candidates because duplicate candidates are removed.

In $INTER16 \times 16\_IBS$ block mode, base and enhancement predictors are jointly searched within search range as described in Section 4.2.4. Thus, for M base predictor candidates, equal numbers of matching enhancement predictors are found. Finally, the R-D costs of M base and enhancement predictor pairs are calculated and compared with R-D costs of other block modes in H.264/AVC (R-D mode decision). Encoded information in $INTER16 \times 16\_IBS$ includes reference indices and motion vectors for base and enhancement predictors as well as the weight indices for each segment. Encodings of reference indices and motion vectors for base and enhancement predictor follow H.264/AVC standards.

To exploit the correlation in motion vectors from neighboring blocks, different motion vector predictors (mvp) are used for QT block modes in H.264/AVC. Because $INTER16 \times 16\_IBS$ is inserted as an additional block mode, we follow the mvp definition in H.264/AVC and it is modified only when $INTER16 \times 16\_IBS$ has been chosen in the neighboring blocks or it is tested in the current macroblock.

Tab. 4.6: Comparison of signaling bits for motion vector of enhancement predictor. Data is collected by encoding 15 frames of *Foreman* sequence (IPPP). In ($A \rightarrow B$), A is the average number of signaling bits for motion vector (mv) when the mvp of the enhancement predictor is set to the mvp of $INTER16 \times 16$. B is the average number of signaling bits for mv when the mvp of the enhancement predictor is chosen from 6 mvp schemes.

QP	Average bits for mv of QT block mode	Average bits for mv of IBS BaseP	Average bits for mv of IBS EnhP
20	20.8 → 20.5	7.0 → 6.7	7.6 → 6.5
24	14.9 → 14.8	6.8 → 6.6	7.1 → 6.1
29	9.4 → 9.5	6.4 → 6.3	6.2 → 5.3

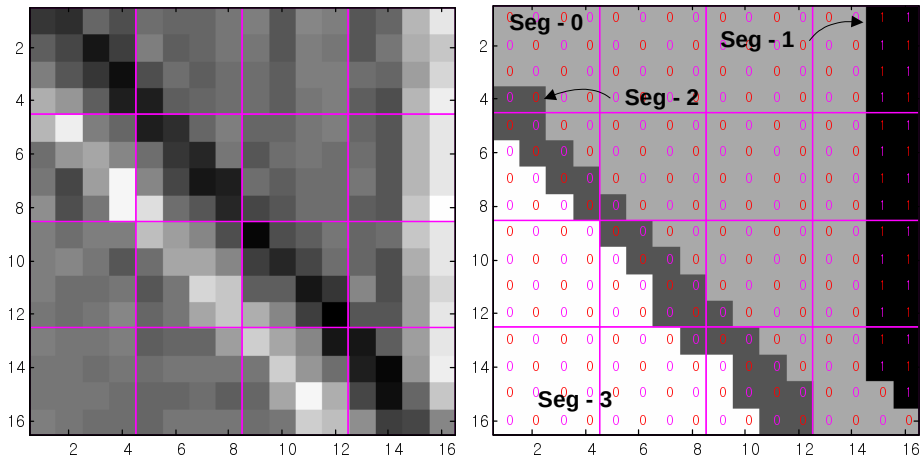
Firstly, assume that the QT block mode is tested in the current macroblock. If neighboring blocks do not use $INTER16 \times 16\_IBS$, the original mvp definition from H.264/AVC is used to find the mvp of current macroblock. If $INTER16 \times 16\_IBS$ is used in the neighboring blocks, it is regarded as $INTER16 \times 16$ with a motion vector from base predictor and the mvp of the current macroblock follows H.264/AVC. Secondly, assume that $INTER16 \times 16\_IBS$ is tested in the current macroblock. Base predictor uses the same mvp as $INTER16 \times 16$. For enhancement predictor, to investigate which mvp improves the coding efficiency most, 6 different mvp's are defined and tested. Tab. 4.5 shows the relative frequencies of occurrence of these 6 mvp schemes. When searching for the best enhancement predictor for a given base predictor, all mvp candidates are tested and the one with minimum distortion J_{enh} is chosen as the best mvp for enhancement predictor. In this experiment, bits signaling mvp selection are not counted so the simulation results can be regarded as an upper bound. As a comparison to this upper bound, the mvp for the enhancement predictor is fixed as the mvp of $INTER16 \times 16$, which is selected most as shown in Tab. 4.5. Tab. 4.6 shows that on average about

Tab. 4.7: Comparison of IBS results when the mvp of the enhancement predictor is set to (a) the mvp of $INTER16 \times 16$ QT block mode and chosen from (b) 6 mvp schemes (upper bound).

QP	PSNR: (a) → (b)	Bit rate: (a) → (b)
20	42.9123 → 42.9234	63367 → 62979
24	40.0678 → 40.0587	33053 → 32763
29	36.8543 → 36.8702	14534 → 14338

1 bit is reduced in signaling the mv of enhancement predictor. However, this reduction is not enough to be reflected into the overall coding gains. As can be seen in Tab. 4.7, less than 0.05 dB gains are achieved by the proposed upper bound, where the same data in Tab. 4.6 is used. Therefore, we conclude that there is no significant improvement in rate-distortion sense and the mvp of enhancement predictor is fixed as the mvp of $INTER16 \times 16$. In summary, if $INTER16 \times 16\_IBS$ is used in neighboring blocks, it is treated the same as if it were $INTER16 \times 16$ with the motion vector used as base predictor. If $INTER16 \times 16\_IBS$ is tested in the current macroblock, the mvp of $INTER16 \times 16$ is used for both base and enhancement predictor.

Weight indices $\{0, 1, 2\}$ which correspond to base, enhancement and average predictor respectively, are binarized and encoded by variable length code in R-D mode decision and binary arithmetic code in bit stream coding. The weight indices are signalled following the order of the segment indices that is defined by raster scanning from the top left corner to the bottom right corner of macroblock. When a pixel is found during the raster scanning, which does not belong to the segment already found, the segment of that pixel is assigned the next index. This segment numbering is repeated until all segments are covered in a macroblock.



(a) Predictor difference of base and enhancement predictor

(b) Segmentation from predictor difference and chosen weight indices

Fig. 4.8: Example of predictor difference and segmentation from *Foreman* sequence. The segment indices are shown, which are decided by raster scanning from the top left corner to bottom right corner of the macroblock.

In Fig. 4.8, an example of predictor difference between base and enhancement predictor is shown, with its corresponding segment information. Predictor difference shown in Fig. 4.8 (a) is scaled to show the difference clearly. Note that the segmentation shown in Fig. 4.8 (b) captures large predictor differences efficiently. Segment 0, 2 and 3 choose the weight index 0, base predictor and segment 1 chooses weight index 1, enhancement predictor. For the macroblock of this example, we signal $INTER16 \times 16\_IBS$ block mode first. Then, the reference index and motion vector for base and enhancement predictor is sent. Finally, four weight indices for each segment are sent. Note that the number of segments and the segments themselves are not transmitted but extracted at the decoder using base and enhancement predictor information. Prediction by IBS achieves 30% SSD reduction as compared with the best predictor based on a quad-tree for the example of Fig. 4.8.

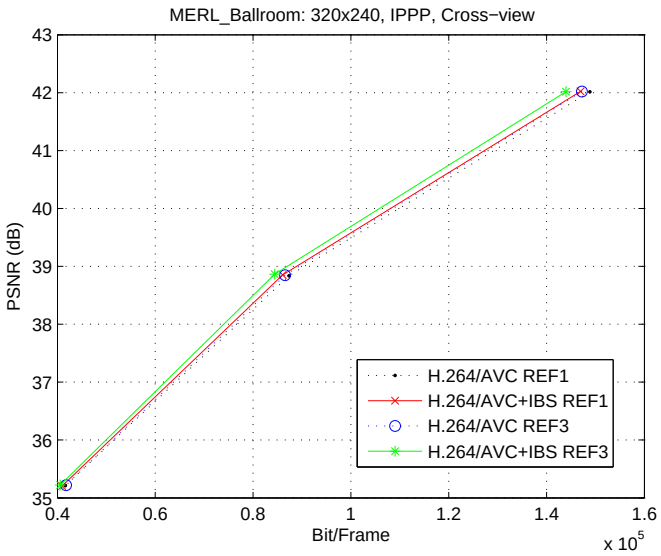


Fig. 4.9: MERL_Ballroom with 1 and 3 reference

4.4.2 Simulation Results

Both multi-view video (*MERL_Ballroom*, 320(w)x240(h)) and standard video sequences (*Foreman*, 352(w)x288(h)) are tested. In *MERL_Ballroom*, each anchor has 8 views coded IPPP PPPP and 2 anchors at different time stamps (0, 10) are tested. In *Foreman*, 15 frames are coded as IPPP. Encoding conditions of H.264/AVC and H.264/AVC+IBS are the same except that in H.264/AVC+IBS, $INTER16 \times 16\_IBS$ is tested as an additional inter block mode. QP 20, 24, 29 are used with ± 32 search range with quarter-pel and CABAC enabled. As can be seen in Figs. 4.9 and 4.10, 0.1-0.2 dB gains are achieved in *MERL_Ballroom* and 0.2-0.4 dB gains from *Foreman*. Note that gains by IBS increase with the number of references.

To see how the prediction gains achieved by IBS are reflected into R-D gains, in Tab. 4.8, average distortions and bits are shown for blocks best predicted by IBS in R-D mode decision. Improvements in prediction quality by IBS shown in the reduction of SSD_p are translated into reduction in residual coding bits and

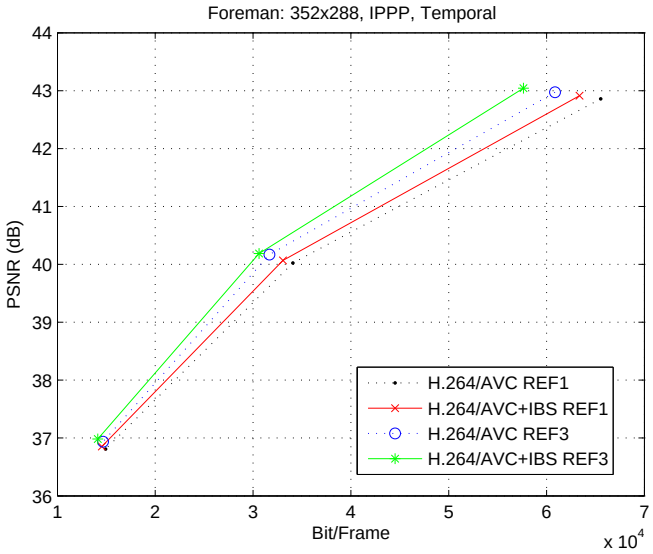


Fig. 4.10: Foreman with 1 and 3 references

SSD in reconstructed frame, SSD_r. Note that typically the bits needed to signal motion vectors are reduced because only two predictors are used in IBS (while a QT approach could use more than two vectors). Extra bits are needed to signal weights when using IBS.

Gains are not encouraging in *MERL_Ballroom*. Firstly, due to the noisy background of *MERL_ Ballroom*, predictor difference results in noisy segments, which increases signaling bits for weight indices as shown in Tab. 4.8. Secondly, for implicit block segmentation, it is assumed that references are not corrupted or mismatches including illumination and focus do not exist between frames. As shown in [17], there *exist* illumination mismatches between frames in different views. When two different segments with non-zero DC level exist in a 4x4 or 8x8 DCT block as shown in Fig. 4.11, this leads to increases in high frequency components so that residual coding bits increase. Also this may create artificial boundaries within a block. Note that in Tab. 4.8, 20% reduction in SSD_p is translated into only 9% reduction in residual bits in *MERL_Ballroom* while 12% reduction in SSD_p

Tab. 4.8: Comparison of data by QT and IBS from *MERL_Ballroom* and *Foreman* with QP 20. Data is averaged for the macroblocks where IBS is the best mode from 14 P-frames in each sequence. $A \rightarrow B$ means 'data by QT' $\rightarrow$ 'data by IBS'. SSD_p and SSD_r are SSD between the original and predictor and between the original and reconstruction, respectively. Bit_{res}, Bit_{mv} and Bit_w are bits for residual, motion/disparity vectors and weight indices respectively.

Sequence	SSD_p	SSD_r	Bit_{res}	Bit_{mv}	Bit_w
MERL_ Ballroom	12403 $\rightarrow$ 9885 *(20%)*	1463 $\rightarrow$ 1464 *(0%)*	364 $\rightarrow$ 333 *(9%)*	19 $\rightarrow$ 17 *(10%)*	0 $\rightarrow$ 11.5
Foreman	3209 $\rightarrow$ 2817 *(12%)*	1077 $\rightarrow$ 1052 *(2%)*	149 $\rightarrow$ 135 *(9%)*	23 $\rightarrow$ 16 *(32%)*	0 $\rightarrow$ 7.6

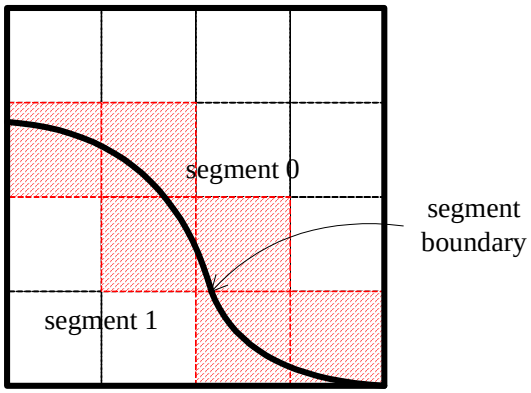


Fig. 4.11: AC increases by 4x4 or 8x8 block DCT due to the unequal DC residual between different segments in IBS.

is translated into 9% reduction in residual bits and 2% reduction in SSD_r in *Foreman*. Combined with illumination compensation [17], the performance of IBS for cross-view prediction could be improved.

4.5 Conclusions

In this chapter, implicit block segmentation based on the predictors available at the decoder is proposed. Given two candidate block predictors, segmentation is applied to the predictor difference. Different weighted sums of predictors are selected for

each segment and signaled to the decoder. Implementation in H.264/AVC shows encouraging results in *Foreman*, where illumination mismatches are not present. Combining IBS with mismatch compensation tools would increase the coding efficiency in cross-view prediction. Areas of future work include improvements to the segmentation strategy where most of computational complexity of IBS comes and efficient search techniques to allow searching for pairs of predictors.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Firstly, the 2-D dependency problem that arises in MVC was addressed in Chapter 2. Because both cross-view and temporal correlations are exploited to improve coding efficiency in MVC, 2-D dependencies are present in MVC. Optimal bit allocation is possible based on 3-D trellis expansion but with significant complexity during data generation process. To reduce the complexity, monotonicity property is extended to 3-D trellis expansion and from the correlation between quantizers of anchor and non-anchor frames, the number of quantizer candidates for non-anchor frames is limited. With proposed bit allocation scheme, 0.5 - 1 dB gains are achieved.

Next, the illumination mismatch problem in multi-view video was covered in Chapter 3. Even with sophisticated calibration, it is not possible to ensure all cameras in an array are calibrated perfectly, which causes global brightness mismatches among different views. Even with perfect camera calibration, an object may appear differently due to the different depths and perspectives of the objects

with respect to each camera causing local mismatches. The accuracy of disparity search is degraded by these brightness variation between frames, leading to the degradation of coding efficiency. To compensate both global and local mismatches, a block level illumination compensation (IC) model is proposed. Because different portions of a video frame can undergo different illumination changes, block by block activation of IC model is proposed. For efficient transmission, IC parameters are quantized and binary arithmetic coded. It is shown that IC requires about 64% additional calculation within motion/disparity search. Simulation results of cross-view prediction show 0.2 - 0.8 dB gains. IC techniques are applied to both temporal and cross-view prediction in MVC and achieve higher coding efficiency as compared to WP. It is also shown how IC and ARF can be combined to compensate both illumination and focus mismatches in MVC. The combined system achieves 0.5 - 1.3 dB gains in cross-view prediction of three test sequences.

In Chapter 4, an implicit block segmentation (IBS) method was proposed in order to improve the quality of prediction. Block based motion/disparity estimation and compensation provides a good balance between prediction accuracy and rate overhead. However most of the object boundaries are not perfectly aligned with block boundaries, which makes motion/disparity search difficult and reduces coding efficiency. Given two candidate block predictors, from the observation that distortion can be reduced further where two predictors differ most, segmentation is applied to the block of predictor difference. For each segment, weighted sum of predictors with minimum distortion is decided. Additional overheads for each block include the locations of two predictors and weight indices for each segment. Segment information can be retrieved implicitly by repeating the segmentation for the predictor difference at the decoder. IBS is implemented as an additional block mode in H.264/AVC reference codec and achieves 0.1 - 0.4 dB gains in cross-view

prediction of *Ballroom* and *Foreman*. The more references are available, the more the coding efficiency of IBS improves.

5.2 Future Work

Although each chapter addresses different problems of predictive coding in MVC, these techniques can be combined to provide a unified solution. For example, IBS is proposed assuming there are no mismatches between frames. However in cross-view prediction, there *are* illumination mismatches. Therefore, applying IC in each segment by IBS helps to find the correct match and improve overall coding gains. On top of the block level compensations by IC and IBS, 2-D dependent bit allocation can be applied in order to optimize available resources in frame level.

In this work, it is assumed that only multi-view sequences are available without any other information. However, when auxiliary information, e.g., camera parameters or depth information, is available, efficiency of multi-view video coding can be further improved. For example, instead of sending all the views, only video sequences corresponding to a subset of views are transmitted along with depth information, from which intermediate views can be interpolated. Because disparities in cross-view prediction are caused by the different depths of the objects and camera perspectives, if camera parameters and object depths are known, it can be used to help disparity estimation/compensation faster and more accurate.

Bibliography

[1] "Call for proposals on multi-view video coding," ISO/IEC JTC1/SC29/WG11 MPEG Document N7327, Jul. 2005.

[2] "Description of core experiments in MVC," ISO/IEC JTC1/SC29/WG11 MPEG Document W8019, Montreux, Switzerland, Apr. 2006.

[3] M. Accame, F. D. Natale, and D. Giusto, "Hierarchical block matching for disparity estimation in stereo sequences," in *Proc. IEEE International Conference on Image Processing (ICIP)*, vol. 2, Washington, USA, Oct. 1995, pp. 23–26.

[4] E. H. Adelson and J. R. Bergen, *Computational Models of Visual Processing*. Poznan, Poland: MIT Press, 1991.

[5] H. Aydinoglu and M. Hayes, "Compression of multi-view images," in *Proc. IEEE International Conference on Image Processing (ICIP)*, vol. 2, Austin, TX, Nov. 1994, pp. 385–389.

[6] J. M. Boyce, "Weighted prediction in the H.264/MPEG AVC video coding standard," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 3, Vancouver, Canada, May 2004, pp. 789–792.

[7] F. Chang, C. Chen, and C. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Computer Vision and Image Understanding (CVIU)*, vol. 93, no. 2, pp. 206–220, Feb. 2004.

[8] G. Chen, J. H. Kim, J. Lopez, and A. Ortega, "Response to call for evidence on multi-view video coding," ISO/IEC JTC1/SC29/WG11 MPEG Document M11731, Hong Kong, China, Jan. 2005.

[9] O. D. Escoda, P. Yin, D. Congxia, and L. Xin, "Geometry-adaptive block partitioning for video coding," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, Apr. 2007, pp. 657–660.

[10] C. Fehn, N. Atzpadin, M. Muller, O. Schreer, A. Smolic, R. Tanger, and P. Kauff, "An advanced 3DTV concept providing interoperability and scalability for a wide range of multi-baseline geometries," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Oct. 2006, pp. 2961–2964.

[11] T. Fujii, T. Kimoto, and M. Tanimoto, "Ray space coding for 3-D visual communication," in *Picture Coding Symposium (PCS)*, Meloburne/Australia, Mar. 1996.

[12] Y. He, J. Ostermann, M. Tanimoto, and A. Smolic, "Introduction to the special section on multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1433–1435, Nov. 2007.

[13] E. Hung and R. D. Queiroz, "On macroblock partition for motion compensation," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Oct. 2006, pp. 1697–1700.

[14] (2006, Jul.) Software implementation of H.264: JM Version 10.2. The Image Communication Group at Heinrich Hertz Institute Germany. [Online]. Available: http://iphome.hhi.de/suehring/tml/index.htm

[15] K. Kamikura, H. Watanabe, H. Jozawa, H. Kotera, and S. Ichinose, "Global brightness-variation compensation for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 8, pp. 988–1000, Dec. 1998.

[16] S. Kang, C. Zitnick, M. Uyttendaele, S. Winder, and R. Szeliski, "Free-viewpoint video with stereo and matting," in *Picture Coding Symposium (PCS)*, San Francisco, USA, Dec. 2004.

[17] J. H. Kim, P. Lai, J. Lopez, A. Ortega, Y. Su, P. Yin, and C. Gomila, "New coding tools for illumination and focus mismatch compensation in multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1519–1535, Nov. 2007.

[18] J. H. Kim, P. Lai, A. Ortega, Y. Su, P. Yin, and C. Gomila, "Results of CE2 on multi-view video coding," ISO/IEC JTC1/SC29/WG11 MPEG Document M13720, Klagenfurt, Austria, Jul. 2006.

[19] S. Kim and R. Park, "Fast local motion-compensation algorithm for video sequences with brightness variations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 4, pp. 289– 299, Apr. 2003.

[20] H. Kimata, M. Kitahara, K. Kamikura, and Y. Yashima, "Free-viewpoint video communication using multi-view video coding," NTT, Yokosuka-shi, Japan, Tech. Rep. F0282C, 2-8, 2004.

[21] P. Lai, Y. Su, P. Yin, C. Gomila, and A. Ortega, "Adaptive filtering for cross-view prediction in multi-view video coding," in *Proc. SPIE Visual Communication and Image Processing (VCIP)*, vol. 6508, San Jose, CA, Jan. 30-Feb. 1 2007.

[22] Y. Lee, J. Hur, Y. Lee, K. Han, S. Cho, N. Hur, J. Kim, J. H. Kim, P. Lai, A. Ortega, Y. Su, P. Yin, and C. Gomila, "CE11 : Illumination compensation," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG JVT-U052, Hangzhou, China, Oct. 2006.

[23] D. Liu, Y. He, S. Li, Q. Huang, and W. Gao, "Linear transform based motion compensated prediction for luminance intensity changes," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, Beijing, China, May 2005, pp. 304–307.

[24] S. Liu and C.-C. J. Kuo, "Joint temporal-spatial bit allocation for video coding with dependency," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 15–26, Jan. 2005.

[25] J. Lopez, J. H. Kim, A. Ortega, and G. Chen, "Block-based illumination compensation and search techniques for multiview video coding," in *Picture Coding Symposium (PCS)*, San Francisco, USA, Dec. 2004.

[26] J. Lopez, "Block-based compression techniques for multiview video coding," Master's thesis, Universitat Politecnica de Catalunya, Barcelona, Spain, 2005.

[27] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[28] W. Matusik and H. Pfister, "3D TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 814–824, Aug. 2004.

[29] D. Mount. KMlocal: A testbed for K-means clustering algorithms based on local search Version: 1.7.1. Dept of Computer Science at University of Maryland. [Online]. Available: http://www.cs.umd.edu/∼mount/Projects/KMeans/

[30] K. Mueller, P. Merkle, A. Smolic, and T. Wiegand, "Multiview coding using avc," ISO/IEC JTC1/SC29/WG11 MPEG Document M12945, Bangkok, Thailand, Jan. 2006.

[31] U. Neumann, T. Pintaric, and A. Rizzo, "Immersive panoramic video," in *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, Marina del Rey, California, USA, 2000, pp. 493–494.

[32] W. Niehsen and S. Simon, "Block motion estimation using orthogonal projection," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Lausanne, Switzerland, Sep. 1996.

[33] M. Orchard, "Predictive motion-field segmentation for image sequence coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 1, pp. 54–70, Feb. 1993.

[34] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 533–545, Sep. 1994.

[35] N. Sebe, M. S. Lew, and D. P. Huijsmans, "Toward improved ranking metrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1132–1143, Oct. 2000.

[36] Y. Sermadevi and S. S. Hemami, "Efficient bit allocation for dependent video coding," in *Proc. Data Compression Conference (DCC)*, Mar. 2004, pp. 232–241.

[37] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, Sep. 1988.

[38] R. Shukla, P. Dragotti, M. Do, and M. Vetterli, "Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 343–359, Mar. 2005.

[39] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3D video and free viewpoint video - technologies, applications and MPEG standards," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2006, pp. 2161–2164.

[40] Y. Su, P. Yin, C. Gomila, J. H. Kim, P. Lai, and A. Ortega, "Thomson's response to MVC CfP," ISO/IEC JTC1/SC29/WG11 MPEG Document M12969/2, Bangkok, Thailand, Jan. 2006.

[41] K. M. Uz, J. M. Shapiro, and M. Czigler, "Optimal bit allocation in the presence of quantizer feedback," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, Apr. 1993, pp. 385–388.

[42] A. Vetro, Y. Su, H. Kimata, and A. Smolic, "Joint multiview video model (JMVM) 2.0," ISO/IEC JTC1/SC29/WG11 MPEG Document N8459, Hangzhou, China, Oct. 2006.

[43] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[44] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[45] W. Woo and A. Ortega, "Optimal blockwise dependent quantization for stereo image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 6, pp. 861–867, Sep. 1999.

[46] P. Yin, H.-Y. C. Tourapis, A. Tourapis, and J. Boyce, "Fast mode decision and motion estimation for JVT/H.264," in *Proc. IEEE International Conference on Image Processing (ICIP)*, vol. 3, Sep. 2003, pp. 853–856.

Appendix A

Comparison between MSD and MAD in Motion/Disparity Search

Let $\bar{\mathbf{P}} = [P_0, P_1, ..., P_{N-1}]^T$ be a candidate predictor for the original signal $\bar{\mathbf{X}} = [X_0, X_1, ..., X_{N-1}]^T$, for residual error $(X_i - P_i)$, the sum of squared difference (SSD) and the sum of absolute difference (SAD) metrics are defined as

$$SSD = \frac{1}{N}\sum_{i=0}^{N-1}(X_i - P_i)^2 \tag{A.1}$$

$$SAD = \frac{1}{N}\sum_{i=0}^{N-1}|X_i - P_i| \tag{A.2}$$

It is known that SSD and SAD are justified as an error metric from maximum likelihood perspectives when the error follows normal and Laplace distribution, respectively [35]. In block motion search, due to the complexity of multiplication in SSD, SAD is commonly used as a search metric. For example, in illumination compensation (IC) in Chapter 3, scale and offset parameters are calculated using SSD but for the motion/disparity search, SAD after compensation is adopted. Also in implicit block segmentation (IBS) in Chapter 4, SAD is adopted instead of SSD during motion/disparity search. In this appendix, from the statistical modeling of residual error $(X_i - P_i)$, we evaluate conditions for the motion search results by SAD to be equal to those obtained with SSD.[1]

Let p be the prediction of original signal x, then mean squared difference (MSD) and mean absolute difference(MAD) are defined as

$$MSD = E\{(x-p)^2\} \tag{A.3}$$

$$MAD = E\{|x-p|\}, \tag{A.4}$$

[1] We believe there would be the similar evaluations to Appendix A but the concepts and terms used in this analysis help understanding Appendix B and C thus, we start from scratch.

which are statistically equal to SSD and SAD, respectively. Let p_0 and p_1 be two candidate predictors for the original signal x. Then their respective residual errors are denoted n_0 and n_1:

$$\begin{aligned} n_0 &= x - p_0 \\ n_1 &= x - p_1. \end{aligned} \tag{A.5}$$

Let MSD_i and MAD_i denote MSD and MAD by p_i, respectively. If the mean and the variance of n_i are denoted as μ_i and σ_i^2, from (A.3)

$$MSD_i = E\{(x - p_i)^2\} = E\{n_i^2\} = \mu_i^2 + \sigma_i^2. \tag{A.6}$$

If $\mu_i \sim 0$ or $\frac{\mu_i}{\sigma_i} \sim 0$, from (A.6)

$$MSD_i = E\{(x - p_i)^2\} = E\{n_i^2\} = \mu_i^2 + \sigma_i^2 \sim \sigma_i^2. \tag{A.7}$$

Thus if $\sigma_0^2 < \sigma_1^2$, p_0 will be chosen based on the MSD distortion measure. Note that the result is derived from the second order statistics of n_0 and n_1 without any assumption about a specific probability model.

Due to the absolute operation, MAD can not be found directly from the second order statistics. In this appendix, MAD is derived for the statistical models of (i) 'normal' and (ii) 'Laplace' distributions. In Fig. A.1, the distribution of residual errors ($n = x - p$) from coding results of *Foreman* sequence is compared with normal and Laplace distributions using mean and variance from coding results, which verifies that both distributions are good approximations of real data. Note that during the motion search, the quality of predictor improves by the error metric and converge to the best predictor, thus the predictors used in Fig. A.1 are the best matches to the original signal in each block.

(i) For normal distribution model $n \sim N(\mu, \sigma^2)$,

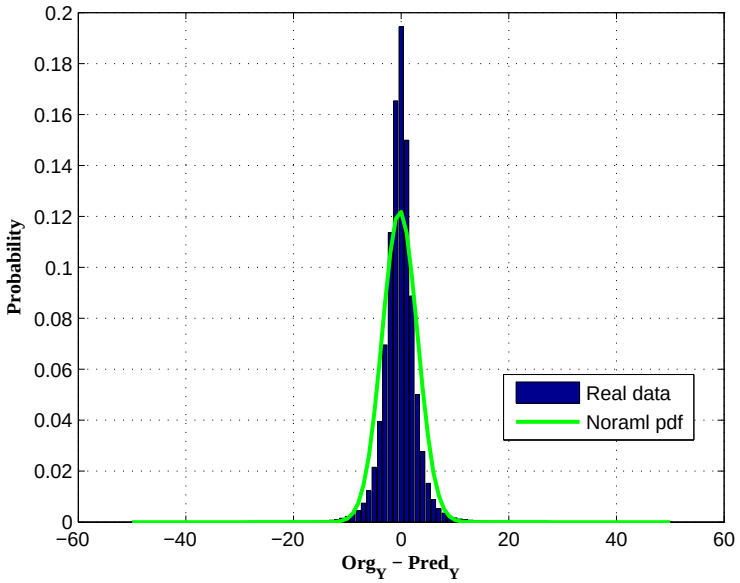
$$\begin{aligned} E\{|n|\} &= \int_{-\infty}^{\infty} \frac{|n|}{\sqrt{2\pi\sigma^2}} e^{-\frac{(n-\mu)^2}{2\sigma^2}} dn \\ &= \sqrt{\frac{2\sigma^2}{\pi}} e^{-\frac{\mu^2}{2\sigma^2}} - \mu \left(1 - 2Q(-\frac{\mu}{\sigma})\right) \end{aligned} \tag{A.8}$$

where

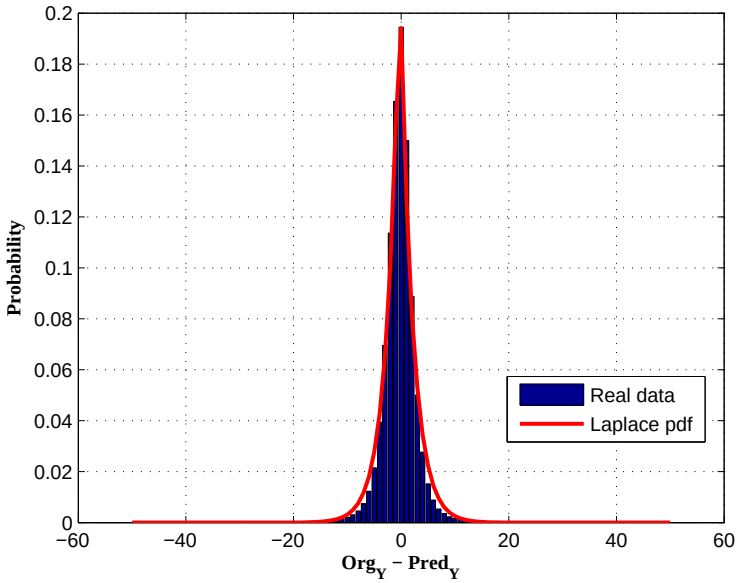
$$Q(c) = \int_c^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt.$$

Therefore,

$$MAD_i = E\{|n_i|\} = \sqrt{\frac{2\sigma_i^2}{\pi}} e^{-\frac{\mu_i^2}{2\sigma_i^2}} - \mu_i \left(1 - 2Q(-\frac{\mu_i}{\sigma_i})\right) \tag{A.9}$$



(a) Normal distribution



(b) Laplace distribution

Fig. A.1: Comparison of normal and Laplace distribution with real data obtained by encoding *Foreman* sequences (CIF). Data is collected from 7 P frames coding using QP 20, ±32 search range and quarter-pixel precision by *JSVM 8.4*. The differences between original and predictor data are obtained only for luminance. Mean and variance are -0.25 and 10.66 respectively.

If $\mu_i \sim 0$ or $\frac{\mu_i}{\sigma_i} \sim 0$,

$$MAD_i \sim \sqrt{\frac{2\sigma_i^2}{\pi}} \tag{A.10}$$

For signals with $\sigma_0^2 < \sigma_1^2$, from (A.10), p_0 is selected as a better estimate of x than p_1. Therefore, when the residual errors n_0 and n_1 follow the normal distribution with $\mu_i \sim 0$ and/or $\frac{\mu_i}{\sigma_i} \sim 0$, MSD and MAD give the same result.

(ii) If n follows a Laplace distribution, probability distribution function (*pdf*) is defined as

$$f_{\mathbf{n}}(n) = \frac{1}{2a} e^{-\frac{|n-\mu|}{a}} \tag{A.11}$$

where $\mu = E\{n\}$ and $\sigma^2 = 2a^2 = E\{(n-\mu)^2\}$ and

$$\begin{aligned} E\{|n|\} &= \int_{-\infty}^{\infty} \frac{|n|}{2a} e^{-\frac{|n-\mu|}{a}} dn \\ &= a e^{-\frac{|\mu|}{a}} + |\mu| \end{aligned} \tag{A.12}$$

Therefore, using $a = \sqrt{\frac{\sigma^2}{2}}$

$$MAD_i = \sqrt{\frac{\sigma_i^2}{2}} e^{-\frac{|\mu_i|}{\sigma_i}\sqrt{2}} + |\mu_i| \tag{A.13}$$

If $\mu_i \sim 0$ or $\frac{\mu_i}{\sigma_i} \sim 0$,

$$MAD_i \sim \frac{\sigma_i}{\sqrt{2}} \tag{A.14}$$

For signals with $\sigma_0^2 < \sigma_1^2$, from (A.14) p_0 is selected as a better estimate of x than p_1. Therefore, when the residual error n_0 and n_1 follows Laplace distribution with $\mu_i \sim 0$ and/or $\frac{\mu_i}{\sigma_i} \sim 0$, MSD and MAD give the same result.

In conclusion, for 'normal' and 'Laplace' distributions, if $\mu_i \sim 0$ and/or $\frac{\mu_i}{\sigma_i} \sim 0$ are satisfied, SSD and SAD would provide the same searching capability. As can be seen in Fig. A.1, the conditions '$\mu_i \sim 0$' and/or '$\frac{\mu_i}{\sigma_i} \sim 0$' would be satisfied in most video sequences.[2]

[2] Note that above analysis is derived 'statistically' but in block motion/disparity search, there might be blocks such that the accurate predictor to the original signal is hard to find (e.g., occluded or uncovered regions) thus, the conditions '$\mu_i \sim 0$' and '$\frac{\mu_i}{\sigma_i} \sim 0$' are not satisfied. In these cases, if $|\mu_i| \gg 0$ or $|\frac{\mu_i}{\sigma_i}| \gg 0$, MSD ($= \mu_i^2 + \sigma_i^2$) tends to be large thus, instead of $INTER$ block mode where motion/disparity search is performed, $INTRA$ block mode would be used where the comparison between SSD and SAD in motion search has no meanings.

Appendix B

Additional Weight Selection in IBS

In this appendix, it is shown statistically why $(\frac{1}{2}, \frac{1}{2})$ has been included in W in addition to $(1, 0)$ and $(0, 1)$ which correspond to p_0 and p_1 respectively.

Let x be the original pixel predicted by two pixel predictor p_0 and p_1 respectively and corresponding residual errors are represented by noise signal n_0 and n_1.

$$n_0 = x - p_0$$
$$n_1 = x - p_1$$

Let the mean and variance of n_i be denoted $\mu_i = E\{n_i\}$ and $\sigma_i^2 = E\{(n_i - \mu_i)^2\}$, respectively. An additional predictor p_a is defined as a weighted sum of p_0 and p_1; $p_a = \alpha_0 p_0 + \alpha_1 p_1$. Let n_a denote the residual error by p_a thus, $n_a = x - p_a$. With the constraint $\alpha_0 + \alpha_1 = 1$ to make $p_a = p_0$ with $\alpha_0 = 1$ and $p_a = p_1$ with $\alpha_1 = 1$,

$$\begin{aligned} n_a = x - p_a &= (\alpha_0 + \alpha_1)x - (\alpha_0 p_0 + \alpha_1 p_1) \\ &= \alpha_0(x - p_0) + \alpha_1(x - p_1) \\ &= \alpha_0 n_0 + \alpha_1 n_1. \end{aligned}$$

Therefore, the mean and the variance of n_a are

$$\begin{aligned} \mu_a &= E\{n_a\} = \alpha_0\mu_0 + \alpha_1\mu_1 \\ \sigma_a^2 &= E\{(n_a - \mu_a)^2\} = \alpha_0^2\sigma_0^2 + 2\alpha_0\alpha_1\sigma_c^2 + \alpha_1^2\sigma_1^2 \end{aligned} \tag{B.1}$$

where $\sigma_c^2 = E\{(n_0 - \mu_0)(n_1 - \mu_1)\}$. The residual energy corresponding to p_a is quantified as

$$\begin{aligned} MSE_a &= E\{n_a^2\} = \mu_a^2 + \sigma_a^2 \\ &= (\alpha_0\mu_0 + \alpha_1\mu_1)^2 + \alpha_0^2\sigma_0^2 + \alpha_1^2\sigma_1^2 + 2\alpha_0\alpha_1\sigma_c^2 \\ &= (\alpha_0(\mu_0 - \mu_1) + \mu_1)^2 + \alpha_0^2\sigma_0^2 + (1-\alpha_0)^2\sigma_1^2 + 2\alpha_0(1-\alpha_0)\sigma_c^2 \\ &= \alpha_0^2((\mu_0 - \mu_1)^2 + \sigma_0^2 + \sigma_1^2 - 2\sigma_c^2) - 2\alpha_0(\sigma_1^2 + \mu_1^2 - \sigma_c^2 - \mu_0\mu_1) + \sigma_1^2 + \mu_1^2 \\ &= \alpha_0^2(\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2) - 2\alpha_0\tilde{\sigma}_1^2 + \sigma_1^2 + \mu_1^2 \\ &= (\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2)\left(\alpha_0 - \frac{\tilde{\sigma}_1^2}{\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2}\right)^2 + \sigma_1^2 + \mu_1^2 - \frac{\tilde{\sigma}_1^4}{\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2} \end{aligned} \tag{B.2}$$

where $\tilde{\sigma}_i^2 = \sigma_i^2 + \mu_i^2 - \sigma_c^2 - \mu_0\mu_1$ for $i \in \{0, 1\}$. By setting to zero the gradient of MSE_a with respect to α_0 in eq. (B.2), the optimal α_0 and α_1 can be found as

$$\begin{aligned} \alpha_0 &= \frac{\tilde{\sigma}_1^2}{\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2} = \frac{E\{n_1(n_1 - n_0)\}}{E\{(n_0 - n_1)^2\}} \\ \alpha_1 &= \frac{\tilde{\sigma}_0^2}{\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2}, \end{aligned} \tag{B.3}$$

and the minimum MSE_a is

$$\begin{aligned} MMSE_a &= \sigma_1^2 + \mu_1^2 - \frac{\tilde{\sigma}_1^4}{\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2} \\ &= \sigma_c^2 + \mu_0\mu_1 + \frac{\tilde{\sigma}_0^2\tilde{\sigma}_1^2}{\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2}. \end{aligned} \tag{B.4}$$

Due to the computational complexity of multiplication and division in (B.3) and signaling overhead of α_0, a weight can be pre-selected and only the weight index minimizing the distortion can be signaled. From the constraints that weights are non-negative and $\alpha_0 + \alpha_1 = 1$, α_0 should be in $(0, 1)$ and a straightforward selection would be $\frac{1}{2}$ which corresponds to the average of p_0 and p_1. With respect to the computational complexity, $\frac{1}{2}$ is the most efficient weight between $(0, 1)$ because in the calculation of new predictor p_a, only the sum of p_0 and p_a followed by shift operation is needed as can be seen in chapter 4.3.

If α is defined as the weight most frequently chosen, it can be found as

$$\alpha = \arg\max_{0<\alpha_0<1} P\left\{MSE_a < MSE_0 \quad \& \quad MSE_a < MSE_1\right\}. \tag{B.5}$$

From (B.2),

$$\begin{aligned} & MSE_a < MSE_0 \\ & \Leftrightarrow \alpha_0^2(\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2) - 2\alpha_0\tilde{\sigma}_1^2 + \sigma_1^2 + \mu_1^2 < \sigma_0^2 + \mu_0^2 \\ & \Leftrightarrow \frac{1-\alpha_0}{1+\alpha_0} < \frac{\tilde{\sigma}_0^2}{\tilde{\sigma}_1^2} \end{aligned} \tag{B.6}$$

$$\begin{aligned} & MSE_a < MSE_1 \\ & \Leftrightarrow \alpha_0^2(\tilde{\sigma}_0^2 + \tilde{\sigma}_1^2) - 2\alpha_0\tilde{\sigma}_1^2 + \sigma_1^2 + \mu_1^2 < \sigma_1^2 + \mu_1^2 \\ & \Leftrightarrow \frac{\tilde{\sigma}_0^2}{\tilde{\sigma}_1^2} < \frac{2-\alpha_0}{\alpha_0}. \end{aligned} \tag{B.7}$$

From (B.6) and (B.7), (B.5) is equal to

$$\alpha = \arg\max_{0<\alpha_0<1} P\left\{\frac{1-\alpha_0}{1+\alpha_0} < \frac{\tilde{\sigma}_0^2}{\tilde{\sigma}_1^2} < \frac{2-\alpha_0}{\alpha_0}\right\}. \tag{B.8}$$

If m pixel residuals in the segment are regarded as m sample observations from independent normal random variable $n_i \sim N(0, \kappa_i^2)$ with $i \in \{0, 1\}$, the sum of residual energy by p_i in the segment would be $(m-1)s_i^2$ where s_i^2 is the sample variance from n_i. Replacing $\tilde{\sigma}_i^2$ with s_i^2 in (B.8) and noting that $\chi_i^2 = \frac{(m-1)s_i^2}{\kappa_i^2}$ has a chi-square density function with $\nu_i = m-1$ degrees of freedom,

$$\alpha = \arg\max_{0<\alpha_0<1} P\left\{\frac{1-\alpha_0}{1+\alpha_0} < \frac{\kappa_0^2\chi_0^2}{\kappa_1^2\chi_1^2} < \frac{2-\alpha_0}{\alpha_0}\right\}. \tag{B.9}$$

Because n_0 and n_1 are assumed to be independent, χ_0^2 and χ_1^2 are independent chi-square random variables with ν_0 and ν_1 degrees of freedom, respectively, and then $F = \frac{\chi_0^2/\nu 0}{\chi_1^2/\nu_1}$ has an F-distribution with ν_0 numerator degrees of freedom and ν_1 denominator degrees of freedom. With $\nu_0 = \nu_1 = m-1$,

$$\alpha = \arg\max_{0<\alpha_0<1} P\left\{\frac{1-\alpha_0}{1+\alpha_0} < \frac{\kappa_0^2}{\kappa_1^2}F < \frac{2-\alpha_0}{\alpha_0}\right\}. \tag{B.10}$$

The probability in (B.10) is calculated for various values of three parameters. First, for six different values of $m \in \{10, 50, 100, 150, 200, 250\}$, the probabilities are calculated fixing $\frac{\kappa_0^2}{\kappa_1^2}$ and α_0. Then their average is shown in the Tab. B.1 with respect to each $\frac{\kappa_0^2}{\kappa_1^2}$ and α_0. The range of $\frac{\kappa_0^2}{\kappa_1^2}$ is limited to $\left[\frac{1}{3}, 3\right]$ because most of $\frac{\kappa_0^2}{\kappa_1^2}$ lies between $\left[\frac{1}{3}, 3\right]$ as can be seen in the example of Fig. C.1. As shown in the last row of Tab. B.1, the average probability over m and $\frac{\kappa_0^2}{\kappa_1^2}$ is the highest for

Tab. B.1: The probability in (B.10) is calculated changing three parameters, (i) m, (ii) $\frac{\kappa_0^2}{\kappa_1^2}$, and (iii) α_0. The average of probabilities for $m = \{10, 50, 100, 150, 200, 250\}$ is shown with respect to different $\frac{\kappa_0^2}{\kappa_1^2}$ and α_0. The last row shows the average over $\frac{\kappa_0^2}{\kappa_1^2}$.

	$\alpha_0 = 0.1$	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\frac{\kappa_0^2}{\kappa_1^2} = 3$	0.994	0.987	0.966	0.892	0.500	0.130	0.050	0.028	0.017
2	0.983	0.987	0.983	0.970	0.934	0.774	0.353	0.108	0.048
$\frac{3}{2}$	0.965	0.978	0.983	0.980	0.969	0.942	0.840	0.500	0.168
1	0.828	0.935	0.965	0.977	0.980	0.977	0.965	0.935	0.828
$\frac{2}{3}$	0.168	0.500	0.840	0.942	0.969	0.980	0.983	0.978	0.965
$\frac{1}{2}$	0.048	0.108	0.353	0.774	0.934	0.970	0.983	0.987	0.983
$\frac{1}{3}$	0.017	0.028	0.050	0.130	0.500	0.892	0.966	0.987	0.994
	0.572	0.646	0.734	0.809	**0.827**	0.809	0.734	0.646	0.572

$\alpha_0 = \frac{1}{2}$. If we take the probability of $\frac{\kappa_0^2}{\kappa_1^2}$ in Fig. C.1 into account, the probabilities corresponding to $\frac{\kappa_0^2}{\kappa_1^2} = 1$ are weighted most, which favors the weight $\frac{1}{2}$ more.

Although the weight $\frac{1}{2}$ is found with the assumption that n_0 and n_1 are independent and follow the normal distributions with zero mean, it is the most efficient weight which has the least computational complexity for p_a. Therefore in IBS, the additional weight is defined to be $\frac{1}{2}$.

Appendix C

Comparison between MSD and MAD for Weight Selection

In order to avoid the complexity by multiplication in SSD distortion measure, SAD is adopted in weight selection for each segment in IBS. In this appendix, from the statistical modeling of residual error, we study the penalty incurred for using SAD instead of SSD in IBS weight selection.

Let p_a be the weighted sum of p_0 and p_1 as

$$p_a = \alpha_0 p_0 + \alpha_1 p_1$$

with $\alpha_0 + \alpha_1 = 1$ and $0 < \alpha_0 < 1$. Then, associated residual error n_a is represented as

$$n_a = \alpha_0 n_0 + \alpha_1 n_1.$$

Let μ_i and σ_i^2 be the mean and the variances of n_i. Then from (B.1),

$$\begin{aligned} \mu_a &= E\{n_a\} = \alpha_0\mu_0 + \alpha_1\mu_1 \\ \sigma_a^2 &= E\{(n_a - \mu_a)^2\} = \alpha_0^2\sigma_0^2 + 2\alpha_0\alpha_1\sigma_c^2 + \alpha_1^2\sigma_1^2 \end{aligned} \tag{C.1}$$

where $\sigma_c^2 = E\{(n_0 - \mu_0)(n_1 - \mu_1)\}$.

With three predictors (p_0, p_1 and p_a) available in each segment of the macroblock, p_a will be selected when it gives the minimum distortion. For MSD distortion measure, from (B.6) and (B.7)

$$\begin{aligned} & MSD_a < MSD_0 \quad \& \quad MSD_a < MSD_1 \\ \Leftrightarrow \quad & E\{n_a^2\} < E\{n_0^2\} \quad \& \quad E\{n_a^2\} < E\{n_1^2\} \\ \Leftrightarrow \quad & \frac{\alpha_1^2}{1-\alpha_0^2} < \frac{\tilde{\sigma}_0^2}{\tilde{\sigma}_1^2} < \frac{1-\alpha_1^2}{\alpha_0^2} \end{aligned} \tag{C.2}$$

where $\tilde{\sigma}_i^2 = \sigma_i^2 + \mu_i^2 - \sigma_c^2 - \mu_0\mu_1$ for $i \in \{0, 1\}$. If (i) n_0 and n_1 are uncorrelated and (ii) $\mu_i \sim 0$ and/or $\frac{\mu_i}{\sigma_i} \sim 0$, then $\tilde{\sigma}_i^2 = \sigma_i^2 + \mu_i^2 - \sigma_c^2 - \mu_0\mu_1 \sim \sigma_i^2$ thus (C.2) is equal to

$$\frac{\alpha_1^2}{1-\alpha_0^2} < \frac{\sigma_0^2}{\sigma_1^2} < \frac{1-\alpha_1^2}{\alpha_0^2}. \tag{C.3}$$

For MAD distortion measure,

$$\begin{aligned} & MAD_a < MAD_0 \quad \& \quad MAD_a < MAD_1 \\ \Leftrightarrow \quad & E\{|n_a|\} < E\{|n_0|\} \quad \& \quad E\{|n_a|\} < E\{|n_1|\} \\ \Leftrightarrow \quad & E\{|\alpha_0 n_0 + \alpha_0 n_1|\} < E\{|n_0|\} \quad \& \quad E\{|\alpha_0 n_0 + \alpha_0 n_1|\} < E\{|n_1|\}. \end{aligned} \tag{C.4}$$

Due to the absolute operation, it is not straightforward to find a generic closed form of solution for (C.4). Therefore, we solve (C.4) assuming (i) normal and (ii) Laplace distribution models for n_0 and n_1.

(i) For normal distribution model $n_i \sim N(\mu_i, \sigma_i^2)$, from (A.8) with $\mu_i \sim 0$ and/or $\frac{\mu_i}{\sigma_i} \sim 0$

$$E\{|n_i|\} = \sqrt{\frac{2\sigma_i^2}{\pi}} e^{-\frac{\mu_i^2}{2\sigma_i^2}} - \mu_i \left(1 - 2Q(-\frac{\mu_i}{\sigma_i})\right) \sim \sqrt{\frac{2\sigma_i^2}{\pi}} \tag{C.5}$$

If n_0 and n_1 are uncorrelated, from (C.1), $\sigma_a^2 = \alpha_0^2\sigma_0^2 + \alpha_1^2\sigma_1^2$. Therefore, $E\{|n_a|\} \sim \sqrt{\frac{2\sigma_a^2}{\pi}} = \sqrt{\frac{2(\alpha_0^2\sigma_0^2 + \alpha_1^2\sigma_1^2)}{\pi}}$. Thus (C.4) is equal to

$$\begin{aligned} & MAD_a < MAD_0 \quad \& \quad MAD_a < MAD_1 \\ \Leftrightarrow & \sqrt{\frac{2\sigma_a^2}{\pi}} < \sqrt{\frac{2\sigma_0^2}{\pi}} \quad \& \quad \sqrt{\frac{2\sigma_a^2}{\pi}} < \sqrt{\frac{2\sigma_1^2}{\pi}} \\ \Leftrightarrow & \frac{\alpha_1^2}{1-\alpha_0^2} < \frac{\sigma_0^2}{\sigma_1^2} < \frac{1-\alpha_1^2}{\alpha_0^2}. \end{aligned} \tag{C.6}$$

From (C.3) and (C.6), it is shown that if the residual errors by p_0 and p_1 are uncorrelated and follow the normal distributions respectively with $\mu_i \sim 0$ and/or $\frac{\mu_i}{\sigma_i} \sim 0$, the same weight index is selected by both MSD and MAD.

(ii) If n_0 and n_1 follow Laplace distributions, from (A.11) and (A.12), MAD_0 and MAD_1 are found as

$$\begin{aligned} MAD_0 = E\{|n_0|\} &= \int_{-\infty}^{\infty} \frac{|n_0|}{2a} e^{-\frac{|n_0 - \mu_0|}{a}} dn_0 \\ &= ae^{-\frac{|\mu_0|}{a}} + |\mu_0| = \sqrt{\frac{\sigma_0^2}{2}} e^{-\frac{|\mu_0|}{\sigma_0}\sqrt{2}} + |\mu_0| \end{aligned} \tag{C.7}$$

$$
\begin{aligned}
MAD_1 = E\{|n_1|\} &= \int_{-\infty}^{\infty} \frac{|n_1|}{2b} e^{-\frac{|n_1-\mu_1|}{b}} dn_1 \\
&= be^{-\frac{|\mu_1|}{b}} + |\mu_1| = \sqrt{\frac{\sigma_1^2}{2}} e^{-\frac{|\mu_1|}{\sigma_1}\sqrt{2}} + |\mu_1|.
\end{aligned}
\tag{C.8}
$$

Probability density function (pdf) of n_a are calculated for $n_0 \perp n_1$ as

$$
\begin{aligned}
f_{\mathbf{n_a}}(n_a) &= \int_{-\infty}^{\infty} \frac{1}{\alpha_0} f_{\mathbf{n_0}}(\frac{n_a - \alpha_1 n_1}{\alpha_0}) f_{\mathbf{n_1}}(n_1) dn_1 \\
&= \frac{1}{2((a\alpha_0)^2 - (b\alpha_1)^2)} \left(a\alpha_0 e^{-\frac{|n_a-\mu_a|}{a\alpha_0}} - b\alpha_1 e^{-\frac{|n_a-\mu_a|}{b\alpha_1}} \right)
\end{aligned}
\tag{C.9}
$$

thus, MAD_a is given as

$$
\begin{aligned}
MAD_a = E\{|n_a|\} &= \int_{-\infty}^{\infty} \frac{|n_a|}{2((a\alpha_0)^2 - (b\alpha_1)^2)} \left(a\alpha_0 e^{-\frac{|n_a-\mu_a|}{a\alpha_0}} - b\alpha_1 e^{-\frac{|n_a-\mu_a|}{b\alpha_1}} \right) dn_a \\
&= \frac{(a\alpha_0)^3 e^{-\frac{|\mu_a|}{a\alpha_0}} - (b\alpha_1)^3 e^{-\frac{|\mu_a|}{b\alpha_1}}}{(a\alpha_0)^2 - (b\alpha_1)^2} + |\mu_a|
\end{aligned}
\tag{C.10}
$$

With the assumption that '$\mu_i \sim 0$' or '$\frac{\mu_i}{\sigma_i} \sim 0$ and $\mu_0 \sim \mu_a \sim \mu_1$', from (C.7) and (C.10),

$$
\begin{aligned}
& MAD_a < MAD_0 \\
& \Leftrightarrow \frac{(a\alpha_0)^2 + (b\alpha_1)^2 + a\alpha_0 b\alpha_1}{a\alpha_0 + b\alpha_1} < a \\
& \Leftrightarrow \frac{a}{b} < \frac{-\alpha_1 - \sqrt{\alpha_1^2 + 4\alpha_0\alpha_1}}{2\alpha_0} \quad \text{or} \quad \frac{a}{b} > \frac{-\alpha_1 + \sqrt{\alpha_1^2 + 4\alpha_0\alpha_1}}{2\alpha_0}
\end{aligned}
\tag{C.11}
$$

and from (C.8) and (C.10),

$$
\begin{aligned}
& MAD_a < MAD_1 \\
& \Leftrightarrow \frac{(a\alpha_0)^2 + (b\alpha_1)^2 + a\alpha_0 b\alpha_1}{a\alpha_0 + b\alpha_1} < b \\
& \Leftrightarrow \frac{\alpha_0 - \sqrt{\alpha_0^2 + 4\alpha_0\alpha_1}}{2\alpha_0} < \frac{a}{b} < \frac{\alpha_0 + \sqrt{\alpha_0^2 + 4\alpha_0\alpha_1}}{2\alpha_0}
\end{aligned}
\tag{C.12}
$$

Tab. C.1: Sub-optimality of MAD with respect to MSD

Range of $\frac{\sigma_0^2}{\sigma_1^2}$	Predictor by MSD	Predictor by MAD
(a) $\left(\frac{1}{3}, 0.382\right)$	p_a	p_0
(b) $(2.618, 3)$	p_a	p_1

Thus, from (C.11) and (C.12)

$$\begin{aligned} & MAD_a < MAD_0 \quad \& \quad MAD_a < MAD_1 \\ & \Leftrightarrow \frac{-\alpha_1 + \sqrt{\alpha_1^2 + 4\alpha_0\alpha_1}}{2\alpha_0} < \frac{a}{b} < \frac{\alpha_0 + \sqrt{\alpha_0^2 + 4\alpha_0\alpha_1}}{2\alpha_0} \\ & \Leftrightarrow \left(\frac{-\alpha_1 + \sqrt{\alpha_1^2 + 4\alpha_0\alpha_1}}{2\alpha_0}\right)^2 < \frac{\sigma_0^2}{\sigma_1^2} < \left(\frac{\alpha_0 + \sqrt{\alpha_0^2 + 4\alpha_0\alpha_1}}{2\alpha_0}\right)^2 \end{aligned} \tag{C.13}$$

For the additional weight used in Chapter 4, $\alpha_0 = \alpha_1 = \frac{1}{2}$,

$$\begin{aligned} & MSD_a < MSD_0 \quad \& \quad MSD_a < MSD_1 \\ & \Leftrightarrow \frac{\alpha_1^2}{1-\alpha_0^2} < \frac{\sigma_0^2}{\sigma_1^2} < \frac{1-\alpha_1^2}{\alpha_0^2}. \\ & \Leftrightarrow \frac{1}{3} < \frac{\sigma_0^2}{\sigma_1^2} < 3 \end{aligned} \tag{C.14}$$

and

$$\begin{aligned} & MAD_a < MAD_0 \quad \& \quad MAD_a < MAD_1 \\ & \Leftrightarrow \left(-\frac{1}{2} + \sqrt{\frac{5}{4}}\right)^2 < \frac{\sigma_0^2}{\sigma_1^2} < \left(\frac{1}{2} + \sqrt{\frac{5}{4}}\right)^2 \\ & \Leftrightarrow 0.382 < \frac{\sigma_0^2}{\sigma_1^2} < 2.618 \end{aligned} \tag{C.15}$$

Therefore, MSD and MAD select a different predictor in two separate intervals as in Tab. C.1. For (a), p_a is the predictor with minimum distortion by MSD but p_0 is chosen by MAD. Similarly, for (b) p_a is the predictor with minimum distortion

by MSD but p_1 is chosen by MAD. The sub-optimal choice by MAD increases the distortion as

$$\begin{aligned} &\text{(a)} \quad \Delta D_0 = MSD_0 - MSD_a = (1-\alpha_0^2)\sigma_0^2 - \alpha_1^2\sigma_1^2 = \frac{3}{4}\sigma_0^2 - \frac{1}{4}\sigma_1^2 \\ &\text{(b)} \quad \Delta D_1 = MSD_1 - MSD_a = (1-\alpha_1^2)\sigma_1^2 - \alpha_0^2\sigma_0^2 = \frac{3}{4}\sigma_1^2 - \frac{1}{4}\sigma_0^2. \end{aligned} \tag{C.16}$$

From two intervals in Tab. C.1, it can be derived that

$$\begin{aligned} &\text{(a) } 2.618\sigma_0^2 < \sigma_1^2 < 3\sigma_0^2 \\ &\text{(b) } 2.618\sigma_1^2 < \sigma_0^2 < 3\sigma_1^2 \end{aligned} \tag{C.17}$$

thus,

$$\begin{aligned} &\text{(a)} \quad 0 < \Delta D_0 < 0.10\sigma_0^2 \sim 0.11 MSD_a \\ &\text{(b)} \quad 0 < \Delta D_1 < 0.10\sigma_1^2 \sim 0.11 MSD_a. \end{aligned} \tag{C.18}$$

Therefore, when the residuals follow Laplace distribution, MAD will choose p_0 or p_1 which is not the predictor with minimum distortion (p_a). The distortion by this sub-optimal choice can be higher than the optimal distortion MSD_a by up to 11%. Fig. C.1 demonstrates the distribution of $\frac{\sigma_0^2}{\sigma_1^2}$ based on the coding results of *Foreman* with QP 24. In this example, $\frac{\sigma_0^2}{\sigma_1^2}$ is clustered around 1 and about 68% and 82% lies between $\left(\frac{1}{2}, 2\right)$ and between $\left(\frac{1}{3}, 3\right)$, respectively. The probability that $\frac{\sigma_0^2}{\sigma_1^2}$ belongs to the interval (a) and (b) in Tab. C.1 is equal to $P(\frac{1}{3} < \frac{\sigma_0^2}{\sigma_1^2} < 0.382 \quad \& \quad 2.618 < \frac{\sigma_0^2}{\sigma_1^2} < 3) = 6.7\%$. Thus, the penalty using MAD instead of MSD is negligible practically.

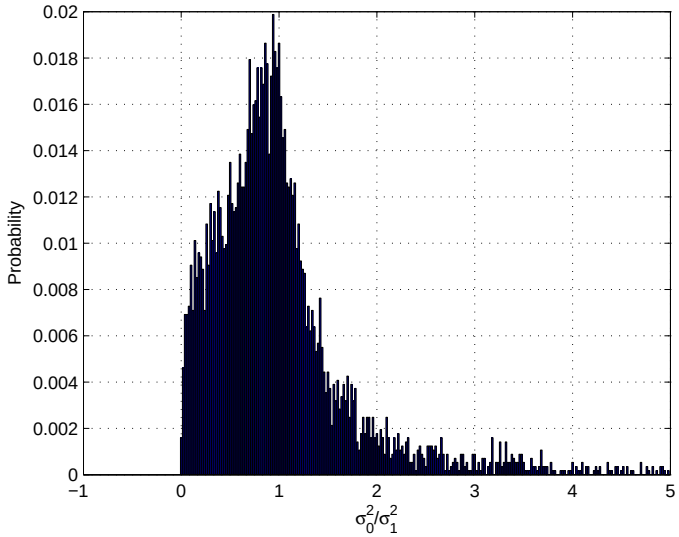


Fig. C.1: Distribution of $\frac{\sigma_0^2}{\sigma_1^2}$ from IBS coding results of *Foreman* with QP 24