Title

Quality, drift, and delay issues in multiple reference frame video coding

Permalink

Author

Leontaris, Athanasios

Publication Date

2006

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Quality, Drift, and Delay Issues in Multiple Reference Frame Video Coding

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Communication Theory and Systems)

by

Athanasios Leontaris

Committee in charge:

Professor Pamela C. Cosman, Chair
Professor Laurence B. Milstein
Professor Truong Nguyen
Professor Alon Orlitsky
Professor Geoffrey M. Voelker

2006

The dissertation of Athanasios Leontaris is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2006

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my advisor Professor Pamela Cosman for her guidance over the past several years. One of my uncles once said that one can only learn from people that are better than oneself. Prof. Cosman is definitely such a person.

In addition to supervising my work, Prof. Cosman was instrumental in helping me improve my public speaking skills. I can still recollect my first public presentation at U.C. San Diego which was a humbling experience. Furthermore, I admire her meticulousness, attention to detail, and analytical thought. I am grateful that she worked hard trying to instill these qualities in me.

Throughout my studies at U.C. San Diego I had the opportunity to participate at international conferences and also work as an intern at research laboratories. None of that would have been possible without her support. I am deeply grateful for the time that she always had available for discussing research problems with me. Last, I also thank her for her patience and understanding. I can only learn from her example.

I would also like to thank my dissertation committee members, Prof. Larry Milstein, Prof. Truong Nguyen, Prof. Alon Orlitsky and Prof. Geoffrey Voelker for their time, assistance, and advice. Prof. Milstein was inspirational as an instructor of digital communications and Prof. Nguyen made wavelets and filter banks an especially engaging course. Prof. Orlitsky was critical in my selection of U.C. San Diego. I was torn between two options and the reminder email he sent five years ago tipped the scales.

I am also grateful for Prof. Voelker's insightful questions during my candidacy exam, approximately one year ago, that helped improve some rough edges of my work.

I am also grateful to Dr. Amy Reibman at AT&T Labs-Research and Dr. Takayuki Nakachi at NTT Labs for their mentoring and being so supportive during my internships.

I thank my fellow researchers in the Information Coding Laboratory for the friendly and conducive atmosphere that made this dissertation possible. Special thanks to Yushi Shen, Sandeep Kanumuri, and Mayank Tiwari for being great co-workers as well as friends.

Finally, I would like to dedicate this dissertation to my beloved parents, for their unbounded love and support through all the years, as well as my beloved grandmother and uncle who recently passed away.

Chapter II of this dissertation, in full, is a reprint of the material as it appears in A. Leontaris and P. C. Cosman, "Video Compression for Lossy Packet Networks with Mode Switching and a Dual Frame Buffer," *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 885-897, July 2004. I was the primary author and the co-author Dr. Cosman directed and supervised the research which forms the basis for Chapter II.

Chapter III of this dissertation, in full, is a reprint of the material as it appears in A. Leontaris, V. Chellappa, and P. C. Cosman, "Optimal Mode Selection for a Pulsed-Quality Dual Frame Video Coder," *IEEE Signal Processing Letters*, vol. 11, no. 12, pp. 952-955, December 2004. Co-author Vijay Chellappa and I contributed equally towards

this publication. Co-author Dr. Cosman directed and supervised the research which forms the basis for Chapter III.

Chapter IV of this dissertation, in full, is a reprint of the material as it appears in A. Leontaris and P. C. Cosman, "Drift-Resistant SNR Scalable Video Coding," accepted in October 2005 for publication in the *IEEE Transactions on Image Processing*. I was the primary author and the co-author Dr. Cosman directed and supervised the research which forms the basis for Chapter IV.

Chapter V of this dissertation, in full, is a reprint of the material as it appears in A. Leontaris and P. C. Cosman, "Compression Efficiency and Delay Trade-Offs for Hierarchical B-Pictures and Pulsed-Quality Frames," which is under preparation for submission to a journal. I was the primary author and the co-author Dr. Cosman directed and supervised the research which forms the basis for Chapter V.

VITA

1977	Born, Thessaloniki, Greece
2000	Diploma, Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece
2000–2001	Research Assistant, Informatics and Telematics Institute, Thessaloniki, Greece
2002	M.S., Electrical Engineering (Communication Theory and Systems), University of California, San Diego
2002–2006	Research Assistant, University of California, San Diego
Summer 2004	Intern, AT&T Labs-Research, Florham Park, New Jersey
Summer 2005	Intern, NTT Network Innovation Labs, Yokosuka, Japan
2006	Ph.D., Electrical Engineering (Communication Theory and Systems), University of California, San Diego

PUBLICATIONS

A. Leontaris and P. C. Cosman, “Video Compression with Intra/Inter Mode Switching and a Dual Frame Buffer,” in *Proc. IEEE Data Compression Conference*, Snowbird, Utah, March 25-27, 2003.

A. Leontaris and P. C. Cosman, “Dual Frame Video Encoding with Feedback,” in *Proc. 37th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, November 9-12, 2003.

A. Leontaris and P. C. Cosman, “Video Compression for Lossy Packet Networks with Mode Switching and a Dual Frame Buffer,” *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 885-897, July 2004.

A. Leontaris and P. C. Cosman, “Optimal Per-Pixel Estimation for Scalable Video Coding,” in *Proc. IEEE International Conference on Image Processing*, Singapore, October 24-27, 2004.

A. Leontaris, V. Chellappa, and P. C. Cosman, “Optimal Mode Selection for a Pulsed-Quality Dual Frame Video Coder,” *IEEE Signal Processing Letters*, vol. 11, no. 12, pp. 952-955, December 2004.

A. Leontaris and P. C. Cosman, “Drift-Resistant SNR Scalable Video Coding,” accepted for publication in *IEEE Transactions on Image Processing*, October 2005.

A. Leontaris and P. C. Cosman, “End-To-End Delay for Hierarchical B-Pictures and Pulsed Quality Dual Frame Video Coders,” accepted for presentation in the *IEEE International Conference on Image Processing*, Atlanta, Georgia, October 8-11, 2006.

FIELDS OF STUDY

Major Field: Electrical Engineering
Studies in Communication Theory and Systems.
Professor Pamela C. Cosman.

ABSTRACT OF THE DISSERTATION

Quality, Drift, and Delay Issues in Multiple Reference Frame Video Coding

by

Athanasios Leontaris

Doctor of Philosophy in Electrical Engineering

(Communication Theory and Systems)

University of California, San Diego, 2006

Professor Pamela C. Cosman, Chair

The efficiency of modern video compression is largely a result of motion-compensated prediction (MCP) that exploits temporal correlation in the signal. Multiple reference frames for MCP improve compression efficiency and error resilience and require new paradigms in (a) reference frame selection, (b) bit rate allocation among the frames, (c) application to scalable video codecs, and (d) delay requirements of these codecs. We study all four of these aspects in this dissertation.

A dual-frame video coder employs two past reference frames (one short-term frame and one long-term frame available for prediction) for MCP. Dual-frame video codecs benefit greatly from near-optimal intra/inter mode switching within a rate- distortion framework. We show that such a scheme improves the error resilience of the

coder. We improve the mode-switching algorithm with the use of half-pel motion vectors. Furthermore, we investigate the effect of feedback in making more efficient mode-switching decisions.

In previous work, it was shown that uneven assignment of quality to frames, to create high-quality (HQ) long-term reference frames, can enhance the performance of a dual-frame encoder. Here, we demonstrate the performance advantages of optimal mode selection among such HQ frames for video transmission over noisy channels.

We investigate dual frame prediction for both base and enhancement layers of an SNR scalable video coder, with pulsed quality allocation in the base layer. Furthermore, a per-pixel drift estimation algorithm is introduced, where the encoder estimates the potential drift at the enhancement layer recursively and chooses coding modes accordingly.

Real-time video applications require tight bounds on end-to-end delay. Hierarchical bi-directional prediction requires buffering in the encoder input and output. Dual frame prediction with pulsed quality requires buffering at the encoder output. Both codecs involve uneven bit rate allocation that affects the encoder and decoder buffering requirements. We derive an efficient rate allocation for hierarchical B-pictures, and investigate the trade-off between delay and compression efficiency. Furthermore, we discuss the effect of the temporal prediction distance and prediction branch truncation.

Chapter I

Introduction

I.A Video Coding

Video communication is characterized by the huge amount of input data. A typical standard definition (SD) television (TV) channel has spatial resolution of 720×480 pixels, three color components (RGB), 8 bits per component, and is shown at 30 frames per second. The required transmission rate is 31 megabytes per second (248 megabits per second). Given that multiple (tens or even hundreds of) channels have to be accommodated simultaneously, compression is needed to reduce the number of transmitted bits. Widely-used lossy video compression sacrifices signal fidelity, while preserving visual quality.

Initially reserved for niche applications such as video-conferencing, and constrained to corporate and research use, digital video compression was used in the Video CD (VCD) through the ISO/IEC MPEG-1 digital video compression standard in 1986 (MPEG stands for Motion Picture Experts Group). The VCD used a spatial resolution of 352×240 pixels, almost comparable to that of the analog VHS tape. Furthermore, a single VCD could not hold more than 60 minutes of video. Widespread home use took hold in 1996, when the Digital Video Disk (DVD) with MPEG-2 compression was introduced. The move from the 650-megabyte VCD to the 8-gigabyte DVD enabled a four-fold increase in spatial resolution, at 720×480 pixels, the ability to fit over 120 minutes of content in a single disk, as well as the addition of high fidelity multi-channel digital sound.

The mandated transition of terrestrial and satellite TV to High Definition Television (HDTV) is a recent development. There are currently two widely used HDTV resolution formats: “720p” that refers to 1280×720 spatial resolution with progressive scan, and “1080i” that refers to to 1920×1080 spatial resolution with interlaced scan. Progressive scan denotes display of the video in raster-scan order on an entire frame basis. In interlaced scan, each frame is split into even and odd line fields and each field is displayed successively. MPEG-2 was the first standard to add support for efficient interlaced coding and is still used extensively for digital HDTV broadcast. However, the almost six-fold jump in data size motivated new research.

The first attempt to improve upon MPEG-2 was the ISO/IEC MPEG-4 Part 2 standard [2] which, although finalized in 2000, was not significantly deployed either in digital TV broadcast or video content packaged media. Even though it gained fame as the MP3 of video through the XviD and DivX codecs, its performance for HDTV scenarios was still found inadequate. As a result, a new standard, called H.264 (by ITU-T) [3] and MPEG-4 Part 10 AVC (by ISO/IEC), was developed that achieves the same reconstruction Peak Signal-to-Noise Ratio as MPEG-2 at less than half the bit rate requirements. We next discuss how video compression is achieved.

I.A.1 Intra-Frame Coding

The simplest form of video coding is intra-frame coding. In this arrangement, the current frame is encoded independently of previous and future frames. One such practical example is Motion JPEG (MJPEG) which is widely used in modern digital cameras. It consists of using a JPEG image coder on each incoming frame. Each frame is encoded as an independent image.

In short, JPEG operates as follows: (a) calculates the Discrete Cosine Transform (DCT) for each 8×8 pixel block, (b) quantizes (rounds off) the DCT coefficients given some Quantization Parameter (QP), which scales a quantization matrix applied to the 8×8 block DCT coefficients, (c) scans the quantized block DCT coefficients in a zig-zag order (meant to maximize the chance of consecutive zeros) and encodes the resulting runs of zeros, and levels, using a Variable Length Code (VLC) table. This type of entropy coding is called Run-Length Coding (RLC).

Intra-frame coding is computationally efficient and allows random access to every frame in the sequence. Furthermore, it is robust to errors: an error in a frame only affects that particular frame, since temporal prediction is not used. However, all the above are gained at a substantial cost in compression efficiency. The reason is that intra-frame coding does not exploit the high temporal correlation inherent in image sequences. Inter-frame video coders exploit this correlation.

I.A.2 Inter-Frame Coding: Motion Compensation

Two frames of an artificial video sequence are shown in Fig. I.1(a). To compress these two frames, one can encode them independently using a still image coder such as JPEG as discussed above. The two pictures are similar (temporally correlated), hence more compression can be obtained if we use the previous frame in some way to help with the coding of the current frame. One way to do this is to encode the difference between the two frames (i.e., subtract the previous frame from the current frame and encode that difference) rather than the current frame itself. In this case, the entire previous frame is called the prediction of the current frame.

This technique is shown in the first row of Fig. I.1(b). For sequences with little motion this technique ought to perform well; the difference between two similar frames is mostly zero, and is highly compressible. In Fig. I.1(a) for example, most of the ground and the bottom part of the tree on the right will be highly compressed since the difference for these areas will be zero. However the clouds have moved from one frame to the next, so the difference in those parts is non-zero, and will require many bits to represent. The key to achieving further compression is to compensate for this motion, by forming a better prediction of the current frame. This process of motion-compensated prediction of the current frame, and subsequent compression of the difference between the actual and predicted frames, is often called *hybrid coding*, and it forms the core of all MPEG-derived video compression standards.

When a camera pans or zooms, this causes global motion, meaning that all pixels in the frame (or almost all pixels) are apparently in motion in some related way,

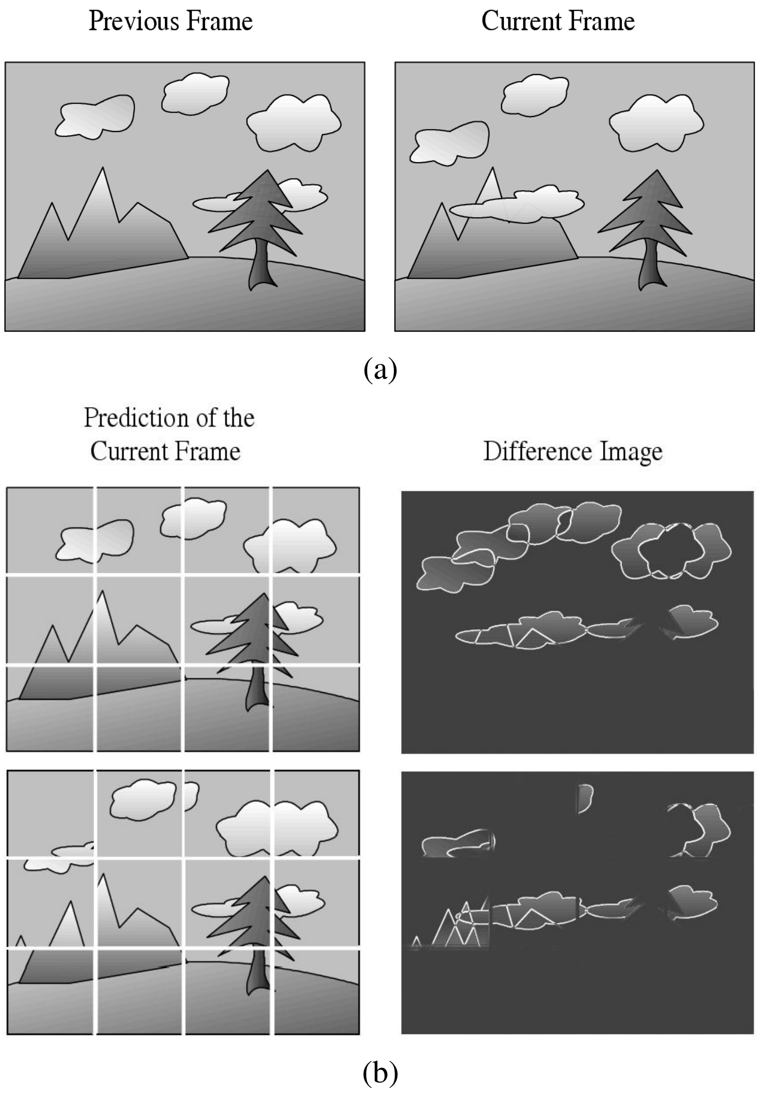


Figure I.1: Motion Compensated Prediction. (a) The previous and the current frame of the video sequence. (b) The top row shows the prediction which is the unaltered previous frame and the resulting difference image that has to be coded. The bottom row shows the equivalent prediction and error image for motion compensated prediction. The reduction of the error is apparent.

differing from the values they had in the previous frame. When the camera is stationary but objects in the scene move, this is called local motion. To compensate for local motion, a frame is typically subdivided into smaller blocks of pixels, in which motion is assumed to consist of uniform translation.

In block-based motion-compensated prediction (MCP), for each block in the current frame, a motion vector (MV) can be transmitted to the decoder to indicate which block in the previous frame is the best match block for the given block in the current frame, and therefore forms the best prediction of it. The MV points from the origin (top-left corner) of the current block to the origin of its best match block in the previous frame. MVs are essentially addresses of the best match blocks in the previous frame. If the MV is $(0, 0)$, then the best match block is the co-located block, the one with the same spatial coordinates.

As Fig. I.1(b) shows, the upper right cloud in the current frame is successfully compensated with a block in the previous frame with a non-zero MV. The block on the tree finds its best match with the $(0, 0)$ MV. The MVs are determined by doing motion search, also known as motion estimation (ME), in the previous frame. Assuming a search range of $[-16, +16]$ pixels for each spatial (horizontal and vertical) component, $(2 \times 16 + 1)^2 = 1089$ potential best match blocks can be referenced and have to be evaluated. The MV v, that minimizes either the sum of absolute differences (SAD) or the mean squared error (MSE) between the block in the current frame and the block in the previous frame that is referenced by v, is selected and transmitted.

To form the motion-compensated prediction frame for the current frame, the blocks that are addressed through the MVs are copied from their original spatial location to the location of the block in the current frame, as shown in Fig. I.1(b). This is subsequently subtracted from the current frame to yield the motion-compensated difference frame. Obviously, if the motion-compensated prediction frame is very similar to the current frame, then the difference frame will have most pixels close to zero, and hence be easier to compress, compared to doing MJPEG or direct subtraction of the previous frame. The difference information is typically transmitted to the decoder by transforming it using DCT, rounding off the coefficients to some desired level of accuracy (a process called quantization) and sending unique variable-length codewords to represent these rounded-off coefficients. Along with this difference information, the MVs are transmitted to the decoder, requiring some additional bit rate of their own. Still, the rate requirements are much less than without the use of MCP for video content with realistic temporal correlation.

The decoder uses the MVs to obtain the motion compensated prediction of the current frame from the previous frame, which has already been decoded. Then, the decoded difference frame is added to the motion-compensated prediction to yield the current decoded frame. This frame however will not be identical to the original one, because of the quantization used on the difference frame. MCP for a block is also known as inter-coding since inter-frame redundancy is used to achieve compression. In general, inter coding enables higher compression ratios but is less error resilient than intra coding. Video frames that use exclusively intra-coding to encode all blocks are called I-Frames, while frames that allow the use of either intra- or inter-coding are known as

P-Frames. P-frames have been traditionally constrained to reference past frames. Finally, B-Frames/Pictures allow bidirectional prediction from past and future frames in addition to intra- or inter-coding.

Block-based MCP traditionally made use of the previous frame to search for prediction blocks for the current frame. But the search does not have to be limited to the previous frame alone.

I.A.3 Multiple Reference Frames for Motion Compensation

Multiple-frame motion-compensated prediction has recently found its way into the new H.264/AVC video coding standard. Devised mainly for increasing compression efficiency, it exhibits other useful properties such as enhanced error resilience. We consider multiple frame prediction (MFP) where the current block is predicted from multiple reconstructed frames.

A multiple frame predictor with N reference frames as shown in Fig. I.2, requires in general N times the computational and memory complexity of a standard single-frame prediction scheme. Background information spanning a large temporal space is available. This is illustrated in Fig. I.2, where the macroblock in frame n (the current frame to be encoded) that contains the tree trunk is unable to find a suitable reference block in frame $n-1$, since the trunk has been occluded. A good match, however, is available in frame $n-6$. Compared to predicting from the previous frame alone, we now need a way to signal to the decoder which reference frame has to be used to reconstruct each block. Assuming we have access to N buffered reconstructed frames

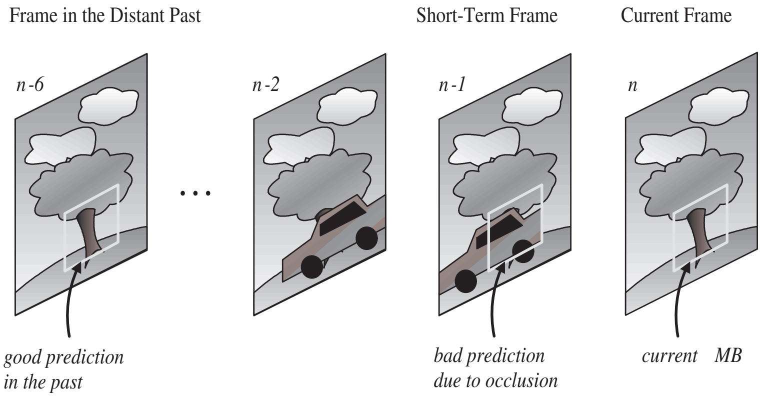


Figure I.2: Multiple Frame Motion-Compensated Prediction. The availability of multiple reference frames improves motion compensation by providing good matches even when occlusion is temporarily present.

we need to transmit $\lceil \log_2 N \rceil$ bits per block to signal the decoder.

Methods for multiple frame prediction can be categorized according to how many reference frames they employ and which ones. We use the term MFP to denote schemes where the reference frames are actual reconstructed frames, and the term background coding to denote schemes where one or more of the reference frames is a synthetic one that is constructed to resemble the image background.

For MFP, we first consider methods that use more than one reference frame within a causal sliding window of past frames, including the previous frame. Such a system with three references would predict frame n from frames $n-1$, $n-2$, and $n-3$. An advantage of this arrangement is the simplicity of its implementation with a First-In First-Out (FIFO) frame buffer. These are termed sliding-window methods.

The second category includes methods where the additional references consist of frames back in the past, say frame $n-N$, where $N > 1$, intended to capture long-term

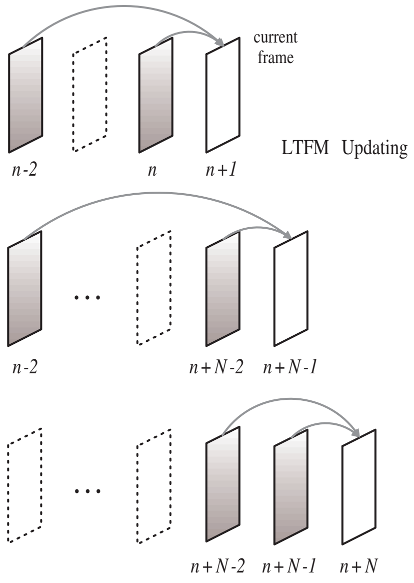


Figure I.3: A Long-Term/Short-Term Frame Memory Scheme. The best match block is located either in the previous frame or in a long-term frame memory that stores an older frame that is not necessarily the one previous to the short-term frame, or finally, can be a linear combination (multihypothesis) of both.

video content. These are termed long-term methods. The previous frame is buffered in the Short Term Frame Memory (STFM), and the long-term frame in the Long-Term Frame Memory (LTFM), which is periodically updated. Suppose we are currently encoding frame n, and suppose the LTFM, which gets updated every N frames, has just been updated to be frame $n-2$. Frame n will now reference frames $n-1$ and $n-2$. Then, frame $n+1$ will predict from frames n and $n-2$. Thus, frame $n-2$ will continue to be buffered until frame $n+N$ is the current frame where the LTFM will be flashed with frame $n+N-2$. Fig. I.3 depicts this scheme.

Why Do Multiple Reference Frames Make Sense for Compression?

Often cited [4, 5] reasons for the improved compression performance of MFP over single-frame prediction (SFP), many of which are also valid for background coding, can be summarized as follows:

1. Periodic occurrence of motion. Useful instances of the same object could exist in some past frame.

2. Uncovered background. Moving objects occlude parts of the background not available in the previous frame. However, due to motion, that part may be uncovered in some frame farther back in the past.

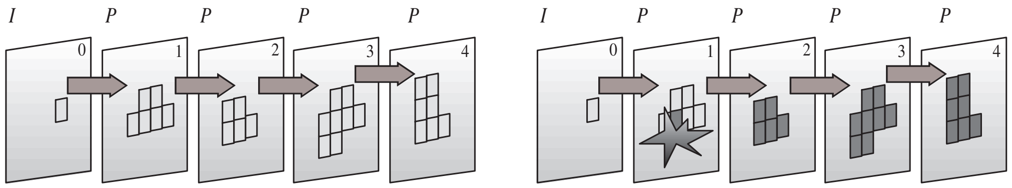
3. Alternating camera scenes due to camera (global) periodic movement, such as camera shaking, where a frame farther back in the past could easily be a much better prediction of the current frame than the immediate previous one.

4. Sampling grid. In past frames we often have access to versions of the current frame that correspond to arbitrary fractional pixel displacements, which cannot be obtained by traditional fractional pixel motion compensation.

5. Lighting and shadow changes. Small (periodic) variations of the global luminance can render the previous frame of little use as a prediction reference. At the same time moving objects cast shadows on other objects or the background with the same undesirable effect.

6. Noise introduced in the signal due to camera distortion, or environmental factors.

In summary, the extension of the block codebook alphabet with additional frames/blocks is the primary reason behind the increased performance of multiple frames,



(a) (b)

Figure I.4: Error Resilience for a Motion-Compensated Video Codec. The large arrows denote motion-compensated prediction. Blocks on the left of an arrow are prediction references of blocks on the right. (a) The last seven blocks in the current frame are all decoded correctly because all of the blocks across frames 0, 1, 2, and 3 were decoded correctly. (b) The loss of a single block in frame 1 corrupts all blocks referencing it in frame 2 and the error propagates until everything in its path has been corrupted.

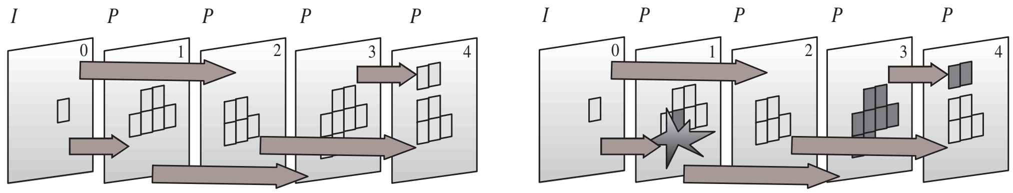
as first pointed out in [6] and [7]. More choices from which to choose can increase the probability for a better match. Still, even if infinite memory and computational resources are available, the bit rate overhead required to index the reference frame would grow prohibitively large. Hence there is a point of diminishing returns.

Multiple Frames for Error Resilience

In hybrid video coding, error corruption in one frame can propagate to subsequent frames via motion compensated prediction. This is illustrated in Fig. I.4. The dark squares denote corrupted blocks.

Multiple reference frames reduce the error probability for a decoded frame transmitted over an error-prone channel. The intuition behind this is illustrated in Fig. I.5.

Hence multiple references help not only in compression but also in error resilience. In general, the main issues in multiple frame prediction are (a) the type and number of reference frames, and (b) the criterion used to select the reference frame. We



(a) (b)

Figure I.5: Error Resilience for a Multiple-Frame Motion-Compensated Video Codec. (a) Multiple arbitrary prediction paths are created since there is no restriction as to which frame each block references. (b) The loss of a single block again causes error propagation, but in this case it is easier to contain it, and blocks in frames 2 and 4 survive unharmed. The "domino effect" we witnessed in the case of traditional single-frame MCP is avoided now.

now provide a quick overview on the development of these techniques.

I.A.4 Background Prediction Literature

Chronologically, MFP received attention after the development of background coding methods. The high computational cost associated with motion estimation and the high procurement cost of computer hardware imposed computational and memory constraints that favored prediction from at most two reference frames; the previous and some kind of background frame.

During the mid 1980's, video compression was aimed at video-conferencing, often at speeds as low as 28kbps. Researchers were hard pressed to make every transmitted bit count. Video-conferencing consists of people talking in front of a quite static background (an office, a blackboard, etc.). Small gestures and movements can temporarily occlude parts of the background, but these are often uncovered again later on. This problem could be fixed by using not only the previous frame for reference blocks, but using as well a secondary background memory that stored content deemed to be static.

Most of the techniques update small portions of the background frame, which can be either macroblocks, blocks or single pixels. The first known work in this genre attempted to identify background/object pixels [8]. If the difference between the pixel in the current frame and the one in the previous frame was below a threshold, the pixel was classified as background, and the corresponding background memory pixel was updated. A more complex approach followed in [9], that made use of image segmentation. In [10], the background detection was accomplished by counting the number of times a pixel is unchanged between two frames. Compared to [8], information from more than one frame is thus used to decide whether the current pixel is static or not. An evolution of the updating algorithm of [10] appeared in [11]; here the pixel is considered unchanged if the Sum of Absolute Differences (SAD) for a square neighborhood around the pixel in the current and previous frame is below a threshold. A similar change detector was later employed in [12].

A novel element to this approach appeared in [13], where the authors buffered a map of the quantization parameter (QP) that was used to encode each pixel. A low value of QP means the information is rounded off with great accuracy, whereas a high value of QP means that a coarse approximation was done in the compression of that information. The goal was to refresh the background memory with high quality pixel values.

Vector quantization (VQ) was used to select, buffer, and remove the frame elements (blocks) in [7] within a so-called image library. Image blocks were fed to a Linde-

Buzo-Gray VQ design algorithm, which merged the functions of obtaining and removing the no-longer needed image blocks. In contrast, the selection of relevant blocks in [14] was done by satisfying a SAD-based criterion of frame coherence, while the removal of irrelevant blocks from the memory used a FIFO scheme where blocks with the lowest priority were removed first. Both selection and removal algorithms used reconstructed frames and hence the decoder was able to replicate the encoder's operation without the need to explicitly transmit the library as in [7].

Multiple Reference Frames for Compression

Block-based MFP uses either a sliding-window or a long-term frame arrangement. The first study [6] used up to eight past frames. It was shown that the MSE of the prediction error is lower compared to prediction solely from the previous frame. An actual video coder with MFP was then presented in [15], where up to fifty past frames were used for prediction by modifying a H.263 video codec.

Long-term methods were initially devised to keep the computational and memory complexity low and enable an easier theoretical formulation and efficiency analysis. The first such example used just two reference frames for prediction [16] and is shown in Fig. I.3. A similar scheme was employed in [17], while in [18] every N-th frame is coded at a somewhat higher bit rate.

Different approaches were adopted to select the best match block. In [6] the best match was selected by minimizing the SAD. Similarly in [16, 18] the best inter match block was selected by minimizing the SAD prediction error, while the decision to use inter or intra was made by selecting the mode with the lowest SAD, by slightly biasing against intra.

In [15], rate-constrained motion estimation was applied to select the MV and the temporal reference by minimizing the rate-distortion (RD) Lagrangian Cost. In the framework of Lagrangian Minimization, the cost $J = D + \lambda R$ is the sum of the distortion measure D associated with a coding decision and the estimated or actual bit rate R required to encode according to that decision. The rate is multiplied by the Lagrangian multiplier λ which seeks to optimize the selection. In this particular case J is comprised of the MV (including the temporal reference) coding cost and the resulting MSE distortion. A further RD cost minimization is then used to select the coding mode. In contrast, this dissertation (Chapter II) employed RD optimization for joint mode and reference frame selection, but not for the spatial MV as in the two-step RD selection scheme of [15].

Multiple Reference Frames for Error Resilience: Block-Based

Compared to using MFP for improving compression efficiency, where the number and type of frames is important, in the error resilience case, the critical part is the reference frame selection. Here we investigate cases where each block in a frame uses a different temporal reference.

Multiple ($N > 2$) frames were used as references in [19, 20, 21, 22]. Deciding which frame to use as reference has to take into account error resilience. The decision is a trade-off between error resilience and compression efficiency, and is solved with rate-distortion techniques in [19]. The distortion calculation becomes now an estimation problem as potential corruption is modeled probabilistically to obtain an estimate of the resulting distortion. Different approaches have been proposed on how to estimate the distortion due to errors.

Branches of an event tree were employed to model error propagation in [19]. Each tree leaf represents a single event, i.e. "first frame received and second frame lost". To limit complexity since modeling the n-th frame's distortion requires a 2^n-leaf tree, the tree was pruned in [19]. However, if the status of a transmitted frame is known, the event tree can be reinitialized with that decoded frame as its root. Hence assuming a feedback delay of $d \ll n$ frames, the tree will now have 2^d leaves.

In [19, 15], first the reference frame is determined per block with rate- constrained motion estimation, and then the coding mode with another RD decision. A simpler scheme was employed for reference frame selection in [20], where the frame minimizing the SAD metric over all reference frames is selected.

In a very different approach, presented in [21] and [22], multiple frames were buffered but the decision which to use was controlled by feedback signals. Reference frame selection was avoided in [23], where periodic key frames (either I or P-pictures) were afforded forward error correction (FEC) to selectively improve error resilience, and it was proposed that periodic key P-pictures employ long-term MCP to predict exclusively from previous key pictures. The regular P-frames (without FEC) were constrained to only reference frames as old as their past key picture, creating thus a separate prediction path that increased error resilience.

Valuable theoretical insight into the error resilience of multiple frame prediction schemes is given in [20], which first proved using Markov chains that MFP reduces the error probability for a decoded frame transmitted over an error-prone channel. Intuition, as in Fig. I.5, also pointed to that conclusion.

Multiple Reference Frames for Error Resilience: Frame-Based

Here the reference frame is selected on a frame basis. The first application was video transmission over multiple network paths, whose error statistics may vary. Two separate transmission paths were investigated in [24, 25, 26, 27]. Frames transmitted along one path could reference frames transmitted along the other in [27, 26]. In [25], however, the predictor was constrained to only reference frames sent along the same path. Two important decisions have to be reached: from which reference frame to predict the current frame, and along which path to send it.

Both [25] and [26] make use of feedback, in contrast to [27]. The reference selection scheme proposed in [24] is extended in [25] to employ path diversity for streaming applications, adopting the former's RD-optimized decision scheme as well as a distortion estimator. The use of feedback was critical though in [26] in allowing prediction from frames in different paths. The strategy employed was to use the last frame that is believed to be reliable.

A new technique for reference picture selection was proposed in [27] which also avoided the use of feedback. Dynamic programming was used to solve the problem with the help of graphs that described potential reference frame selections. Furthermore, the authors did not seek to minimize the expected distortion (PSNR) of the reconstructed frames, rather the goal was to optimize the number of correctly received frames.

I.B Motivation

Multiple frame prediction is beneficial both for compression efficiency as well as for error resilience. To design a good multiple reference frame predictor, there are two critical parts: (a) the type and number of the multiple reference frames, and (b) the decision mechanism that selects the reference frame. For example, different criteria have to be employed when the goal is compression efficiency compared to criteria for error resilience.

We thus investigate optimal reference frame and coding mode decision algorithms for the case of bit stream error resilience. Furthermore, the use of long-term reference frames changes the individual significance of each frame. In traditional MCP from the previous frame, assuming roughly equal entropy among frames, all frames will require approximately the same bit rate. For long-term frames, that are referenced for quite a long time after they are displayed, this rate allocation model may no longer be good. We investigate this problem.

Most MPEG-derived codecs are fixed rate coders. Fixed rate coders produce a bit stream that has to be available in its entirety to be decoded successfully. Otherwise decoding fails outright or serious visual errors are introduced. Fixed-rate coders benefit from near-optimal rate-distortion performance. Another category of coders, called SNR scalable coders, exhibits the highly desirable characteristic of producing an embedded bit stream. An embedded bit stream can be truncated at almost any point and still enable successful decoding of the encoded image sequence, albeit at a lower fidelity, without introducing errors. We intend to investigate the use of multiple reference frames in scalable video codecs. A special case of scalable codecs suffers from drift. We also address this issue.

Last, we intend to take a closer look at the influence of multiple reference frames on delay. For example, in the proposed case of long-term frame prediction, the long-term frame is afforded more bit rate compared to the rest of the frames, and this causes an increase in delay. We are thus motivated to explore this trade-off of compression efficiency for delay.

I.C Thesis Outline

In Chapter II, a dual-frame video coder employs two past reference frames (one short-term frame and one long-term frame available for prediction) for MCP. Dual-frame video codecs benefit greatly from near-optimal intra/inter mode switching within a rate-distortion framework. We show that using a dual-frame buffer together with intra/inter mode switching improves the compression performance of the coder. We improve the mode-switching algorithm with the use of half-pel motion vectors. In addition, we investigate the effect of feedback in making more informed and effective mode-switching decisions.

In Chapter III, we investigate pulsed-quality for long-term frames. The dual-frame encoder can have advantages both in distortion-rate performance and in error resilience. In previous work, it was shown that uneven assignment of quality to frames, to create high-quality (HQ) long-term reference frames, can enhance the performance of a dual-frame encoder. Here, we demonstrate the performance advantages of optimal mode selection among HQ frames for video transmission over noisy channels.

In Chapter IV, we investigate dual frame prediction for both base and enhancement layer of an SNR scalable video coder, with pulsed quality allocation in the base layer. In addition, we address the problem of enhancement layer drift estimation. An optimal per-pixel drift estimation algorithm is introduced. The encoder recursively estimates the potential drift and chooses coding modes accordingly.

In Chapter V, we treat delay. Real-time video applications require tight bounds on end-to-end delay. Hierarchical bi-directional prediction requires buffering frames in the encoder input and output buffer. Dual frame prediction with pulsed quality requires buffering at the encoder output. Both codecs involve uneven bit rate allocation that affects the encoder and decoder buffering requirements. We derive an efficient rate allocation for hierarchical B-pictures. In addition, we discuss the effect of the temporal prediction distance, and delay trade-offs for prediction branch truncation. Finally, we investigate the trade-off between delay and compression efficiency.

In the Conclusions section, we enumerate our contributions in this dissertation for each individual chapter, and discuss open problems and potential future work. We note that partial conclusions are also given at the end of each individual chapter.

Chapter II

Video Compression for Lossy Packet Networks with Mode Switching and a Dual Frame Buffer

Packet-switched networks have become ubiquitous and form the backbone of the Internet. These networks have been designed with delivery of data in mind [28]. Thus, protocols such as TCP provide guaranteed transmission of packets but are not well suited for real-time delivery of streaming video content [29]. UDP on the other hand is widely used for streaming video through higher level protocols such as RTP. Due to time constraints imposed by real-time operation, it is not feasible to retransmit packets which were lost due to network congestion or buffer overflows. Consequently, packet losses can severely corrupt an unprotected bitstream. The transmitted bitstream has to be organized so as to minimize corruption and error propagation due to dropped packets. Error resilience can be improved in two ways: (a) with the use of MFP, and (b) the intelligent selection of intra or inter coding for each block.

Rate-distortion based techniques for optimal coding mode selection were studied in [30], [31], [32], [33] and [34]. For the case of error resilience, one seeks to estimate the resulting distortion due to potential packet drops. A novel algorithm for calculating estimated distortion due to packet losses was introduced in [35] and will be described in Section II.A. Robust video transmission was studied in [36], [37] and [19]. In [19], long-term memory motion-compensated prediction was used, and distortion due to error propagation was modeled as a tree where each leaf represented different decoded versions of the same frame. The final computationally tractable model that was adopted by the authors used only three branches, reducing the accuracy of the model. Feedback performance was also investigated. In [38], K encoder/decoder pairs were simulated under K different error patterns to model potential errors. However, even for $K = 30$ that was used, convergence was not guaranteed and the distortion estimation algorithm exhibits $O(K)$ complexity.

In this chapter, we show how using a dual frame buffer together with an algorithm for intra/inter mode switching decisions can lead to improved compression performance. We first examine performance assuming no feedback is present, and then we experiment with a more refined updating that takes into account feedback signals to effectively synchronize the long-term frame buffers of both the encoder and decoder.

The chapter is organized as follows. In Section II.A we review the ROPE algorithm [35] for distortion estimation. In Section II.B, we show how this algorithm can be used in the context of a dual frame buffer. The use of half-pel motion vectors is covered in Section II.C. Results in the absence of feedback are presented in Section II.D. In Section II.E we describe the feedback extensions, with experimental results given in Section II.F. Complexity is analyzed in Section II.G. Finally, conclusions are drawn in Section II.H.

II.A ROPE Algorithm

Recent attempts to switch coding modes according to error robustness criteria can be found in [39], [37], [38] and [35]. Our work makes use of the Recursive Optimal per-Pixel Estimate (ROPE) algorithm [35] which provides distortion estimates, which are then used for mode decision in hybrid video coders operating over packet erasure channels. In general, inter-mode achieves higher compression efficiency than intra-mode, at the cost of potentially severe error propagation. A single error in a past frame may corrupt all subsequent frames if inter-coding is used repeatedly. This error propagation can only be stopped by transmitting and successfully receiving an intra-coded macroblock. The problem that arises is how to optimally select between intra- and inter-coding for each macroblock, such that both error resilience and coding efficiency are achieved.

We assume that the video bitstream is transmitted over a packet erasure channel (lossy packet network). Each frame is partitioned into Groups Of Blocks (GOB). Each GOB contains a single horizontal slice of macroblocks (MBs) and is transmitted as a single packet. Each packet can be independently received and decoded, due to resynchronization markers. Thus, loss of a single packet wipes out one slice of MBs, but keeps the rest of the frame unharmed.

Let p be the probability of packet erasure, which is also the erasure probability for each single pixel. When the erasure is detected by the decoder, error concealment is applied [40], [41]. The decoder replaces the lost macroblock by one from the previous frame, using as motion vector (MV) the median of the MVs of the three closest macroblocks in the GOB above the lost one. If the GOB above has also been lost (or the 3 nearest MBs were all intra-coded and therefore have no motion vectors), then the all-zero $(0, 0)$ MV is used, and the lost macroblock is replaced with the co-located one from the previous frame.

We will now summarize the ROPE algorithm [35] in some detail as these equations will prove useful in elaborating our proposed method. Within this section we make use of the notation and equations from [35]. Frame n of the original video signal is denoted f_n, which is compressed and reconstructed at the *encoder* as $\hat{f}_n$. The decoded (and possibly error-concealed) reconstruction of frame n at the receiver is denoted by $\tilde{f}_n$. The encoder does not know $\tilde{f}_n$, and treats it as a random variable.

Let f_n^i denote the original value of pixel i in frame n, and let $\hat{f}_n^i$ denote its *encoder* reconstruction. The reconstructed value at the *decoder*, possibly after error concealment, is denoted by $\tilde{f}_n^i$. The expected distortion for pixel i is:

$$d_n^i = E\{(f_n^i - \tilde{f}_n^i)^2\} = (f_n^i)^2 - 2f_n^i E\{\tilde{f}_n^i\} + E\{(\tilde{f}_n^i)^2\} \qquad \text{(II.1)}$$

Calculation of d_n^i requires the first and second moments of the random variable of the estimated image sequence $\tilde{f}_n^i$. To compute these, recursion functions are developed in [35], in which it is necessary to separate out the cases of intra- and inter-coded MBs.

For an intra-coded MB, $\tilde{f}_n^i = \hat{f}_n^i$ with probability $1 - p$, corresponding to correct receipt of the packet. If the packet is lost, but the previous GOB is correct, the concealment based on the median motion vector leads the decoder to associate pixel i in the current frame with pixel k in the previous frame. Thus, $\tilde{f}_n^i = \tilde{f}_{n-1}^k$ with probability $p(1-p)$. Finally, if both current and previous GOB-packets are lost, $\tilde{f}_n^i = \tilde{f}_{n-1}^i$ (occurs with probability p^2). So the two moments for a pixel in an intra-coded MB are [35]:

$$E\{\tilde{f}_n^i\} = (1-p)(\hat{f}_n^i) + p(1-p)E\{\tilde{f}_{n-1}^k\} + p^2 E\{\tilde{f}_{n-1}^i\} \tag{II.2}$$

$$E\{(\tilde{f}_n^i)^2\} = (1-p)(\hat{f}_n^i)^2 + p(1-p)E\{(\tilde{f}_{n-1}^k)^2\} + p^2 E\{(\tilde{f}_{n-1}^i)^2\} \tag{II.3}$$

For an inter-coded MB, let us assume that its true motion vector is such that pixel i is predicted from pixel j in the previous frame. Thus, the encoder prediction of this pixel is $\hat{f}_{n-1}^j$. The prediction error, e_n^i, is compressed, and the quantized residue is $\hat{e}_n^i$. The encoder reconstruction is:

$$\hat{f}_n^i = \hat{f}_{n-1}^j + \hat{e}_n^i \tag{II.4}$$

The encoder transmits $\hat{e}_n^i$ and the MB's motion vector. If the packet is correctly received, the decoder knows $\hat{e}_n^i$ and the MV, but must still use its own reconstruction of pixel j in the previous frame, $\tilde{f}_{n-1}^j$, which may differ from the encoder value $\hat{f}_{n-1}^j$. Thus, the decoder reconstruction of pixel i is given by:

$$\tilde{f}_n^i = \tilde{f}_{n-1}^j + \hat{e}_n^i \tag{II.5}$$

Again, the encoder models $\tilde{f}_{n-1}^j$ as a random variable. The derivation of the moments is similar to the intra-coded MB for the last two cases, but differs for the first case where there is no transmission error (probability $1-p$). The first and second moments of $\tilde{f}_n^i$ for a pixel in an inter-coded MB are then given by:

$$E\{\tilde{f}_n^i\} \quad = \quad (1-p)\left(\hat{e}_n^i + E\{\tilde{f}_{n-1}^j\}\right) + p(1-p)E\{\tilde{f}_{n-1}^k\} + p^2E\{\tilde{f}_{n-1}^i\} \tag{II.6}$$

$$\begin{aligned} E\{(\tilde{f}_n^i)^2\} \quad &= \quad (1-p)\left((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-1}^j\} + E\{(\tilde{f}_{n-1}^j)^2\}\right) \\ &+ \quad p(1-p)E\{(\tilde{f}_{n-1}^k)^2\} + p^2E\{(\tilde{f}_{n-1}^i)^2\} \end{aligned} \tag{II.7}$$

These recursions are performed at the *encoder* in order to calculate the expected distortion at the *decoder*. The encoder can exploit this result in its encoding decisions, to optimally choose the coding mode for each MB. The expectation for each pixel is calculated as a weighted sum (due to the probabilities) of pixel expectations from the previous frame, prediction residuals and intra coefficients.

II.A.1 Rate-Distortion Framework

The ROPE algorithm estimates the expected distortion, due to both compression and transmission errors, to be used for optimal mode switching. The encoder switches between intra- or inter-coding on a macroblock basis, in an optimal fashion for a given bit rate and packet loss rate. The goal is to minimize the total distortion D subject to a bit rate constraint R. Using a Lagrange multiplier λ, the ROPE algorithm minimizes the total cost $J = D + \lambda R$. Individual MB contributions to this cost are additive, thus it can be minimized on a macroblock basis. Therefore, the encoding mode for each MB is chosen by minimizing

$$\min_{(mode,QP)} J_{MB} = \min_{(mode,QP)} (D_{MB} + \lambda R_{MB}) \tag{II.8}$$

where the distortion D_{MB} of the MB is the sum of the distortion contributions of the individual pixels. Rate control is achieved by modifying λ as in [42]. Both the *coding mode* and the *quantization step size* QP are chosen to minimize the Lagrangian cost. This is computationally complex for the encoder, but it enhances coding efficiency. The resulting bitstream is compatible with a standard compliant decoder.

We note that while the ROPE algorithm is optimal under the given assumptions, there is potential for improvement by incorporating the motion vector choice into the rate-distortion framework, or by correctly estimating distortion for half-pel vectors (the algorithm only models distortion for integer motion vectors).

II.B Dual Frame Buffer Extension

Our research has focused on using a dual frame buffer together with optimal mode switching within a rate-distortion framework. The basic use of the dual frame buffer is as follows. While encoding frame n, the encoder and decoder both maintain two reference frames in memory. The short-term reference frame is frame $n-1$. The long-term reference frame is, say, frame $n-k$, where k may be variable, but is always greater than 1. Each macroblock can be encoded in one of three coding modes: intra coding, inter coding using the short-term buffer (inter-ST-coding), and inter coding using the long-term buffer (inter-LT-coding). This is illustrated in Fig. II.1. The choice among these three will be made using an extended version of the ROPE algorithm, as described below. Once the coding mode is chosen, the syntax for encoding the bit stream is almost identical to the standard case of the single frame buffer. The only modification is that, if inter coding is chosen, a single bit will be sent to indicate use of the short-term or long-term frame.

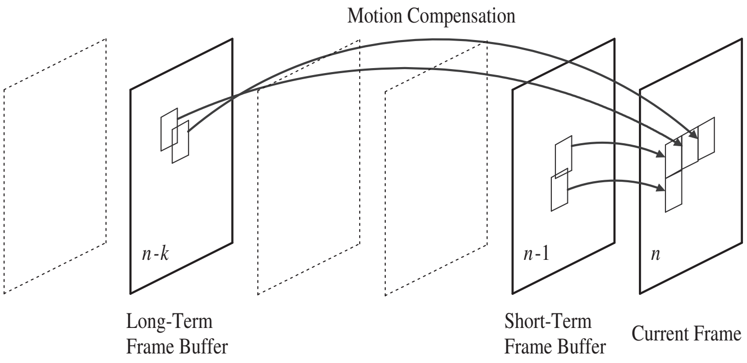
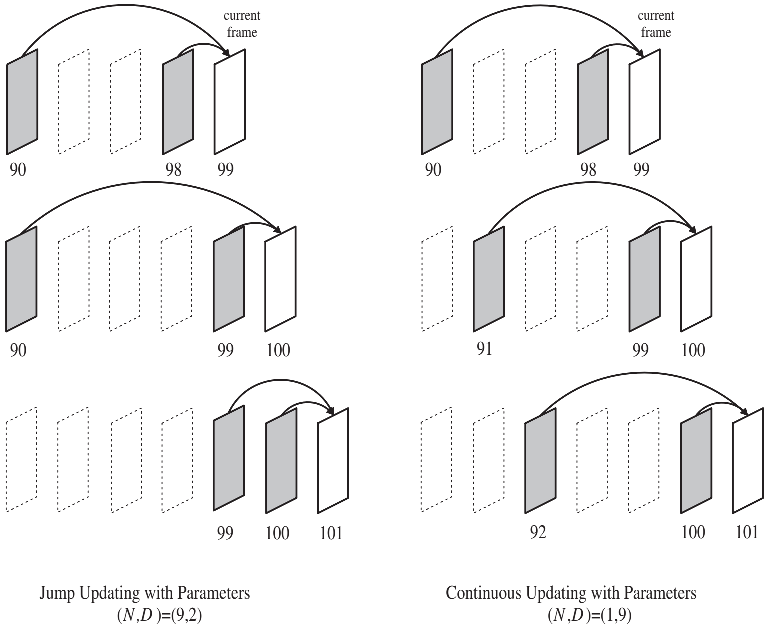


Figure II.1: Dual Frame Buffer Motion Compensation.

The choice among the three coding modes does not, of course, need to be done using an extension of the ROPE algorithm. A naive approach would be to use a traditional distortion estimator that evaluates the distortion from motion compensation and quantization alone. However, experimental results showed a substantial advantage (up to 3-4dB) to using a rate-distortion-based decision with a ROPE distortion estimator instead of a rate-distortion-based decision with a traditional distortion estimator. This was true for both single frame and dual frame coders. Given the substantial benefit to using the ROPE distortion estimator over a traditional distortion estimator, this chapter focuses on extending the ROPE algorithm to work with a dual frame coder, and comparing it against a single frame ROPE coder.

We now describe how the long term reference frame is chosen. In one approach, which we call *jump updating*, the long term reference frame varies from as recent as frame $n-2$ to as old as frame $n-N-1$. When encoding frame n, if the long-term reference frame is $n-N-1$, then, when the encoder moves on to encoding frame $n+1$, the short-term reference frame will slide forward by one to frame n, and the long-term reference frame will jump forward by N to frame $n-1$. The long-term reference frame will then remain static for N frames, and then jump forward again. We refer to N as the jump update parameter. This approach was adopted in [16].

A novel approach, which we call *continuous updating*, entails continuously updating the long-term frame buffer so that it contains a frame with a fixed temporal distance from the current buffer. Therefore, the buffer always contains the $n-D$ frame for each frame n. We refer to D as the continuous update parameter. These two approaches are depicted in Fig. II.2.



Jump Updating with Parameters (N,D)=(9,2)

Continuous Updating with Parameters (N,D)=(1,9)

Figure II.2: Two different dual frame buffer approaches. In the top row, frame 99 is being predicted from frames 98 and 90. In the middle row, the current frame to be encoded is frame 100. With the jump updating approach, frames 99 and 90 are used for prediction. With the continuous updating approach, frames 99 and 91 are used. However, as we examine the bottom row we observe that jump updating takes place when 101 is encoded. Thus, the new long-term frame buffer will be frame 99, while for the continuous updating approach we will use 92.

We note that both jump updating and continuous updating can be viewed as special cases of a more general (N, D) updating strategy, in which the long term reference frame jumps forward by an amount N to be the frame at a distance D back from the current frame to be encoded, and then remains static for N frames, and jumps forward again. For general (N, D) updating, a frame k might have an LT frame as recent as frame $k-D$ or as old as frame $k-N-D+1$. In our definition of jump updating, N can be selected freely for each sequence, and $D = 2$, (meaning that when updating occurs, the LT frame jumps forward by N to become frame $n-2$). In continuous updating, D can be selected freely for each sequence and N is fixed at 1. Clearly the most general updating strategy would have no fixed N or D; rather the long term frame buffer would be updated irregularly when needed, to whatever frame is most useful. In our trials, (N, D) remain fixed while coding one sequence.

Let us now elaborate on how the choice is made among the coding modes. As before, we use f_n, $\hat{f}_n$, and $\tilde{f}_n$ to denote the original frame n, the encoder reconstruction of the compressed frame, and the decoder version of the frame, respectively. We assume that the long-term frame buffer was updated l frames ago. Thus, it contains $\hat{f}_{n-l}$ at the transmitter and $\tilde{f}_{n-l}$ at the receiver. The expected distortion for pixel i in frame n is given by Equation II.1.

To compute the moments in Equation II.1, the recursion steps for pixels in intra-coded and inter-ST-coded MBs are identical to the corresponding steps in the original ROPE algorithm. For a pixel in an inter-LT-coded MB, we assume that the true motion vector of the MB is such that pixel i in frame n is predicted from pixel j in frame $n-l$, where $l > 1$. The encoder prediction of this pixel is $\hat{f}^j_{n-l}$. The prediction error e^i_n is compressed, and the quantized residue is denoted by $\hat{e}^i_n$. The encoder reconstruction of the pixel is:

$$\hat{f}^i_n = \hat{e}^i_n + \hat{f}^j_{n-l} \qquad \text{(II.9)}$$

As the receiver does not have access to $\hat{f}^j_{n-l}$, it uses $\tilde{f}^j_{n-l}$:

$$\tilde{f}^i_n = \hat{e}^i_n + \tilde{f}^j_{n-l} \qquad \text{(II.10)}$$

When the MB is lost, the median motion vector from the three nearest MBs is calculated and used to associate pixel i in the current frame with pixel k in the previous frame. Using the same arguments as in the original ROPE algorithm, we compute the first and second moments of $\tilde{f}_n^i$ for a pixel in an inter-LT-coded MB,

$$E\{\tilde{f}_n^i\} = (1-p)\left(\hat{e}_n^i + E\{\tilde{f}_{n-l}^j\}\right) + p(1-p)E\{\tilde{f}_{n-1}^k\} + p^2E\{\tilde{f}_{n-1}^i\} \quad \text{(II.11)}$$

$$\begin{aligned} E\{(\tilde{f}_n^i)^2\} &= (1-p)\left((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-l}^j\} + E\{(\tilde{f}_{n-l}^j)^2\}\right) \\ &+ p(1-p)E\{(\tilde{f}_{n-1}^k)^2\} + p^2E\{(\tilde{f}_{n-1}^i)^2\} \end{aligned} \quad \text{(II.12)}$$

We note that error concealment is still done using the *previous* frame $n-1$ and not the long-term frame. This is done regardless of whether the three MBs above are inter-ST-coded or inter-LT-coded, or some combination of the two. The motion vectors may be uncorrelated. If the upper GOB is also lost, we conceal the MB using the co-located block from the previous frame.

Using an additional reference frame (LT) has some drawbacks with respect to motion vector compression efficiency when compared to single frame. There is a bit rate loss due to inaccurate prediction of MVs from the neighboring and potentially uncorrelated MVs. By neighboring motion vector, we mean the motion vector of the MB on the left of the one being coded. During coding, we do not predict MVs using the MVs above because we wish the GOBs to be decodable independently of each other. The first

MB of each GOB uses no prediction for the MVs. For those MBs where the MV points to the same reference frame as the neighbor (and only for those MBs), we obtain a MV coding efficiency equal to that of single frame approaches. As an alternative approach, we tried predicting the MV using the neighbor only when the neighbor corresponded to the same reference frame. When the neighbor did not use the same reference, the MV would be coded without prediction. Experimentally, this did not do as well. The explanation for this is that with relatively small values for N, MVs pointing to either the short-term or the long-term frame buffer tend to have similar values, so it is better to use them for prediction than to code MVs without prediction. But they are not as similar as are MVs in single frame motion compensation, so there is still a loss in MV compression efficiency. As will be seen in the results section, this loss in MV compression efficiency is more than made up for in other ways by the dual frame coder.

Compression efficiency will also suffer due to the need to transmit one bit for every inter-coded MB to specify the frame buffer. (This overhead could be reduced by using run length coding on the bits, but we do not do this as it incurs penalties in terms of buffering at the decoder and a risk of catastrophic error if the RLC encoded frame buffer selection stream is lost.) Nonetheless, as experimental results will show, the rate-distortion optimization models these additional bits, and is still able to yield superior compression performance.

The requirement to encode and decode this additional bit (for selecting between ST/LT), clearly makes this proposed scheme *not* a H.263+ compliant codec. Since H.264 already supports multiple frame prediction, there is no compliance problem. However, a straightforward application of ROPE on H.264 without any modifications is not wise. Apart from the half-, quarter- and eighth-pel accuracy present within H.264, which would have to be modeled (see Section II.C), there is also the problem of the loop filter, and additional concealment modes, which would require evaluating multiple product expectations (correlations).

Since the quantization parameter QP takes values from 1 to 31, the coder optimizes over 62 potential combinations of coding modes (intra or inter) and quantization parameters by calculating the estimated distortion using ROPE. With the extra coding mode inter-LT, the search for optimal coding parameters is conducted over 93 combinations. There is a computational increase of approximately 50% for the rate-distortion optimization portion of the encoder. Furthermore, motion estimation complexity is approximately doubled. Hence the total encoding time of the modified encoder is roughly 1.8 times that of the baseline ROPE encoder. Further analysis on computational complexity is provided in Section II.G.

II.C Half-Pixel Approximation Extension

The use of integer motion vectors limits the reference choices in the previous frame. Most video codecs show a performance advantage when half-pel motion vectors are implemented, as the encoder is now presented with many more options in the search for the best-match block. The use of an additional reference frame likewise presents the encoder with more options for the best match block. We wished to see how the gains from an additional frame buffer compared to those from adding a half-pel grid, and also whether the two approaches could be used together for greater benefit.

The use of a half-pel grid in a standard video codec requires the generation of the half-pel values using some kind of interpolation, and then requires a four-fold increase in the motion vector search. However, simply adding a half-pel grid within the ROPE algorithm, and attempting to run the optimal mode switching over it, incurs a far more substantial complexity penalty than this, as discussed below.

Since the accurate use of a half-pel grid is prohibitive, another approach would be to use a half-pel grid only for finding and transmitting motion vectors, but to leave it out of the ROPE distortion calculation altogether. This is what is done in [35], which we call the Unmodeled Half-Pel, and it provides some improvement over the use of strictly integer motion vectors. However, as we will now discuss, an approximate modeling of the half-pels within the ROPE algorithm provides further improvement, while avoiding the computational complexity of the fully accurate modeling of a half-pel grid in ROPE.

We assume that error concealment is still done using only the integer portion of the motion vectors, and therefore Equations II.2 and II.3 for the intra-coded MBs are unchanged. Returning to Equations II.6 and II.7 for the inter-coded MBs, we see the terms $\hat{e}_n^i$, $E\{\tilde{f}_{n-1}^k\}$, $E\{\tilde{f}_{n-1}^i\}$, $E\{(\tilde{f}_{n-1}^k)^2\}$ and $E\{(\tilde{f}_{n-1}^i)^2\}$ remain unchanged. However, the calculation of $E\{\tilde{f}_{n-1}^j\}$ and $E\{(\tilde{f}_{n-1}^j)^2\}$ has become critical. Pixel coordinate j now points to a position in an interpolated grid that covers an area four times that of the original image.

For this calculation, we differentiate among three types of pixels on the half-pel grid: pixels that coincide with actual (original) pixel positions (called integer-indexed pixels, they do not need to be interpolated), pixels that lie between two integer-indexed pixels (either horizontally or vertically), and pixels that lie diagonally between four integer-indexed pixels. We use bilinear interpolation, so the interpolated value is simply the average of the two or four neighboring integer-indexed pixels.

For the integer-indexed pixels, the recursion equations are identical to those of the baseline ROPE algorithm, and the estimation is optimal.

II.C.1 Horizontally or Vertically Interpolated Pixel

For a horizontally or vertically interpolated pixel, we assume that j on the interpolated pixel domain corresponds to a pixel that was interpolated using pixels k_1 and k_2 in the original pixel domain. We define the following abbreviations; Let $\mu_i = E\{\tilde{f}_{n-1}^{k_i}\}$ denote the estimate (mean-value) of the pixel with coordinates k_i in frame $n-1$, $\mu_{i,j} = E\{\tilde{f}_{n-1}^{k_i}\tilde{f}_{n-1}^{k_j}\}$ denote the correlation (expectation of product) between pixels k_i and k_j, and $\sigma_i = E\{(\tilde{f}_{n-1}^{k_i})^2\}$ denote the mean squared value of pixel k_i. The first moment is computationally tractable:

$$E\{\tilde{f}_{n-1}^{j}\} = \frac{1}{2}\left[1 + \mu_1 + \mu_2\right] \tag{II.13}$$

But the expression for the second moment is:

$$E\{(\tilde{f}_{n-1}^{j})^2\} = \frac{1}{4}\left[1 + \sigma_1 + \sigma_2 + 2\mu_1 + 2\mu_2 + 2\mu_{1,2}\right] \tag{II.14}$$

The last term requires calculating the correlation of matrices whose horizontal/vertical dimension equals the number of pixels in the image. This is computationally infeasible for images of typical size. The second moment can be bounded using the cosine (Cauchy-Schwartz) inequality:

$$E\{(\tilde{f}^{j}_{n-1})^2\} \quad \leq \quad \frac{1}{4}\left[1+\sigma_1+\sigma_2+2\mu_1+2\mu_2+2\sqrt{\sigma_1\sigma_2}\right] \tag{II.15}$$

and we will approximate it by setting the inequality to be an equality. This worked well, perhaps because the (image domain) pixel values are always positive, and so correlations tend to be close to the upper bound, which was also verified by our experimental results. During our simulations, we also experimented with multiplying the Cauchy-Schwartz-derived upper bound with various constants $c < 1$, such as $c = 0.50$, however, this did not always perform as well as the upper bound.

II.C.2 Diagonally Interpolated Pixel

For a diagonally interpolated pixel, we assume that j on the interpolated pixel grid is the result of interpolating pixels k_1, k_2, k_3 and k_4 in the original pixel domain. The first moment can be computed exactly as:

$$E\{\tilde{f}^{j}_{n-1}\} = \frac{1}{4}\left[2+\mu_1+\mu_2+\mu_3+\mu_4\right] \tag{II.16}$$

The accurate but intractable expression for the second moment is:

$$\begin{aligned} E\{(\tilde{f}^j_{n-1})^2\} &= \frac{1}{16}[4+\sum_{i=1}^{4}(\sigma_i+4\mu_i)+2\,(\mu_{1,2}+\mu_{1,3}+\mu_{1,4} \\ &+ \mu_{2,3}+\mu_{2,4}+\mu_{3,4})] \end{aligned} \tag{II.17}$$

Applying the same approximation as with the horizontal/vertical case, we obtain:

$$\begin{aligned} E\{(\tilde{f}^j_{n-1})^2\} &\leq \frac{1}{16}[4+\sum_{i=1}^{4}(\sigma_i+4\mu_i)+2\sqrt{\sigma_1\sigma_2}+2\sqrt{\sigma_1\sigma_3}+2\sqrt{\sigma_1\sigma_4} \\ &+ 2\sqrt{\sigma_2\sigma_3}+2\sqrt{\sigma_2\sigma_4}+2\sqrt{\sigma_3\sigma_4}] \end{aligned} \tag{II.18}$$

and again we use this upper limit to approximate the second moment. Hence we obtained the Cauchy-Schwartz Approximation.

II.C.3 Distortion Estimation

We investigated the accuracy of our distortion approximation for half-pel motion vectors. The enhanced accuracy provided when the Cauchy-Schwartz inequality is employed, is depicted in Fig. II.3. To obtain this graph we constrained the mode decisions to use distortion only due to quantization. No estimated distortion was used so as to make the encoder independent of the accuracy of either method. The encoder optimized its stream only with regard to compression efficiency, employing half-pel motion vectors and applying errors with $p = 10\%$. Concurrently, the original ROPE algorithm and the modified one with the Cauchy-Schwartz Approximation estimated the resulting distortion. Our modification enables a more accurate estimate.

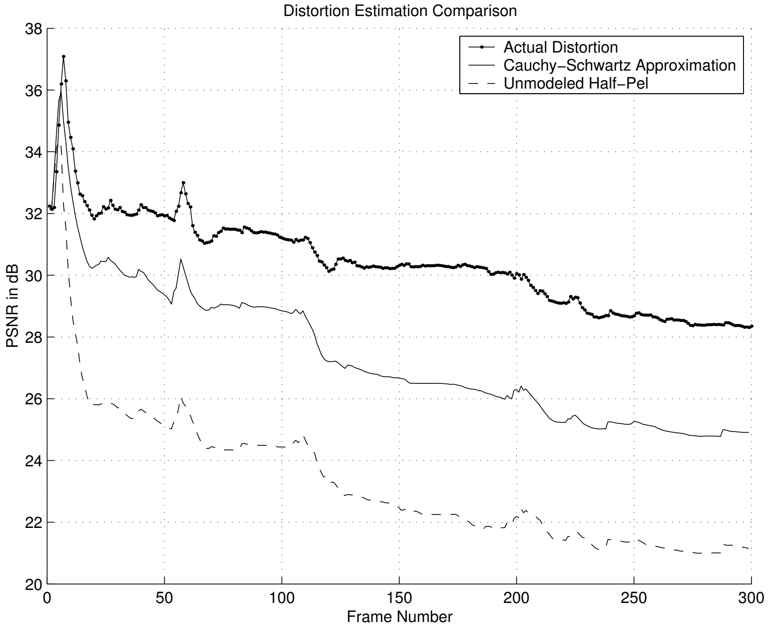


Figure II.3: Distortion Estimation Comparison.

For integer motion vectors, the distortion estimation of the classical ROPE algorithm is very accurate, within 0.1-0.2dB of the actual distortion. In Fig. II.3, where half-pel vectors are applied, such an accuracy can no longer be obtained. Nevertheless, the gain in estimation accuracy by using the Cauchy-Schwartz Approximation instead of the Unmodeled Half-Pel is quite noticeable.

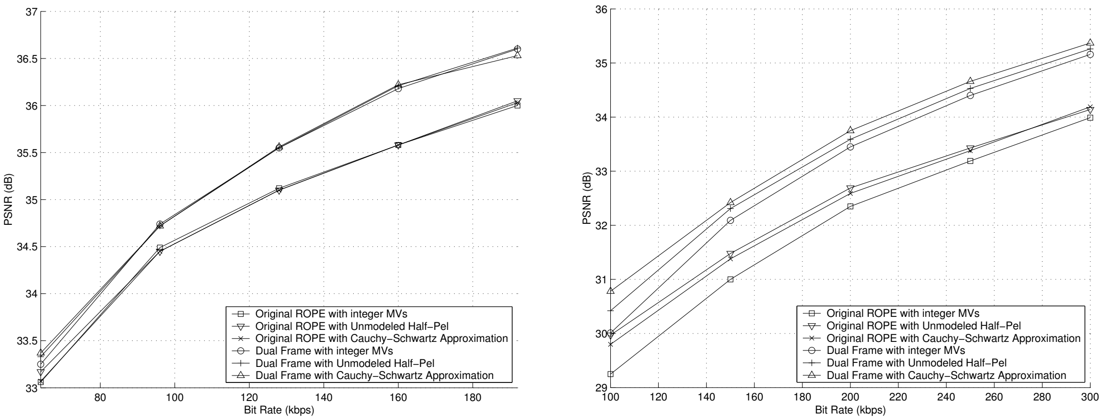
II.D Results in the Absence of Feedback

We modified an existing H.263+ video codec [43], [44] in two ways. In the case of single-frame (SF) motion compensation, we used the ROPE algorithm to estimate distortion for mode switching decisions. The resulting bitstream is fully compliant with the H.263+ standard. Secondly, we modified the H.263+ codec to make use of one additional (long-term) frame buffer. For both the single frame and dual frame cases, we measured the performance for integer and half-pel motion vectors. The half-pel results are of two types: one where the half-pel vectors are used but are not modeled in the recursive error equations, and the other where the half-pel vectors are used and are modeled using the approximations given above. We refer to these as the Unmodeled Half-Pel and Cauchy-Schwartz Approximation.

We use N to denote the jump update parameter, and D to denote the temporal distance of the long-term frame buffer in the continuous updating case. N and D were kept small to increase MV correlation, and thus improve MV coding efficiency. The GOB-packet error probability was tested with values of $p = 0.05$, 0.10, 0.15, 0.20 and 0.25. The resulting dual frame encoder is not standard compliant [44], as it must send an additional bit for every inter-coded MB to signal the use of the short-term or long-term frame buffer. The test sequences used are standard QCIF (176×144) image sequences at frame rates of 10, 15 and 30 fps. The results shown have been averaged using 100 random channel realizations (error patterns) to achieve performance consistency. The same error patterns were used for all codec variants.

II.D.1 PSNR vs. bit rate

In Fig. II.4(a) we examine the performance of the variants for “hall”. This particular sequence is rather static and does not benefit from the use of half pel MVs (the percentage of nonzero MVs per frame is less than 4%). The gains of dual frame increase with bit rate and quickly reach 0.6dB. A different situation is depicted in Fig. II.4(b) for "news", where even the lowest performing dual frame version easily provides higher PSNR than any single-frame approach does. Gains begin at 0.8dB for low rates and quickly reach 1.2dB.



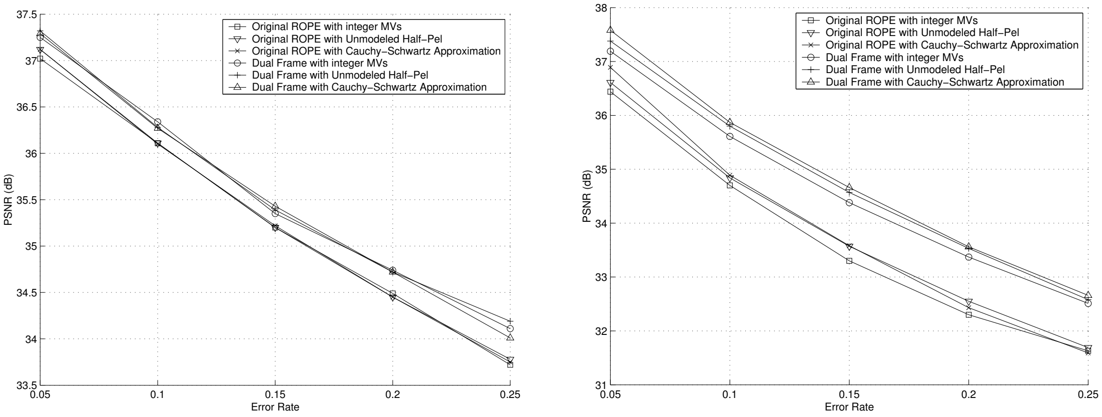
(a) (b)

Figure II.4: PSNR performance vs. bit rate. (a) "Hall" QCIF sequence at 15 fps, with continuous update parameter $D = 3$ and packet loss rate $p = 20\%$. (b) "News" QCIF at 30fps with continuous update parameter $D = 5$ and packet loss rate $p = 15\%$.

Simulations using "carphone" and "container" yielded an average improvement of 0.4 and 0.6dB, respectively.

II.D.2 PSNR vs. packet loss rate

Fig. II.5(a) depicts the performance for "hall" QCIF at 15fps. As we pointed out for Fig. II.4(a), there is no gain by using half pels. Dual frames outperform, for these particular parameters, single-frame by up to 0.5dB. The gain increases slightly with p. Similarly, in Fig. II.5(b) we can observe how packet losses affect performance for the "news" image sequence at 15fps. For both single and dual frame methods, Cauchy-Schwartz provides a slight advantage. The performance gap between single and dual frame approaches is approximately 0.8dB for $p = 0.05$ and reaches 1dB as the error rate increases.



(a) (b)

Figure II.5: PSNR performance vs. error rate. (a) “Hall” QCIF sequence at 15 fps, with continuous update parameter $D = 3$ and bit rate 96kbps. (b) “News” QCIF at 15 fps with continuous update parameter $D = 3$ and a bit rate of 200kbps.

Gains of 0.4-0.5dB were similarly obtained for “carphone” and “silent”. We also observed that errors are far more destructive in a lower frame rate case than in a higher frame rate one. When adjacent frames are more distant temporally, they are less correlated, and the respective motion vectors have generally higher and more varying values, and are thus more difficult to predict. Hence error concealment that uses estimated or all-zero MVs does much worse compared to the full frame-rate case.

II.D.3 Motion Vector Optimization

For comparison purposes we provide some experimental results where the selection of the MVs was also incorporated within the R-D Mode Decision, at an enormous computational cost. The search for optimal coding parameters is conducted over 89373 combinations (31 quantization parameter values, 3 coding modes, and $31 \times 31 = 961$ possible motion vectors) rather than just over 93. The results are, however, a good indication of the optimal attainable performance. Indicative experimental results can be seen in Fig. II.6 where only Quantization Distortion was employed, and not the one estimated by ROPE. We can comment that even for high motion sequences such as "Foreman", the gain of 0.35dB is definitely not worth the enormous increase in computational complexity. One of the reasons for the small gain is that MSE is often not a reliable measure of block similarity compared to SAD, and secondly, that motion estimation does sufficiently well at finding near-optimal MVs, so that exhaustive RD search will not yield much.

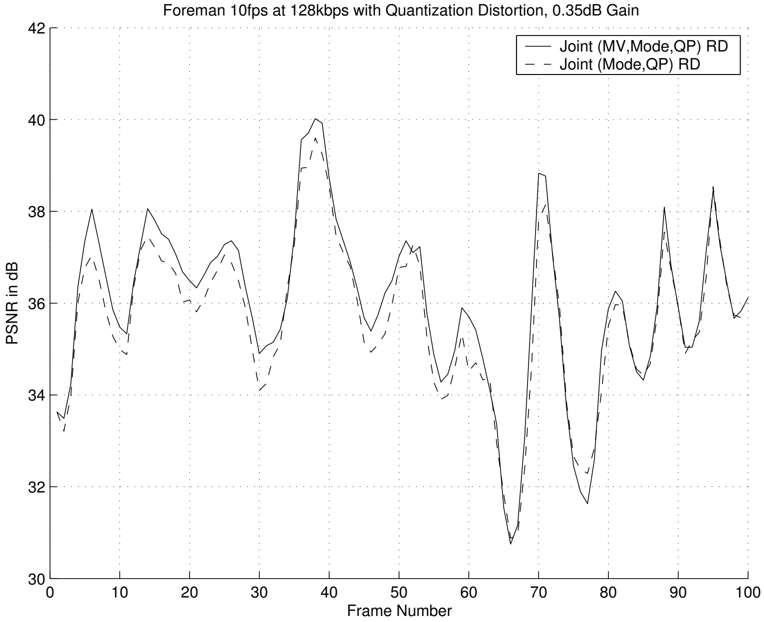


Figure II.6: Rate-Distortion optimization of MV selection.

II.E Feedback Extensions

Experimental results in [35] showed that the intelligent use of feedback information (acknowledgement of received packets) can lead to substantial improvements in performance. The ROPE algorithm estimates reconstructed pixel values that incorporate potential error propagation due to packet losses. The estimates of pixel values are made by using Equations 2, 6 and 11 for intra, inter-ST, and inter-LT coded blocks respectively. These estimates are initialized at the beginning of the video sequence by assuming that the first frame is always received unharmed. Let i be the current frame's index. Using feedback with a fixed delay d, the encoder can have perfect knowledge of the decoder's $(i - d)$-th reconstructed frame. We will use the term "re-decode" to describe the encoder's process of using the feedback information to decode a past frame so that it is identical to the decoder's version of that frame. As the encoder knows which GOBs were received intact and which ones were dropped, it can simulate the decoder's operation exactly, including error concealment. A "re-decoded" frame is one at the encoder that is identical to the decoder version, whereas we use the term "estimate" to describe a frame at the encoder for which the feedback information is not yet available, so the encoder is forced to estimate the decoder version. With feedback information, estimates of pixel values in intermediate frames are still made using Equations 2, 6 and 11 for intra, inter-ST, and inter-LT coded MBs as before, however now the information about past decoder frames required by these equations can be reinitialized using the ACKed/NACKed re-decoded frames. Then the encoder can recalculate the pixel estimates much more reliably and track potential errors for the last d frames. The actual prediction residuals or intra coefficients are fed into the ROPE estimation algorithm where the reference frames are either ROPE estimates that also were calculated recursively, or re-decoded frames. This approach was applied to a traditional single-frame reference video coder in [35] with positive results. However, it lends itself to considerable improvement through the use of a dual frame buffer.

An example is illustrated in Fig. II.7. Here, the jump update parameter and the feedback delay are respectively $N = 2$ and $d = 5$. The jump update parameter $N = 2$ means that frame 0 will be the long term reference for frames 2 and 3, frame 2 will be the long term reference for frames 4 and 5, and frame 4 will be used for frames 6 and 7. Frames that serve as long-term frame buffers for future frames are highlighted with a thicker black outline.

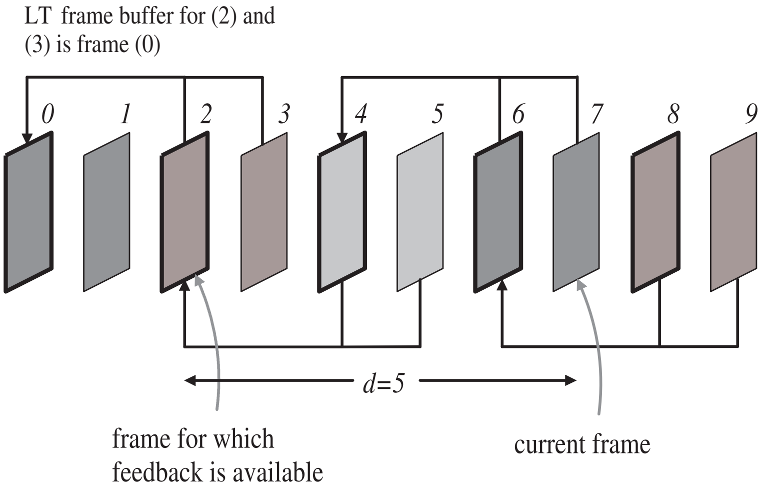

LT frame buffer for (2) and (3) is frame (0)
0
1
2
3
4
5
6
7
8
9
d=5
frame for which feedback is available
current frame

Figure II.7: Example of Approach A where $N = 2$ and $d = 5$.

Since $d = 5$, at the start of encoding frame 7, frame 2 will be re-decoded, and this newly re-decoded frame can be promptly used to update the estimates of frames 3, 4, 5, and 6. For encoding frame 7, the long-term frame is frame 4, and the short-term one is frame 6, and the new estimates of these two frames will be used by the encoder to calculate the expected distortion due to packet drops for frame 7. This jump updating, which we call Approach A, outperforms both the single-frame feedback variants, and the dual frame case without feedback, as the feedback allows us to improve the estimates of the ST and LT frames.

An alternative approach is to make the long term frame buffer move forward to contain the closest *exactly known* frame, that is the $(i - d)$ frame. The feedback here allows us to improve the estimate of the ST frame, and reduce the estimation error for the LT frame to zero. We ensure that *both* the encoder and decoder long-term frame buffers always contain an *identical* reconstruction. With a delay of d, we can use either a general (N, D) updating strategy with $D = d$ and $N > 1$ (Approach B), or a continuous updating strategy with $D = d$ and $N = 1$ (Approach C). An example of Approach B for $N = 2$ and $d = 5$ is depicted in Fig. II.8. In Fig. II.8 frame 12 is currently being encoded. Its LT frame is frame 7 which has also been re-decoded. However, re-decoding frame 7 required the re-decoded versions of frames 1 and 6, its ST and LT frames, respectively. Now we can obtain the estimates of 8, 9, 10 and 11. For frame 8 the re-decoded 7 and the re-decoded 3 will be required. For 9 we will need *estimated* 8 (ST) and re-decoded 3 (LT). For 10 we will need estimated 9 and re-decoded 5. Similarly, 11 needs estimated 10 and re-decoded 5.

By synchronizing the long-term frame buffers at the transmitter and receiver,

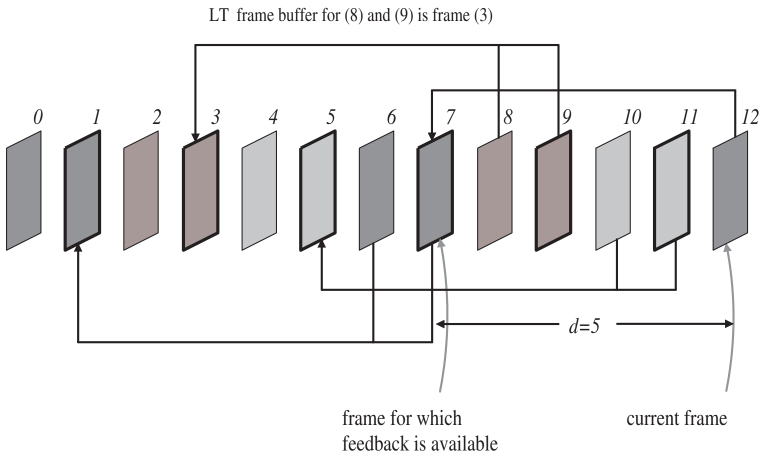

LT frame buffer for (8) and (9) is frame (3)
0
1
2
3
4
5
6
7
8
9
10
11
12
d=5
frame for which feedback is available
current frame

Figure II.8: Example of Approach B where $N = 2$ and $d = 5$.

we can totally eliminate drift errors caused by packet drop accumulation. Inter-LT encoded macroblocks, if they arrive, will be reconstructed in an identical manner at the encoder and decoder. Normally, this is only guaranteed by transmitting intra-coded macroblocks. Here, however, feedback signals enable us to use the long-term frame buffer as an additional error robustness factor without sacrificing greatly in compression efficiency.

This is the major difference from the original ROPE plus feedback case. Instead of using feedback only to improve the distortion estimate and therefore the mode selection, we now, in addition, use this information to re-decode the LT frame at the encoder and thus improve motion estimation, and use a more realistic reference frame. As we will see, the codec performs very well under a variety of conditions.

II.F Feedback Results

As before, 100 random channel realizations were run. In addition to examining performance as a function of bit rate and of packet loss rate, we now wish to study the behaviour of the codec for varying values of delay d.

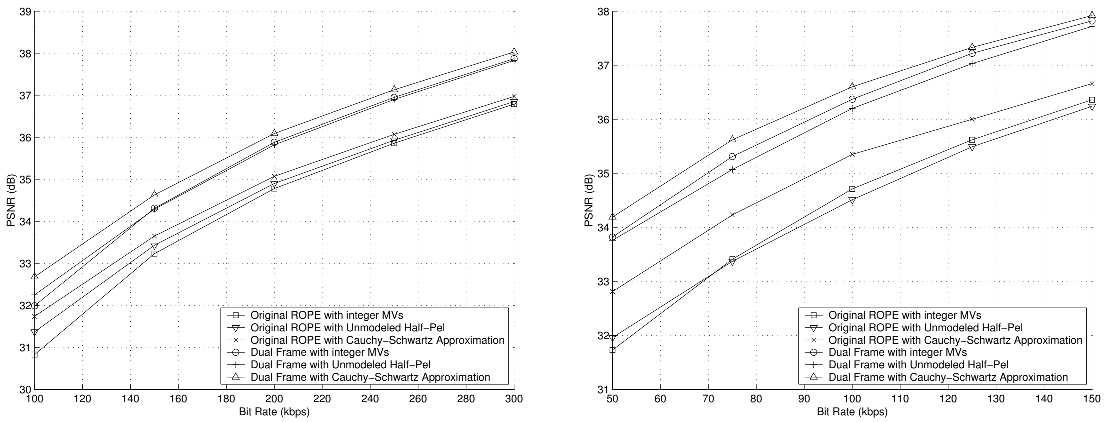
We note that continuous updating outperformed jump updating for 4 out of 6 image sequences. In particular, both Approaches B and C outperformed Approach A. However, all of them consistently outperformed single-frame ROPE with feedback.

II.F.1 PSNR vs. bit rate

Fig. II.9(a) shows results for the “News” sequence at 30 fps with delay parameter $d = 5$ and continuous updating (Approach C). For low rates in particular, performance is significantly enhanced through the use of half-pels. With the Cauchy-Schwartz Approximation, the PSNR improvement between single and dual frames grows from 0.9dB to more than 1.1dB as the bit rate increases. The dual frame approach exhibits consistent performance gains as more rate is allocated. The single frame variants, however, do not yield comparable gains.

In Fig. II.9(b), “container” QCIF at 15 fps and $d = 3$ was examined with Approach C. This sequence benefits considerably from the Cauchy-Schwartz Approximation. In the single frame case, the performance advantage is almost equal to 1dB. Simulations show a PSNR gain of almost 1.2dB in favor of the dual frame scheme.

Experimental results for the “carphone” sequence showed a 0.5dB advantage, while “silent” QCIF showed a performance difference of 0.6dB for high bit rates.



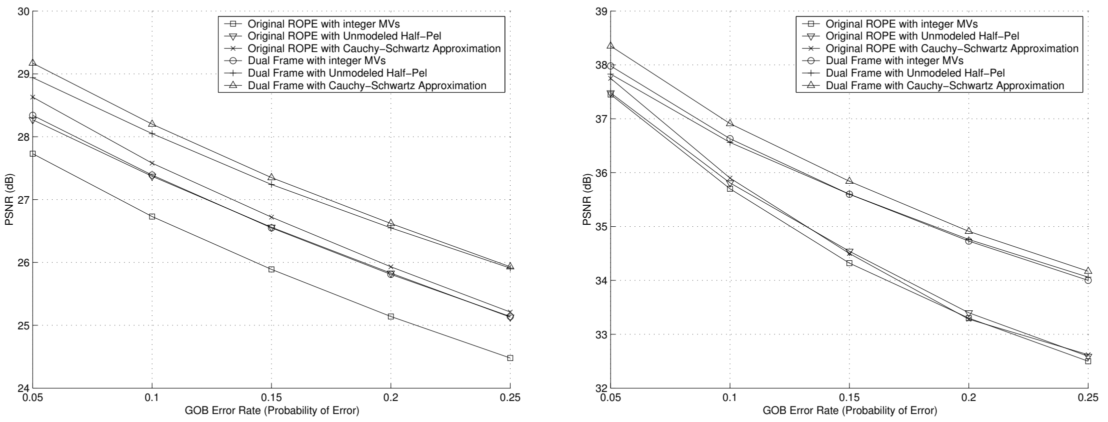
(a) (b)

Figure II.9: PSNR performance vs. bit rate. (a) "News" QCIF sequence at 30fps, with continuous updating, a feedback delay $d = 5$ and packet loss rate $p = 10\%$. (b) "Container" QCIF at 15fps with continuous updating, a feedback delay $d = 3$ and packet loss rate $p = 10\%$.

II.F.2 PSNR vs. packet loss rate

Fig. II.10(a) shows the PSNR performance for "foreman" QCIF for $N = 1$ and delay $d = 3$ for a bit rate of 100kbps (Approach C). Single frame ROPE in conjunction with the Unmodeled Half-Pel outperforms dual frame with integer motion vectors. However, when half-pel vectors are applied to dual frame, our coder outperforms the original by up to 0.7dB. Every dual frame variant outperforms the corresponding single-frame one. The difference between the Cauchy-Schwartz versions stands at 0.5dB at $p = 5\%$ and reaches more than 0.7dB at $p = 25\%$.

In Fig. II.10(b) ("news" image sequence at 30 fps, $d = 6$, 300kbps, Approach C) we see that half-pixel motion estimation provides negligible to no gain against the integer version. While the performance improvement stands at only 0.6dB at $p = 0.05$, it increases as the error rate does and reaches 1.1dB at $p = 0.10$, 1.3dB at $p = 0.15$,



(a) (b)

Figure II.10: PSNR performance vs. packet loss rate. (a) “Foreman” QCIF sequence at 30 fps, with continuous updating, a feedback delay $d = 3$ and a bit rate of 100kbps. (b) “News” QCIF at 30fps with continuous updating, a feedback delay $d = 6$ and a bit rate of 300kbps.

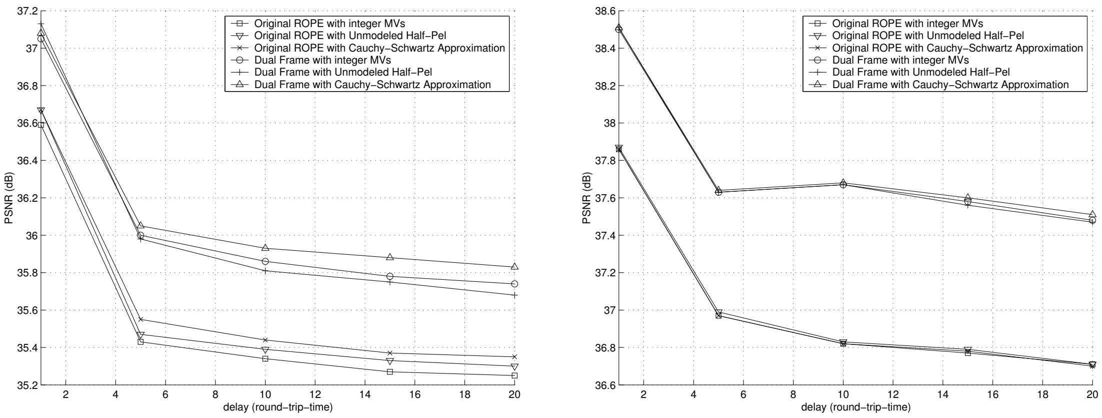
and roughly 1.6dB at $p = 0.20$ and $p = 0.25$, which means the dual frame renders the bitstream more robust to severe packet losses.

Additional simulations on the image sequences “carphone” and “silent” at 15 and 10 fps, respectively, yielded gains of approximately 0.5dB.

II.F.3 PSNR vs. delay

Fig. II.11(a) examines the behaviour of the image sequence “silent” at a frame rate of 10 fps, an error probability $p = 0.10$, a bit rate of 150kbps, and using Approach C. Performance is constant for $d > 10$. The positive effect of the Cauchy-Schwartz Approximation reaches 0.15dB for dual frames. The dual frame variants show an advantage of more than 0.5dB over single frame variants.

Performance results for “hall” sequence at 10fps, Approach C, are depicted in Fig. II.11(b). The Cauchy-Schwartz Approximation fails to provide any gain in the



(a) (b)

Figure II.11: PSNR performance vs. delay. (a) “Silent” QCIF sequence at 10 fps, with continuous updating, packet loss rate $p = 10\%$ and a bit rate of 150kbps. (b) “Hall” QCIF at 10fps, with continuous updating, packet loss rate $p = 10\%$ and a bit rate of 90kbps.

dual frame case, which is attributed to the fact that “hall” is a relatively static video sequence with limited motion. However, we see that dual frame outperforms single-frame versions by a margin that ranges from 0.65 to more than 0.8dB.

At the same time “carphone” produced gains of 0.45dB, while “news” yielded an improvement ranging from 1.4-2dB in favor of the dual frame variants.

II.F.4 Dual Frame and Half Pel Comparison

The experimental results show that in the majority of cases, the performance gains from using Cauchy-Schwartz for ROPE estimation and from using dual reference frames are approximately additive. However, the gain by only using dual frame is much more substantial than that of using only Cauchy-Schwartz. If just one of them would be implementable, then it is a question of computational and memory constraints, where the complexity analysis shows clearly that a dual frame reference is costlier.

II.G Complexity Analysis

Motion Estimation (ME) is a major bottleneck in any video coder design, and implementation of real-time video codecs, especially for wireless and power-limited devices, places a limit on the use of computational resources. However, it is not the sole one. We now analyze the computational and memory requirements of our proposed scheme. In this section, we consider multiple frame prediction with n reference frames, so we can compare with our dual frame scheme ($n = 2$).

II.G.1 Computational Requirements

No Feedback. The *encoder* performs Motion Estimation (ME), Motion Compensation (MC), ROPE Estimation, Rate-Distortion Optimization, Motion Vector Coding and Coefficient Quantization and De-Quantization. As we will see, some of those parts invoke other functions within their execution.

The ME segment entails searching for the best match 16×16 macroblock (MB) over a range of $[-15, 15]$. Hence, the *optimal* integer motion vector is obtained. Motion Vector selection is further refined by searching over a range of $[-1, 1]$ for the best half-pel refinement vector. Let us denote its computational complexity as C_{ME} for each MB. *All* complexities C presented in this analysis are *per MB*. The MC segment reconstructs a MB at a computational cost of C_{MC}. It is obvious that $C_{MC} \ll C_{ME}$.

In the ROPE estimation segment we have to differentiate among two cases: intra and inter. Intra MBs are relatively easier to estimate at a computational cost which we denote as C_{ROPE}^{intra}. For inter MBs with the Cauchy-Schwartz Approximation let us denote the complexity as C_{ROPE}^{inter}, where $C_{ROPE}^{inter} > C_{ROPE}^{intra}$. While ROPE complexity is not greater than ME, it is still quite substantial.

The computational cost for Coefficient Quantization (CQ) also includes the DCT Forward Transform which is denoted as C_{CQ}^{DCT}. The Coefficient De-Quantization (DQ) and Inverse DCT Transform (IDCT) have comparable complexity $C_{CQ}^{DCT} \simeq C_{DQ}^{IDCT}$. The CQ/DCT and DQ/IDCT complexities are negligible compared to ME or ROPE estimation, $C_{CQ}^{DCT}, C_{DQ}^{IDCT} \ll C_{ROPE}, C_{ME}$.

Rate-Distortion Optimization is the most complex since it makes use of the previous segments. Assuming n reference frames ($n = 2$ for our case), ME is run n times. Thus the cost of encoding one MB is $C_{MB} = n \times C_{ME} + C_{RD}$, where C_{RD} is the RD cost. In order to optimize for 31 possible QP parameters and $n + 1$ modes (intra and n inter), the ROPE estimation part, together with CQ/DCT and DQ/IDCT are run for $31 \times (n + 1)$ times. MC is however run only for $31 \times n$ times since intra modes do not require it. We obtain:

$$C_{RD} = (31 \times (n+1))(C_{ROPE} + C_{CQ}^{DCT} + C_{DQ}^{IDCT}) + 31 \times n \times C_{MC} \tag{II.19}$$

Thus, R-D optimization increases linearly with n, just as ME does.

Feedback. In the case of Feedback, one additional part is present, Decoder Tracking. The encoder takes advantage of feedback ACK/NACK signals to reconstruct past frames exactly the way they were reconstructed at the decoder side (re-decoding). The intermediate frames between the re-decoded one and the current cannot be re-decoded since no feedback exists for them, but we re-derive the ROPE estimates for them using the last re-decoded frame as a starting point. If d is the feedback delay, the complexity is $C_{DT} = C_{MC} + d \times C_{ROPE}$. We observe that it is invariant with respect to n.

II.G.2 Memory Requirements

We denote the number of pixels in the image as S. We assume grayscale images at 1 byte (unsigned char) per pixel. Assume again n reference frames.

No Feedback. The encoder needs to buffer $n + 1$ images (1 current, 1 ST and $n - 1$ LTs) that require S bytes each. In addition we need to buffer $n + 1$ ROPE estimates which, however, are stored as floats, thus requiring $4 \times S$ bytes each (a float is stored using 4 bytes).

Feedback. Tracking past frames requires some additional buffering. The obvious buffered frames are the last acknowledged one and *its* previous one, in unsigned char format. This happens for the single-frame case. In the dual and multiple frame case, the buffering requirements multiply.

For example, in Approach A in Fig. II.7, consider the encoding of frame 7. Frame 2 has just been re-decoded, and we wish to use this re-decoded frame to improve the estimates of frames 3, 4, 5, and 6. First of all, re-decoded frames 0 and 1 must have been buffered in order to re-decode frame 2, as they were the LT and ST frames for frame 2. After re-decoding frame 2, the encoder can purge re-decoded frame 1, since that will no longer be needed. However, re-decoded frame 0 (since it is the LT frame for frame 3) must be kept until the ACK/NACK information arrives for frame 3. Re-decoded frames 0 and 2 are used to improve the estimate of frame 3. Re-decoded frame 2 and estimated frame 3 are then used to improve the estimate of frame 4. Re-decoded frame 2 and estimated frame 4 are used to improve the estimate of frame 5. Lastly, estimated frames 4 and 5 are used to improve the estimate of frame 6. Now the encoder can encode frame 7. So, in this example, the largest number of frames being buffered at any given time is 7 (that is, frames 0, 1, 2, 3, 4, 5, and 6) in addition to the frame to be encoded (frame 7).

In general, buffering requirements for the feedback case increase linearly with n and can be a significant impediment to implementation.

Let us now consider an example of computational and memory requirements for Fig. II.7. We assume $N = 2$ and $d = 5$. As the analysis in the Memory Requirements Subsection showed, we need to buffer 8 frames (the current one, 3 re-decoded ones and 4 estimates). If we had instead used traditional single-frame encoding with ROPE estimation then only 4 frames would need to be buffered (the current one, one estimate, the current and the previous re-decoded). Thus, we obtain a 100% increase in memory complexity for these *particular* parameters. Computational complexity also increases. C_{ME} increases by 100% and C_{RD} by 50% compared to single frame. Given that C_{ME} represents roughly 65% of total complexity and C_{RD} the remaining 35%, we calculate that the increase in computational complexity when going to a dual frame scheme is 82.5%.

II.G.3 Conclusions on Complexity

Both the memory requirements and computational requirements of using ROPE within a multiple frame framework are large, growing linearly with n (the number of reference frames). However, past research [15] as well as our own simulations confirmed that the performance gains grow sub-linearly with n; there is quickly a point of diminishing returns after which increasing n produces trivial or no gains. Our simulations showed an advantage of up to 0.35dB for $n = 6$ (5 LT frames) with ROPE compared to using dual frame with ROPE, for certain sequences. However, our experiments showed that most of the gain over single-frame ROPE is obtained through dual frame ROPE. Since most of the performance gain can be captured by using only two reference frames, whereas the complexity grows linearly with n, we chose to use $n = 2$.

II.H Conclusions and Future Work

The addition of a long-term frame buffer for motion compensation improves the encoder's compression efficiency and renders the bitstream more robust to packet drops. At the same time, using only a single extra frame buffer keeps the computational complexity relatively low. An inter/intra mode switching algorithm coupled with the additional frame buffer provides a very robust and efficient bitstream. The experimental results showed that when feedback is employed, dual frame schemes consistently outperform single-frame ones, and the advantage tends to become more apparent as the bit rate or the packet loss rate grows large. In the case where feedback is not available, dual frame when used together with the Cauchy-Schwartz Approximation outperforms all other variants, and for most of the cases, the advantage is more pronounced as the packet error rate grows large.

With visual inspection of the reconstructed sequences, it is apparent that the dual frame predictor provides a noticeably smoother viewing experience. Background details are preserved, and packet losses generally affect only macroblocks with high motion, unlike in the single frame reconstruction where visual distortion encompasses the entire picture.

A pre-determined and fixed value of the jump updating parameter N is not optimal for all sequences. Future work will concentrate on finding good rules for choosing N and D for a general (N, D) updating scheme, and for choosing LT frames in an irregular updating scheme. It would be desirable to know which update parameter is best for a given sequence. Some sequences exhibit long-term statistics that could be best captured by using relatively large update parameters or by setting a constant distance frame buffer in the remote past.

II.I Acknowledgements

We would like to thank Prof. Kenneth Rose and Dr. Rui Zhang for making available their source code for the ROPE algorithm. This work was supported in part by the National Science Foundation, the Center for Wireless Communications at UCSD, and the CoRe program of the State of California.

Chapter II of this dissertation, in full, is a reprint of the material as it appears

in A. Leontaris and P. C. Cosman, “Video Compression for Lossy Packet Networks with Mode Switching and a Dual Frame Buffer,” *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 885-897, July 2004. I was the primary author and the co-author Dr. Cosman directed and supervised the research which forms the basis for Chapter II.

Chapter III

Optimal Mode Selection for a Pulsed-Quality Dual Frame Video Coder

In Chapter II we showed that rate-distortion optimal mode and reference frame selection improves error resilience. In other recent work [45], periodic high-quality frames were generated, and retained as the long-term frames in a dual frame encoding approach. In this chapter, we combine the high-quality construction and buffering of long-term frames from [45] with the optimal mode selection approach from Chapter II and show that the combination provides a significant advantage for lossy packet networks. For reasons of simplicity and reduced computational complexity, we made use of half-pel motion compensation to allow easier calculation of the distortion estimate, and the loop filter was disabled as well. Both greater motion vector accuracy and loop filtering could be enabled in our scheme, with some minor modifications and approximations (for the second moments) in the distortion estimation.

This chapter is organized as follows: Section III.A describes how distortion was estimated and optimal mode selection was performed for a dual frame encoder. Section III.B discusses the High Quality (HQ) updating approach. In Section III.C, we provide experimental results for the combination of these two approaches. The chapter is concluded with Section III.D.

III.A Optimal Mode Selection for a Dual Frame Coder

The optimal mode selection algorithm and the dual frame buffer for motion compensated prediction is identical to the one presented in Section II.B of Chapter II. The only major difference of the mode selection here, compared to Chapter II, is that we use H.264 here, and therefore need to change the approximation used for half-pixel motion vectors, to account for the 6-tap interpolation filters used in H.264. These filters are applied first horizontally and then vertically. The net result is that the first moment estimate of a diagonally interpolated half-pixel will be found as the sum of as many as 36 estimates (a 6×6 pixel support area). For the second moment calculation however, the numbers become unmanageable if the Cauchy-Schwartz-based solution of Chapter II is used without modification. We adopted the following approach to ease the computational burden: Cauchy-Schwartz approximation is used for horizontal filtering only, whereas the vertical filtering is treated as summing non-random variables. We apply the filter on the second moments as if they were constants, and obtain the second moment approximated estimate. Accurate calculation of the vertical filtering step would have required close to $\sum_{\alpha=1}^{36} \alpha = 666$ additions (36 pixel second moments and all possible pairwise correlations since all filter coefficients are nonzero), compared to just $6 + 21 = 27$ that we now do.

Given the distortion estimate, the encoder switches between intra, inter-ST or inter-LT coding on a macroblock basis, in an optimal fashion for a given bit rate and packet loss rate. The goal is to minimize the total distortion subject to a bit rate constraint. Individual macroblock contributions to this cost are additive, thus it can be minimized on a macroblock basis. Therefore, the encoding mode for each MB is chosen by minimizing:

$$\min_{(mode)} J_{MB} = \min_{(mode)} \left(D_{MB} + \lambda R_{MB}\right) \tag{III.1}$$

where $D_{MB} = \sum_{i \in MB} d_n^i$ and R_{MB} denote per MB distortion and rate, respectively, and λ is the Lagrange multiplier. The coding *mode* (*intra*, *inter-ST* and *inter-LT*) is chosen to minimize the Lagrangian cost. Because of the uneven quality levels (discussed in the next section) being assigned in the current work, contrary to what was done in Chapter II, we do not optimize over the Quantization Parameter (QP). We instead use the one chosen for that particular frame (or MB), manually or through a rate allocator.

III.B High Quality Updating

A question that arises when designing such a system is the choice of the optimal update parameter N for a given image sequence, frame rate and bit rate. It depends heavily on the sequence's characteristics, such as occlusion effects and scene changes. Thus, the problem is one of computer vision and image sequence characterization. An optimal solution will require significant computational resources. In [45] it was proposed not to attempt to *select* an optimal frame to be buffered as long-term, but rather to reformulate the problem as one of *constructing* a good frame explicitly. In this approach, every N frames, one frame is coded with additional bit rate at the expense of other regular frames. This frame is then buffered and used as the long-term reference frame for the subsequent N frames.

A second issue is how to allocate bit rate to the long-term frame. In this chapter, as in [45], the rate allocation was heuristic. We used a quantization parameter for the long-term frame that was lower by 7 compared to the quantization parameter for the regular frames. One plausible *optimal* solution to the problem would be to consider a block of N frames, and optimize the QP or rate of the long-term frame and the QP or rate of the rest over *all* N frames by applying rate-distortion optimization for all possible combinations of rates/QPs. However, the computational complexity would be immense, and there would be a delay of N frames.

The actual transmission of extra bits for the periodic high-quality frames can be accomplished in two ways. One can incur extra delay for the high-quality frames, by using extra transmission time for them, sending more information at the same average bit rate. In another realization that is not prone to delays, the sender could use extra channel bandwidth for a short period of time, to send the extra bits for the high-quality frames.

It was found in [45] that using a dual frame encoder with periodic high-quality frames (pulsed quality) that could be used as long term frames provided about a 0.6 dB advantage over a regular dual frame encoder where the long-term reference frames have the same quality as other frames.

III.C Experimental Results

The previous work with pulsed-quality dual frames considered only transmission over noiseless channels [45]. In this chapter, our goal is to use the pulsed-quality creation of long-term reference frames, but for packet erasure channels, and to use the optimal mode selection approach of Chapter II to optimally select among *intra* coding, *inter-low-quality* coding, and *inter-high-quality* coding based on estimates of the distortion that arises both from packet erasures and from the difference in low and high quality frames.

The dual frame buffer with high quality updating was implemented with the H.264/AVC reference software version JM 7.4. The encoder was modified, using functionality built into the standard. The resulting video codec produces a fully standard compliant H.264 bitstream. To limit the complexity in implementing the per-pixel estimator from [35] we employed half-pixel motion vectors. For similar reasons, the loop filter was disabled. Since we deal with standard QCIF imagery at 176×144 pixels, we took each slice (packet) to contain 11 MBs, so each variable-length packet is equivalent to a horizontal slice of MBs.

A multiple frame buffer of size two was employed. In the case of regular updating, the first frame is the previous and the second a long-term one. When high quality updating is used, the sole difference is that the frames selected to be buffered as long-term ones have been explicitly coded with a lower QP compared to all others. In our simulations we code the long-term frames with a QP that is lower by 7 compared to the general QP for the entire sequence. The long-term frame was updated every 10 frames (updating parameter $N = 10$). This fixed value was selected experimentally as in [45] and is a compromise between the optimal values for many sequences.

Packet losses corrupt the bit stream. The loss of a packet translates to losing a horizontal slice of MBs. Packets can be decoded independently of one another. Error concealment is applied by using the median of the motion vectors of the three upper MBs to conceal from the previous frame. If the upper slice has been lost as well, then we just copy the co-located MB from the previous frame. The error concealment is modeled within the distortion estimation equations.

A hundred different random patterns were used to obtain the displayed results. Since the regular dual frame coder takes no account of the possibility of packet losses in choosing between the coding modes, it underutilizes the intra mode, and performs very poorly in a lossy environment. Thus, we also investigated the performance of random intra refresh algorithms.

The QP value was selected empirically in [45] after extensive experimentation for standard video sequences, and, although it cannot be said to be optimal, it was found to be an acceptable compromise for a range of image sequences. Constraining the QP selection and comparing against intra-coded long-term frames, we found that the present scheme outperformed intra-coded long-term frames for most of the cases by up to 1dB.

Fig. III.1(a) illustrates the system's performance for the "carphone" image sequence, which consists of a man talking on his videophone in a moving car. For a packet loss rate of 10% the "zero intra" coding proves extremely error-prone. It has very few intra-coded MBs, and a $p = 10\%$ packet loss ratio drops the PSNR down to 21dB from more than 32dB. Increasing the allocated rate has no effect since the absence of intra-coded MBs totally compromises the bitstream's resilience.

Attempting to protect the bitstream by employing some intra-coded MBs, we used a random intra refresh update. We experiment by forcing 20, 33 and 45 random intra-coded MBs *per-frame*. Performance increases with the transmission rate, due to the added protection of the intra MBs, particularly as the number of intra-coded MBs increases.

However, we observe that providing high-quality to the long-term frames does less well than regular quality for these heuristic intra refresh approaches. This is likely due to the fact that the high-quality frames are depriving other frames of their share of rate, and the random intra refresh macroblocks also deprive other MBs of their share of rate; the competing effects of these heavy rate users hurt the final performance. Similar conclusions can be drawn in Fig. III.1(b).

However, with the use of rate-distortion optimal mode selection among *intra* coding, *inter-low-quality* coding, and *inter-high-quality* coding, the pulsed-quality dual frame approach outperforms regular dual frame updating by as much as 1-1.5dB throughout Figs. III.1(a)-(b). For Fig. III.1(b) in particular, the performance gap increases with the bit rate.

We then investigated the performance of the system for varying packet loss ratios. Figs. III.2(a)-(b) demonstrate that HQ updating holds a comfortable lead over regular updating for packet loss ratios ranging from 1% to 25%. The performance gain varies from 0.6-1.5dB depending on the image sequence characteristics. It is in general higher for low-motion sequences such as “mother-daughter” but still can reach 1 dB for active sequences such as “carphone”. The QPs were chosen so as to achieve the same ($\pm 5\%$) total bit rate for the graph points.

We experimented with using more than 2 reference frames, and found, as in Chapter II, that expanding the reference buffer size beyond 2 frames produces sharply diminishing returns. In particular, we tried (1) two long-term (high-quality) frames plus one short-term frame, (2) two short-term frames and one long-term (high quality) frame, and (3) one long-term and four short-term frames. In all cases, the gains over the dual frame high-quality case were quite small (on the order of 0.1-0.2 dB). It appears that, for the sequences we tried, the immediate past frame captures most of the benefit that short term references can provide (high correlation with current frame), and a single high-quality frame captures most of the benefit that the high-quality long-term past can provide.

Subjective quality evaluation showed that the periodic coding of frames at high quality is only rarely noticeable for the error-free case, and for the error-prone case it is completely masked by the error concealment and propagation.

III.D Conclusion

In conclusion, the dual-frame coder with pulses of high quality provided to the long-term frame, when used in conjunction with random intra refresh, performs *less well* than the regular dual frame coder where long-term frames are chosen from among regular quality frames. The optimal mode selection does significantly better than random intra refresh for both regular quality and pulsed-quality dual frame coders, and the pulsed-quality approach performs *much better* than the regular dual frame approach when used with the optimal mode selection. This result says that the method of creating or choosing a long-term reference frame (pulsed quality or regular quality) and the method of choosing, for each macroblock, whether or not to use that long-term reference frame (or use the short-term or intra mode) can work together synergistically or can oppose each other. The gains in performance ranged from 0.6 to 1.6dB.

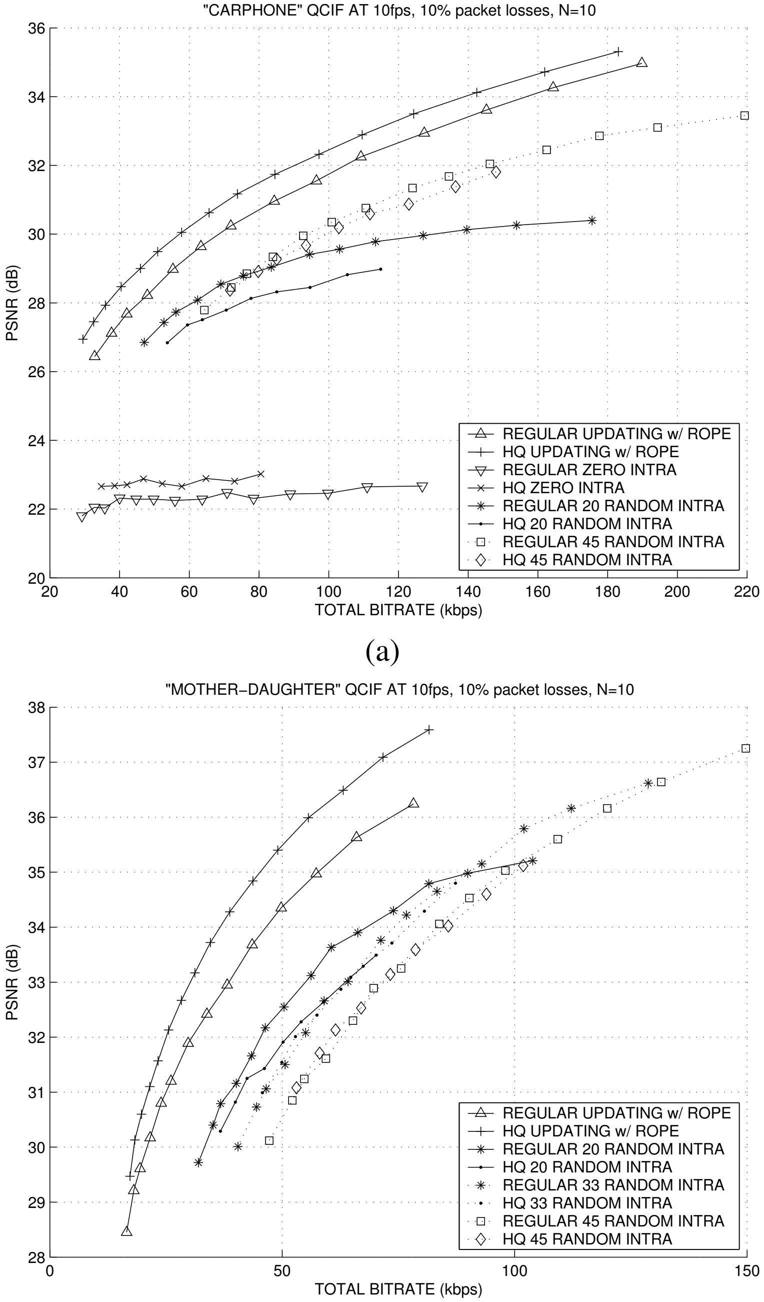
The results point to the superiority of high quality (pulsed quality) over regular updating for lossy packet network video transmission with a dual frame coder. This gain comes at trivial extra computational and implementation cost and can be easily deployed in a standard compliant H.264 codec.

In our work, the heuristic allocation we used worked remarkably well for all combinations of image sequences and bit rates, but there is still much that has not been modeled and optimized. Future work will concentrate on finding an efficient explicit rate control mechanism to allocate rate to the long-term and regular frames. Finding good update parameters is at least as challenging.

III.E Acknowledgements

This work was supported in part by the National Science Foundation, the Office of Naval Research, and the UC Discovery Grant program of the State of California.

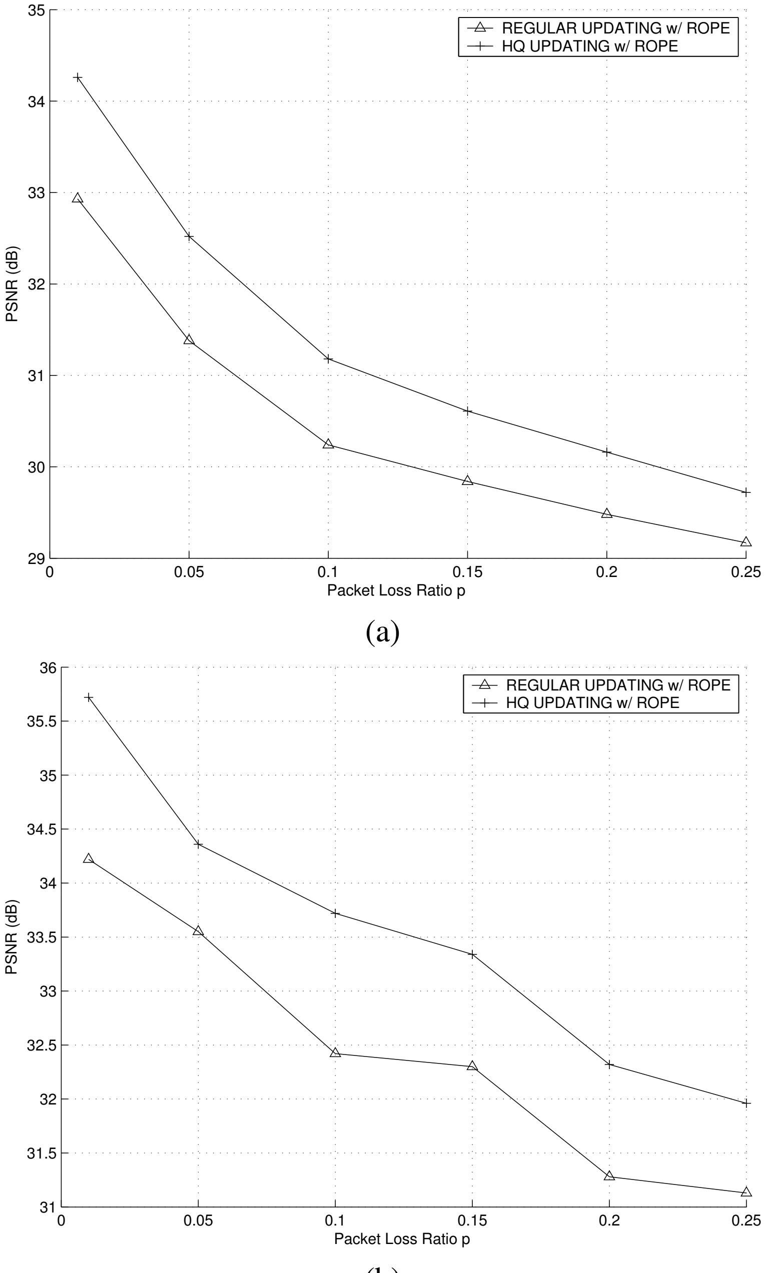
Chapter III of this dissertation, in full, is a reprint of the material as it appears in A. Leontaris, V. Chellappa, and P. C. Cosman, “Optimal Mode Selection for a Pulsed-Quality Dual Frame Video Coder,” *IEEE Signal Processing Letters*, vol. 11, no. 12, pp. 952-955, December 2004. Co-author Vijay Chellappa and I contributed equally towards this publication. Co-author Dr. Cosman directed and supervised the research which forms the basis for Chapter III.



(a)

(b)

Figure III.1: Packet loss ratio $p = 10\%$. (a) Image Sequence “carphone”. (b) Image Sequence “mother-daughter”.



(a)

(b)

Figure III.2: PSNR performance vs. packet loss rate. (a) Image Sequence “carphone” QCIF at 10fps, $N = 10$, 122.5kbps. (b) Image Sequence “mother-daughter” QCIF at 10fps, $N = 10$, 34.4kbps.

Chapter IV

Drift-Resistant SNR Scalable Video Coding

Fine Granular Scalable (FGS) video coding has emerged as an important research topic in recent years. Instead of compressing for a given target rate, it is desirable to compress for a range of bit rates at which the sequence can be potentially decoded. This is critical for internet video streaming, because there is usually no guarantee of constant bandwidth. One can extract multiple versions of the same video, at different levels of quality, from a single compressed file, and then stream them to recipients with different bit rate requirements. FGS was recently accepted for inclusion into the state-of-the-art scalable video codec jointly developed by ISO and ITU-T [46]. The first standardized effort on FGS video coding was the MPEG-4 FGS Signal-to-Noise Ratio scalability extension [47]. The base layer consists of a standard single-layer MPEG-4 bitstream while the enhancement layer (EL) is coded with the bitplane technique and references only the base layer reconstruction of the image. Bitplane coding provides a completely embedded stream that can be arbitrarily truncated to fit the available bandwidth.

In [48], Wu et al. introduced progressive fine granularity scalability (PFGS), which uses an additional EL reference frame to improve motion prediction. Assuming availability of the base layer and EL references, the frames being encoded alternate between those two layers as reference. In [49], performance was improved by selecting the reference layer on a macroblock basis, called MB-PFGS. At the same time, He et al. [50] combined H.264/AVC with MB-PFGS to produce a scalable coder that outperformed MPEG-4 FGS, using both base and EL information during motion estimation. PFGS suffers from drift due to possible loss of the previous EL. A drift estimation technique was proposed in [51]. The drift was not modeled probabilistically, hence could not be used to estimate first or higher order moments of the enhancement reference pixels. We thus propose a new drift estimate that is valid for all moments of the pixel values.

To further reduce drift and improve compression we investigate incorporating multiple frame prediction into FGS scalable video coding. The earliest attempt is found in [52] which used the previous five frames as additional references. Another approach to multiple references is found in [53]. Two frames (one is the short-term) are buffered and reference frame selection is biased in favor of the farthest frame. The non-selected short-term frames are not referenced by future frames. A separate approach with multiple references that makes use of leaky prediction to constrain drift was presented in [54], where the drift error was modeled as the worst possible.

In this chapter, we apply pulsed quality allocation to periodically updated long-term frames used for dual frame prediction as proposed in Chapter III. Uneven quality allocation is applied only to the base layer. The chapter is organized as follows. Section IV.A gives an overview of the EL coding modes, and describes our algorithm for optimal per-pixel estimation. Section IV.B discusses the implementation of the recursive estimation and Section IV.C presents the dual frame prediction scheme. Experimental results are presented and discussed in Section IV.D. The chapter concludes in Section IV.E.

IV.A Optimal Per-Pixel Estimation of Drift

Base layer macroblocks (MBs) are encoded with one of the many possible modes defined in the H.264 standard. For the EL however, every MB can be encoded with three possible coding modes (Fig. IV.1(a)) [49]. Top dark gray squares denote base layers, bottom light gray squares denote enhancement references, and white squares with dashed lines denote partially decoded (top) or higher (bottom) enhancement layers. Base layer MBs are always reconstructed exclusively from previous base layers. Black arrows denote prediction, while white arrows denote reconstruction. We note that hereon "prediction" refers to the MC prediction at the encoder side, while "reconstruction" stands for the MC prediction at the decoder side.

The first coding mode is *LPLR*, where an enhancement MB is predicted and reconstructed from the previous base layer. Using this mode, and assuming that the base layer is always received in its entirety, no prediction/reconstruction mismatch is possible and drift from previous frames is stopped. The coding efficiency is degraded due to the low quality motion compensation and reference.

The two other coding modes involve prediction from the EL reference. In *HPHR*, the enhancement MB is both predicted and reconstructed from the EL reference. This yields high compression, provided the previous enhancement reference was received in its entirety. If not, we have drift. To counter this, in *HPLR* mode, prediction still takes place from the enhancement reference, but reconstruction now uses the previous base layer. The quality is lower than HPHR, but drift is contained. At the decoder side, the modes “LPLR” and “HPLR” are identical, since in both modes the base layer reference is used for reconstruction. Thus, only one bit is needed to signal an enhancement layer mode.

Consequently, selecting HPHR provides best quality with drift, LPLR yields low quality without drift, while HPLR is a trade-off between those two. Leaky prediction [55] uses as a prediction reference a weighted superposition of the EL and BL predictions. Quality is a trade-off, and while drift exists, it attenuates to zero over time provided the EL weighting is sufficiently small. In our scheme the suppression of drift is a problem of coding mode decision.

Let n be the number of the current frame, and (i, j) the spatial coordinates of the pixel we seek to estimate. The motion vector that points to the prediction block in frame $n-1$ is denoted (v_x, v_y). Let $(i + v_x, j + v_y) = (\alpha, \beta)$. Let f_k denote the probability that the received EL portion has been truncated at rate R_k (i.e., available bandwidth at a particular moment is R_k), for $k = 0$ to $N-1$, where $R_l < R_k$ for $l < k$, and N is the number of operational rates. Let R_{ER} denote the enhancement reference rate. Even if rate $R > R_{ER}$ is available to the decoder, the enhancement reference will still be decoded at rate R_{ER}. The frame decoded at rate R will be used only for display purposes by the decoder. It is left out of the decoding loop. Disregarding the effects of the loop filter and quarter-pixel accurate motion compensation used in baseline H.264, we observe that, at the decoder, a reconstructed EL reference pixel $\tilde{p}_{er}^{n}(i,j)$ at frame n and spatial coordinates (i,j) can be written for LPLR and HPLR modes as:

$$\tilde{p}_{er}^{n}(i,j) = p_{b}^{n-1}(\alpha,\beta) + \tilde{r}^{n}(i,j) \tag{IV.1}$$

where $p_{b}^{n-1}(\alpha,\beta)$ is a motion-compensated base layer pixel of frame $n-1$, which is a deterministic value known by both encoder and decoder, since the BL is assumed to be received in full. Term $\tilde{r}^{n}(i,j)$, the reconstructed residue from the received part of the EL, can vary according to channel conditions and thus has to be modeled, by the encoder, as a random variable. This residue differs for LPLR and HPLR because of separate references, though the equations are unaffected. For HPHR we obtain:

$$\tilde{p}_{er}^{n}(i,j) = \tilde{p}_{er}^{n-1}(\alpha,\beta) + \tilde{r}^{n}(i,j) \tag{IV.2}$$

Term $\tilde{p}_{er}^{n-1}(\alpha,\beta)$ is the motion-compensated pixel in the EL reference of frame $n-1$, which has to be considered random by the encoder, since the encoder cannot know if the received portion of the EL was enough to reconstruct the enhancement reference frame in full. We seek the expected values (*first moments*) of these random variables. Due to space constraints we derive this only for HPHR:

$$E\{\tilde{p}^n_{er}(i,j)\} = E\{\tilde{p}^{n-1}_{er}(\alpha,\beta) + \tilde{r}^n(i,j)\} = E\{\tilde{p}^{n-1}_{er}(\alpha,\beta)\} + E\{\tilde{r}^n(i,j)\} \qquad \text{(IV.3)}$$

If the last term, the residual, is calculated, then our recursive estimate is complete. We use l to denote that value among the possible truncation rates where $R_{l-1} < R_{ER} \leq R_l$, and obtain:

$$E\{\tilde{r}^n(i,j)\} = \sum_{k=0}^{l-1} f_k r^n_k(i,j) + r^n_{ER}(i,j) \sum_{k=l}^{N-1} f_k \qquad \text{(IV.4)}$$

where $r^n_k(i,j)$ denotes the enhancement residue truncated at rate R_k, and $r^n_{ER}(i,j)$ the enhancement residue required to reconstruct the enhancement reference in full. For $k \geq l$, we set $r^n_k(i,j) = r^n_{ER}(i,j)$ since the truncated rate is enough to fully recover the enhancement reference. Per-pixel recursive estimation was previously shown to be effective in packet loss scenarios [35]. However, one needs the *second moment* of the random variable as well, to calculate the mean *squared* error during mode decision. From Eq. IV.2:

$$\begin{aligned} E\{(\tilde{p}^n_{er}(i,j))^2\} &= E\{\big(\tilde{p}^{n-1}_{er}(\alpha,\beta) + \tilde{r}^n(i,j)\big)^2\} && \text{(IV.5)} \\ &= E\{(\tilde{p}^{n-1}_{er}(\alpha,\beta))^2\} + E\{(\tilde{r}^n(i,j))^2\} + 2E\{\tilde{p}^{n-1}_{er}(\alpha,\beta)\tilde{r}^n(i,j)\} \end{aligned}$$

To obtain the third term we *assume* that prediction reference $\tilde{p}^{n-1}_{er}$ is *uncorrelated* with the residue $\tilde{r}^n$:

$$E\{(\tilde{p}_{er}^n(i,j))^2\} = E\{(\tilde{p}_{er}^{n-1}(\alpha,\beta))^2\} + E\{(\tilde{r}^n(i,j))^2\} + 2E\{\tilde{p}_{er}^{n-1}(\alpha,\beta)\}E\{\tilde{r}^n(i,j)\} \tag{IV.6}$$

The second moment of the residual is:

$$E\{(\tilde{r}^n(i,j))^2\} = \sum_{k=0}^{l-1} f_k(r_k^n(i,j))^2 + (r_{ER}^n(i,j))^2 \sum_{k=l}^{N-1} f_k \tag{IV.7}$$

Using Eqs. IV.3 and IV.4, we recursively estimate the first moment, and with Eqs. IV.6 and IV.7, we estimate the second moment for HPHR blocks. For LPLR and HPLR, the residual estimates Eq. IV.4 and Eq. IV.7 remain the same. For the first moment instead of Eq. IV.3 we write:

$$E\{\tilde{p}_{er}^n(i,j)\} = E\{p_b^{n-1}(\alpha,\beta) + \tilde{r}^n(i,j)\} = p_b^{n-1}(\alpha,\beta) + E\{\tilde{r}^n(i,j)\} \tag{IV.8}$$

and for the second moment instead of Eq. IV.6 we use:

$$E\{(\tilde{p}_{er}^n(i,j))^2\} = \left(p_b^{n-1}(\alpha,\beta)\right)^2 + E\{(\tilde{r}^n(i,j))^2\} + 2p_b^{n-1}(\alpha,\beta)E\{\tilde{r}^n(i,j)\} \tag{IV.9}$$

These equations are used at the encoder to estimate drift optimally. This algorithm is called DEPP (Drift Estimate Per-Pixel).

IV.B Drift Estimate Algorithm Implementation

Mode selection for the EL is accomplished as in [49]. Instead of employing the intact enhancement reference, we use our recursive per-pixel estimates. Let $h^n(i,j)$ denote a pixel in the original current frame n at position (i,j). Let $r_e^n(i,j) = h^n(i,j) - p_{er}^{n-1}(\alpha,\beta)$ denote the prediction residual from the EL reference, and $r_b^n(i,j) = h^n(i,j) - p_b^{n-1}(\alpha,\beta)$ denote the prediction residual from the base layer. Term p_{er}, without the tilde, is the *intact* EL reference, and not an estimate. We now disregard frame indices and spatial coordinates to simplify notation. The base layer codec quantizes r_b and sends the quantized $\hat{r}_b$ to the receiver. In [49], the coding mode is selected as LPLR over either HPLR or HPHR, if:

$$\|r_b - \hat{r}_b\| < \|\tilde{r}_e - \hat{r}_b\| \tag{IV.10}$$

The DCT residues encoded in the enhancement layer are $r_b - \hat{r}_b$ for the LPLR mode, and $r_e - \hat{r}_b$ for either HPHR or HPLR. We calculate $\tilde{r}_e = h - E\{\tilde{p}_{er}\}$ using our per-pixel estimates. Since our estimate $E\{\tilde{p}_{er}\}$ is going to be worse than the actual EL reference prediction p_{er}, doing this will slightly bias in favor of the LPLR mode.

If either HPLR or HPHR mode was selected for the EL block, we follow the approach in [49] and select HPHR over HPLR when the following inequality is satisfied:

$$\|h - p_{er}\| \times c < \|p_b - p_{er}\| \tag{IV.11}$$

where c is a constant that is fine-tuned empirically. Eq. IV.11 trades-off distortion (left side) for possible drift (right side). In this expression from [49], we replace p_{er} with the estimated predictions p_b^{n-1} or $E\{\tilde{p}_{er}^{n-1}\}$, depending on the EL coding mode. The encoder takes $N = 1$, so $f_0 = 1$, meaning that only one truncation rate R_0 is assumed to occur, and that rate is assumed to be insufficient for proper reconstruction of the enhancement layer reference: $R_k < R_{ER}$. We finally note that $||.||$ denotes mean squared error (MSE); hence the need to obtain the second moments of our estimates.

We recursively estimate the EL references with Eq. IV.3, IV.8, IV.4 (first moment), and Eq. IV.6, IV.9, IV.7 (second moment). During mode selection, we only use the estimated predictions p_b^{n-1} and $E\{\tilde{p}_{er}^{n-1}\}$ and do not add the partial residue. Only after the EL bitstream has been fully produced, we update the estimates using the above mentioned equations, in contrast with the ROPE packet loss estimation algorithm [35] that uses the current estimates for mode selection. Due to the scalable nature of our codec this is not feasible, since the calculation of the current estimates requires the truncation of the enhancement layer under construction, and every single enhancement mode decision we make changes the way the final layer will look. We instead employ the predictions from the previous estimated reference. More complex implementations of our approach are possible if we know additional statistics (additional and more accurate f_k values) about the channel, or if we employ approximations of the truncated residuals to update the estimates at intermediate rate points prior to mode decision.

IV.C Dual Frame Prediction and Pulsed Quality Allocation

In dual frame prediction from Chapter II, two reference frames, one short- and one long-term, are used for motion compensation. The long-term frame is periodically updated every N frames. In Chapter III, pulsed quality (allocation of additional bit rate) was proposed for the long-term frames (while keeping the rest of the frames at a lower quality), leading to improved performance in error-prone scenarios.

Here, we investigate periodic long term frames, both with even and with uneven (pulsed) quality. Pulsed rate allocation takes place only at the base layer level. However, since we desire roughly equal-length base layers, we incur some extra delay for the pulsed frames, as shown in Fig. IV.1(b), where a delay of 1 frame is observed. The bitstream is displayed first on top as it is encoded and on the bottom as it is transmitted. Flattening the bandwidth and transmitting at constant rate ensures a constant-length base layer. The decoder receives this group of frames, extracts the overlayed rate belonging to the long-term frame, and then decodes them. Ensuring a constant and low average-rate base layer guarantees that it will not surpass the lowest rate threshold (imposed by the bottleneck channel; e.g: 64kbps if the operational range includes ISDN). Otherwise, the rate pulses could surpass this threshold.

The encoder selects the reference frame and block through an exhaustive search whose goal is to minimize prediction distortion. We minimize the following prediction distortion measure from [50]:

$$SAD = SAD_b + \lambda_1 SAD_{er} + \lambda_2 \|p_b - p_{er}\| \tag{IV.12}$$

where SAD_b is the prediction distortion from the base layer and SAD_{er} is the prediction distortion from the EL reference. The last term is identical to the one in Eq. IV.11 with the sole difference that $\|.\|$ denotes here SAD calculation. The λ's are constants with values $\lambda_1 = 1.2$ and $\lambda_2 = 0.05$. Eq. IV.12 is used both for block motion estimation as well as for reference frame selection. The rate-distortion constrained scheme of the H.264 test model was not used. Minimizing just SAD_b would lead to sub-optimal reference frame selection because we are not necessarily going to use LPLR mode for all macroblocks in the frame. The motion vectors (MVs), reference indices, and motion partitioning are encoded in the base layer and are re-used when coding the EL. The EL encodes the FGS residuals and the block coding mode.

IV.D Experimental Results and Discussion

We employed the H.26L-PFGS video codec, comprised of an H.264 TML9 base layer codec and an EL codec with MPEG-4 FGS syntax. A uniform Quantization Parameter (QP) value was applied to all blocks of the base layer: $QP = 25$ for "carphone" and "foreman", $QP = 27$ for "container" and "mother-daughter". We measured the performance of the scalable codec by truncating the enhancement bit rate of each frame in 250 byte intervals (chunks). For sequences encoded at a frame rate of 10fps, this translates to bitrate intervals of 20kbps, while for sequences encoded at 30fps this translates to 60kbps. The bit rate horizontal axis in Fig. IV.2 and IV.4(b) corresponds to the total transmission bit rate, comprised of the base layer that naturally varies, but has been encoded so that it provides an acceptable visual quality (usually a PSNR value close to 30-31dB), and the additional EL bit rate that comes in 250-byte chunks. The leftmost point in the curves of Fig. IV.2 corresponds to the base layer plus one 250-byte chunk.

Integer motion vectors are used for motion estimation and compensation. The loop filter is used but not modeled in our per-pixel estimates due to the high complexity. The use of integer MVs enabled optimal calculation of the estimates. Regarding efficient techniques for adapting per-pixel estimates to fractional pixel motion vectors, see [56] and Chapter II. We set $f_0 = 1$ for $R_0 = 0.65 \times R_{ER}$, meaning that regardless of how many 20kbps/60kbps chunks of enhancement layer bits are received at the decoder side, the encoder runs its recursions by always assuming that network conditions force the enhancement layer to be truncated at some 65% of the rate needed for full reconstruction of the enhancement reference. The encoder is thus made to *assume* that there is drift on every enhancement reference, whether or not there actually is. Values greater than 0.65 would lower performance for low rates and raise it for higher rates.

All proposed schemes employ IPPP structure in both base and enhancement layer. The entropy coder was CABAC. We investigate both the scheme in [50] referred to as REGULAR, and our proposed scheme DEPP. The only difference between them is the modeling of drift. For each of the two schemes, three codec configurations are evaluated. The SF codecs employ single-frame prediction using the previous frame as the reference. The LT codecs employ periodic updating of an additional long-term frame buffer, hence two reference frames are available during motion compensation. We recall that the reference frame is fixed for both layers and the decision is made at the *base* *layer* encoding step. Hence a block in the enhancement layer will be predicted from the same (enhancement) frame as the base layer block was predicted from. No additional reference frame index is transmitted in the enhancement bitstream. Finally, the HQ codecs family employs pulsed quality on the long-term frame. The long-term frame is encoded with a finer quantization parameter QP_L than the rest of the frames which are instead coded with a coarser quantization parameter QP_S to ensure the same average bit rate as with SF and LT codecs. In our simulations the updating period has been fixed to 5. The following QP combinations were used for each of the evaluated sequences: $(QP_L, QP_S) = (23, 26)$ for "carphone" and "foreman", $(QP_L, QP_S) = (23, 29)$ for "container" and "mother-daughter". After searching over a range to determine a good value of the factor c in Eq. IV.11, $c_{REGULAR}$ was fixed to 13 for static sequences (detected through motion vectors) and 4 for dynamic sequences. While optimizing the parameter individually for each sequence is not realistic, we consider that it is realistic that the encoder would be able to make this simple binary categorization to choose one of two values of the parameter. Then $c_{DEPP} = 0.5 \times c_{REGULAR}$ was used. The same value was used for SF, LT and HQ versions of the codec.

Fig. IV.2 shows results for uniform truncation rate: all enhancement layers are truncated at the same bit length. In Fig. IV.2(a), all three curve families (SF, LT, and HQ), and SF in particular, show gains of 1dB for DEPP at low to medium bitrates, compared to their respective REGULAR curves. The performance loss at high rates is negligible. A similar case is observed in Fig. IV.2(b) where this time the gains at low rates are smaller. REGULAR HQ and LT perform well at high rates hinting at the usefulness of multiple frame references for this sequence. DEPP again underperforms for high rates. Recall that c was optimized for SF codecs so our claims for LT and HQ are conservative and not representative of the maximum achievable performance. For reference we show the performance of the non-scalable SF codec ("FIXED RATE") with integer motion vectors. It is apparent that the generic FGS methodology achieves SNR scalability at a significant cost in compression efficiency.

In Fig. IV.2 we observe a "knee" in the curves where the slope changes significantly. This point corresponds to the EL reference truncation rate. It does not depend on the expected rate used by the drift estimation, which is why the knee occurs in both the REGULAR and DEPP curves. The reason for the knee is as follows: Up to the EL reference rate, having more rate for the EL helps improve both the prediction reference and the final display. If however, the rate received is greater than the EL reference, the decoder will still only use the prescribed reference. So the extra rate is used only for final display purposes, but does not help with any prediction, which is why the slope is lower for that portion of the curves.

The Scalable Video Codec JSVM 2.0 [46] that incorporates FGS is also evaluated with IBPBPB structure (low delay) and integer motion vectors (performance suffers 1-1.5dB compared to quarter-pixel vectors). It outperforms the older H.26L-PFGS codec as was expected, due to more advanced entropy coding and motion prediction. Last, we investigate performance when quarter-pel motion vectors are enabled while DEPP still models vectors as integers. The "DEPP QR" is now handicapped due to inaccurate modeling of the motion compensation process and this shows in Fig. IV.2(a).

For Fig. IV.2(b) however, “DEPP QR” performs well compared to “REGULAR QR”.

Next, in Fig. IV.3 we investigate performance at arbitrary truncated rates on a per frame basis. Due to space constraints we omit the LT codecs from this comparison. From both figures we observe that DEPP is always better than REGULAR, which was expected since the truncation rate was low to medium. However, we also observe the substantial gain through the use of pulsed quality (HQ). For sequences with repetitive image content such as “mother” and “container” we observe gains of 1-1.5dB. Note that pulsing the quality does not create artificially high variations in PSNR: similar PSNR spikes are found in the SF variants as well. Finally, we observe for the REGULAR codecs that their performance deteriorates with time steadily, in contrast to the DEPP codecs that are inherently resistant to drift. The PSNR values inside the legend boxes are the averaged values over the entire sequence.

Finally, in Fig. IV.4(b) we investigate variable bandwidth scenarios. The left, center, and right points in the curves in Fig. IV.4(b) correspond to the bit rate truncation patterns 1, 2, and 3, in Fig. IV.4(a), respectively. The EL reference truncation rate is depicted with a straight line. The shape of the time-varying truncation rate patterns was chosen to resemble TCP/IP behavior. Fig. IV.4(b) shows that DEPP performs well, though the margin against REGULAR is not as high as previously. DEPP LT is not noticeably better than DEPP SF. The reason is the low quality long-term reference base layer, whose SAD contributes to reference frame and MV selection in Eq. IV.12. Furthermore, the low-quality BL makes the evaluation of fractional pixel displacements [57] - a primary reason for the compression efficiency of multiple frame prediction -

hard. Once it is pulsed, we observe impressive gains in the HQ codecs. Last, the "QR" curves use quarter-pixel MVs while the recursive estimates model them as integer only. Thus, DEPP outperforms REGULAR, though with a smaller margin.

The additional computations consist of two parts: FGS decoding (inverse DCT and inverse quantization) that yields the intermediate decoded residual, and the recursive updating step for each of the moments once the EL bitstream has been fully produced. The complexity of FGS decoding is very close to that of FGS encoding since the operations are simply reversed. The complexity of the updating step is essentially equal to the complexity of the algorithm in [35], which is comparable to applying DCT and encoding. As we track two moments, the updating complexity is estimated to be twice the decoding complexity. The overall complexity of our scheme is thus approximately three times the decoding complexity. We found that execution time is increased by just 3% when DEPP is employed.

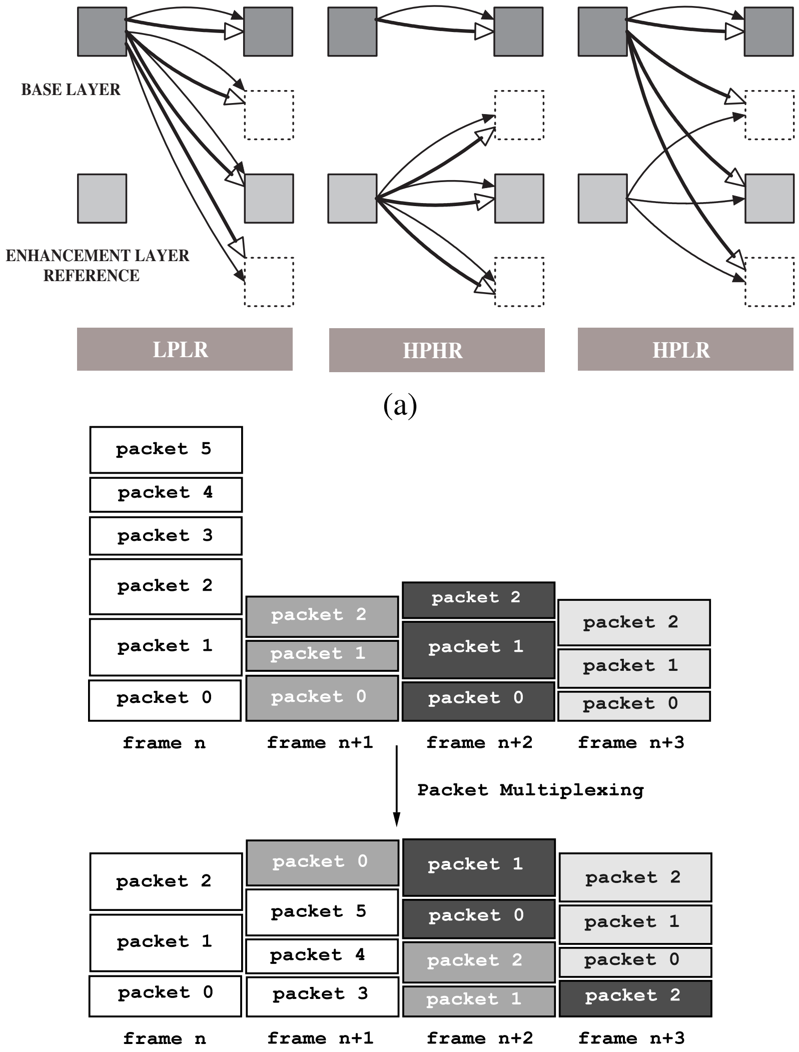
IV.E Conclusion

The proposed drift estimation approach yielded performance gains of about 1dB for most sequences across low to medium rates, with negligible loss at high rates. This was true even though the encoder persisted with a simplistic assumption about the truncation rates, an assumption that did not hold true in the actual simulations, for which the enhancement reference truncation rates varied substantially. The reason is that even for a crude channel description, it is better to assume some amount of drift and estimate its effect rather than disregarding it altogether. Pulsed-quality long-term frame prediction was shown to be advantageous for low-to-medium rates and video content with sufficient temporal redundancies.

Future work can include modeling drift in the evolving SVC standard [46]. FGS is used in an LPLR coding approach that encodes base layer motion-compensated residuals to achieve SNR scalability. Prediction from EL frames, similarly to HPLR and HPHR coding modes, can be used to improve the compression efficiency of the FGS layer, introducing potential drift.

IV.F Acknowledgements

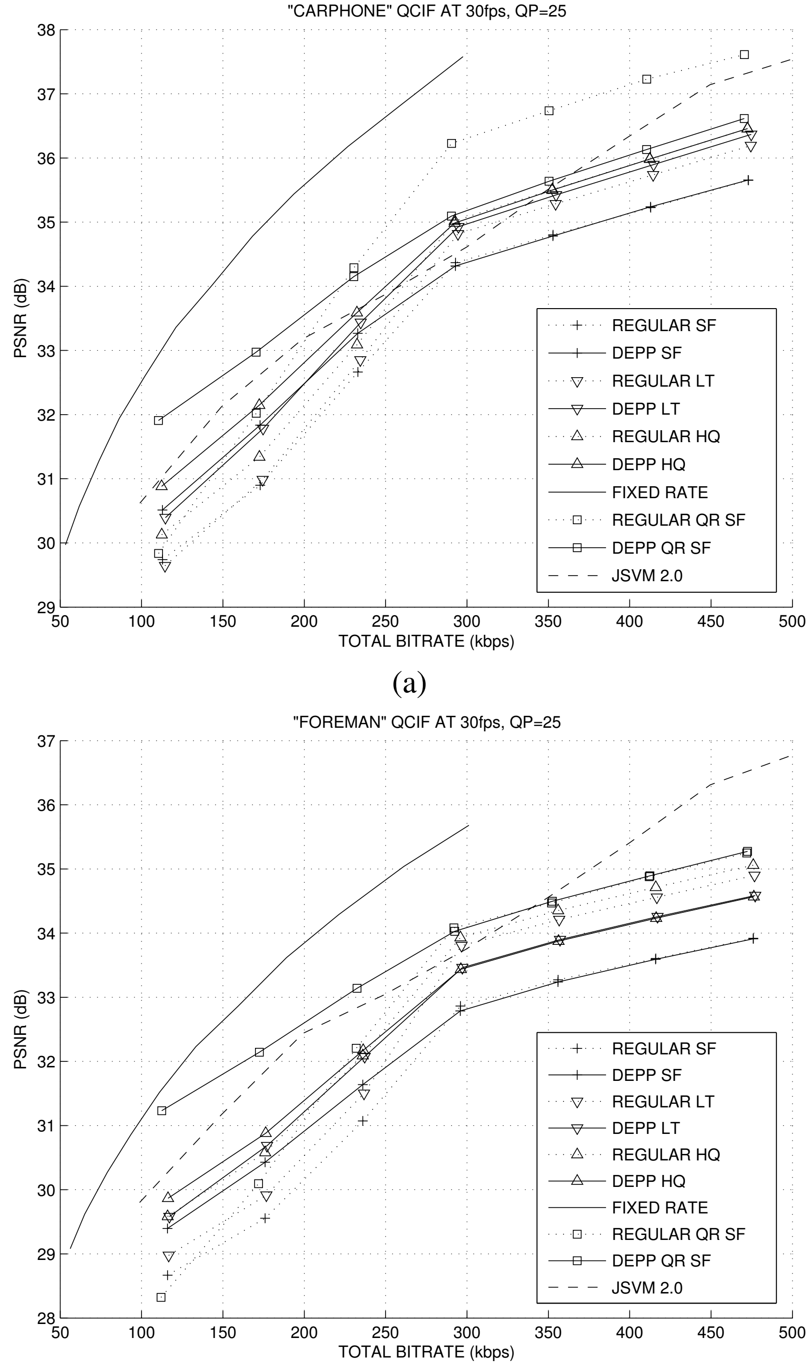
Chapter IV of this dissertation, in full, is a reprint of the material as it appears in A. Leontaris and P. C. Cosman, “Drift-Resistant SNR Scalable Video Coding,” accepted for publication in *IEEE Transactions on Image Processing*, October 2005. I was the primary author and the co-author Dr. Cosman directed and supervised the research which forms the basis for Chapter IV.



(a)

(b)

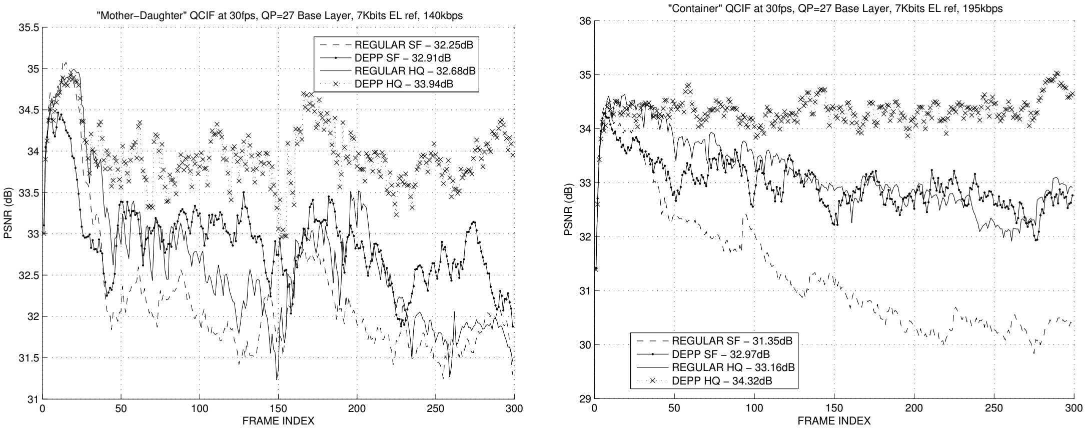
Figure IV.1: (a) Enhancement layer coding modes. (b) Bitstream generation and transmission with delay in pulsed quality framework.



(a)

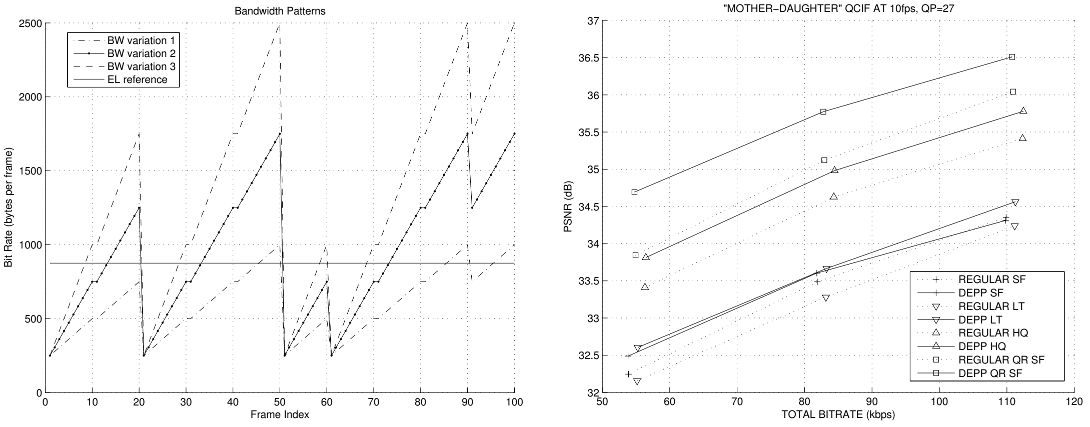
(b)

Figure IV.2: Constant bit rate (CBR) truncation experimental PSNR performance vs. total received bit rate for (a) “Carphone” at 30fps. (b) “Foreman” at 30fps.



(a) (b)

Figure IV.3: Constant bit rate (CBR) truncation experimental PSNR performance vs. frame number for (a) “Mother-Daughter” at 30fps. (b) “Container” at 30fps.



(a) (b)

Figure IV.4: Variable bit rate (VBR) truncation experiment. (a) Time-varying bit rate truncation pattern. (b) PSNR performance vs. total bit rate received for “Mother-daughter” at 10fps.

Chapter V

Compression Efficiency and Delay Trade-Offs for Hierarchical B-Pictures and Pulsed-Quality Frames

Constraining delay is critical for real-time communication and live event broadcast. Live television broadcast should have a delay of no more than one second in many cases. Interactive video-phone communication should have a maximum end-to-end delay of no more than 300ms. For traditional predictive coding techniques (the IPPP coding structure), the end-to-end delay is low. A frame is captured, encoded in real-time, briefly buffered, and then transmitted. After brief buffering, the decoder decodes the bits and displays.

Throughout this chapter, the term *rate allocation* refers to the bit distribution on a frame basis. The allocation of this bit budget to individual blocks within a frame, by varying the quantization parameter (QP), is denoted *rate control*. We also use the terms frame and picture interchangeably. Assuming that content is largely stationary and we operate under very tight delay constraints, then rate allocation is straightforward: every frame receives the same number of bits so that a total bit rate constraint is satisfied. Compression efficiency can be improved either by increasing the buffering delay (bit rate allocated to each frame can vary) or when more flexible motion-compensated prediction (MCP) structures are used. These include prediction structures that use additional reference frames, as well as structures that use frames from both the past and the future.

By filtering across frames or by using bidirectional prediction, compression performance improves because the temporal correlation among several neighboring frames is better exploited, but additional delay is incurred. An example is motion- compensated temporal filtering (MCTF). Trade-offs of delay and compression in MCTF video codecs were investigated in [58]. In that work, delay was reduced by selectively removing the update step. Recently, the update step was removed from the working draft of the Scalable Video Coding extension to H.264/AVC [46]. The end-to-end delay trade-off for MCTF was studied in [59]. Delay is an issue for hierarchical bi-predictive structures as well. The delay in the hierarchical case depends on the size of the Group of Pictures (GOP), and cannot be reduced by removing update steps while keeping GOP size intact.

One can also have increased delay when using a single-direction (forward) prediction scheme. The codec proposed in Chapter III employs two reference frames, one short-term (ST) and one long-term (LT). The LT frame is afforded extra bits; it is high (pulsed) quality. At a given constant transmission bit rate, these frames will take longer to transmit, introducing delay. The rest of the frames are starved to achieve the rate constraint. Compression efficiency was improved for certain image sequences, but delay was not studied in that work.

The studies in [58, 59] did not take into account the effect of the encoder output and the decoder input buffering requirements which are non-trivial. Here, we model both delays. In this chapter, we study the delay for LT pictures with pulsed quality, as well as for hierarchical B-frames for varying GOP size. MCTF structures are not evaluated as they were found to be in most practical cases inferior to hierarchical B-frames [60]. MCTF, being inherently open-loop, cannot outperform the latter closed-loop scheme. Rate-control is used in all codecs to ensure that the delay budget is enforced: no buffer overflows occur.

Even when perfect rate control is possible, i.e. each frame receives exactly its pre-allocated number of bits, extra buffering delay at the encoder output and decoder input is incurred when the bit rate is distributed unevenly among the frames. We investigate the effect of uneven bit rate distribution on delay. Bit rate allocation hence affects both compression efficiency as well as buffering delay. We are interested in trading off compression efficiency for less delay. This is possible if one has access to a sufficiently accurate rate-distortion model. Given constraints on bit rate and buffering delay, such a model can yield an efficient rate allocation within a GOP.

To obtain a model for hierarchical prediction we need to account for the temporal prediction distance. In [61] a rate-distortion model was presented that modeled the two-dimensional video signal as a wide-sense stationary process. The rate and distortion were calculated as functions of the power spectral density of the prediction error. This model introduced the concept of motion-compensation accuracy, which was investigated in depth in [62]. Although in [61], motion compensation accuracy represented the level of fractional-pel accurate MCP, in this chapter, we intend to use this accuracy to also model the temporal prediction distance. Intuitively, the farther we attempt to predict from, the less accurate the predictor becomes. This is manifested in increased displacement error.

The chapter is organized as follows: In Section V.A we define the end-to-end delay and describe several motion-compensated prediction structures. We calculate some useful bounds for delay constraints in the case of uneven bit rate allocation in Section V.B. The motivation and the fundamental challenges of deriving a rate allocation scheme for hierarchical B-pictures, as well as the resulting rate allocation scheme are presented in Section V.C. In Section V.D we study the structural delay trade-off. The adopted rate control schemes for each codec are discussed in Section V.E. Experimental results and conclusions follow in Section V.F. Finally, the chapter is concluded in Section V.G.

V.A End-to-End Delay

End-to-end delay involves delay at the source encoder, channel encoder, channel decoder, and source decoder, as well as transmission and propagation delay. We assume a propagation delay of zero, and we assume a lossless channel, so we do not include any channel coding. We further ignore the actual computation time at the encoder and decoder, limiting our scope to the buffers at the source encoder, shown in Fig. V.1, the transmission delay, and the buffers at the source decoder.

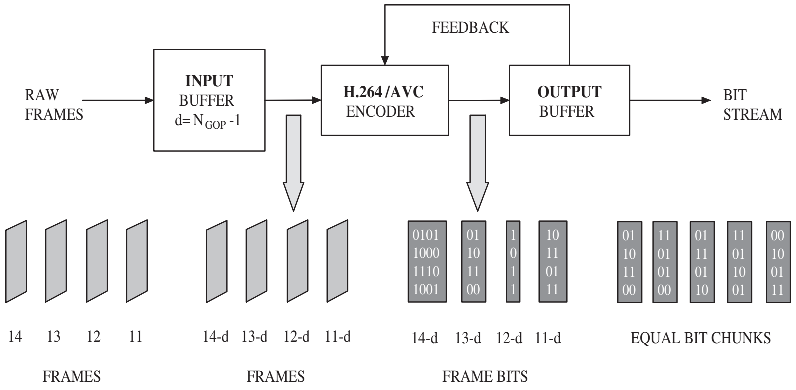


Figure V.1: The encoder input buffer and output buffer introduce delay.

The first delay is at the encoder input buffer, and this delay depends on the motion-compensation structure used, and varies in increments of whole frame durations. The encoder converts the frame into a bit stream instantaneously and then starts writing the bits to the encoder output buffer at a constant rate. If frame i is encoded with b_i bits, then it is written into the buffer at a rate of $30b_i$ bits/sec, as the video is input at a rate of 30 frames per second, so the bits for each frame get written during 1/30th of a second. The rate $30b_i$ may be more or less than the average source coding rate r. The encoder output buffer is a “leaky bucket”: it is continuously drained at the constant average source coding bit rate r.

The encoder output buffer determines how tightly the rate allocation and rate control must operate. With a constant source coding rate of r bits per second, each frame could have the same exact $\frac{r}{30}$ bits per frame, and then the output buffer could be in fact of zero length. Bits leave the buffer as soon as they enter it. Still, even in that case, 33ms are required for a frame to leave the encoder and arrive at the decoder in its entirety (for 30 frames per second one frame is displayed for 33ms). This is termed the transmission delay D_{TX}. But with an output buffer of zero length, the encoder could not respond to a scene cut or to high motion by using more bits, and by giving fewer bits to static scenes. Allowing the encoder output buffer to be larger leads to higher video quality.

We will require our rate control to live within the buffer without any frame skipping and without producing overflows or underflows. During the time (33ms) that the bits for frame i are fed into the output buffer, $\frac{r}{30}$ bits will drain out of it. Therefore the rate control must ensure that the length in bits of any single frame is no longer than the buffer size plus $\frac{r}{30}$ bits, and indeed, is no longer than the space remaining in the buffer at that particular time plus $\frac{r}{30}$ bits.

The decoder is a mirror image of Fig. V.1. Bits are buffered in a decoder input buffer, which is the same size as the encoder output buffer, a common assumption made in [63]. Decoded frames are buffered at the output prior to display. In our model which excludes delays from computation, the source coding end-to-end delay D_{e2e} depends on the four buffers and can be written as:

$$D_{e2e} = D_{enc}^{in} + D_{enc}^{out} + D_{TX} + D_{dec}^{in} + D_{dec}^{out} \tag{V.1}$$

where subscripts indicate encoder or decoder, and superscripts indicate input or output buffers.

We investigate three types of encoders: predictive IPPP coding (IPPP), long-term prediction with pulsed quality (PULSE), and hierarchical B-pictures (HIER). The codecs are now described in detail.

The *IPPP* codec, shown in Fig. V.2, is based on the Joint Model (JM) 10.1 reference software of the H.264/AVC video coding standard [64]. Frames are encoded predictively in an I-P-P-P structure.

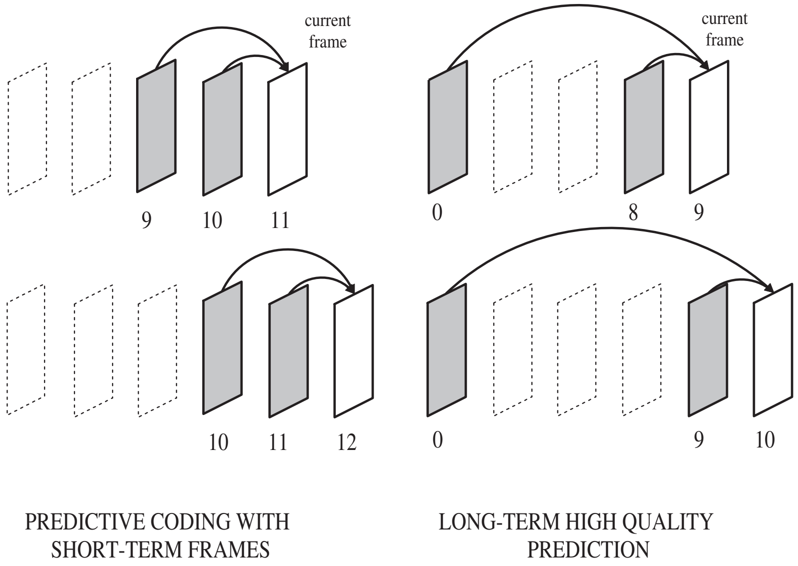

current frame
9
10
11
10
11
12
current frame
0
8
9
0
9
10
PREDICTIVE CODING WITH SHORT-TERM FRAMES
LONG-TERM HIGH QUALITY PREDICTION

Figure V.2: The PULSE and the IPPP motion-compensated structures. The arrows denote motion-compensated prediction.

The *PULSE* codec, shown in Fig. V.2, uses a short-term (ST) reference frame and a long-term (LT) reference frame for motion-compensated prediction as described in Chapter III. It is based on a modified version of the JM 10.1 reference software. The LT reference frame is periodically updated every U frames and is afforded more bits than the regular frames. Let N_{GOP} denote the number of frames in a GOP. Both for IPPP as well as for PULSE we have $N_{GOP} = 1$.

The *HIER* coder uses hierarchical motion-compensated prediction. These prediction structures, called hierarchical B-pictures, are composed of more than one temporal resolution level (a hierarchy). The simplest case is the well known IBPBP prediction structure. Examples for $N_{GOP} = 2$ (IBPBP) and $N_{GOP} = 4$ are shown in Fig. V.3. Fig. V.4 illustrates a $N_{GOP} = 8$ structure. HIER coders have by definition $N_{GOP} > 1$. The number of hierarchical temporal levels is given by $\log_2 N_{GOP} + 1$. This coder was implemented with the JSVM 3.3.1 reference software [46]. Hierarchical structures benefit from prediction both from the "future" and the "past". This is particularly advantageous in cases of global motion and camera pan as shown in [65]. Note that the "closed loop" approach [60] was used for hierarchical prediction: B-frames are predicted from the reconstructed reference frames and not the original ones as traditionally done in MCTF.

With hierarchical B-frames, the encoder cannot begin to encode a frame until the entire GOP is available for processing, so for GOP size equal to N_{GOP} the encoder begins processing one frame while $N_{GOP} - 1$ frames are in the encoder input buffer. Thus $D^{in}_{enc} = (N_{GOP} - 1)t_{fr}$, where t_{fr} is the display time duration of a frame. In all our experiments we encode at 30 frames per second, so $t_{fr} = 33$ms. The transmission time is $D_{TX} = t_{fr}$. We assume $D^{out}_{enc} = D^{in}_{dec}$. Finally, the output/display decoder delay

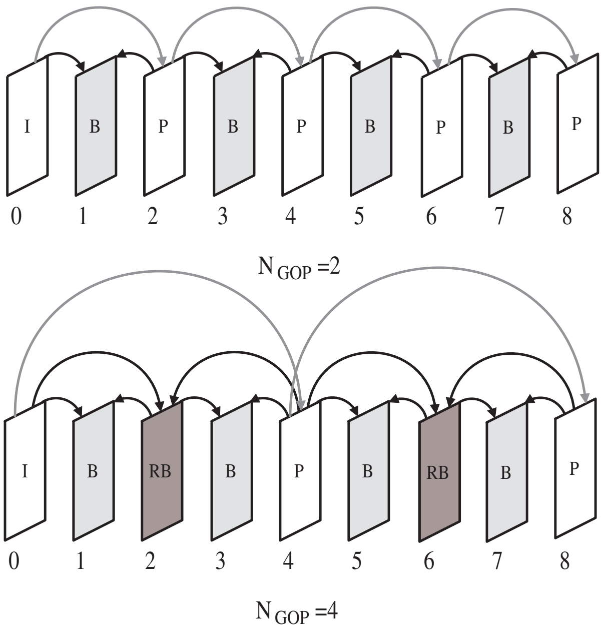


Figure V.3: Hierarchical bi-predictive motion-compensated structures. The arrows denote motion-compensated prediction. The RB frames are B-frames that can be used as *references*.

is again: $D_{dec}^{out} = (N_{GOP} - 1)t_{fr}$. We thus rewrite Eq. V.1 as:

$$\begin{aligned} D_{e2e} &= (N_{GOP} - 1)t_{fr} + 2 \times D_{enc}^{out} + t_{fr} + (N_{GOP} - 1)t_{fr} \\ &= 2 \times D_{enc}^{out} + (2 \times N_{GOP} - 1)t_{fr} \end{aligned} \tag{V.2}$$

If the rate allocation could achieve exactly $\frac{r}{30}$ bits per frame, then no buffering would be needed at the encoder output or decoder input, and the above result shows that the delays for N_{GOP} equal to 1, 2, 4 would be 1, 3, and 7, respectively, times the frame duration of 33ms. In the next section, we estimate the encoder output buffer size for perfect rate control and uneven rate allocation.

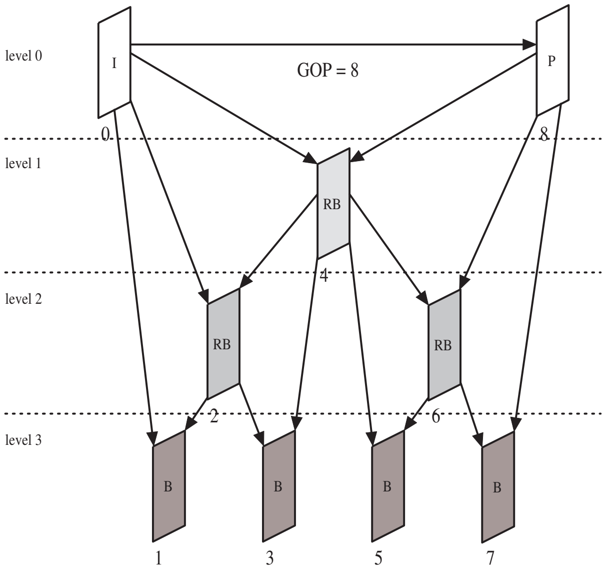


Figure V.4: Hierarchical B-Pictures for $N_{GOP} = 8$.

V.B Delay Calculation for Uneven Rate Allocation

In this section we calculate the delay due to buffering at the encoder output. Both for pulsed quality dual frame video coding, as well as for hierarchical B-pictures coding, the problem of increased encoder output buffering is unavoidable when good compression efficiency is desired. Pulsed-quality by definition involves uneven rate allocation, while a HIER coder can in theory be allocated equal bits per frame, but doing so severely degrades compression efficiency.

The following assumptions are made: (a) bit rate is controlled by varying the QP and block coding mode to achieve the allocated target rate for this frame, and (b) the rate allocation mechanism is accurate enough to ensure that within some block of N frames it allocates exactly $N \times R$ rate, where R the constant bit rate. The block of N frames associated with the rate allocation periodicity is not necessarily the same as the GOP associated with the prediction structure periodicity. We will refer to the block of N frames as GOP-R. The length N of the GOP-R is not necessarily equal to N_{GOP}. Recall that for both IPPP and PULSE we have $N_{GOP} = 1$, while $N_{GOP} > 1$ implies a hierarchical B-picture prediction structure. Here, in the context of rate allocation, term N is equal to: (i) 1 for the IPPP case, (ii) U (the long-term frame updating period) for the PULSE case, and (iii) N_{GOP} for the HIER case.

Let x_i, where $0 \leq x_i \leq N \times R$, denote the rate allocated to frame i. Let $x_M = \max_{i \in [0,N-1]} x_i$ denote the maximum value of x_i. Let B denote the encoder output buffer length in bits. To avoid a buffer overflow during encoding, the necessary condition is:

$$B = \max_{j \in [0,N-1]} \left(\sum_{i=0}^{j} x_i - j \times R \right) \tag{V.3}$$

The encoder can estimate the encoder output buffer length from the bit rate allocation. A useful and intuitive lower bound can be written as

$$B > \max(R, x_0, x_M) \tag{V.4}$$

The buffer size should be larger than the first frame, the largest frame (in terms of bits), and the constant rate per frame R. We note that very often the first frame in a GOP-R is the frame that is allocated the highest single number of bits in the GOP-R.

In video coding systems where the initial frame is pulsed and the latter frames are starved (not necessarily equal in size), the ensuing delay depends on the rate allocated to the pulsed frame. Hence the decision on the rate allocated to the pulsed frame trades-off delay for compression performance. This result is valid both for hierarchical B-pictures coding with GOP-R size $N = N_{GOP}$, as well as for pulsed quality dual frame coding with GOP-R size N equal to updating period $N = U$. It translates a delay constraint into a rate allocation constraint.

Allocating excessive rate to a frame, and starving the rest, can not only decrease performance, but can also dramatically increase delay due to buffering. In the next section (Section V.C) we seek to find a good theoretical rate allocation for a hierarchical structure. For pulsed quality we derive experimentally an efficient rate allocation scheme later in this chapter.

V.C Proposed Framework for Rate Allocation

V.C.1 Motivation

The original rate allocation in the JSVM scalable video coder uses a single QP for the entire frame. The QP value allocated to a bi-directionally predicted picture (B-picture) is higher (coarse quantization) than the average of the QPs used to encode the two references. This arrangement guarantees high compression efficiency. However, rate allocation under tight delay constraints cannot use the same QP for the entire frame. A different QP is allocated every, say, 11 blocks, in order to achieve a rate constraint. We can only set the bit rate for each frame. The per-block QP decisions seek to avoid buffer overflow and underflow and satisfy the target rate.

Our goal is to establish the bit rate allocation for different hierarchical levels with B-pictures, shown in Fig V.4. It is known that bi-directional prediction (multi-hypothesis prediction with two hypotheses) attenuates the prediction error energy by half, compared to uni-directional prediction [1]. Furthermore, we have to take into account the temporal distance from the reference frames. We found through experiments that the efficiency of the bi-directional prediction of a frame depends on the distance from its references. The QP allocation algorithm in JSVM ignores this distance. However, the temporal distance determines the motion compensation accuracy and hence the resulting prediction error. Our goal is to find a good theoretical model on the influence of prediction temporal distance to compression efficiency.

We now discuss our main assumptions: (a) frames within a temporal decomposition level have similar entropy, (b) closed-loop coding, and (c) high rate operation.

We assume that the image sequence is correlated enough so that frames within the same temporal decomposition level have similar entropies and can be afforded the same number of bits. We seek a solution that does not depend on video content: fixed proportion of bits for each temporal decomposition level, divided equally among the frames of the level. Given the overall bit-budget and the proportions, it is straightforward to calculate the exact rates. The requirement for fixed ratios is a result of computational and delay constraints: the complexity and delay needed to optimize the rate allocation for each sequence are prohibitive.

We are primarily interested in rate allocation for closed-loop hierarchical B-pictures. Closed-loop refers to using as references the previously reconstructed versions of the frames. It always outperforms open-loop prediction. Obtaining however the optimal rate allocation for closed loop prediction is significantly harder. It is essentially a problem of dependent quantization [66]. Rate allocation for MCTF has been studied before [67] and is relatively easy to obtain since MCTF is open-loop (motion compensation is accomplished using the *original* frames). Still, those approaches did not take into account the temporal distance between the frames and lack any delay constraints. They are primarily modeling the error attenuation due to temporal filtering and are not appropriate for this work since we cannot afford the delay and the computational complexity to analyze the signal and derive near-optimal rate allocation.

We assume operation at high rates. It was shown in [68] that closed-loop prediction at high rates does not alter the signal significantly. Hence the effect of quantization error on prediction efficiency can be neglected for sufficiently fine quantization [69]. It was then suggested in [70] that using a closed-loop video coder with the optimal open-loop rate allocation performs close to the optimal closed-loop rate allocation. We will use the theory, originally developed in [61], and later extended to multi-hypothesis prediction in [1], to model rate-distortion behavior in hierarchical B-picture prediction. Open-loop compression efficiency for MCTF-like structures was investigated in [71] using the methodology of [61]. That approach, however, does not yield bit allocations on a frame basis and it also assumes a KLT transform which is not realistic in either traditional MCTF or closed-loop B-pictures.

The main reason we prefer the model of [61] over other approaches for bit allocation, is that results presented in [61, 1] depend on the motion compensation accuracy. We propose that this accuracy is a direct function of the temporal distance between the reference and the predicted frame. A second reason is the modeling of multi-hypothesis prediction coding efficiency (B-pictures involve two hypotheses).

V.C.2 Theoretical Background

We will briefly now outline the Rate-Distortion (R-D) modeling scheme from [61, 1]. Let

ω_x denote horizontal frequency,

ω_y denote vertical frequency,

$\Phi_{ss}(\omega_x, \omega_y)$ denote the signal power spectrum of the input video signal,

$F(\omega_x, \omega_y)$ denote the frequency response of the "loop filter",

$P(\omega_x, \omega_y)$ denote the two-dimensional (2-D) Fourier transform of the displacement error p.d.f.,

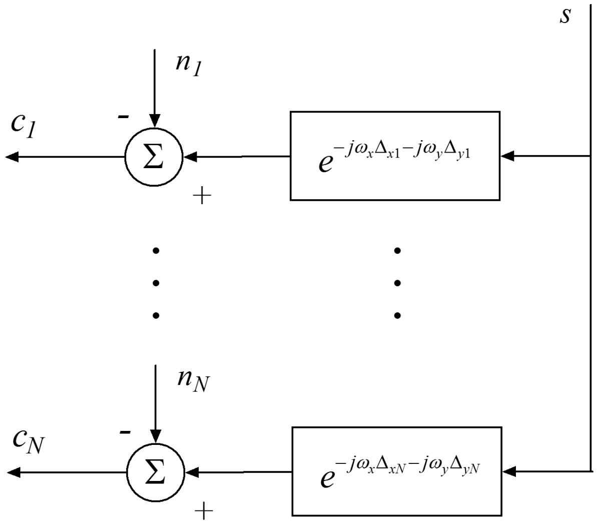
$\Phi_{nni}(\omega_x, \omega_y)$ denote the power spectrum of residual noise component i that cannot be predicted by motion compensation,

$\Re\left(\cdot\right)$ denote the real part of a complex number, and

D denote the distortion resulting from encoding a signal with R bits per sample.

The original signal $s(x, y)$ is predicted by convolving the spatial 2D convolution filter $f(x, y)$ with the hypotheses $c(x, y)$ (reference frames). The prediction error can then be written as:

$$e(x, y) = s(x, y) - f(x, y) * c(x, y) \tag{V.5}$$



s
n_1
c_1
-
Σ
+
$e^{-j\omega_x\Delta_{x1}-j\omega_y\Delta_{y1}}$
n_N
c_N
-
Σ
+
$e^{-j\omega_x\Delta_{xN}-j\omega_y\Delta_{yN}}$

Figure V.5: Signal model for multi-hypothesis prediction from [1].

The signal model is depicted in Fig. V.5, borrowed from [1]. The c_i are the hypotheses which are assumed to be versions of the original source signal s, corrupted with white noise n_i, and also shifted in 2D by Δ_{xi} in the horizontal direction and Δ_{yi} in the vertical direction. The Δ_x and Δ_y are also modeled as random variables. In this work we assume that the p.d.f. $P(\omega_x, \omega_y)$ of the displacement error Δ_x and Δ_y is a function of the temporal prediction distance. If the power spectral density of the prediction error $\Phi_{ee}(\omega_x, \omega_y)$ is known, then the error variance σ_e^2 is given from Parseval's relation as:

$$\sigma_e^2 = \frac{1}{4\pi^2} \int_{-\pi f_{sx}}^{+\pi f_{sx}} \int_{-\pi f_{sy}}^{+\pi f_{sy}} \Phi_{ee}(\omega_x, \omega_y) d\omega_x d\omega_y \qquad \text{(V.6)}$$

where terms f_{sx} and f_{sy} are the spatial sampling frequencies in the horizontal and the vertical direction. The well-known rate distortion function for memoryless coding is:

$$R(D) = \frac{1}{2}\log_2\left(\frac{\sigma_e^2}{D}\right) \tag{V.7}$$

in bits per sample (pixel). The power spectrum $\Phi_{ee}(\omega_x, \omega_y)$ is calculated for N-hypothesis prediction in [1]. Since in this work we are studying single and double hypothesis prediction, we need the expressions for $N = 1$ and $N = 2$. For $N = 1$ hypotheses (P-pictures) Equation (23) in [1] yields the following expression:

$$\begin{aligned}\frac{\Phi_{ee}(\omega_x, \omega_y)}{\Phi_{ss}(\omega_x, \omega_y)} &= 1 + \left|F_{(1)}(\omega_x, \omega_y)\right|^2 - 2\Re\{F(\omega_x, \omega_y)P_{(1)}(\omega_x, \omega_y)\} \\ &\quad + \frac{\Phi_{nn1}(\omega_x, \omega_y)}{\Phi_{ss}(\omega_x, \omega_y)}\left|F_{(1)}(\omega_x, \omega_y)\right|^2 \end{aligned} \tag{V.8}$$

For $N = 2$ hypotheses (B-pictures) the power spectral density is given from Eq. (23) in [1] as:

$$\frac{\Phi_{ee}(\omega_x, \omega_y)}{\Phi_{ss}(\omega_x, \omega_y)} = 1 - 2\Re\left\{F_{(2)}\begin{bmatrix} P_{(2,1)} \\ P_{(2,2)} \end{bmatrix}\right\} + F_{(2)}\begin{bmatrix} 1+\alpha_1 & P_{(2,1)}P^*_{(2,2)} \\ P_{(2,2)}P^*_{(2,1)} & 1+\alpha_2 \end{bmatrix} F^H_{(2)} \tag{V.9}$$

Terms α_i for each hypothesis are given in Equation (22) in [1] as:

$\alpha_i = \Phi_{nni}(\omega_x, \omega_y)/\Phi_{ss}(\omega_x, \omega_y)$, where the power spectrum of the signal s is found in Equation (19) in [62] as:

$$\Phi_{ss}(\omega_x, \omega_y) = \frac{2\pi{\sigma_s}^2}{\omega_0^2}\left(1 + \frac{\omega_x^2 + \omega_y^2}{\omega_0^2}\right)^{-\frac{3}{2}} \tag{V.10}$$

${\sigma_s}^2$ is the variance of the original signal s. The noise power spectrum is: $\Phi_{nn}(\omega_x, \omega_y) = \sigma_n^2$. Terms $P_{(1)}(\omega_x, \omega_y)$, $P_{(2,1)}(\omega_x, \omega_y)$, and $P_{(2,2)}(\omega_x, \omega_y)$ are critical and relate to motion compensation accuracy. The first is the displacement error p.d.f. for a P-picture, while the latter two correspond to each of the hypotheses in a B-picture. In [61, 1] they are not parameterized with distance and are assumed i.i.d. with p.d.f.:

$$p(\Delta_x, \Delta_y) = \frac{1}{2\pi\sigma_\Delta^2} e^{-\frac{\Delta_x^2 + \Delta_y^2}{2\sigma_\Delta^2}} \tag{V.11}$$

The Fourier transform of the above probability density function is calculated as:

$$P(\omega_x, \omega_y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(\Delta_x, \Delta_y) e^{-j\omega_x \Delta_x} e^{-j\omega_y \Delta_y} d\omega_x d\omega_y = e^{-2\pi\sigma_\Delta^2(\omega_x^2 + \omega_y^2)} \tag{V.12}$$

Finally, terms $F_{(1)}(\omega_x, \omega_y)$ and $F_{(2)}(\omega_x, \omega_y)$ from Equations V.8 and V.9 represent the Fourier transform of the spatial filters $f(x, y)$ for single and double hypothesis. For $N = 1$ hypotheses (P-pictures) the Fourier transform is equal to $F_{(1)}(\omega_x, \omega_y) = 1$ (we assume no loop filtering) since $f_{(1)}(x, y) = \delta(x, y)$. For $N = 2$ hypotheses it is set to $F_{(2)}(\omega_x, \omega_y) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$ since $f_{(2)}(x, y) = \begin{bmatrix} \frac{1}{2}\delta(x, y) & \frac{1}{2}\delta(x, y) \end{bmatrix}$. The hypotheses are simply averaged.

We continue the derivation of the power spectral densities of Eq. V.8 and Eq. V.9. To simplify notation, we adopt $\Lambda = (\omega_x, \omega_y)$ from [61]. For $N = 1$ hypothesis we obtain the following expression:

$$\Phi_{ee}^{(1)}(\Lambda) = \Phi_{nn}(\Lambda) + 2\Phi_{ss}(\Lambda)(1 - P_{(1)}(\Lambda)) \tag{V.13}$$

For $N = 2$ we derive:

$$\Phi_{ee}^{(2)}(\Lambda) = \Phi_{ss}(\Lambda)(1 + 0.5(1 + \frac{\alpha_1}{2} + \frac{\alpha_2}{2} + P_{(2,1)}(\Lambda)P_{(2,2)}(\Lambda)) - P_{(2,1)}(\Lambda) - P_{(2,2)}(\Lambda)) \tag{V.14}$$

We assume that both hypotheses are distorted equally yielding $\alpha_1 = \alpha_2 = \alpha = \frac{\sigma_n^2}{\Phi_{ss}(\Lambda)}$. Terms $P_{(2,1)}(\Lambda)$ and $P_{(2,2)}(\Lambda)$ represent the motion compensation accuracy for each of the two hypotheses. They are also assumed to be equal, so we define $P_{(2)}(\Lambda) = P_{(2,1)}(\Lambda) = P_{(2,2)}(\Lambda)$. Given the above two assumptions we finally obtain the following expression for the prediction error power spectral density:

$$\Phi_{ee}^{(2)}(\Lambda) = \frac{1}{2}\Phi_{nn}(\Lambda) + \frac{1}{2}\Phi_{ss}(\Lambda)(3 + P_{(2)}^2(\Lambda) - 4P_{(2)}(\Lambda)) \tag{V.15}$$

Comparing Equation V.13 with Equation V.15 we notice that the power spectral density of the residual error n is attenuated by $\frac{1}{2}$. This is an intuitive property of the selected averaging filter $F(\omega_x, \omega_y)$. It is obvious that if σ_n^2 is larger then there is a larger benefit in going to two hypotheses. Another property that is easy to identify is that if $P_{(2)}(\Lambda) = P_{(1)}(\Lambda)$ then two hypotheses are always better than a single hypothesis since: $\Phi_{ee}^{(1)}(\Lambda) - \Phi_{ee}^{(2)}(\Lambda) = \frac{1}{2}[\Phi_{nn}(\Lambda) + \Phi_{ss}(\Lambda)(1 - P^2(\Lambda))] \geq 0$. Naturally, there can be cases where the previous inequality is not upheld. These are cases where $P_{(2)}(\Lambda) \neq P_{(1)}(\Lambda)$.

V.C.3 Proposed Estimate

We parameterize $P(\Delta_x, \Delta_y)$ with respect to the temporal prediction distance by setting $\sigma_\Delta = \sigma_\Delta(\Delta_t)$. In a hierarchical B-picture system with $L+1$ temporal decomposition levels, the prediction distance Δ_t decreases as the temporal resolution increases. For $L = 1$, which corresponds to a GOP size of 2 and is the well known IBPBPB structure, the distance Δ_t is 2 frames at level 0 and 1 frame at level 1. Let l denote the temporal decomposition level. We can write the prediction distance with respect to the level and the GOP size N_{GOP} as:

$$\Delta_t(l) = 2^{-l} N_{GOP}, l \in [0, L] \tag{V.16}$$

We assume full binary temporal decomposition, hence: $N_{GOP} = 2^L$. Let us assume that $\sigma_\Delta(\Delta_t)$ is known, and that the optimal open-loop rate allocation is sufficiently close to the optimal closed-loop allocation. This allows us to find the optimal rate allocation with a single R-D curve for each of the $L + 1$ hierarchical levels. In reality, the rate allocation for each higher level (with smaller index) has a direct effect on the R-D curve of the immediately lower level as shown in [66]. The R-D curve is shifted for different quantization levels (or bit rate allocations) of its reference frames. Since, however, we operate at high rates and strict optimality is not our objective, we can generate $L + 1$ R-D curves. The rate allocation is obtained by constraining the same distortion D across all $L + 1$ curves.

The above technique relies on estimating $\sigma_\Delta(\Delta_t)$. We therefore encoded several sequences for varying frame rates where we constrained equal rate on a frame basis. Two of those sequences are shown in Fig. V.6. We used the JM 10.1 AVC reference software to derive these encoded sequences. Single-frame/hypothesis prediction was used.

The rate control operated on groups of 11 macroblocks. The different temporal distances were obtained by varying the temporal sub-sampling ratio. The graphs present the distortion as mean squared error (not PSNR) versus and bits per sample (pixel) and the curves correspond to different temporal distances. A frame rate of 30fps leads to $\Delta_t = 1$, 15fps has $\Delta_t = 2$, 7.5fps leads to $\Delta_t = 4$, and so on. The figures show that the mean squared error σ_e^2 curves move to the right and to the top with an approximately logarithmically-spaced rate. Hence σ_e^2 seems to be a logarithmic function of the temporal distance Δ_t. We now need to establish the relationship between Δ_t and σ_Δ.

Replacing Φ_{ee} in Equation V.6 with the expression derived in Equation V.13 we obtain the theoretical performance for a single-hypothesis hybrid video encoding system with memoryless encoding of the prediction error. This is illustrated in Fig. V.7. Term $\sigma_\Delta \omega_0$ is varied at equal spaces and produces approximately logarithmically-spaced rate-distortion functions. This is very important as it mirrors the behavior of R-D curves in our experimental investigation. We conclude that for fixed ω_0 the standard deviation of the motion compensation displacement error σ_Δ varies *linearly* with the temporal prediction distance Δ_t. Therefore, we can estimate σ_Δ as:

$$\tilde{\sigma}_\Delta(\Delta_t) = \alpha + \beta(\Delta_t - 1) \tag{V.17}$$

The next challenge is to estimate the constants α and β. The AVC reference software uses quarter-pel accurate motion compensation. From [1] we know that the value of $\tilde{\sigma}_\Delta(1) = \alpha$ is calculated to be approximately 0.0702 for quarter-pel MCP. For half-pel motion compensation it doubles to 0.1404, and again doubles to 0.2808 for integer-pel motion estimation. Parameter β was estimated by fitting the theoretical curves in Fig. V.7 to experimental data, some of which were presented in Fig. V.6. We note that the above estimated parameters α and β are valid for the specific values of ω_0, σ_s, and σ_n we selected.

The final rate-distortion model is written as:

$$R_l = \frac{1}{2}\log_2\left(\frac{\sigma_e^2(\Delta_t, N)}{D \times (1 + \epsilon \times (l+1))}\right) \tag{V.18}$$

Recall that l is the decomposition level. Parameter N denotes the number of hypotheses used to predict that frame. Term σ_Δ is calculated using the linear model from Equation V.17 and is then plugged into $\Phi_{ee}^{(N)}(\omega_x, \omega_y)$ to yield $\sigma_e^2(\Delta_t, N)$. The motivation behind adding the term $\epsilon \times (l+1)$ to the denominator of Equation V.18 has to do with the fact that hybrid video coding is closed-loop and thus a case of dependent video coding. Frames at temporal level l are being predicted from frames of level $l-1$ or less. However, these reference frames have already been quantized and the R-D curves of the current frame will have shifted as a result. The constant parameter ϵ was empirically set to a small value: $\epsilon = 0.1$.

The obtained rate R values are used to establish bit rate ratios among temporal levels. Assuming for example $N_{GOP} = 4$, we have three temporal levels: 0, 1, and 2. Level 0 contains the P frames for which $\Delta_t = 4$ and $N = 1$. Level 1 contains RB frames (RB in H.264/AVC notation are B frames that can be referenced during motion compensation) for which $\Delta_t = 2$ and $N = 2$. Finally, at level 2 we have B frames (in H.264/AVC notation it means that these frames cannot be used as references) for which the temporal distance is $\Delta_t = 1$ and $N = 2$. Using our algorithm we fix a common D, say a mean squared error of 20, that corresponds to a PSNR of 35.12dB. We thus obtain R_0, R_1, and R_2. Finally, we encode the sequence by allocating $R_0 \times c$ bits per frame to frames of level 0, $R_1 \times c$ bits per frame to frames of level 1, and $R_2 \times c$ bits per frame to frames of level 2.

Given an average bandwidth constrain of R_f bits per frame we can easily calculate the parameter c as follows:

$$c = \frac{2^L R_f}{1 + \sum_{i=1}^{L} 2^{i-1} r_i} \tag{V.19}$$

V.D Structural Delay Trade-Off

So far we have studied the delay trade-off with respect to encoder output buffering that is a direct result of the rate allocation scheme. However, from Equation V.1 we note that end-to-end delay is also a function of structural delay, which is non-trivial in hierarchical prediction systems. The structural delay is not however monolithic: it can be reduced. Doing so reduces compression efficiency as well. A study on reduced structural delay appeared in [59] for MCTF systems, and reduced structural delay has been integrated into the JSVM scalable video software. In this section we study the effect of *branch removal* from hierarchical B-picture coders.

To illustrate our examples we will use the case where GOP size is set to 4.

Such a prediction structure is illustrated on the left of Figure V.8. Frame 2 is predicted from frames 0 and 4, then frame 1 is predicted from frames 0 and 2, and frame 3 is predicted from frames 2 and 4. Assume now that the prediction of frame 2 from frame 4 is removed, as shown in the middle of Figure V.8. This brings down the structural delay by one half: the truncated GOP size 4 structure has $N_{GOP} = 2$ instead of 4. The structure is similar to a GOP size 2 structure, shown on the right of Figure V.8. There are only two differences: (a) it still has 3 hierarchical levels and allows more granular temporal scalability or network condition adaptability (frames 1, 2, and 3 can be dropped without affecting the reconstruction of frame 4) and (b) frame 4 instead of being predicted from frame 2 as for GOP size 2, is predicted from frame 0. This means that compression performance will be worse than a GOP size 2 structure, since the temporal prediction distance for frame 4 increases. Hence for hierarchical B-pictures the trade-off of compression efficiency for delay effectively becomes a trade-off of compression efficiency for increased temporal scalability and bitstream resilience and decreased delay.

The rate allocation scheme presented in Section V.C.3 can be used to derive an efficient bit rate distribution in those cases where prediction branches are truncated. The scheme will adapt the parameter σ_{Δ} to reflect the removal of the branch and reflect single over double hypothesis. Some indicative experimental results are provided in Section V.F.

V.E Rate Control and Implementation

For the *IPPP* codec we derive from Eq. V.2 the end-to-end delay as $D_{e2e} = 2D_{enc}^{out} + t_{fr}$. Recall that both IPPP and PULSE have $N_{GOP} = 1$. Two short-term reference frames were used for motion compensation. The rate control algorithm is the one included in the JM 10.1 reference software and described in detail in [72].

For the *PULSE* codec the updating period was chosen as $U = 5$, and we allocated two or three times as many bits to the LT frames as to the regular frames. The exact number of bits is calculated adaptively so that the overall rate constraint is satisfied. The decisions to allocate twice or thrice the short-term bits and to set $U = 5$ are not optimal. Better performance could be achieved by optimizing these parameters, but exploring this large parameter space is beyond the scope of this chapter. As in the IPPP case, the end-to-end delay is $D_{e2e} = 2D_{enc}^{out} + t_{fr}$, but now the output buffer will be larger for good performance, since the high quality frames require more bits.

In PULSE the rate allocation is similar to that in [72] with some critical modifications. We do not allocate rate to ST and LT frames from a common budget. The budget is divided into two bins: the ST and the LT rate bins. Two separate rate control "paths" for ST and LT frames draw bits from the respective bins. They however share the constraint on the encoder buffer status taking care to avoid a buffer overflow. In each rate control path, the buffer limit is enforced both (a) by modifying the QP so as to achieve the target rate but also (b) by the last-resort measure of forcing SKIP coding modes on blocks when the buffer is about to overflow. We switch the QP on a basis (*basic unit*) of 11 macroblocks (MB). The quadratic model of [72] selects a QP for this *basic unit* that *ought* to avoid an overflow. Since the quadratic model is only an estimate, there are many cases where SKIP modes must be invoked. Signalling a SKIP mode for a MB involves transmitting two bits. In that case, the reconstructed MB is a motion-compensated prediction from a previous reference frame. The motion vector is obtained through spatial prediction of neighboring motion vectors.

Concerning the HIER coder, we note that the JSVM 3.3.1 reference software does not include rate control for the base layer. As a result, we adopted the rate control scheme from the JM software [72]. This rate control scheme was not designed with hierarchical motion-compensation structures in mind. It addresses B-frames in non-hierarchical structures.

While the rate of the P-frames is strictly controlled by changing the QP in terms of basic units, the B-frames are allocated a single QP value for the entire frame. Thus, the rate is not explicitly controlled. In general, for constant QP allocation, a B-frame will be noticeably smaller than its neighboring P-frames, due to the efficiency of bi-directional prediction. Furthermore, the rate control of [72] allocates a QP incremented by two over the average QP of the neighboring P-frames. The B-frame is thus guaranteed to be smaller than the neighboring P-frames. To ensure accurate rate control under tight delay constraints, we adopt the rate control approach of the PULSE codec, with multiple rate-control paths. For a hierarchical stream the number of rate control bins is equal to the number of temporal decomposition levels. For example, for $N_{GOP} = 4$, we obtain three bins: one for the P frames, a second one for the RB frames, and, last, a third for the B frames. Frames draw their bits only from their corresponding rate control bin. Still, as in the PULSE case, all three bins share the same buffer and thus the same constraint. In cases where the rate control is close to triggering a buffer overflow we strongly increase the QP.

We note that large values of N_{GOP} such as 8 and 16 are possible (the H.264/AVC specification allows up to 16), but preliminary trials showed that the gain in PSNR is small compared to the dramatic increase in end-to-end delay. Still they offer better temporal scalability, error resilience, and bit stream adaptability.

We note that for IPPP and PULSE, the output frame order is $[012345678]$. For the $N_{GOP} = 2$ case, the order is $[021436587]$. For $N_{GOP} = 4$, the order is $[042138657]$. Last, for the truncated GOP size 4 case, the order is identical to that for $N_{GOP} = 2$.

All investigated video codecs are fully compatible with H.264/AVC [64]. In fact, all results in Section V.F were obtained with the JM 10.1 reference decoder for all JM and JSVM bit streams. Although we ignored the encoder complexity during delay calculation, we constrained it to be approximately equal for all four types of streams. The hierarchical codecs use one short-term reference frame for the P-frames. However, the B-frames are encoded with bi-directional prediction, the complexity of which is comparable to prediction from two reference frames. Additional iterations (by fixing the forward motion vector as we are optimizing the backward motion vector, and then reversing the process and iterating it), needed for bi-directional prediction to converge, may lead to greater complexity. The PULSE codec uses two reference frames. We hence used two short-term reference frames for the IPPP codec as well.

V.F Results

V.F.1 Proposed Rate Allocation

The efficiency of the scheme proposed in Section V.C.3 is illustrated in Figure V.9(b) where we present encoding results for three different rate allocations: (a) a trivial uniform rate allocation where each frame, irrespective of its temporal level and prediction distance, receives the same number of bits, (b) an intuitive allocation where all B frames, irrespective of temporal level, receive half the rate of the P frames, and (c) our proposed scheme. We observe that the rate allocation model we obtained theoretically outperforms the other two heuristic schemes. We note that allocation (b) becomes less efficient than allocation (a) as the bit rate increases. The performance delta for allocation (c) versus the rest tends to increase with increasing rate, which is attributed to our high rate assumptions used for deriving the scheme in Section V.C.3.

V.F.2 Delay Trade-Offs

We investigated the performance of the four codecs for a variety of video sequences: *Carphone* includes localized motion of various kinds. Still, the majority of the activity is due to the instability of the camera inside the car. There is repetitive translational global motion. *Mobile* has high frequency content and the motion is mostly global due to the horizontal camera pan. *Flower* also has high frequency content, and the motion is again global. However, this time the motion is not planar/translational since objects come closer to the camera.

In Figs. V.10 and V.11 we show video quality versus end-to-end delay. The bit rate is fixed for all curves displayed within a graph of Figs. V.10 and V.11. The delay was varied by allocating different numbers of bits to the encoder output buffer (D_{enc}^{out}). Performance increases with delay and GOP size. $N_{GOP} = 4$ outperforms $N_{GOP} = 2$, which in turn outperforms $N_{GOP} = 1$, both IPPP and PULSE. The truncated GOP size 4 is underperforms $N_{GOP} = 2$ as was predicted theoretically. Last, PULSE is better than IPPP.

We observe in Figs. V.10 and V.11 that the IPPP codec achieves good performance at a delay of around 51ms. We note that the minimum delay in our system is the transmission time $t_{fr} = 33$ms. So good performance at 51ms means that the IPPP codec needs only $(51 - 33)/2 = 9$ms of delay in the encoder output buffer (and the same at the decoder input buffer) to allow sufficient flexibility in rate control decisions.

The PULSE codec achieves good performance at a delay of around 77ms for $2\times$ pulsing. We note that the minimum delay that guarantees good performance can be calculated from U and the long-term to short-term bit budget ratio. The PSNR performance depends on the image sequence. For the highly active “Flower” there is no gain over the IPPP codec. Significant gains are however observed for the “Carphone” and “Mobile” sequences. For $3\times$ pulsing, both the delay, at 117ms, as well as the performance is slightly higher.

Moving to the $GOP = 2$ case, we observe that the end-to-end delay needs to be at least 170ms for good performance. There is no performance gain over PULSE for “Carphone”. However, impressive gains are observed in “Mobile” and “Flower”. It is thus evident that hierarchical structures can be very advantageous in static sequences or sequences with global motion.

The truncated $GOP = 4$ codec exhibits considerably lower end-to-end delay compared to the $GOP = 4$ codec due to the removal of the backward prediction branch, but the delay is still somewhat higher than the $GOP = 2$ codec. Even though the structural delay is indeed the same, the truncated $GOP = 4$ codec suffers from the fact that the anchor P frames (e.g. 0, 4, 8, ...) have to be afforded more rate compared to the inner P frames (e.g. 2, 6, 10, ...) because they are predicted from a larger temporal distance. The rate allocation scheme we developed in Section V.C.3 gives us an approximate estimate of the required increase in rate to compensate for the drop in motion compensation efficiency. From the graphs we see that, as expected, this structure is not beneficial. However, it provides additional temporal scalability for much lower delay compared to $GOP = 4$.

The increase of the GOP size to 4 increases delay considerably to more than 300ms. Apart from increased GOP delay, the anchor P-frames get large contributing further to delay. The three B-frames in each GOP need many fewer bits to be encoded. In terms of performance gain, “Carphone” and “Mobile” benefit the most. The largest gain in “Mobile” is attributed not only to its global motion but also to the fact that it is translational.

V.G Conclusions

We studied end-to-end delay versus compression efficiency trade-offs for video encoders with varying GOP size. The end-to-end delay depends on the structural and the buffering delay. The buffering delay was found to be a function of the rate allocation. We hence investigated the effect on delay of allocating more bits to some frames than the rest. All these codecs are H.264/AVC compliant. We implemented a robust rate control algorithm for the PULSE codec as well as for the hierarchical B-pictures. The work in Chapter III used constant QPs without any consideration of rate or delay constraints. Here we operated under both constraints.

A theoretical framework was derived for rate allocation in the context of varying temporal distances and number of prediction hypotheses. We found that the standard deviation of the motion compensation displacement error σ_Δ varies approximately *linearly* with the temporal prediction distance Δ_t.

We investigated constraints in structural delay through prediction branch truncation for lower delay. This leads to worse compression efficiency but is efficient in terms of scalability and bit stream adaptability. Our rate allocation scheme was used to find an efficient bit distribution for these cases.

The study of the delay trade-offs yielded the following conclusions:

(a) IPPP performs well at low delay applications and for sequences with high motion.

(b) PULSE is advantageous for relatively static sequences with repetitive content.

(c) $N_{GOP} > 1$ structures benefit from static sequences and from sequences with global motion.

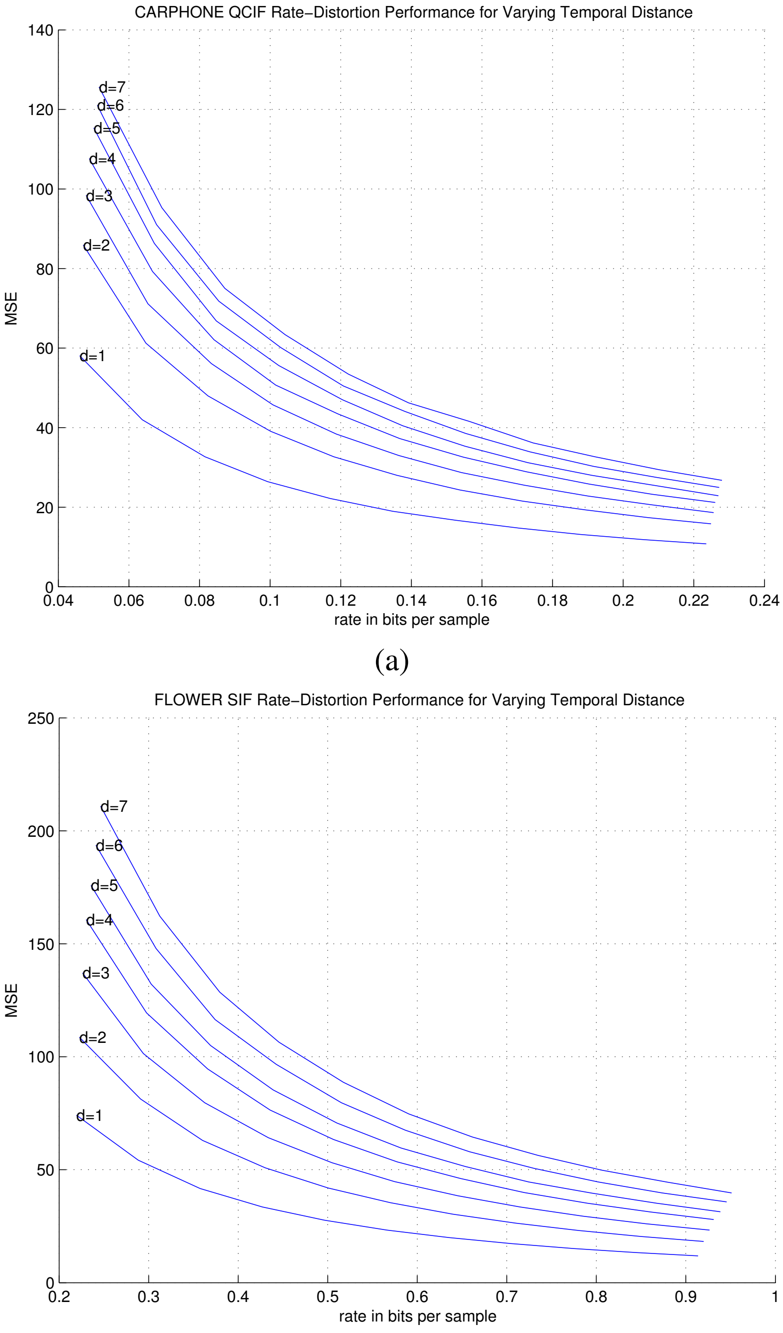
(d) As N_{GOP} increases, the gain is non-trivial only if the sequence is either static, or if the global motion is translational. In general, to achieve the optimal R-D performance, a switching mechanism, such as the proposed Adaptive GOP Size for H.264/AVC, has to be used to adapt the length of the GOP or the amount of pulsing according to the sequence statistics.

(e) For the sequences we evaluated, the delay thresholds are as follows: between 51ms and 77ms IPPP is the best choice, between 77ms and 170ms PULSE performs well, the large space between 170ms and 300ms is dominated by $N_{GOP} = 2$, and for delays larger than 300ms then $N_{GOP} = 4$ is the best choice. Delays larger than 300ms are only however useful in cases of live event broadcast or streaming of stored content. They are prohibitive for real-time communication.

(f) The truncated $GOP = 4$ codec underperforms the $GOP = 2$ codec but has similar delay with the added advantage of increased temporal scalability.

V.H Acknowledgements

Chapter V of this dissertation, in full, is a reprint of the material as it appears in A. Leontaris and P. C. Cosman, “Compression Efficiency and Delay Trade-Offs for Hierarchical B-Pictures and Pulsed-Quality Frames,” which is under preparation for submission to a journal. I was the primary author and the co-author Dr. Cosman directed and supervised the research which forms the basis for Chapter V.



(a)

(b)

Figure V.6: PSNR vs. rate (in bits per sample) for varying temporal prediction distance. (a) “Carphone” QCIF 176×144. (b) “Flower” SIF 352×240.

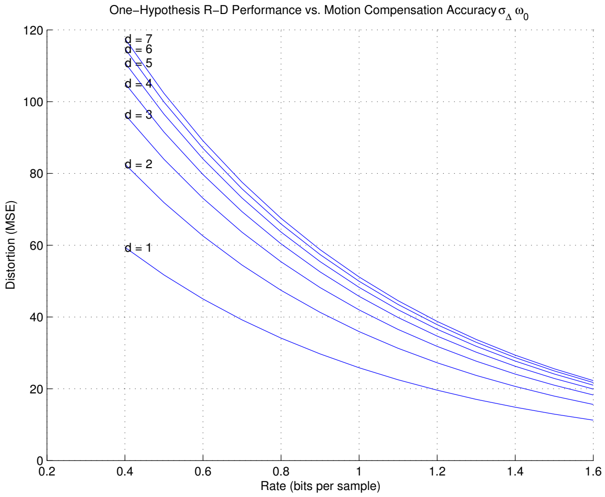


Figure V.7: Theoretical performance for varying $\sigma_\Delta \omega_0$.

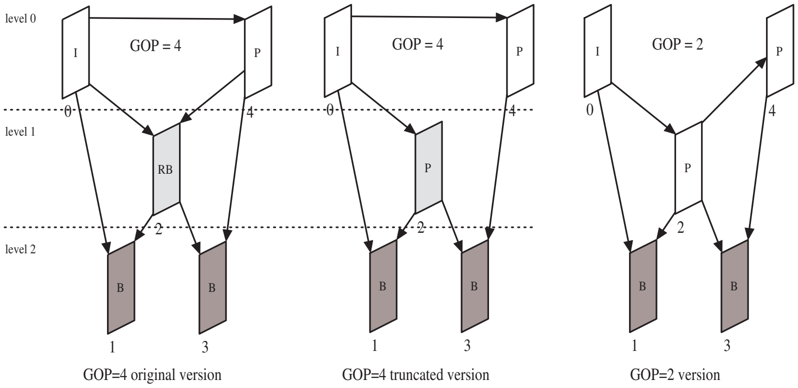
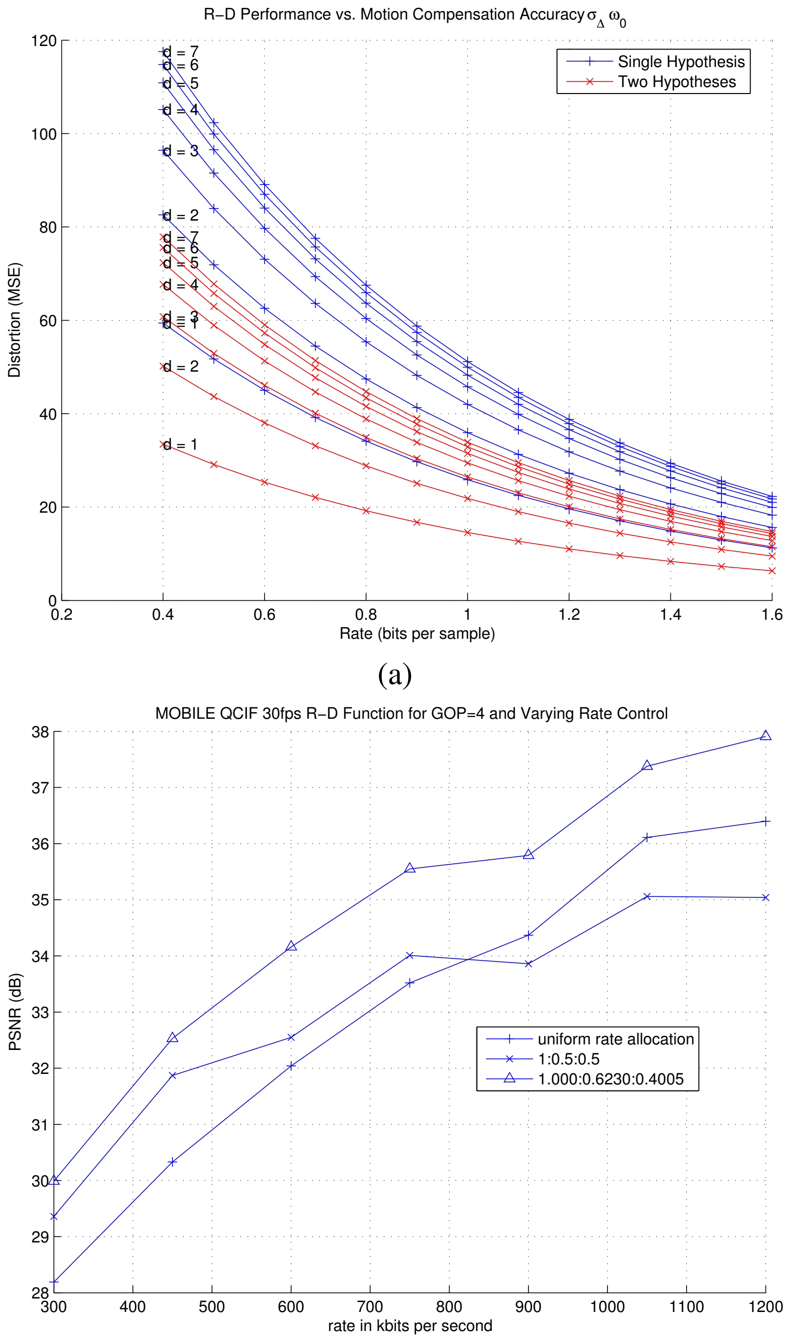


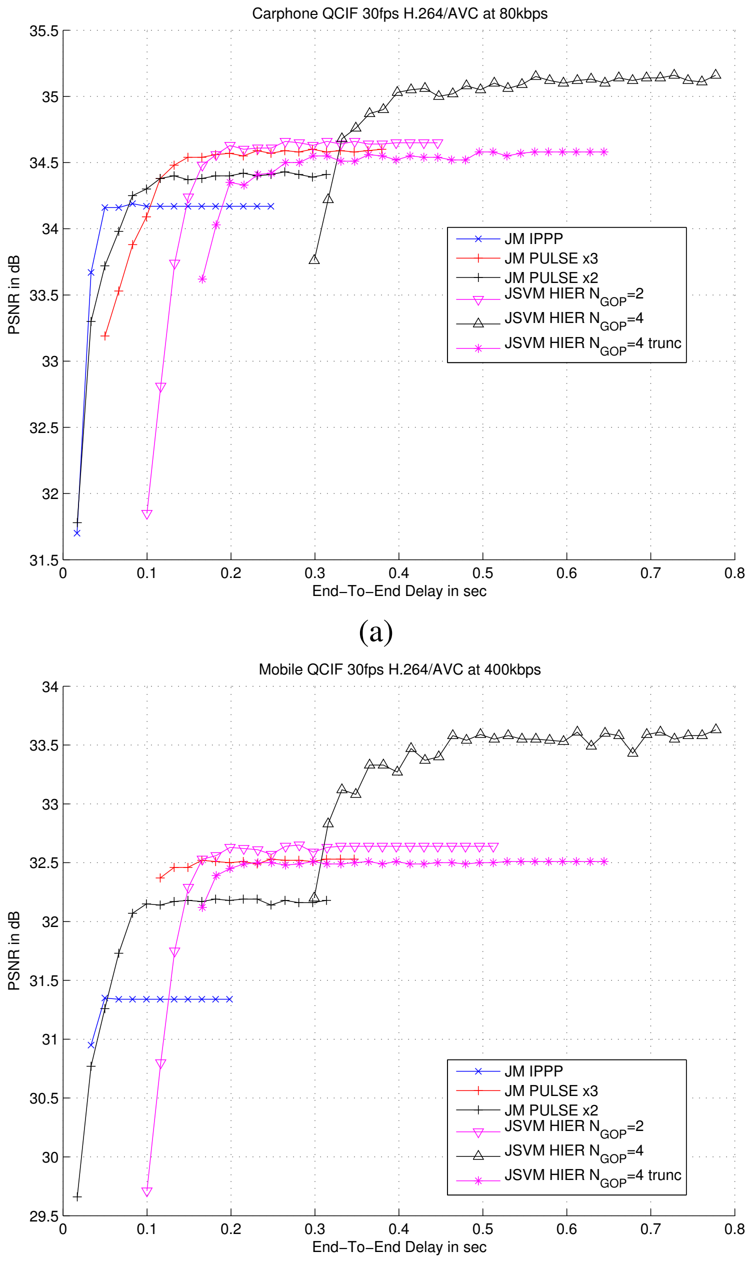
Figure V.8: A $N_{GOP} = 4$ original structure and its truncated version.



(a)

(b)

Figure V.9: (a) Single and double hypothesis R-D behavior calculated theoretically for difference temporal prediction distances. (b) Performance of the proposed rate allocation for “Mobile” QCIF at 30fps with $N_{GOP} = 4$.



(a)

(b)

Figure V.10: PSNR vs. delay (in seconds) for fixed source coding bit rate. (a) “Carphone” QCIF 176×144 at 30fps. (b) “Mobile” QCIF 176×144 at 30fps.

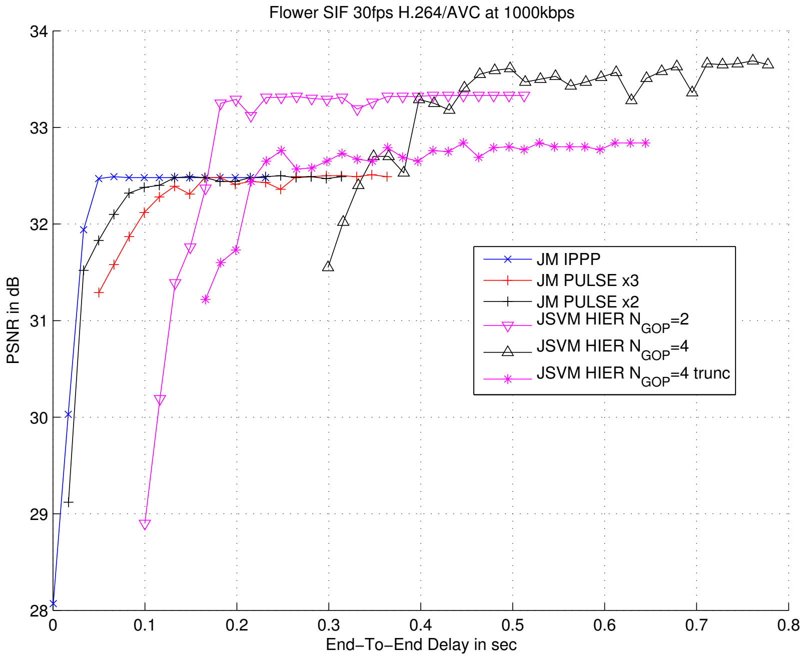


Figure V.11: PSNR vs. delay (in seconds) for fixed source coding bit rate. “Flower” SIF 352×240 at 30fps.

Chapter VI

Conclusions

In this dissertation, we studied several aspects of the use of multiple reference frames for motion-compensated prediction. We investigated (a) rate-distortion optimal coding mode selection for multiple reference frames with respect to error resilience, (b) improved compression efficiency and error resilience through the use of high quality long-term reference frames, (c) the reduction of drift in scalable video coding through the use of multiple reference frames and drift estimation, and, last, (d) the issue of delay in high-quality long-term frame prediction and hierarchical B-pictures. We now enumerate our contributions, categorized according to the chapter in which they appear.

In Chapter II we treated multiple frame prediction for error resilience.

1. The combination of per-pixel distortion estimation, rate-distortion optimal mode selection, and the presence of the long-term reference frame increased error resilience substantially.

2. We proposed a novel algorithm for more accurate distortion estimation at the en-

coder side when half-pixel motion compensation is used.

3. We showed that the additional use of feedback, and, in particular, the use of feedback to render the long-term frame reliable, has a very positive effect for the error robustness of the video bit stream.

In Chapter III we studied high quality long-term reference frames.

1. The application of high-quality to the long-term frames can improve error robustness substantially.
2. The dual-frame coder with pulses of high quality provided to the long-term frame, when used in conjunction with random intra refresh, performs worse than the regular dual frame coder where long-term frames are chosen from among regular quality frames.
3. We modified the distortion estimation algorithm of Chapter II to improve computational complexity and adapt to the new type of codec (H.264) used in this work.

In Chapter IV we investigated drift suppression for fine granularity SNR scalable video coding with multiple frames.

1. We proposed a novel per-pixel drift estimation algorithm for scalable video coding. The algorithm enables estimation of all moments of the pixel-random variable.

2. Pulsed-quality long-term frame prediction was shown to be advantageous for low-to-medium rates and video content with sufficient temporal redundancy.

Finally, in Chapter V we studied end-to-end delay versus compression trade-offs for multiple frame video coding.

1. We investigated the effect on delay of allocating more bits to some frames than the rest. Both the buffer length as well as the average rate were constrained. We implemented a robust rate control algorithm for the PULSE codec as well as for the hierarchical B-pictures.

2. We found that the standard deviation of the motion compensation displacement error σ_Δ varies approximately *linearly* with the temporal prediction distance Δ_t.

3. We investigated constraints in structural delay through prediction branch truncation for lower delay.

4. The study of the delay trade-offs yielded the end-to-end delay thresholds over which each coding variant becomes useful.

VI.A Future Work

In Chapter II, a pre-determined and fixed value of the jump updating parameter N is not optimal for all sequences. Future work concentrates on finding good rules for choosing N and D for a general (N, D) updating scheme, and for choosing LT frames in an irregular updating scheme. It would be desirable to know which update parameter is best for a given sequence. Some sequences exhibit long-term statistics that could be best captured by using relatively large update parameters or by setting a constant distance frame buffer in the remote past.

In Chapter III, the heuristic allocation we used worked well for all combinations of image sequences and bit rates. Future work should concentrate on finding an efficient explicit rate control mechanism to allocate rate to the long-term and regular frames. Finding good update parameters is at least as challenging.

In Chapter IV, future work could include modeling drift in the evolving SVC standard [46]. FGS is used in an LPLR coding approach that encodes base layer motion-compensated residuals to achieve SNR scalability. Prediction from EL frames, similarly to HPLR and HPHR coding modes, can be used to improve the compression efficiency of the FGS layer, introducing potential drift.

In Chapter V, future work is needed in improving the rate allocation model, as well as dynamically adapting the GOP length, the prediction branches and rate allocation to satisfy rate and delay constraints.

Bibliography

[1] B. Girod, "Efficiency analysis of multi-hypothesis motion-compensated prediction for video coding," *IEEE Trans. Image Processing*, vol. 9, no. 2, pp. 173–183, Feb. 2000.

[2] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 19–31, Feb. 1997.

[3] T. Wiegand, "Joint final committee draft for joint video specification H.264," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-D157, July 2002.

[4] Y.-W. Huang, B.-Y. Hsieh, T.-C. Wang, S.-Y. Chien, S.-Y. Ma, C.-F. Shen, and L.-G. Chen, "Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Apr. 2003, pp. 145–148.

[5] Y. Su and M.-T. Sun, "Fast multiple reference frame motion estimation for H.264," in *Proc. IEEE International Conference on Image Processing*, Oct. 2004.

[6] M. Gothe and J. Vaisey, "Improving motion compensation using multiple temporal frames," in *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, vol. 1, May 1993, pp. 157–160.

[7] N. Vasconcelos and A. Lippman, "Library-based image coding," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. v, Apr. 1994, pp. 489–492.

[8] N. Mukawa and H. Kuroda, "Uncovered background prediction in interframe coding," *IEEE Trans. Commun.*, vol. 33, no. 11, pp. 1227–1231, Nov. 1985.

[9] S. Brofferio and V. Corradi, "Videophone coding using background prediction," in *Proc. European Signal Processing Conference*, vol. 2, 1986, pp. 813–816.

[10] D. Hepper and H. Li, "Analysis of uncovered background prediction for image sequence coding," in *Proc. Picture Coding Symposium*, 1987, pp. 192–193.

[11] D. Hepper, "Efficiency analysis and application of uncovered background prediction in a low bit rate image coder," *IEEE Trans. Commun.*, vol. 38, no. 9, pp. 1578–1584, Sept. 1990.

[12] X. Yuan, "Hierarchical uncovered background prediction in a low bit rate video coder," in *Proc. Picture Coding Symposium*, 1993, p. 12.1.

[13] K. Zhang and J. Kittler, "A background memory update scheme for H.263 video codec," in *Proc. European Signal Processing Conference*, vol. 4, Sept. 1998, pp. 2101–2104.

[14] R. Kutka, "Content-adaptive long-term prediction with reduced memory," in *Proc. IEEE International Conference on Image Processing*, vol. 3, Sept. 2003, pp. 817–820.

[15] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70–84, Feb. 1999.

[16] T. Fukuhara, K. Asai, and T. Murakami, "Very low bit-rate video coding with block partitioning and adaptive selection of two time-differential frame memories," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 212–220, Feb. 1997.

[17] A. Leontaris and P. C. Cosman, "Video compression for lossy packet networks with mode switching and a dual-frame buffer," *IEEE Trans. Image Processing*, vol. 13, no. 7, pp. 885–897, July 2004.

[18] V. Chellappa, P. C. Cosman, and G. M. Voelker, "Dual frame motion compensation with uneven quality assignment," in *Proc. IEEE Data Compression Conference*, Mar. 2004, pp. 262–271.

[19] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, "Error-resilient video transmission using long-term memory motion compensated prediction," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 1050–1062, June 2000.

[20] M. Budagavi and J. D. Gibson, "Multiframe video coding for improved performance over wireless channels," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 252–265, Feb. 2001.

[21] S. Fukunaga, T. Nakai, and H. Inoue, "Error-resilient video coding by dynamic replacing of reference pictures," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 3, Nov. 1996, pp. 1503–1508.

[22] Y. Tomita, T. Kimura, and T. Ichikawa, "Error resilient modified inter-frame coding system for limited reference picture memories," in *Proc. Picture Coding Symposium*, Sept. 1997, pp. 743–748.

[23] Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, “Error resilient video coding using unequally protected key pictures,” in *Proc. International Workshop on Very Low Bitrate Video Coding*, Sept. 2003, pp. 290–297.

[24] Y. Liang, M. Flierl, and B. Girod, “Low-latency video transmission over lossy packet networks using rate-distortion optimized reference picture selection,” in *Proc. IEEE International Conference on Image Processing*, vol. 2, Sept. 2002, pp. 181–184.

[25] Y. J. Liang, E. Setton, and B. Girod, “Channel-adaptive video streaming using packet path diversity and rate-distortion optimized reference picture selection,” in *Proc. IEEE Workshop on Multimedia Signal Processing*, Dec. 2002, pp. 420–423.

[26] S. Lin, S. Mao, Y. Wang, and S. Panwar, “A reference picture selection scheme for video transmission over ad-hoc networks using multiple paths,” in *Proc. IEEE International Conference on Multimedia and Expo*, Aug. 2001, pp. 96–99.

[27] G. Cheung, “Near-optimal multipath streaming of H.264 using reference frame selection,” in *Proc. IEEE International Conference on Image Processing*, vol. 3, Sept. 2003, pp. 653–656.

[28] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the internet,” *IEEE/ACM Trans. Networking*, vol. 7, no. 4, pp. 458–472, Aug. 1999.

[29] B. Girod, M. Kalman, Y. J. Liang, and R. Zhang, “Advances in channel-adaptive video streaming,” in *Proc. IEEE International Conference on Image Processing*, vol. 1, Sept. 2002, pp. 9–12.

[30] T. Wiegand, M. Lightstone, D. Mukherjee, T. Campbell, and S. Mitra, “Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp. 182–190, Apr. 1996.

[31] A. Schuster and A. Katsaggelos, “Fast and efficient mode and quantizer selection in the rate-distortion sense for H.263,” *SPIE Proc. Visual Communications Image Processing*, vol. 2727, pp. 784–795, 1996.

[32] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Mag.*, vol. 15, pp. 23–50, Nov. 1998.

[33] G. J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Mag.*, vol. 15, pp. 74–90, Nov. 1998.

[34] G. Côté and F. Kossentini, “Optimal intra coding of blocks for robust video communication over the internet,” *Signal Processing: Image Communication, Special Issue Real-time Video over Internet*, vol. 15, pp. 25–34, Sept. 1999.

[35] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 966–976, June 2000.

[36] D. Wu, Y. T. Hou, B. Li, W. Zhu, Y.-Q. Zhang, and H. J. Chao, "An end-to-end approach for optimal mode selection in internet video communication: Theory and application," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 977–995, June 2000.

[37] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, "Long-term memory motion-compensated prediction for robust video transmission," in *Proc. IEEE International Conference on Image Processing*, vol. 2, 2000, pp. 152–155.

[38] T. Stockhammer, T. Wiegand, and S. Wenger, "Optimized transmission of H.26L/JVT coded video over packet-lossy networks," in *Proc. IEEE International Conference on Image Processing*, vol. 2, Sept. 2002, pp. 173–176.

[39] G. Côté, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 952–965, June 2000.

[40] S. Cen and P. Cosman, "Comparison of error concealment strategies for MPEG video," in *Proc. IEEE Wireless Communications and Networking Conference*, vol. 1, 1999, pp. 329–333.

[41] P. Salama, N. B. Shroff, and E. J. Delp, "Error concealment in MPEG video streams over ATM networks," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 1129–1144, June 2000.

[42] J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled lagrange multiplier method," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 546–558, Sept. 1994.

[43] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 7, pp. 849–865, Nov. 1998.

[44] ITU-T Recommendation H.263 Version 2 ("H.263+"), "Video coding for low bit rate communication," Jan. 1998.

[45] V. Chellappa, P. C. Cosman, and G. M. Voelker, "Dual frame motion compensation with uneven quality assignment," in *Proc. IEEE Data Compression Conference*, Mar. 2004.

[46] J. Reichel, H. Schwarz, and M. Wien, "Scalable Video Coding Working Draft 1," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-N020, Jan. 2005.

[47] W. Li, “Overview of fine granularity scalability in MPEG-4 video standard,” *IEEE Trans. on CSVT*, vol. 11, no. 3, pp. 301–317, Mar. 2001.

[48] F. Wu, S. Li, and Y.-Q. Zhang, “A framework for efficient progressive fine granularity scalable video coding,” *IEEE Trans. on CSVT*, vol. 11, no. 3, pp. 332–344, Mar. 2001.

[49] X. Sun, F. Wu, S. Li, W. Gao, and Y.-Q. Zhang, “Macroblock-based progressive fine granularity scalable video coding,” in *Proc. IEEE Int. Conf. on Mul. and Ex.*, 2001, pp. 461–464.

[50] Y. He, F. Wu, S. Li, Y. Zhong, and S. Yang, “H.26L-based fine granularity scalable video coding,” in *Proc. IEEE ISCAS*, vol. IV, 2002, pp. 548–551.

[51] F. Wu, S. Li, B. Zeng, and Y.-Q. Zhang, “Drifting reduction in progressive fine granularity scalable video coding,” in *Proc. Picture Coding Symposium*, Apr. 2001.

[52] C. Zhu, Y. Gao, and L.-P. Chau, “Reducing drift for FGS coding based on multi-frame motion compensation,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[53] Y. Zhou, X. Sun, F. Wu, H. Bao, and S. Li, “Flexible P-picture (FLEXP) coding for the efficient fine-granular scalability (FGS),” in *Proc. IEEE ICIP*, vol. 3, Oct. 2004, pp. 2071–2074.

[54] J. Ascenso and F. Pereira, “Drift reduction for a H.264-AVC fine grain scalability with motion compensation architecture,” in *Proc. IEEE International Conference on Image Processing*, vol. 4, Oct. 2004, pp. 2259–2262.

[55] S. Han and B. Girod, “Robust and efficient scalable video coding with leaky prediction,” in *Proc. IEEE ICIP*, 2002.

[56] V. Bocca, M. Fumagalli, R. Lancini, and S. Tubaro, “Accurate estimate of the decoded video quality: Extension of ROPE algorithm to half-pixel precision,” in *Proc. Picture Coding Symposium*, Dec. 2004.

[57] A. Chang, O. C. Au, and Y. M. Yeung, “A novel approach to fast multi-frame selection for H.264 video coding,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, May 2003, pp. 413–416.

[58] G. Pau, B. Pesquet-Popescu, M. van der Schaar, and J. Vieron, “Delay-performance trade-offs in motion-ccompensated scalable subband video compression,” in *Proc. Advanced Concepts for Intelligent Vision Systems*, Sept. 2004.

[59] G. Pau, J. Vieron, and B. Pesquet-Popescu, “Video coding with flexible MCTF structures for low end-to-end delay,” in *Proc. IEEE Int. Conf. on Image Proc.*, Sept. 2005.

[60] H. Schwarz, D. Marpe, and T. Wiegand, “Comparison of MCTF and closed-loop hierarchical B pictures,” Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-P059, July 2005.

[61] B. Girod, “The efficiency of motion-compensating prediction for hybrid coding of video sequences,” *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, no. 7, pp. 1140–1154, Aug. 1987.

[62] ——, “Motion-compensating prediction with fractional-pel accuracy,” *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604–612, Apr. 1993.

[63] M. Isnardi, “MPEG-2 video compression,” SMPTE Tutorial Overview, Sarnoff Corporation, Nov. 1999.

[64] T. Wiegand, “Final draft international standard for joint video specification H.264,” JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, Mar. 2003.

[65] M. Karczewicz and Y. Bao, “Need for further AVC test model enhancements,” Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-L034, July 2004.

[66] 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, Sept. 1994.

[67] M. Wang and M. van der Schaar, “Rate-distortion modeling for wavelet video coders,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Mar. 2005, pp. 53–56.

[68] N. Farvardin and J. W. Modestino, “Rate-distortion performance of DPCM schemes for autoregressive sources,” *IEEE Transactions on Image Processing*, vol. 31, no. 3, pp. 402–418, May 1985.

[69] P. Ramanathan and B. Girod, “Rate-distortion analysis for light field coding and streaming,” *EURASIP Signal Processing: Image Communication*, Nov. 2005, submitted.

[70] U. Horn, T. Wiegand, and B. Girod, “Bit allocation methods for closed-loop coding of oversampled pyramid decompositions,” in *Proc. IEEE International Conference on Image Processing*, vol. 2, Oct. 1997, pp. 17–20.

[71] M. Flierl and B. Girod, “Video coding with motion compensation for groups of pictures,” in *Proc. IEEE International Conference on Image Processing*, vol. 1, Sept. 2002, pp. 69–72.

[72] K.-P. Lim, G. Sullivan, and T. Wiegand, “Text description of joint model reference encoding methods and decoding concealment methods,” Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-K049, Mar. 2004.