Transforms for Prediction Residuals in Video Coding

by

Fatih Kamışlı

S.M., Massachusetts Institute of Technology (2006)
B.S., Middle East Technical University (2003)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2010

Author .....
Department of Electrical Engineering and Computer Science
August 6, 2010

Certified by .....
Jae S. Lim
Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Terry P. Orlando
Chairman, Departmental Committee on Graduate Students

Transforms for Prediction Residuals in Video Coding

by

Fatih Kamışlı

Submitted to the Department of Electrical Engineering and Computer Science
on August 6, 2010, in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Typically the same transform, the 2-D Discrete Cosine Transform (DCT), is used to compress both image intensities in image coding and prediction residuals in video coding. Major prediction residuals include the motion compensated prediction residual, the resolution enhancement residual in scalable video coding, and the intra prediction residual in intra-frame coding. The 2-D DCT is efficient at decorrelating images, but the spatial characteristics of prediction residuals can be significantly different from the spatial characteristics of images, and developing transforms that are adapted to the characteristics of prediction residuals can improve their compression efficiency.

In this thesis, we explore the differences between the characteristics of images and prediction residuals by analyzing their local anisotropic characteristics and develop transforms adapted to the local anisotropic characteristics of some types of prediction residuals. The analysis shows that local regions in images have 2-D anisotropic characteristics and many regions in several types of prediction residuals have 1-D anisotropic characteristics. Based on this insight, we develop 1-D transforms for these residuals. We perform experiments to evaluate the potential gains achievable from using these transforms within the H.264 codec, and the experimental results indicate that these transforms can increase the compression efficiency of these residuals.

Thesis Supervisor: Jae S. Lim
Title: Professor of Electrical Engineering

Acknowledgments

Many people have helped make this thesis possible, and I would like to acknowledge their contribution.

First, I would like to thank my academic and thesis supervisor Professor Jae Lim for his supervision and contribution to this thesis. I am very grateful to him for providing me a place in his lab, for the intellectual and financial support he has provided throughout my thesis, and for the advice he has given me about both research and life. I would also like to thank Dr. John Apostolopoulos and Professor Vivek Goyal for serving on my thesis committee. Their feedback and suggestions have helped to improve the quality of this thesis.

Throughout my time in the ATSP group I had the chance to meet many fellow students, researchers and past graduates. During the years I have spent in the group, many students have joined and left the group, such as Brian Heng, Zhenya Gu, Andy Lin and Han Wang, and I would like to thank them for their friendship. I also enjoyed my interactions with visiting scientists Dr. Heung-Nam Kim and Professor Chang Dong Yoo. Special thanks go to Cindy LeBlanc for all her help with so many different things and for making my life here much easier.

I would also like to thank David Baylon for all his support and help during my summer at Motorola. He has been a great friend and mentor willing to lend a hand anytime. Similarly, I would also like to thank Yan Ye for her kindness and help during my summer at Qualcomm. Special thanks also to Wade Wan for his advice, time and effort to help me.

My friend Mutlu has been a great comrade during my time in Boston. We have shared so many things, and I would like to thank him for his friendship, support and for all we have shared. Special thanks for his Thanksgiving dinners – I must admit he is getting better at it every year.

My parents and siblings have been a source of support and love. I would like to thank my parents, Mehmet and Semiha, my siblings Sema, Filiz, Remzi and my newborn nephew Berkay for all their support and faith.

Finally, I am very fortunate to have the support and love of my girlfriend Başak. Living apart many years has been extremely difficult for both of us, and I am deeply indebted to her for her understanding and patience. Her support and encouragement have been an invaluable source of strength at difficult times, and her understanding and love have made this achievement possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One main objective of a video compression system is to represent a video sequence with as few bits as possible while keeping a sufficient level of image quality for a given application. To achieve this goal, encoders exploit redundancy and irrelevancy present in video information. For example, two adjacent frames in a video sequence are likely to be very similar since typical sequences do not change rapidly from one frame to the next. To reduce redundancy in the representation of the video information, encoders use a number of tools which can be grouped into three categories; prediction, transform and entropy coding. Prediction tools predict a local region in the current frame from previously encoded frames or from previously encoded regions of the current frame. For example, temporally adjacent frames are typically highly correlated and motion-compensated prediction is widely used to predict a local region in the current frame from previously encoded frames. The prediction is in many instances not accurate enough and the prediction residual (error) is coded. The prediction residual typically contains some spatial redundancy and a transform is used to remove as much of this redundancy as possible. Finally, the transform coefficients are quantized and entropy coded.

This thesis focuses primarily on the transform component of video compression systems. The most widely used transform in video compression is the 2-D Discrete Cosine Transform (DCT). The 2-D DCT was initially used for image compression, where the prediction component is omitted and the transform is applied directly on the image intensities. Typical images contain many regions with smoothly varying intensity and the 2-D DCT can remove spatial redundancy from such regions well, as a small number of

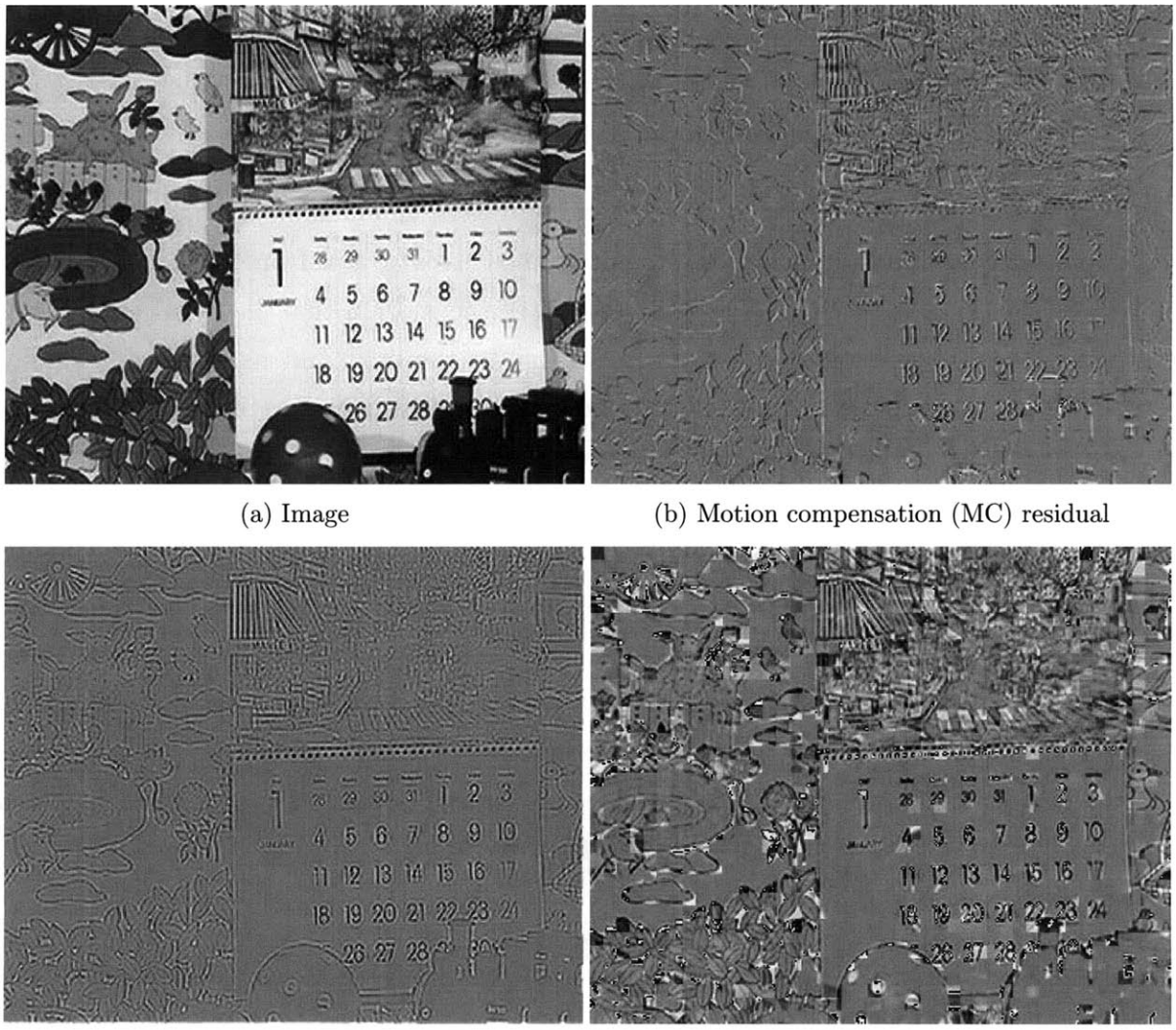
its smoothly varying basis functions is typically sufficient to represent such regions with adequate quality.

The 2-D DCT has also been widely adopted for video compression where it is used to remove spatial redundancy from prediction residuals. Prediction residuals can have significantly different spatial characteristics from images. They are typically not as smooth as images since many prediction methods can predict smooth parts of images well and the prediction errors from the remaining non-smooth parts form a major portion of prediction residuals. Unlike images, the energy of prediction residuals is not distributed broadly in the spatial domain but is concentrated in regions which are difficult to predict. Figure 1-1 shows an image and several different types of prediction residuals. The 2-D DCT may not perform well in many regions of prediction residuals and this thesis investigates the differences between the characteristics of images and prediction residuals and proposes new transforms that are adapted to the characteristics of some types of prediction residuals.

The next section provides a brief introduction to predictive coding of video, introducing different types of prediction methods and residuals. In the second section, we discuss the motivation behind this research and the following section presents an overview of this thesis.

1.1 Predictive Coding of Video

There is a considerable amount of redundant information present in a typical video sequence. In the temporal dimension, two adjacent frames are likely to be highly correlated since typical sequences do not change rapidly from one frame to the next. Similarly, within a single frame each pixel is likely to be correlated with neighboring pixels since most frames contain many regions of smoothly varying intensity. Temporal and spatial redundancies are the major redundancies present in all video sequences, but additional redundancies can be present in different video coding applications. For example, in some video coding applications it is desirable to encode the video information in such a way that a subset of the stream can be used to decode the video at a low spatial resolution and the entire stream can be used to decode the video at a high spatial resolution. In this type of applications, each frame in the high spatial resolution stream is highly cor-



(a) Image

(b) Motion compensation (MC) residual

(c) Resolution enhancement (RE) residual

(d) Intra prediction (IP) residual

Figure 1-1: Frame 10 of mobile sequence at CIF resolution, its MC residual predicted from frame 9 using ful-pel motion estimation with 8x8-pixel blocks, its RE residual obtained from its QCIF resolution version, and its IP residual obtained using 8x8-pixel intra prediction modes in H.264/AVC.

related with the temporally corresponding frame in the low spatial resolution stream. Another type of redundancy is present in 3-D video coding applications, where a scene is captured with two or more closely spaced cameras. Each frame captured with any of the cameras is likely to be highly correlated with the temporally corresponding frames captured with the other cameras. This section continues with a discussion of each type of redundancy, prediction methods used to reduce them, and the prediction residuals obtained from these prediction methods.

To reduce the correlation along the temporal dimension, most video coders use motion compensation to predict a local region in the current frame from previously encoded frames. In this approach, the encoder estimates the motion that the local region, typically a block, has experienced from one frame to the next and uses this information to generate a prediction of the local region by compensating for the motion that has occurred. This approach assumes that the motion is locally translational and it works well in many regions. For example, a typical video sequence contains many stationary regions and this approach works well in such regions. Even in moving regions that are smooth, this approach works well due to the high spatial correlation and the prediction residual in such regions is typically small. However, in moving regions that are not smooth, such as around edges, object boundaries or detailed texture regions, the prediction typically contains large errors. Such regions can be easily seen in Figure 1-1 (b). The residual signal in such regions is concentrated in a fraction of the pixels in local regions and the spatial characteristics of the residual is significantly different from the spatial characteristics of images.

In some video coding applications, it may be desirable to encode the video into multiple streams, referred to as layers. A so-called base layer is coded independent of other layers and contains basic video information. One or more enhancement layers are coded dependent on the base layer and previous enhancement layers (if any), and contain information that can enhance the video to achieve improved quality, frame rate, and/or spatial resolution. In this type of coding approach, the quality of the video scales gracefully with the addition of each enhancement layer, and this type of coding approach is called scalable video coding.

Consider scalable video coding with spatial scalability and one enhancement layer. If a high resolution video sequence is encoded in this way, decoders which support

only standard resolution can decode the base layer and display a standard resolution video, and decoders which support high resolution can also decode the enhancement layer and display a high resolution video. Each frame in the high resolution video is highly correlated with the temporally corresponding frame in the standard resolution video. This correlation is reduced by predicting a frame in the high resolution video through interpolation (upsample and low-pass filter) of the temporally corresponding frame in the standard resolution video. The prediction is typically not accurate and the prediction residual is called resolution enhancement residual. Interpolation works well in smooth regions of the frame and the prediction residual in such regions is typically negligibly small. In other regions which have rapidly changing intensity, such as edges, object boundaries or highly detailed texture regions, interpolation can result in poor prediction and the prediction residual can be large. Such regions with large prediction errors can be easily seen in Figure 1-1 (c). Again, the energy of the residual signal is concentrated in a fraction of pixels in such local regions and the spatial characteristics of the residual is different from the spatial characteristics of images.

In all video coding applications, some frames are not predicted from other frames and are coded independently of other frames. These frames are called I-frames (intra frames) and can provide random access capability into the video bitstream. For example, in a digital television application, if a user switches the channel, the decoder can wait for the next I-frame and decode the bitstream with this I-frame. Prior to the recent video coding standard H.264/AVC, coding of I-frames was similar to coding of images, where the transform is applied directly to image intensities. In H.264/AVC, each block is predicted from reconstructed pixels of previously coded neighboring blocks within the same frame, and the transform is applied on the spatial prediction residual [52]. This type of prediction is called intra prediction (or intra-frame prediction) and the residual is called intra prediction residual. Similar to previously discussed prediction methods, intra prediction works well in smooth regions of a frame but can produce large prediction errors in detailed regions such as textures, edges or object boundaries. Characteristics of prediction residuals in such regions can be different from characteristics of images in such regions. Figure 1-1 (a) and (d) show a frame and its intra prediction residual.

In 3-D video coding, the viewer is provided with visual information that is capable of creating depth perception, in addition to the perception of movement present in regular video coding. One way to create depth perception in the brain is to provide the eyes of

the viewer with two different images, representing two perspectives of the same scene, with a minor deviation similar to the perspectives that both eyes naturally receive. 3-D video coding is based on this principle and multiple closely spaced cameras, such as a linear array of cameras [47], are used to acquire the scene from multiple perspectives. Each frame captured with any of the cameras is likely to be highly correlated with the temporally corresponding frames from neighboring cameras. The correlation between two such frames can be similar to the correlation between two temporally adjacent frames in regular video coding and correlation reduction methods are influenced by motion compensated prediction methods and are known as disparity compensated prediction methods.

In summary, different types of redundancies exist in video information and different types of prediction methods are used to reduce them. Typically these methods work well in smooth regions of frames where spatial correlation is high. Around edges, object boundaries or detailed texture regions, where intensities change rapidly, the prediction residuals can be large. The spatial characteristics of prediction residuals in such regions can be significantly different from spatial characteristics of images. Unlike in images, the energy of prediction residuals in such regions is typically distributed highly unevenly. Many pixels in such regions have negligibly small amplitude because they were predicted well, and pixels with large intensities are not randomly scattered but typically have some spatial structure depending on the specific characteristics of the local region and the prediction method. Transforms that are adapted to these structures are likely to improve the compression efficiency of such regions over the conventionally used 2-D DCT.

1.2 Motivation for Thesis

An important component of image and video compression systems is a transform. A transform is used to transform image intensities. A transform is also used to transform prediction residuals of image intensities, such as the motion compensation (MC) residual, the resolution enhancement (RE) residual in scalable video coding, or the intra prediction (IP) residual in H.264/AVC. Typically, the same transform is used to transform both image intensities and prediction residuals. For example, the 2-D Discrete Cosine

Transform (2-D DCT) is used to compress image intensities in the JPEG standard and MC residuals in many video coding standards. Another example is the 2-D Discrete Wavelet Transform (2-D DWT), which is used to compress images in the JPEG2000 standard and high-pass prediction residual frames in inter-frame wavelet coding [30]. However, prediction residuals have different spatial characteristics from image intensities [18, 16, 9, 28]. This thesis analyzes the differences between the characteristics of images and several types of prediction residuals, and proposes new transforms that are adapted to the characteristics of some types of prediction residuals.

Recently, new transforms that can take advantage of locally anisotropic features in images have been developed [48, 20, 55, 46, 8]. A conventional transform, such as the 2-D DCT or the 2-D DWT, is carried out as a separable transform by cascading two 1-D transforms in the vertical and horizontal dimensions. This approach favors horizontal or vertical features over others and does not take advantage of locally anisotropic features present in images. For example, the 2-D DWT has vanishing moments only in the horizontal and vertical directions. The new transforms adapt to locally anisotropic features in images by performing the filtering along directions where image intensity variations are smaller. This is achieved by resampling the image intensities along such directions prior to a separable transform [20], by performing filtering and subsampling on oriented sublattices of the sampling grid [46], by directional lifting implementations of the DWT [8], or by various other means. Even though most of the work is based on the DWT, similar ideas have been applied to DCT-based image compression [55].

In video coding, prediction residuals of image intensities are coded in addition to image intensities. Many transforms have been developed to take advantage of local anisotropic characteristics of images, however, local anisotropic characteristics of prediction residuals have not been investigated. There are many unanswered questions to this end. For example, are locally anisotropic features also present in prediction residuals? If they are, are they different from the ones in images? How different are they? How would transforms be adapted to these features? What gains could be achievable by adapting the processing/transforms to these features? These are some of the questions that have motivated this work.

Inspection of prediction residuals shows that locally anisotropic features are also present in prediction residuals, yet can have different characteristics from the ones in

images. Unlike in images, a large number of pixels within a local region in prediction residuals have negligibly small amplitudes because they were predicted well. Pixels with large amplitudes concentrate in regions which are difficult to predict. For example, in MC residuals such regions are moving object boundaries, edges, or highly detailed texture regions. Since prediction typically removes the more easily predictable parts of the image signal, the remaining residual signal contains the parts which are difficult to predict, and the spatial characteristics of this signal are different from the characteristics of the original image signal. Transforms which were developed for the original image signal are not efficient for the residual signal, and new transforms adapted to the characteristics of the residual signal are desirable.

Our main goal in this thesis is to explore and utilize the different characteristics of images and prediction residuals. We first analyze the local anisotropic characteristics of images and several different types of prediction residuals. Based on this analysis, we highlight a difference between the local anisotropic characteristics of images and two types of prediction residuals, and develop transforms adapted to these residuals. Experimental results indicate that these transforms can increase the compression efficiency of these residuals and encourage further studies in this direction.

1.3 Overview of Thesis

In the second chapter of this thesis, we provide a review of related previous research. The chapter begins with a survey of commonly used prediction methods in video coding and continues with a discussion of statistical characterizations for images and motion compensated prediction residuals, which consist of studies to differentiate the characteristics of these two signals. Next, a review of coding approaches used to code prediction residuals is presented. Finally, recently developed direction-adaptive transforms that exploit local anisotropic characteristics in images are summarized.

To develop transforms adapted to prediction residuals it is essential to study the characteristics of prediction residuals, and Chapter 3 analyzes the local anisotropic characteristics of several different types of prediction residuals. In particular, a significant difference between the local anisotropic characteristics of images, and MC and RE residuals is highlighted using both visual inspections and statistical analysis of these signals.

Based on the results of the analysis, we propose new transforms for MC and RE residuals in Chapter 4. A sample set of such transforms is provided and discussed for 4x4 and 8x8-pixel blocks.

To examine the performance of the proposed transforms within an actual codec, a number of related aspects need to be carefully designed and we discuss these aspects in Chapter 5. These aspects include coding of the quantized transform coefficients, coding of the side information which indicates the selected transforms for each local region, rate-distortion optimized selection of transforms, and the overall increase in the complexity of the codec.

Chapter 6 presents experimental results to illustrate the compression efficiency of the proposed transforms. We compare encoders with conventional transforms with encoders which have access to both conventional transforms and a set of the proposed transforms. The results of the experiments demonstrate that the proposed transforms can improve the coding efficiency. We present achievable bitrate savings averaged over a range of picture qualities as well as savings specific to lower and higher picture qualities. We also provide other useful information from these experiments that can help understand and improve systems with the proposed transforms.

Finally, Chapter 7 summarizes the thesis and describes possible future research directions.

Chapter 2

Previous Research

In this chapter, we review previous research related to this thesis. In the first section, we provide a survey of prediction methods used in video coding including motion compensation (MC), resolution enhancement (RE), and intra prediction (IP). To develop transforms for prediction residuals, understanding the characteristics of prediction residuals is important and we discuss characterizations of some prediction residuals from the literature in Section 2.2. Section 2.3 reviews commonly used methods to code prediction residuals and presents a summary of methods proposed to improve the coding of some types of prediction residuals. In Section 2.4, we summarize some of the important approaches to developing direction-adaptive transforms for images. These approaches take advantage of local anisotropic features in images. Finally, Section 2.5 discusses how the research in this thesis is related to the above mentioned prior research and how it extends them.

2.1 Prediction Methods

Typical video sequences contain a significant amount of redundant information. In the temporal dimension, two adjacent frames are likely to be highly correlated since typical sequences do not change rapidly from one frame to the next. Within a single frame each pixel is likely to be highly correlated with spatially neighboring pixels since most frames contain many regions of smoothly varying intensity. Temporal and spatial redundancies are the major redundancies present in all video sequences, but additional redundancies

can be present in different video coding applications. For example, in scalable video coding with spatial scalability, it is desirable to encode the video information in such a way that a subset of the stream can be used to decode the video at a low spatial resolution and the entire stream can be used to decode the video at a high spatial resolution. Each frame in the high spatial resolution stream is highly correlated with the temporally corresponding frame in the low spatial resolution stream.

To study the differences of characteristics between images and prediction residuals, it is essential to understand the redundancies present in different video coding applications and the prediction methods to exploit them. This section continues with a more detailed discussion of each type of redundancy, prediction methods used to exploit them, and the prediction residuals obtained from these prediction methods.

2.1.1 Motion Compensation (MC)

To reduce the correlation along the temporal dimension, most video coders use motion compensation to predict a local region in the current frame from previously encoded frames. In this approach, the encoder estimates the motion that the local region has experienced from one frame to the next and uses this information to generate a prediction of the local region by compensating for the motion that has occurred. The most widely used method for motion compensation is block matching. In this method, the current frame is divided into blocks and each block is predicted using the best matching block from a previously encoded frame. The best matching block is indicated to the decoder by transmitting the displacement, often called motion vector, between the block and its best match. To improve the effectiveness of motion compensated prediction from this basic scheme [14, 15], a significant amount of research has been performed. The most important improvements include fractional-pixel MC [13], multihypothesis MC [12, 11], loop filtering [33, 21], overlapped block MC [29, 3, 31] and variable block-size MC [35, 7, 39].

In block matching, the best match can be obtained from a block displaced by integer multiples of the sampling intervals, and this prediction method is called integer-pixel accuracy MC. Since the true motion between frames is unrelated to the sampling intervals, it is expected that the prediction can be improved if fractional-pixel accuracy displacements are used. This improvement is referred to as the accuracy effect in [13]. Typically,

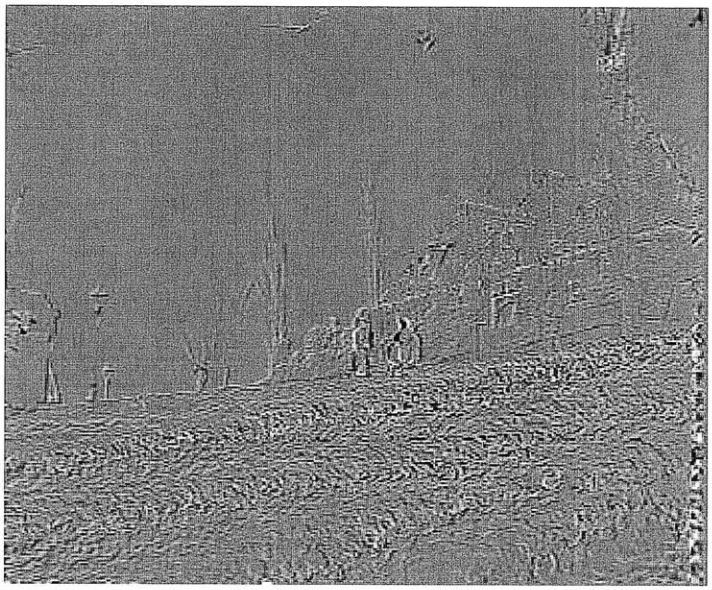
fractional-pixel accuracy prediction is achieved with interpolation using spatial low-pass filtering such as bilinear interpolation. It has been reported that the use of a spatial low pass filter in the predictor can improve the prediction also with integer-pixel accuracy, referred to as the filtering effect [13]. Thus the improvement in the prediction with fractional-pixel accuracy is reported to have two contributors; the accuracy effect and the filtering effect.

State-of-the-art video codecs such as H.264/AVC use quarter-pixel accuracy MC. Figure 2-1 shows MC residuals obtained with integer-pixel accuracy and quarter-pixel accuracy, and it can be observed that the residual obtained with quarter-pixel accuracy MC has smaller energy and thus will require less amount of bits to encode. It is reported in [50] that quarter-pixel MC in H.264/AVC can obtain bitrate savings up to 30% compared to integer-pixel MC. These gains come, of course, at a computational cost. Fractional-pixel values need to be computed and searched.

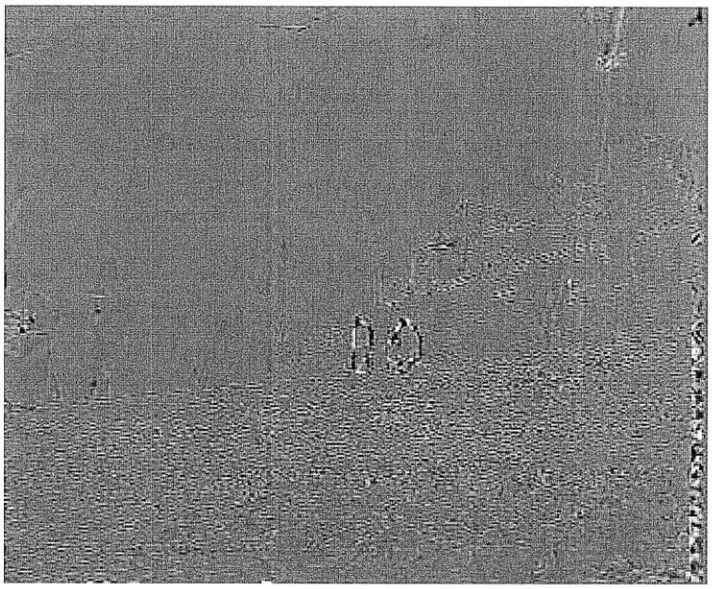
It is possible to obtain multiple motion compensated predictions simultaneously for the current block and use a linear combination of them to form the final prediction signal. This approach is called multihypothesis motion compensation. An example is the B-frame in MPEG-2 where predictions from previously encoded past and future frames can be averaged to predict the current frame. Multihypothesis motion compensation can provide significant coding gains. Experimental results as well as theoretical motivations behind these approaches can be found in [12, 11, 15].

The block-based MC and the block-based 2-D DCT can cause visually annoying artifacts, known as blocking effects, in low bit-rate video coding applications. Blocking effects are visually annoying discontinuities across block boundaries. These discontinuities can also propagate into the interiors of blocks with motion compensated prediction. Figure 2-2 (a) shows a frame which has blocking effects. Reducing blocking effects, often called deblocking, improves the perceived visual quality, as shown in Figure 2-2 (b).

There are two main approaches to reduce blocking effects in video codecs; post filtering and loop filtering [33, 21, 23]. In the post filtering approach, deblocking is performed only at the decoder. Each reconstructed frame is deblocked and displayed. In the loop filtering approach, deblocking is performed both at the encoder and the decoder within the prediction loop. At the encoder, each coded frame is deblocked before it is stored in memory for motion compensated prediction of future frames. At



(a) Integer-pixel accuracy motion compensation residual.



(b) Quarter-pixel accuracy motion compensation residual.

Figure 2-1: Integer and fractional-pixel motion compensation residuals with 8x8 blocks.

(a) Frame with blocking effects. Reconstructed without the loop filter.

(b) Frame with reduced blocking effects. Reconstructed with the loop filter.

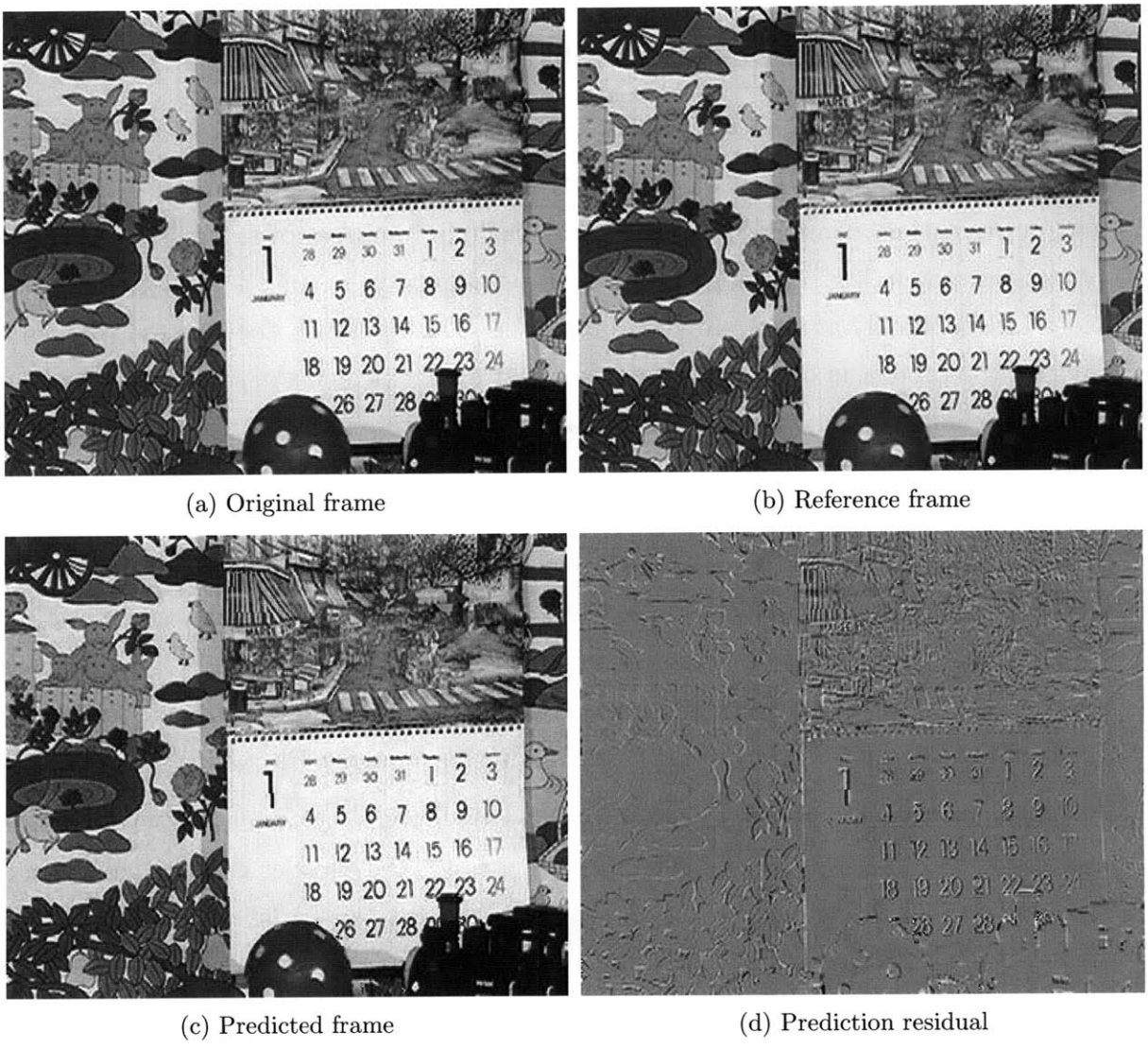
Figure 2-2: Reconstructed frame without and with the loop filter in H.264/AVC.

the decoder, each decoded frame is deblocked, and the deblocked frame is displayed and stored in memory for motion compensated prediction of future frames. Both the encoder and the decoder need to use the same deblocking algorithm so that their prediction loops stay synchronized. Reducing blocking effects provides smoother frames and use of such frames for motion compensated prediction is likely to improve the prediction. Loop filtering can improve the subjective and the objective quality of reconstructed video and is an important tool for low bitrate video coding applications. Figure 2-2 compares reconstructed frames with and without the loop filter in H.264/AVC.

In overlapped block MC, overlapping blocks are used for MC and the blocks are weighted by a smooth window. Use of overlapping blocks with smoothly decaying boundaries can provide smoother predictions and help reduce blocking effects. In addition, due to the overlap, this approach allows to predict each pixel in the current frame using a linear combination of multiple pixels from a previously encoded frame and therefore overlapped block MC is another example of multihypothesis motion compensation [15]. Overlapped block MC can provide coding gains as well as reduce blocking effects [29, 3, 31].

Block-based MC uses the same motion vector for all pixels in a block. However, the true displacements of individual pixels within the block may be slightly different. Transmitting one motion vector for every pixel would require too many bits. One possibility to achieve higher efficiency in this trade-off is to vary the block size adaptively [35, 7, 39]. The optimum size may depend on many factors including the the video content, the amount of motion and the compression ratio. Available block sizes in H.264/AVC are 16x16, 16x8, 8x16 and 8x8, where 8x8 blocks can be further divided into 8x4, 4x8 or 4x4 blocks.

Figure 2-3 shows an original frame, the reference frame used to predict the original frame, the predicted frame, and the prediction residual. The prediction works well in smooth regions and the prediction residual in such regions is negligibly small. In texture regions, such as the calendar picture, the prediction does not work as well as in smooth regions. Around object boundaries or edges, such as the boundary of the ball, the boundary of objects in the background or the numbers on the calendar, the prediction residual is typically large and the energy of prediction residuals in such regions is not distributed throughout the block but is concentrated in a fraction of the pixels within a



(a) Original frame

(b) Reference frame

(c) Predicted frame

(d) Prediction residual

Figure 2-3: Motion compensated prediction. An original frame (Mobile sequence at CIF resolution, frame 10), the reference frame used to predict the original frame (Mobile sequence at CIF resolution, frame 9), the predicted frame (using 8x8-pixel block integer pixel MC), and the prediction residual are shown.

block. These pixels typically follow the boundaries or edges present in the original frame and this property is an important difference between images and motion compensated prediction residuals.

2.1.2 Resolution Enhancement (RE)

In some video coding applications, it may be desirable to have the capability to transmit video information at different quality levels, such as different picture quality, frame rate and/or resolution, commensurate with the available resources of the receivers, such as processing power and/or bandwidth. One possibility is to encode and transmit the video independently at all required quality levels. This approach does not exploit similar information present in other representations and is not efficient. An alternative approach is to use scalable coding, where the video is encoded into multiple streams, called layers. A base layer is coded independent of other layers and contains video information at a basic quality level. One or more enhancement layers are coded dependent on the base layer and previous enhancement layers (if any), and contain information that can enhance the video to achieve improved picture quality (Quality Scalability), frame rate (Temporal Scalability), and/or spatial resolution (Spatial Scalability). A decoder can decode the base layer and achieve basic video quality, or a decoder can decode the base layer and one or more enhancement layers and achieve improved quality.

The encoding and decoding of the base layer operates in the same manner as regular single layer coding. To encode the enhancement layer, the encoder predicts the video at the improved quality from the encoded base layer (or previous layer) video and codes the prediction residual. For example, in temporal scalable coding the enhancement layer contains additional frames that increase frame rate, as shown in Figure 2-4. These frames are predicted from the temporally adjacent frames in the base layer using motion compensated prediction and the prediction residual is coded. In spatial scalable coding, the enhancement layer contains information that can enhance the resolution, as shown in Figure 2-5. The high resolution frames are predicted from the temporally corresponding frames in the base layer through interpolation and the prediction residual is coded. This type of prediction residual is termed resolution enhancement residual.

Figure 2-6 shows an original frame, the low resolution reference frame used to predict the original frame, the predicted frame, and the prediction residual. The prediction

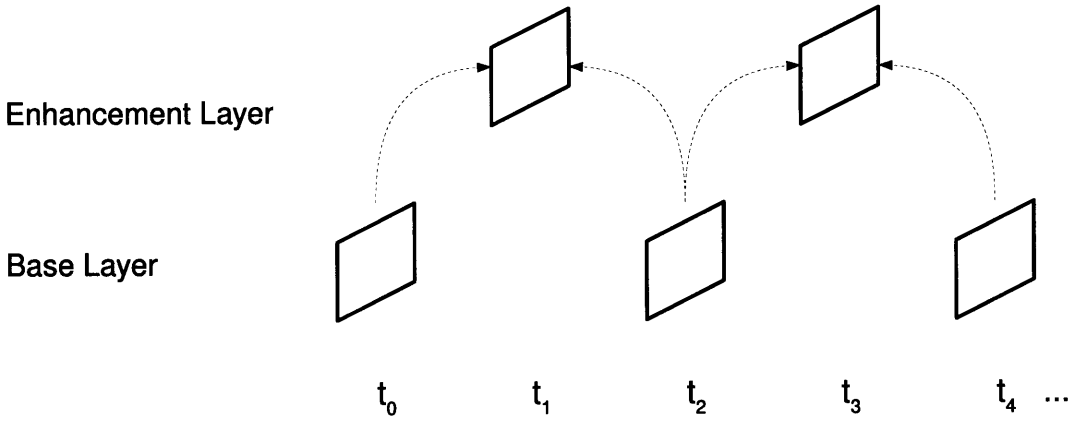


Figure 2-4: Temporal scalability with two layers. The enhancement layer doubles the frame rate and the frames in the enhancement layer are predicted using motion compensated prediction from the base layer.

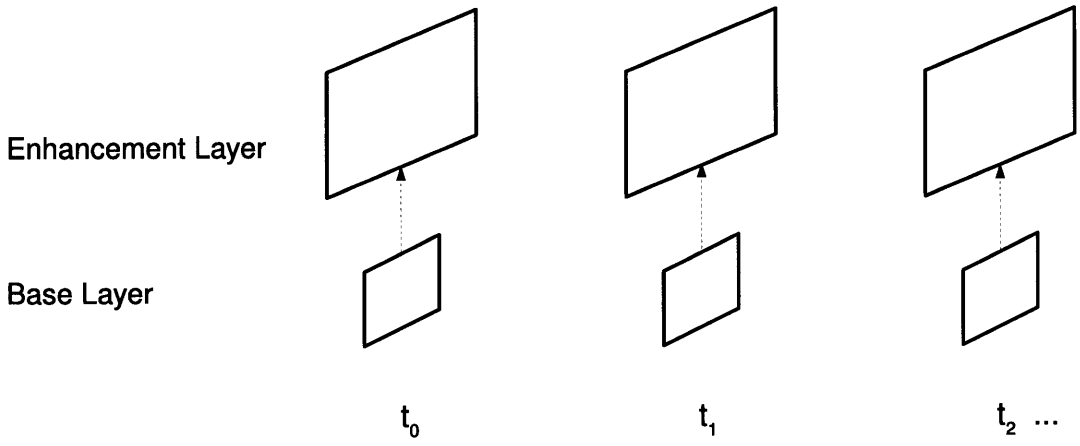
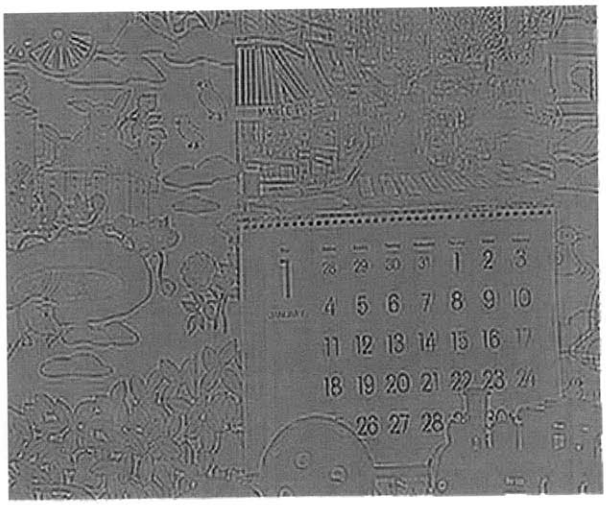


Figure 2-5: Spatial scalability with two layers. The enhancement layer doubles the resolution and the frames in the enhancement layer are predicted through interpolation of base layer frames.

(a) Original frame

(b) Reference frame

(c) Predicted frame



(d) Prediction residual

Figure 2-6: Prediction through interpolation. An original frame (Mobile sequence at CIF resolution, frame 10), the low resolution reference frame used to predict the original frame (Mobile sequence at QCIF resolution, frame 10), the predicted frame (using the interpolation algorithm in H.264/AVC), and the prediction residual are shown.

works well in smooth regions and the prediction residual in such regions is negligibly small. In texture regions, such as the calendar picture, the prediction does not work as well as in smooth regions. Around object boundaries or edges, such as the boundary of the ball, the boundary of objects in the background or the numbers on the calendar, the prediction residual is typically large and the energy of prediction residuals in such regions is not distributed throughout the block but is concentrated in a fraction of the pixels within a block. These pixels typically follow the boundaries or edges present in the original frame and this is an important difference between images and resolution enhancement residuals.

2.1.3 Intra Prediction (IP)

Some frames are not predicted from other frames and are coded independently. These frames are called I-frames (intra frames) and can provide random access capability into the video bitstream. For example, in a digital television application, if a user switches the channel, the decoder can not immediately start decoding because it does not have the previously decoded frames that are used for motion compensated prediction. Instead, the decoder needs to wait for the next I-frame and start decoding the bitstream with this I-frame. I-frames can also stop error propagation by synchronizing the video prediction loop.

In a single frame, each pixel is likely to be highly correlated with neighboring pixels since most frames contain many regions of smoothly varying intensity. To reduce this correlation one approach is to apply a decorrelating transform such as the 2-D DCT on a block of pixels. This approach has been widely used in many video coding standards and in the JPEG image compression standard [49]. An alternative approach is to predict the pixels of the current block from previously coded pixels of neighboring blocks. This approach is called intra prediction and is used in H.264/AVC, the most recent video coding standard. The prediction is typically not accurate and a prediction residual is coded by transforming the residual with a block transform. The intra prediction based approach can provide significant coding gains over the block based transform approach [2].

The prediction is generated using the correlation between the current block and the nearest pixels from previously encoded neighboring blocks. Figure 2-7 shows prediction

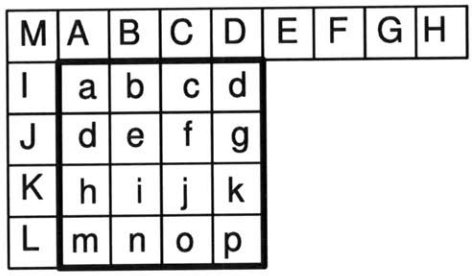
for a 4x4 block. Pixels of the current block are represented with small letters and pixels from neighboring blocks that can be used to generate the prediction are shown in capital letters. For example, if a vertical structure extends from the previously coded upper block to the current block, then the current block is predicted by copying downwards the nearest pixels in the upper block, as shown in Figure 2-8 (a). Similarly, a horizontal structure is predicted as shown in Figure 2-8 (b). In general, there a total of nine 4x4-block prediction modes and eight of them are directional prediction methods (shown in Figure 2-8 (c)) and one predicts the DC value of the block.

Similar to variable block size MC, intra prediction is also performed with varying block sizes. Available block sizes are 4x4, 8x8 (available only in high profile) and 16x16. A smaller block size is likely to result in better prediction but also a larger number of bits to indicate the selected prediction modes. Typically, 4x4-block or 8x8-block prediction is used in busy regions and 16x16-block prediction is used in smooth regions.

Figure 2-9 shows an original frame, the reference frame used to predict the original frame (in case of intra prediction the reference frame is equal to the original frame), the intra predicted frame, and the prediction residual. The prediction works well in smooth regions and the prediction residual in such regions is negligibly small. In texture regions, such as the calendar picture or the leaves in the southwest corner, the prediction can remove the local average value but many features are preserved. Around object boundaries or edges, such as the boundary of the ball or the boundary of objects in the background, the prediction residual is large and is either concentrated along the boundary or in one side of the boundary within a block. In such regions, the energy of the prediction residual is concentrated in a region within a block (which can have a spatial structure), and this can be an important difference between images and intra prediction residuals.

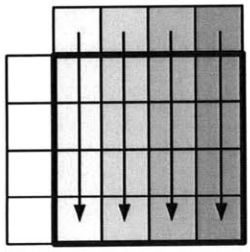
2.2 Characterization of the MC Residual

Section 2.1 introduced a number of redundancies present in video signals and commonly used prediction methods used to reduce them. The prediction is in many instances not accurate and a prediction residual needs to be coded. To code the prediction residuals efficiently, it is important to understand their characteristics. One commonly used ap-

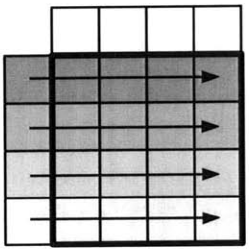


(a) Image

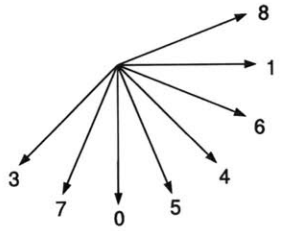
Figure 2-7: 4x4-block intra prediction. Pixels A-M can be used to predict the current block, pixels a-p.



(a) Mode 0.

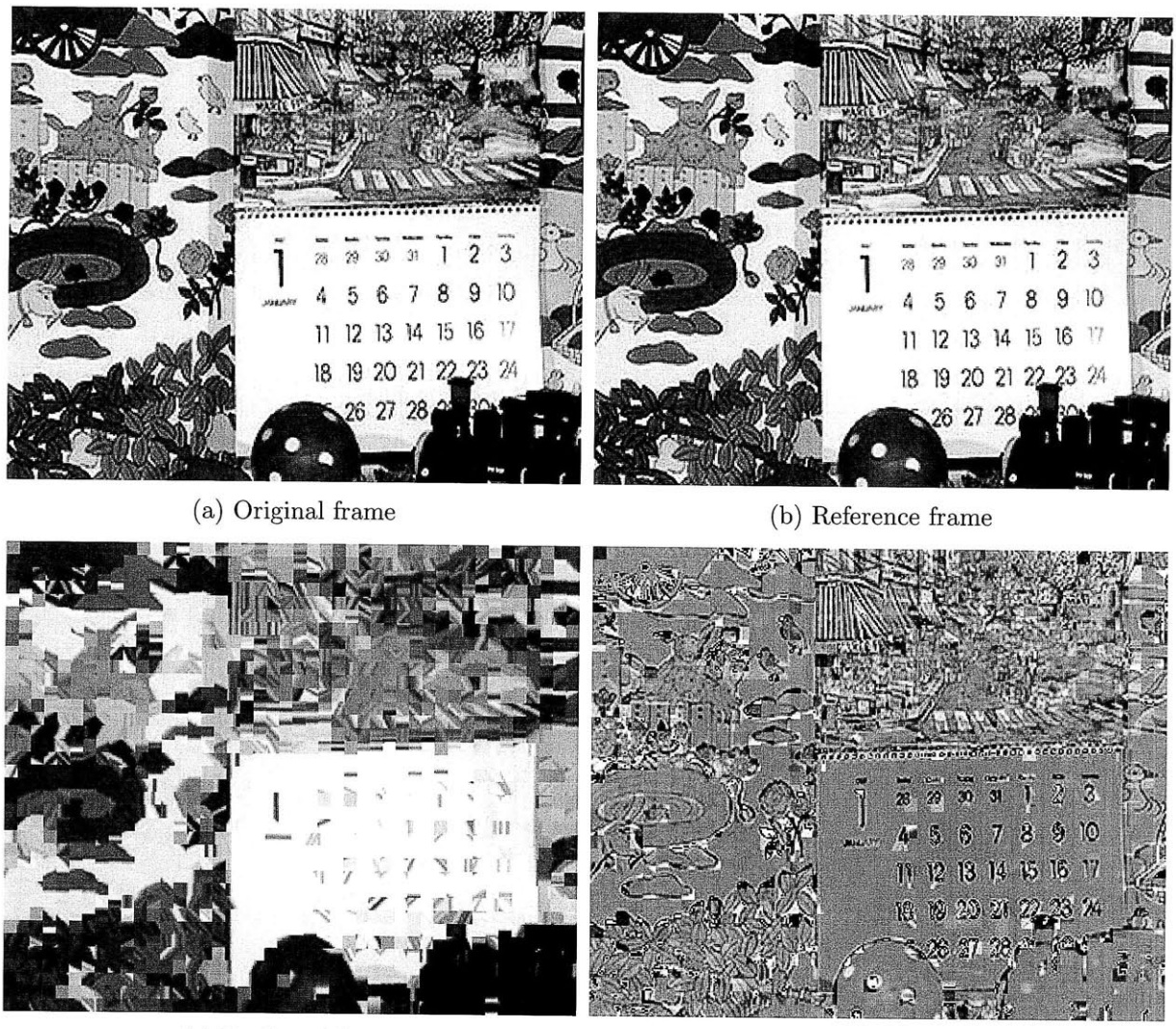


(b) Mode 1.



(c) Directions for prediction.

Figure 2-8: Available 4x4-block intra prediction modes. In the vertical mode (mode 0) the pixels above are copied downward to predict the current block, and in the horizontal mode (mode 1) pixels on the left are copied horizontally. The remaining seven modes copy and/or average neighboring pixels in various orientations.



(a) Original frame

(b) Reference frame

(c) Predicted frame

(d) Prediction residual

Figure 2-9: Intra prediction. An original frame (Mobile sequence at CIF resolution, frame 10), the reference frame used to predict the original frame (same as original frame), the predicted frame (using 8x8-pixel block intra prediction modes in H.264/AVC), and the prediction residual are shown.

proach to characterize such signals is to model them as random processes and estimate their statistical properties, especially their second order statistics. Despite the existence of different types of prediction residuals, characterizations of prediction residuals in the literature focus on the most widely used motion compensated prediction residual. In this section, we first introduce the Markov-1 characterization of images and then review related statistical characterizations for motion compensated prediction residuals.

Markov-1 model A stochastic process has the Markov property if the conditional distribution of the process depends only upon a finite number of past values, and signals having the Markov property are called Markov signals. Many signals can be modeled as Markov signals, including images. The value of a pixel is practically independent of distant pixels given the values of a finite number of neighboring pixels. The simplest case of the Markov property is obtained when the conditional distribution of the process depends upon only a single past value and such signals are called Markov-1 signals. A stationary Markov-1 signal has an auto-covariance given by equation (2.1). Here the parameter I represents the distance between the two points between which the auto-covariance is computed.

$$C(I) = \rho^{|I|} \tag{2.1}$$

A linear transform that can generate uncorrelated transform coefficients when applied on a stationary signal with known auto-covariance is called the Karhunen Loeve Transform (KLT). The KLT depends on the auto-covariance of the signal and is the statistically optimal linear transform to approximate/compress the signal with a minimum number of coefficients in the mean-square-error (MSE) sense. The KLT of a stationary Markov-1 signal can be obtained analytically [1] and this transform becomes the well-known DCT as the correlation reaches its maximum ($\rho \rightarrow 1$) [10].

A 2-D auto-covariance function formed from equation (2.1) using separable construction is given by equation (2.2).

$$C(I, J) = \rho_1^{|I|} \rho_2^{|J|} \tag{2.2}$$

Due to separable construction, the KLT of this auto-covariance is the 2-D DCT (as $\rho_1 \rightarrow 1$, $\rho_2 \rightarrow 1$.) The good performance of the 2-D DCT in image compression is due

to high correlation of neighboring pixels in images and the model in equation (2.2) with $\rho_1 = \rho_2 = 0.95$ has been considered a good approximation for typical images [1].

The separable model in equation (2.2) has also been used to characterize MC residuals and it has been found that the correlations are weaker than in images [34]. Puri and Chen suggested a correlation coefficient of $\rho = 0.5$ [34, 9]. To model the weaker correlations in MC residuals more precisely, a number of models have been proposed including models proposed by Chen [9], Niehsen [28] and Hui [16].

Chen's model An auto-covariance characterization of the MC residual, similar to the Markov-1 characterization of images, has been proposed by Chen et al. in [9]. Chen et al. propose a separable auto-covariance model constructed from a one-dimensional compound covariance model as shown in equation (2.3).

$$C(I) = A \cdot \rho^{|I|} + (1 - A) \cdot \delta(I) \tag{2.3}$$

This model consists of two parts. The first part, $A \cdot \rho^{|I|}$, represents the auto-covariance of a Markov-1 process. The second part, $(1 - A) \cdot \delta(I)$, represents the auto-covariance of white noise. The two parts can separately control the decay for $I \geq 1$ and $I < 1$. Suggested numbers for ρ and A are 0.95 and 0.5, respectively. Figure 2-10 compares this model with the Markov-1 model and it can be seen that Chen's model has a slower decay for $I \geq 1$.

It is possible to derive the KLT of the characterization in equation (2.3) using the KLT of the Markov-1 process. By the definition of white noise, any orthogonal transform is a KLT for white noise. Thus the KLT of the Markov-1 process can be chosen as the KLT of white noise. Then, both parts in equation (2.3) have the same KLT, and using the linearity property of transforms, the KLT of the Markov-1 process becomes the KLT of the characterization in equation (2.3). As a consequence of this result, Chen et al. argue that the DCT remains a near optimum transform for the MC residual.

Niehsen's model In [28], Niehsen et al. proposed another separable model for the MC residual constructed from a different one-dimensional compound covariance model given in equation (2.4).

$$C(I) = A \cdot \rho_0^{|I|} + (1 - A) \cdot \rho_1^{|I|^2} \tag{2.4}$$

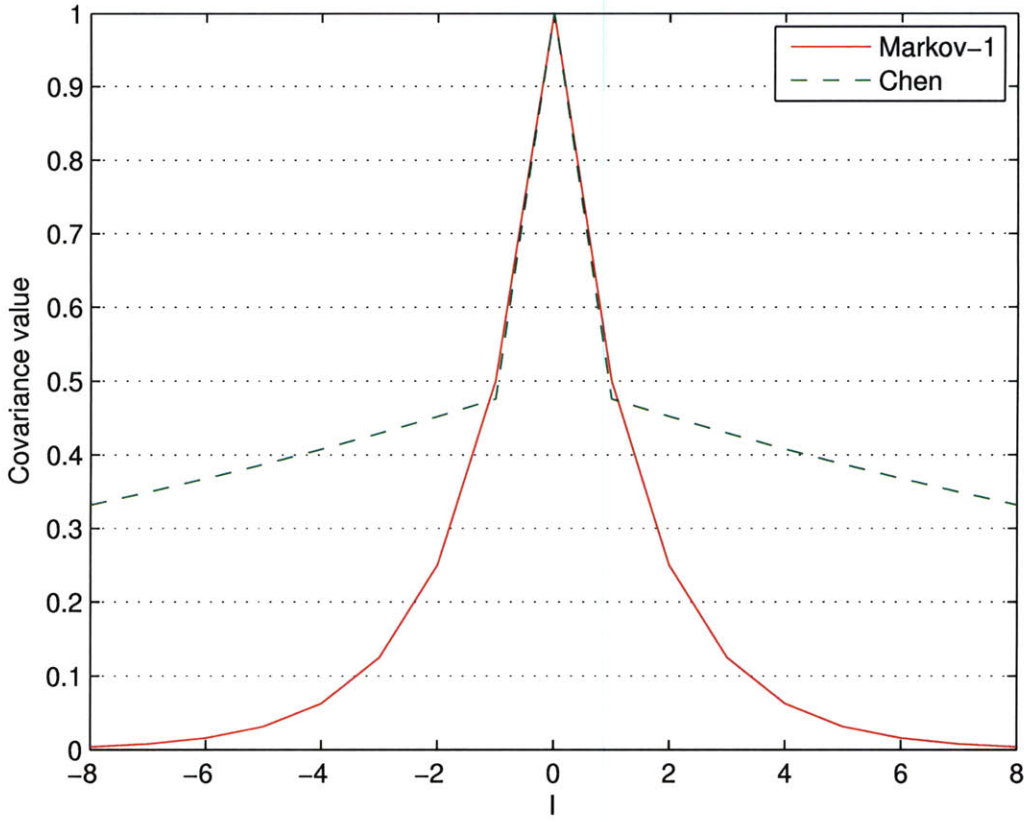


Figure 2-10: Comparison of Chen's compound model in equation (2.3) with the Markov-1 model in equation (2.1) .

The model parameters are given as $A = 0.17$, $\rho_0 = 0.91$ and $\rho_1 = 0.38$. Compared to Chen's model in equation (2.3), the white noise component is replaced by a stochastic process with quadratically exponentially decreasing covariance. Niehsen et al. argue that experimental data shows that the covariance of MC residuals decays quickly for $I \leq 2$ and much slower for $I \geq 2$ [28], and that their model can capture this observation better than Chen's model. Figure 2-11 compares Niehsen's model with Chen's model and the Markov-1 model.

An optimal transform for Niehsen's model in (2.4) is not given. Analytical solutions can usually be derived for simple models. However, Niehsen et al. compare the coding gains of the DCT, and the numerically computed KLT of their model. For a block size of 8, the coding gains are given as 0.97dB and 1.01dB for the DCT and the KLT, respectively. Niehsen et al. conclude that the difference is negligible and the DCT is a

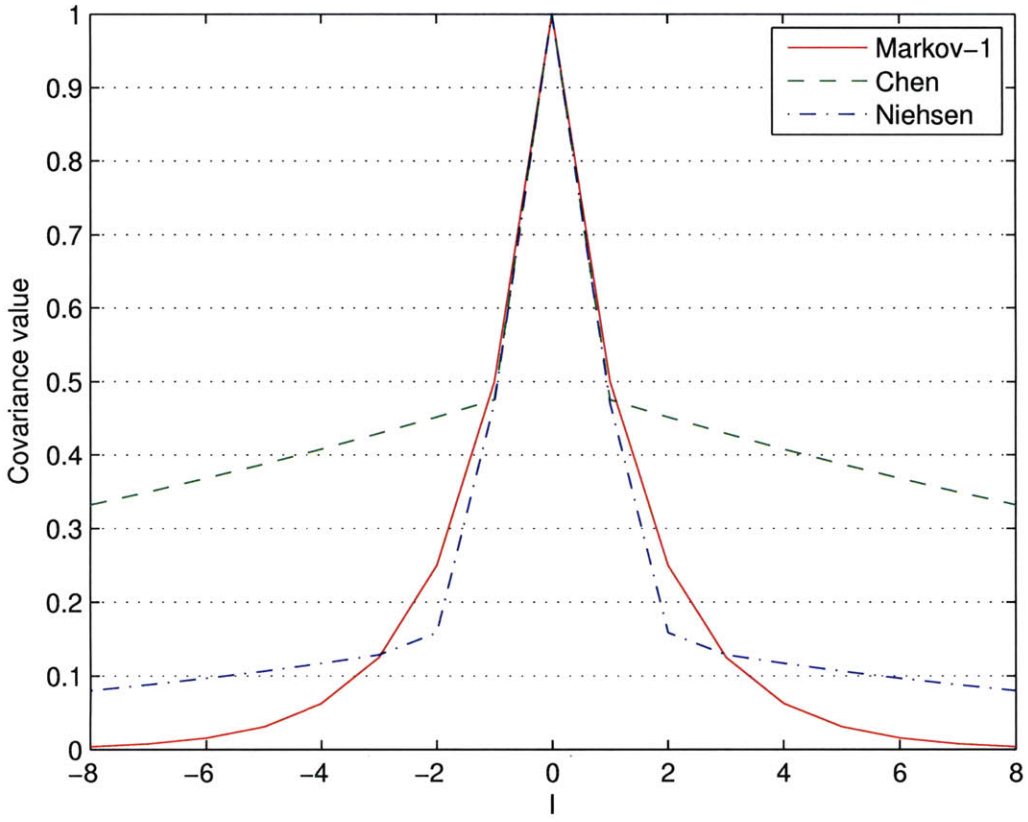


Figure 2-11: Comparison of Niehsen's model in equation (2.4) with Chen's model in equation (2.3) and the Markov-1 model in equation (2.1).

nearly optimum transform for the MC residual.

Hui's model Another characterization for the MC residual is given by Hui et al. in [16]. Instead of proposing a model like in equations (2.3) or (2.4), Hui et al. obtain a characterization by utilizing intra-frame and inter-frame auto-covariances. The MC residual is written as a difference of the current block and the best matching block from a previously encoded frame, as shown in equation (2.5)

$$e(i,j) = f_t(i,j) - f_{t-1}(i+u, j+v). \tag{2.5}$$

Here, $e(i,j)$ represents the MC residual of a block, $f_t(i,j)$ represents the block in the current frame, and $f_{t-1}(i+u, j+v)$ represents the matched block in the previous frame

where (u, v) is the displacement vector. The auto-covariance of the MC residual can be written in terms of the auto-covariances of $f_t(i, j)$ and $f_{t-1}(i + u, j + v)$ and the cross-covariance between $f_t(i, j)$ and $f_{t-1}(i+u, j+v)$, as shown below in equation (2.6).

$$C_e(I, J) = C_t(I, J) + C_{t-1}(I, J) - C_{t,t-1}(I + u, J + v) - C_{t-1,t}(I - u, J - v). \quad (2.6)$$

The current block $f_t(i, j)$ and the best match $f_{t-1}(i + u, j + v)$ are assumed to be Markov-1 processes, and their auto-covariances are

$$C_t(I, J) = C_{t-1}(I, J) = \rho^{|I|} \cdot \rho^{|J|}. \quad (2.7)$$

To obtain the cross-covariance between $f_t(i, j)$ and $f_{t-1}(i + u, j + v)$, it is assumed that the matched block $f_{t-1}(i + u, j + v)$ can be approximated by the current block $f_t(i, j)$ with a reasonable deformation. Specifically, the following approximation is used.

$$f_t(i + m_x, j + n_y) \approx f_{t-1}(i + u, j + v) \quad (2.8)$$

where (m_x, n_y) represents the deformation of each pixel in the current block. It is further assumed that m_x and n_y are independent Gaussian random vectors. From this assumption it follows that

$$C_{t,t-1}(I + u, J + v) = \rho^{|I+m_x|} \cdot \rho^{|J+n_y|} \quad (2.9)$$

$$C_{t-1,t}(I - u, J - v) = \rho^{|I-m_x|} \cdot \rho^{|J-n_y|}. \quad (2.10)$$

Since m_x and n_y are random vectors, the expected values of equations (2.9) and (2.10) are combined with equations (2.6) and (2.7) to obtain the auto-covariance of the MC residual given below in equation (2.11)

$$C_e(I, J) = 2 \cdot \rho^{|I|} \cdot \rho^{|J|} - 2 \cdot E[\rho^{|I-m_x|}] \cdot E[\rho^{|J-n_y|}] \quad (2.11)$$

Hui et al. elaborate further on the result in equation (2.11) and provide approximations for the expected value expressions.

2.3 Coding of Prediction Residuals

Despite the different types of redundancies and prediction methods to reduce them, coding of all prediction residuals is performed similarly. A block-based 2-D DCT is followed by entropy coding of the quantized coefficients. This section first provides a brief overview of the common approach to code prediction residuals, and reviews a number of methods proposed to improve the coding of some types of prediction residuals. A large fraction of the proposed coding improvements in the literature focus on the MC residual and few focus on the IP residual. Improving the coding of RE residuals has received little attention. In this section, we review some approaches proposed for MC and IP residuals.

2.3.1 Conventional Coding Approach

All prediction residuals are coded using a common approach summarized in Figure 2-12. A block-based 2-D DCT is applied, the transform coefficients are quantized, and the quantized coefficients are entropy coded. For example, in H.264/AVC the same transform, quantization and entropy coding method are used to code both MC residuals and IP residuals.

The transform that is used extensively in many video coding standards is the 2-D DCT, and the typical choice for its block size is 8x8. Recently, 4x4-pixel blocks are also used, commensurate with the 4x4-pixel block size for improved prediction such as 4x4-block intra or inter prediction used in H.264/AVC. The implementation of the 2-D DCT has typically been performed in floating point arithmetic (e.g. in MPEG-2), however, transforms that are approximations of the 2-D DCT and enable integer arithmetic implementations without a significant penalty in compression efficiency gained popularity [25].

Quantization involves mapping the values of transform coefficients to another domain which represents them with less accuracy and smaller number of bits. This is typically performed by dividing each transform coefficient by a positive number (quantizer) and rounding the result to an integer. In some video coding standards, such as H.264/AVC, the same quantizer is used for all coefficients within a block and in others, such as MPEG-2, lower frequency coefficients are quantized more finely than higher fre-

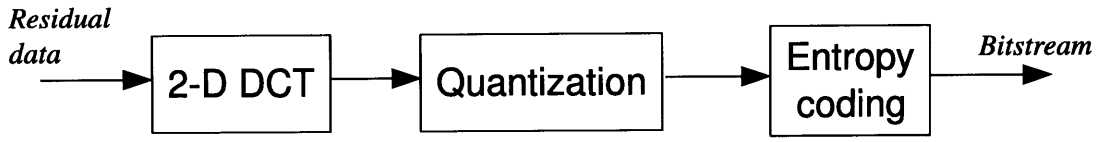


Figure 2-12: Common approach used to code prediction residuals.

quency coefficients due to the increased sensitivity of the human visual system to lower frequencies.

Entropy coding maps values of quantized coefficients to a stream of bits by utilizing their statistical properties. Typically many coefficients within a block are quantized to zero and the positions and values of the nonzero quantized coefficients are coded using either variable length coding (VLC) or arithmetic coding. Previous video coding standards, such as MPEG-2, have used VLC with a single table of codes. Recent video coding standards, such as H.264/AVC, use either VLC with codewords chosen adaptively from multiple tables, or arithmetic coding which can jointly encode a large number of elements and can also easily adapt to the statistics of the particular video sequence being encoded.

2.3.2 Proposed Coding Methods for the MC Residual

Matching-pursuit based coding Neff and Zakhor use an overcomplete set of functions, called dictionary of functions, to code the MC residual [27]. In an overcomplete dictionary, the basis functions are not linearly independent and the expansion of a signal with these basis functions is not unique. The advantage of such an approach is that it becomes possible to represent a signal with fewer basis functions than with a complete set of basis functions. The disadvantage of the approach is the difficulty to find the best representation. Matching-pursuit is one way to find a good representation [24]. In this method, the expansion begins by choosing the dictionary function that produces the largest inner product with the original signal. Then a residual signal is generated by subtracting the projection of the chosen dictionary function on the original signal, from the original signal. Next, the residual signal is expanded in the same way as the origi-

nal signal. This procedure continues iteratively until either a set number of expansion coefficients are generated or some energy threshold for the residual is reached.

The dictionary set that Neff et al. use consists of an overcomplete collection of 2-D separable Gabor functions, constructed from 1-D Gabor functions. Gabor functions are a set of scaled, modulated Gaussian windows of different sizes. If simply described, the dictionary functions are smooth 2-D sinusoids of various rectangular support. Representing the MC residual with these functions has various advantages. A total of 400 dictionary functions, each of which can be placed at any pixel location provides a much larger basis set than the block-DCT basis. This is the primary reason for the superiority of the matching pursuit approach. In addition, these functions die out smoothly at the support boundaries, due to the Gaussian window, and blocking effects are avoided. However, the large and flexible basis set also increases computational complexity. It is reported that this approach increases computational complexity compared to H.263 by a factor of 7 for QCIF resolution videos at 24Kbits/sec. The gains reported range from $0.17dB$ to $0.74dB$.

Second order statistics of the MC residual In [43], Tao and Orchard demonstrate that there is a close relationship between the gradient map of the original frame and the variance map of the MC residual. This relationship is used to encode each block with a transform constructed for that specific block. In particular, an auto-covariance matrix is constructed for each block using equation (2.12).

$$C(I,J) = \rho_r^{|r_I - r_J|} \cdot \rho_c^{|c_I - c_J|} \cdot \sigma_I \sigma_J \tag{2.12}$$

The parameter σ_I is the standard deviation for pixel I indexed by (r_I, c_I), and ρ_r and ρ_c are the correlation coefficients in the vertical and horizontal directions, respectively.

Each of the parameters in equation (2.12) needs to be available at both the encoder and the decoder to construct the same transform. ρ_r and ρ_c are estimated by the encoder and transmitted for each frame. To compute σ_I, the relationship between the gradient map of the previously reconstructed frame and the variance of the MC residual is utilized. In particular, the gradient of pixel I is computed from the previously reconstructed frame, so that the encoder and the decoder stay synchronized. Then a mapping from the gradient to the standard deviation of the MC residual is utilized to obtain the

standard deviation σ_I. The mapping information is transmitted by the encoder on a frame-by-frame level using a small number of bits. Hence, all parameters to compute the auto-covariance matrix in equation (2.12) are obtained. Then the KLT for each block is computed using eigen analysis.

An additional advantage of utilizing the variances of the MC residual pixels is that the variances of the transform coefficients can be obtained as well. This information is utilized by using an adaptive arithmetic coder to encode the transform coefficients.

Mask-based coding of the MC residual Ratakonda et al. develop a method based on the observation that the energy of the MC residual is concentrated in certain areas of the residual image [36]. These areas are predictable and include image edges and boundaries of blocks with non-zero motion vectors. It is proposed that only pixels in such areas of the MC residual are encoded, and other pixels are assumed to be zero.

The method can be summarized in two parts. In the first part, common to both the encoder and the decoder, a mask is generated that determines the pixels in the MC residual with large intensities that need to be coded. In the second part, the encoder freely changes pixel values outside of the mask so that the DCT coefficients of blocks have as few large coefficients as possible while representing the pixels within the mask with adequate fidelity. The decoder reconstructs the residual with the received DCT coefficients, and sets pixel values outside the mask to zero.

The mask for each block is determined using an edge detector on the previously encoded prediction frame. The optimal values of the pixels outside the mask (or equivalently the optimal DCT coefficients of the block) are found using an iterative algorithm based on projections onto convex sets (POCS). The first constraint set consists of assigning the original values to the pixels inside the mask. The second constraint set consists of setting predefined high-frequency DCT coefficients to zero and and reconstructing the block. These two constraints are applied iteratively.

2.3.3 Proposed Coding Methods for the IP Residual

A number of approaches have been proposed to improve the coding of I-frames based on intra prediction [38, 4, 42, 53, 54]. Some of these approaches propose improvements to

generate a more accurate prediction [38, 4, 42] and some of them propose improvements to code the prediction residual more efficiently [53, 54].

In [38, 53], bi-directional intra prediction (BIP) is proposed to improve the prediction. Similar to bi-directional motion compensated prediction in B-frames, predictions from two intra prediction modes are combined [38]. The number of combinations for the two prediction modes is large and some of these are not sufficiently effective and [53] proposes a reduced set of combinations. In [42], template matching (TM) is proposed as an additional intra prediction mode. In TM, five previously coded neighboring pixels of a block are used as a template (see Figure 2-13), and a match for the template within the previously coded areas is searched. Using the best match, a prediction for the 2x2 block is generated as shown in Figure 2-13. The same search algorithm is used at both the encoder and the decoder, eliminating the need for transmission of side information. In [4], displaced intra prediction (DIP) is proposed to extend the intra prediction modes in H.264/AVC. In DIP, the best match for the current block is searched within the previously coded regions and the best match is signaled by transmitting a displacement vector. It is reported that TM and DIP can generate more accurate prediction than the extrapolation based prediction modes within H.264/AVC in regions with complex texture.

Ye and Karczewicz propose directional transforms to code intra prediction residuals more efficiently [54]. Ye et al. argue that after prediction, there is significant directional information left in the prediction residual, and use directional transforms and adaptive scanning of transform coefficients to more effectively code the intra prediction residual.

To obtain the directional transforms, intra prediction residuals from training video sequences are used. Residuals for each prediction mode are gathered and a singular value decomposition is applied, first in the horizontal direction and then in the vertical direction, to obtain a separable transform for each individual prediction mode. The obtained transform matrices are approximated with integer point precision to reduce computational cost. Since each individual transform is used for prediction residuals from a particular prediction mode, no additional information is necessary to indicate the transform used for each block. To code the coefficients obtained with these transforms, adaptive scanning patterns are used which adapt to each transform and the video sequence being encoded. Specifically, as coding of a block is performed, the statistics of

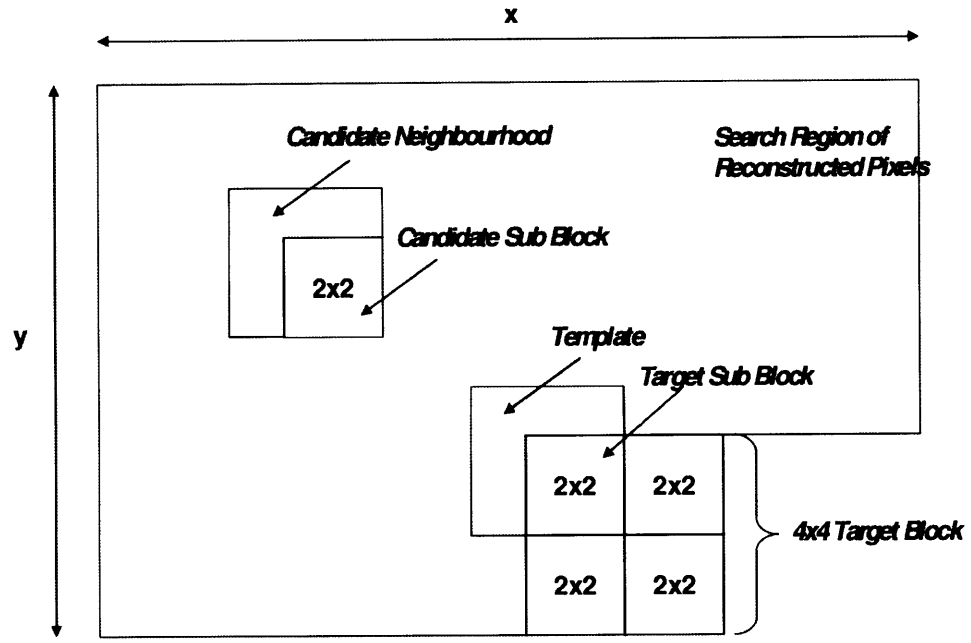


Figure 2-13: Template matching. The best match for the template is searched in the shown search region and a prediction for the 2x2 block is generated using the shown subblock of best match. (Reproduced from [42])

nonzero transform coefficients at each location is recorded separately for each transform and the scanning patterns of the next blocks are derived from these statistics.

2.4 Direction-adaptive Image Transforms

The 2-D Discrete Cosine Transform (DCT) and the 2-D Discrete Wavelet Transform (DWT) are the most commonly used transforms in image compression. These transforms are separable and are carried out by cascading two 1-D transforms in the vertical and horizontal dimensions. This approach favors horizontal or vertical features over others, for example diagonal features. New transforms have been developed that can adapt to the local features by performing the filtering along the direction of dominant local features. This can be achieved by resampling the image intensities along such directions [20], by performing filtering and subsampling on oriented sublattices of the sampling grid [46], by directional lifting implementations of the DWT [8], or by various other means. Even though most of the work is based on the DWT, similar ideas have been applied

to DCT-based image compression [55]. This section reviews some of the important approaches in this area.

Taubman's method One of the earliest approaches to developing direction-adaptive image transforms is presented by Taubman and Zakhor in [44]. In this approach, the image is first divided into smaller partitions. For each partition, a pair of axes is chosen according to the dominant linear features of the partition and the partition is resampled along these axes so that the linear features are aligned either horizontally or vertically. Next, a separable subband decomposition is applied in the resampled domain. The subband samples are quantized and coded, along with the orientation information.

At the receiver, each partition is reconstructed in the resampled domain and the resampling process is inverted using the orientation information. Finally, a local smoothing algorithm is applied at the partition boundaries in order to remove artifacts resulting from independent processing of image partitions.

Bandelets In this approach, directional features of the image are described with geometric flows [20]. A geometric flow is a vector field representing directions in which the signal has regular (i.e. slowly changing) variations in a small neighborhood. To construct orthogonal bases along these flows, they are required to be parallel either vertically or horizontally, and to maintain enough flexibility, this parallel condition is imposed in partitions of the image. Partition sizes can be varied according to the detail of the geometric flow. Figure 2-14 shows a sample image with partitions and geometric flows in each partition.

Resampling is performed in each partition to capture image sample values along the flow lines. A warped 2-D wavelet transform with a subband filtering along the flow lines, which goes across partition boundaries, is performed. Next, bandeletization is performed in each partition to take advantage of the remaining regularity (smooth structure) in the warped wavelet transform coefficients. The subband which contains coefficients from low-pass filtering (or equivalently inner producting with the scaling function) along the geometric flow lines still has regularity in these coefficients along the flow lines. A one-dimensional wavelet transform on these coefficients along the flow lines is called bandeletization and can provide further energy compaction. If all steps of the

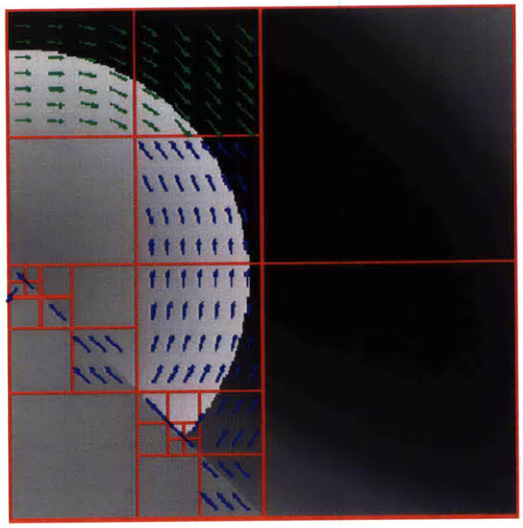


Figure 2-14: Image with partitions of varying sizes and geometric flows in each partition.(Figure reproduced from [20])

processing (the resampling, the warped 2-D wavelet transform and the one-dimensional wavelet transform) in a partition are combined, the aggregate transform is called the bandelet transform and its basis functions are called bandelets.

Directionlets Another approach to direction-adaptive image transforms is presented by Velisavljevic et al. in [46]. These transforms, called directionlets, are constructed from the so-called skewed anisotropic wavelet transforms (S-AWT), which make use of two concepts: directionality and anisotropy. Directionality is achieved by performing the filtering and subsampling steps of the transforms on oriented sublattices of the sampling grid. Anisotropy is obtained by an unbalanced iteration of transform steps along two transform directions.

For a more detailed discussion of how directionality is achieved, consider Figure 2-15. A full-rank integer lattice Λ consists of points obtained from a linear combination of two linearly independent vectors, where both the components of the vectors and the scaling coefficients are integers (see Figure 2-15 (a)). The two independent vectors determine the lattice and the directions of the transform. In Figure 2-15, the vectors are the rows of M_Λ and the directions are $45°$ and $-45°$. The integer lattice can be partitioned into cosets, determined by the shift vectors s_0 and s_1 in Figure 2-15. The closed circles

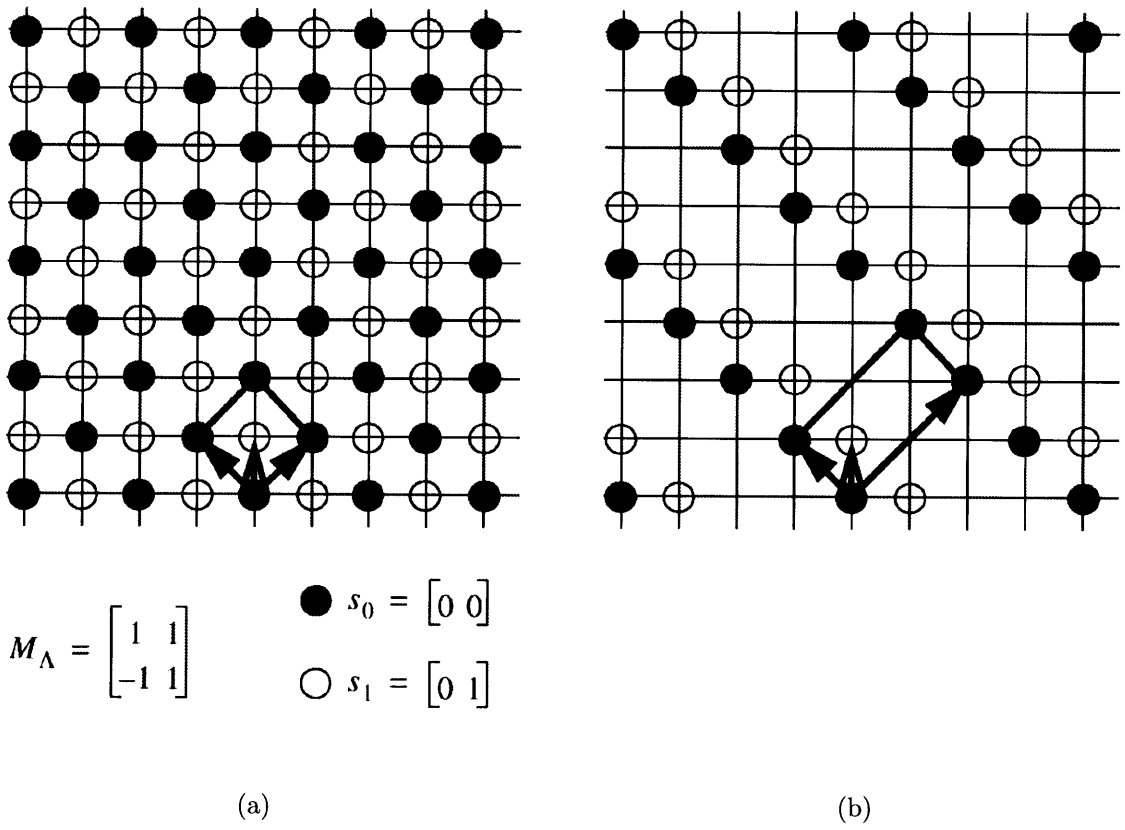
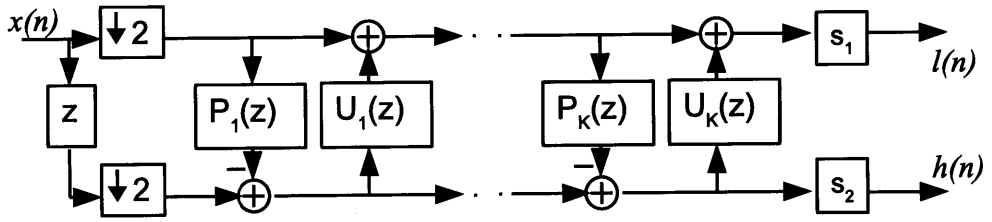


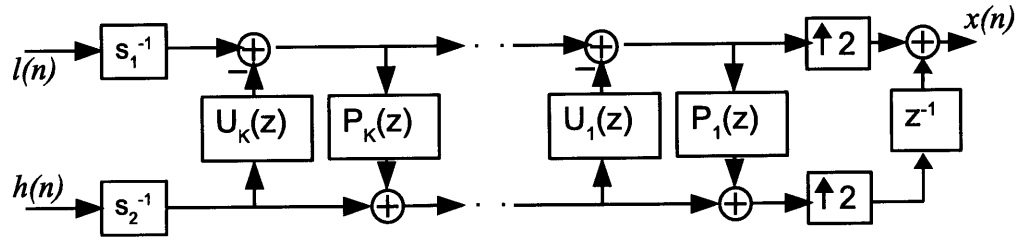
Figure 2-15: Lattice Λ is determined by the generator matrix M_Λ and is partitioned into 2 ($= |det(M_\Lambda)|$) cosets, determined by the shift vectors s_0 and s_1. The two cosets are shown with black and white circles and one dimensional filtering and subsampling is applied along the diagonally aligned points in each coset. The result along 45° is shown in (b). (Figure reproduced from [46])

in Figure 2-15 (a) represent one coset, and the open circles represent another coset. The points aligned diagonally along 45° and −45° in each coset form the so-called co-lines. One-dimensional filtering and subsampling is applied along co-lines separately in each coset. The result along 45° is shown in part (b) of Figure 2-15, and the second one-dimensional filtering can be applied similarly along −45° on the retained pixels.

This example shows how directionlets with vanishing moments are constructed along 45° and −45°. Directionlets along other directions with rational slopes can be obtained similarly. However, with other directions, such as $tan^{-1}1/4$ for example, the points on a co-line become further separated from each other and this reduces the efficiency of the filtering along the co-lines. In an implementation of directionlets in [45], the authors use



(a) Analysis side



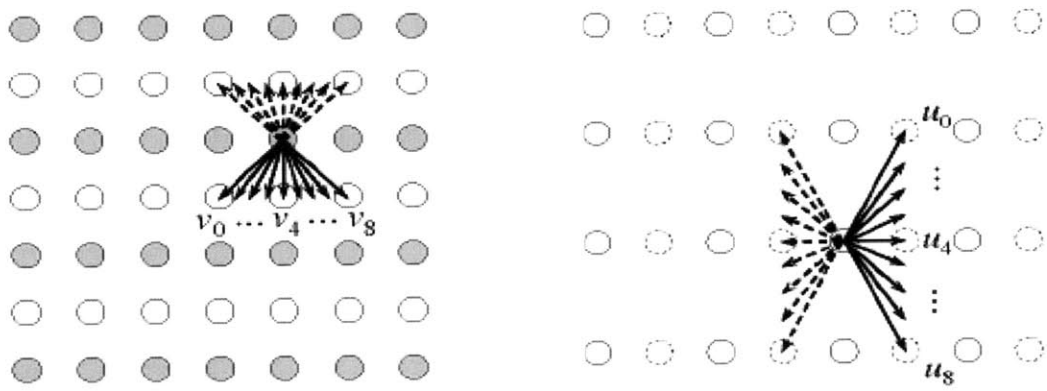
(b) Synthesis side

Figure 2-16: Block diagram of the lifting implementation of the wavelet transform [41].

only horizontal, vertical and diagonal directions.

Lifting-based methods Lifting is a procedure to design wavelet transforms using a series of filtering steps called lifting steps [41]. As shown in Figure 2-16 (a), the signal is first divided into even and odd samples and the odd samples are predicted from the even samples. The residual in the prediction is then used to update the even samples. Any number of prediction and update pairs can be cascaded until the final low-pass and high-pass signals of the transform are obtained. The filters used for prediction and update determine the analysis and synthesis filters of the resulting wavelet transform. No matter how the prediction and update boxes in Figure 2-16 are chosen, this scheme is always invertible and the inverse transform is given in Figure 2-16(b).

To apply a separable 2-D DWT on a 2-D signal using the lifting implementation, 1-D DWT's with lifting implementation in the vertical and horizontal dimensions can be cascaded. Lifting-based wavelet transform with directional prediction is performed by choosing the pixels from which a prediction (or update) is formed in an intelligent manner. These pixels are chosen along a direction which is not necessarily the horizontal or vertical direction as it is the case for the lifting implementation of the separable 2-



(a) Lifting on image samples along the vertical dimension.

(b) Lifting on update samples along the horizontal dimension.

Figure 2-17: Directional prediction options in [48].

D DWT. Figure 2-17 (a) shows several options that can be used along the vertical dimension. To predict the pixels in an odd row, fractional-pixels (interpolated from pixels in the same row) or full-pixels from even rows aligned along a particular direction can be used. In the update step, pixels in even rows are updated from the prediction residuals in odd rows aligned along the same direction. After subsampling in the vertical dimension to form the low-pass and high-pass signals, similar directional prediction and update operations can be performed along the horizontal dimension, separately on the low-pass (Figure 2-17 (b)) and high-pass signals. The low-low signal can be transformed again using directional lifting operations to obtain multilevel directional subband decompositions.

Figure 2-17 shows the directional prediction options proposed in [48]. Other prediction options have also been proposed [8]. In fact, to predict (update) a pixel in an odd (even) row, any pixel from any even (odd) row can be used. Typically, however, nearby pixels are likely to provide better prediction.

Directional DCT The directional block transforms in [55] are 2-D directional DCT's together with a DC separation and ΔDC correction method borrowed from [19]. 2-D directional DCT's are formed by 1-D DCT's along predefined pixels, followed by a second set of 1-D DCT's and DC separation and ΔDC correction computations. DC separation and ΔDC correction are computations introduced to mitigate some undesired proper-

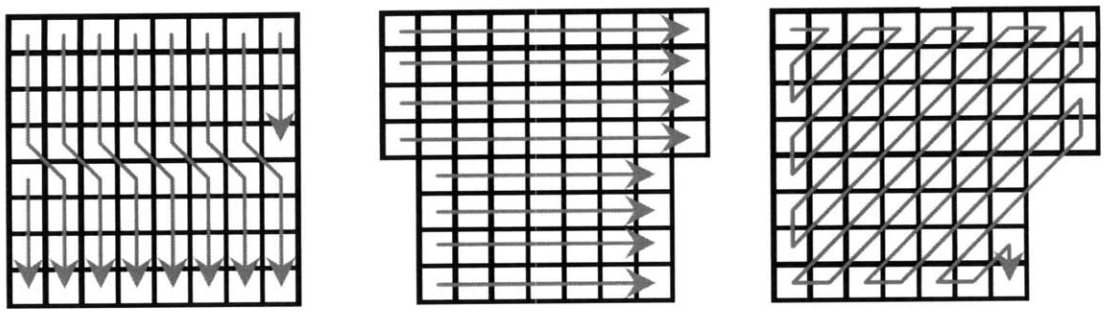
ties of the overall transforms and were taken from the shape adaptive DCT framework proposed for coding arbitrarily shaped objects.

A total of six transforms, each targeting different directions, are proposed in [55]. One way to explain the details of these transforms is to consider one of these transforms, shown in Figure 2-18, as an example. Before the first series of the 1-D DCT's are performed, the mean of the block is computed and subtracted from the block. This is the DC separation step. Then the first series of 1-D DCT's, as shown in Figure 2-18 (a), are applied on the zero-mean block. The arrangement of these DCT's implies that the direction of this transform is close to the vertical direction. All coefficients from these 1-D DCT's are arranged into columns, as shown in Figure 2-18 (b), such that the DC coefficients of each 1-D DCT are positioned in the same row. Then the second series of 1-D DCT's are performed as shown in Figure 2-18 (b). The DC of the DC coefficients is set to zero and instead, the initially computed mean of the block is transmitted. The mean represents the projection of the block onto the constant intensity basis function, which is important in block-based image compression. Note that setting the DC of the DC coefficients to zero can be compensated for in the reverse transform (ΔDC correction) because it is known that this block is a zero-mean block [19]. The proposed scan of the resulting coefficients is shown in part (c) of the figure.

The inverse transform consists of applying the inverses of each step of the forward transform, except that a ΔDC correction step is introduced. The inverse transform starts with arranging the coefficients into the form shown in Figure 2-18 (b), and applying the 1-D IDCT's as shown in the same figure. Then a ΔDC correction parameter is computed and subtracted from all the DC coefficients of the first 1-D IDCT's. Next, the coefficients are arranged into the form shown in Figure 2-18 (b) and 1-D IDCT's are applied. Finally the mean of the block is added back, completing the inverse transform.

2.5 Summary and Extension

Previous sections in this chapter have discussed research in a number of closely connected areas which are all related to the research in this thesis. Section 2.1 discussed different types of redundancies present in video coding, prediction methods to reduce them, and the resulting prediction residuals. Section 2.2 presented approaches to characterize some



(a) First series of 1-D DCT's

(b) Rearrangement and second series of 1-D DCT's

(c) Rearrangement and scan of coefficients

Figure 2-18: Directional DCT.

of these residuals using statistical tools, and Section 2.3 presented methods to code these residuals efficiently. Finally, Section 2.4 discussed several new approaches to develop transforms which can take advantage of local anisotropic features in images.

Traditionally, transforms and coding approaches used for images have also been used for prediction residuals. Studies that explore the differences between images and prediction residuals have used global models which are separable and have also focused mainly on the motion compensated prediction residuals, as discussed in Section 2.2. The new transforms developed for images can take advantage of local anisotropic features in images by adapting the transforms to these features. These approaches are novel and prediction residuals have not been studied from this perspective. This thesis attempts to analyze the characteristics of prediction residual using the perspectives and insights provided by these approaches. In Chapter 3, we study and compare the local anisotropic characteristics of images and several types of prediction residuals, and propose in Chapter 4 new transforms for some types of prediction residuals based on this analysis.

Chapter 3

Analysis of Prediction Residuals

To develop transforms for prediction residuals it is essential to study the characteristics of prediction residuals. This chapter analyzes the characteristics of several different types of prediction residuals and discusses differences of the characteristics between images and these prediction residuals. We first provide an empirical analysis based on visual inspection in Section 3.1. We then present in Section 3.2 a statistical analysis that quantifies the differences.

3.1 Empirical Analysis Based on Visual Inspection

This section presents an empirical analysis of images and prediction residuals based on visual inspection using the images and their prediction residuals shown in Figures 3-1, 3-2 and 3-3. Each figure shows a frame, its MC residual, RE residual, and IP residual. Figure 3-1 shows frame 10 of *mobile* sequence at CIF (352x288) resolution, and its prediction residuals. Figure 3-2 shows frame 26 of *paris* sequence at CIF resolution, and its prediction residuals, and Figure 3-3 shows frame 118 of *basket* sequence at CIF resolution, and its prediction residuals.

A common aspect of all prediction residuals is that smooth regions can be predicted quite well. For example, the prediction residuals of uniform background regions in all three figures are negligibly small. The spatial correlation in smooth regions of images is high and this enables successful prediction. In motion compensated prediction, even if the underlying motion is not exactly translational, the high spatial correlation of

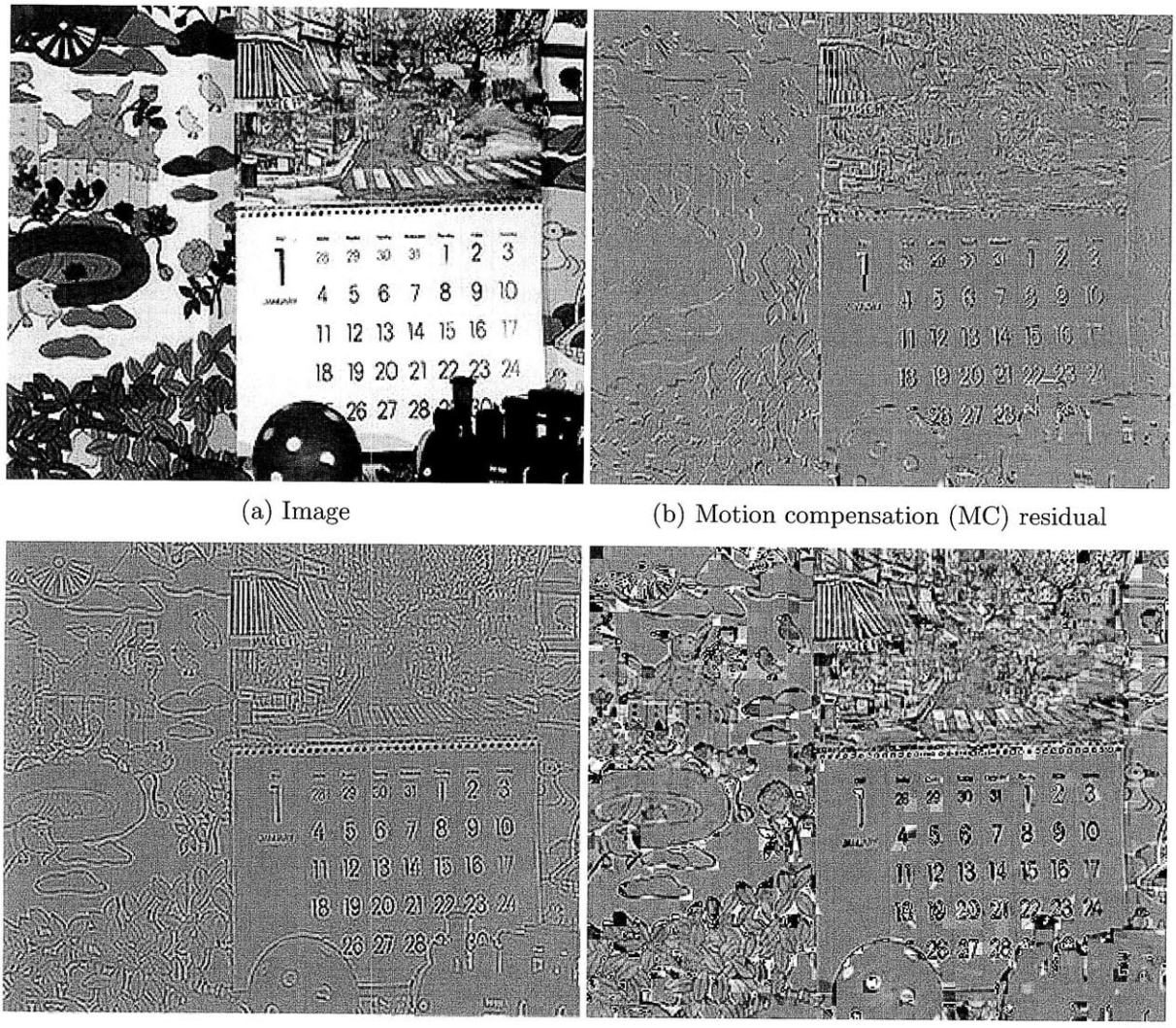
pixels enables a quite accurate match between blocks. Similarly, in interpolation the missing pixel is highly correlated with the neighboring pixels used in low-pass filtering. In intra prediction the pixels within a block to be predicted are highly correlated with the neighboring pixels used for prediction. Since prediction residuals of smooth regions are negligibly small, they are typically not coded.

In texture regions, prediction does not work as well as in smooth regions. For example, in Figure 3-1 the calendar picture contains many fine details and all prediction methods in this region do not work well as can be seen from the residual frames in Figure 3-1 (b),(c) and (d). Even though the local variations in such regions can not be predicted well, the local mean can be predicted well and the local mean of prediction residuals in such regions is typically zero. Except the mean, characteristics of prediction residuals in such regions are similar to characteristics of images.

Prediction also does not work well around object boundaries or edges. Consider the boundary of the ball and the boundary of the objects in the background in Figure 3-1, or the boundary of the peoples' clothes/body in Figures 3-2 and 3-3, or the edges of the letters on the players' shirts in Figure 3-3. In all these regions, the boundaries or edges contain large prediction errors in the residual frames. In motion compensated prediction, motion is typically not exactly translational and this results in a mismatch along an edge or boundary and produces large prediction errors along these structures. Similarly, interpolation can not accurately predict intensities nearby edges or boundaries and large prediction errors are also present along edges or object boundaries in resolution enhancement residuals. Intra prediction also does not perform well around object boundaries and large prediction errors in such regions are present.

Characteristics of images and prediction residuals differ significantly around object boundaries or edges. In particular, consider MC residuals and RE residuals. In these residuals, it is the rapidly changing pixels along the boundary or edge of the original image that can not be predicted well and large prediction errors form along these structures. These structures are 1-D structures and the residuals concentrating on these structures have 1-D characteristics. Such 1-D structures can be easily seen in the MC and RE residuals in Figures 3-1,3-2,3-3. Boundary or edge regions in images, on the other hand, have typically smooth structures on either side of the boundary or edge and their characteristics are 2-D.

In summary, images and prediction residuals have different characteristics. Local regions in images have 2-D anisotropic structures. Local regions in prediction residuals are sparse, meaning many pixels in a local region are typically close to zero because they have been predicted well. The remaining nonzero pixels are usually not randomly scattered but concentrated in regions which are difficult to predict. In this thesis, we focus on MC and RE residuals and in these types of residuals, a major fraction of pixels that can not be predicted well concentrate on object boundaries and edges. Object boundaries and edges are 1-D structures and the residuals concentrating on these structures have 1-D characteristics. As a result, while images have 2-D anisotropic characteristics, MC and RE residuals contain significant amount of local regions with 1-D anisotropic characteristics, and this difference constitutes a major distinction in the characteristics between images and MC or RE residuals. This distinction has been the main inspiration for this thesis and the transforms proposed in Chapter 4 are based on it.



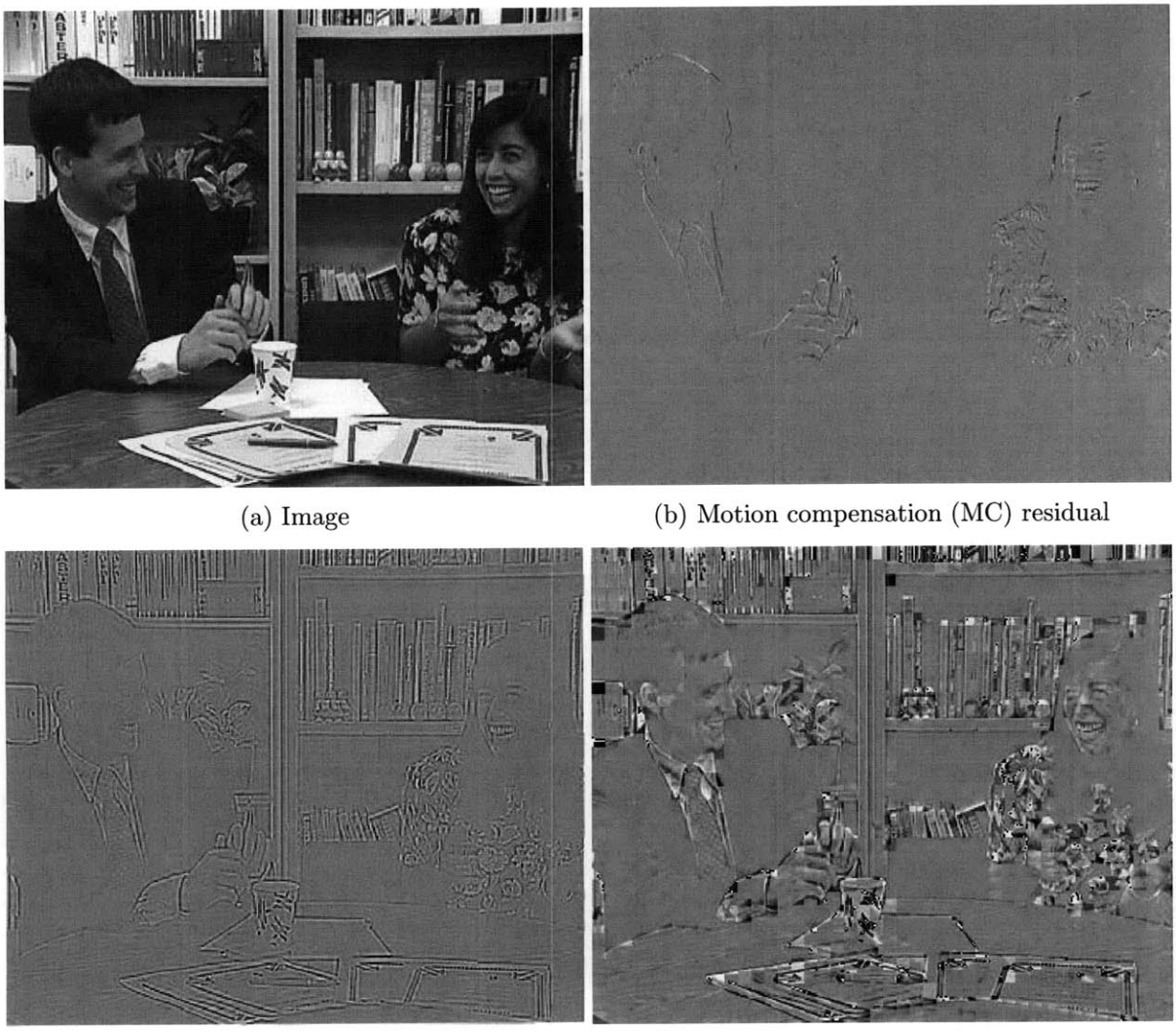
(a) Image

(b) Motion compensation (MC) residual

(c) Resolution enhancement (RE) residual

(d) Intra prediction (IP) residual

Figure 3-1: Frame 10 of mobile sequence at CIF resolution, its MC residual predicted from frame 9 using ful-pel motion estimation with 8x8-pixel blocks, its RE residual obtained from its QCIF resolution version, and its IP residual obtained using 8x8-pixel intra prediction modes in H.264/AVC.



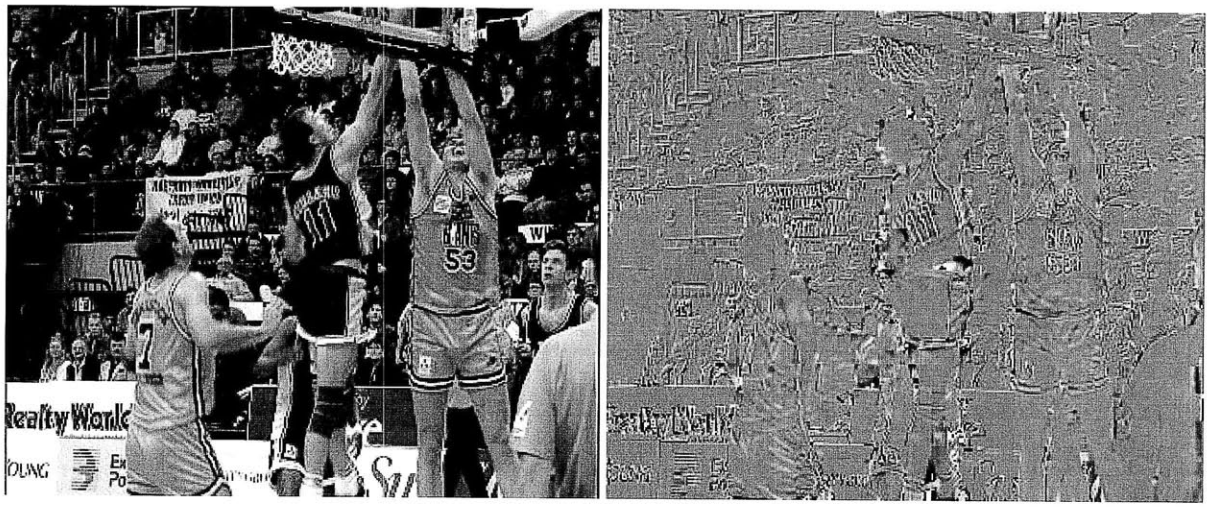
(a) Image

(b) Motion compensation (MC) residual

(c) Resolution enhancement (RE) residual

(d) Intra prediction (IP) residual

Figure 3-2: Frame 26 of paris sequence at CIF resolution, its MC residual predicted from frame 25 using ful-pel motion estimation with 8x8-pixel blocks, its RE residual obtained from its QCIF resolution version, and its IP residual obtained using 8x8-pixel intra prediction modes in H.264/AVC.



(a) Image

(b) Motion compensation (MC) residual

(c) Resolution enhancement (RE) residual

(d) Intra prediction (IP) residual

Figure 3-3: Frame 118 of basket sequence at CIF resolution, its MC residual predicted from frame 117 using ful-pel motion estimation with 8x8-pixel blocks, its RE residual obtained from its QCIF resolution version, and its IP residual obtained using 8x8-pixel intra prediction modes in H.264/AVC.

3.2 Auto-covariance Analysis

As described in the previous section, images and prediction residuals have different characteristics. Local regions in images have 2-D anisotropic structures. Characteristics of local regions in prediction residuals depend on the characteristics of the local regions of the image from which they are obtained and on the prediction method. In MC and RE residuals, a major fraction of pixels that can not be predicted well concentrate on object boundaries and edges, and form 1-D structures along them. Thus a major fraction of local regions in these types of residuals have 1-D characteristics, which is not present in images. In this section, we quantify the difference between images and various types of prediction residuals using an auto-covariance analysis and this analysis also shows the aforementioned difference in characteristics.

Prior characterizations of prediction residuals focus on characterizing the MC residual. Other types of prediction residuals have not received much attention. These characterizations use auto-covariance functions that provide a close fit to experimental data using one global model for the entire MC residual [9, 28, 16]. To show the differences of local anisotropic characteristics between images and prediction residuals, we use two models for the auto-covariance of local regions. One is a separable model and the other generalizes it by allowing the axes to rotate. The ability to rotate allows capturing local anisotropic characteristics. We estimate the parameters of these models from images and prediction residuals shown in Figures 3-1, 3-2 and 3-3 and plot the estimated parameters. These plots provide valuable insights.

3.2.1 Auto-covariance Models

A stationary Markov-1 signal has an auto-covariance given by equation (3.1).

$$C(I) = \rho^{|I|} \tag{3.1}$$

For discrete-time stationary Markov-1 signals, the decorrelating transform can be obtained analytically [1] and this transform becomes the well-known DCT as correlation reaches its maximum ($\rho \rightarrow 1.$) A 2-D auto-covariance function formed from equation

(3.1) using separable construction is given by equation (3.2).

$$C_s(I, J) = \rho_1^{|I|}\rho_2^{|J|} \tag{3.2}$$

Due to separable construction, the decorrelating transform for this auto-covariance is the 2-D DCT (as $\rho_1 \rightarrow 1$, $\rho_2 \rightarrow 1$.) The good performance of the 2-D DCT in image compression is due to high correlation of neighboring pixels in images and $\rho_1 = \rho_2 = 0.95$ has been considered a good approximation for typical images [1].

The separable model in equation (3.2) has also been used to characterize the MC residual and it has been reported that the correlations are weaker than in images [34]. Other models have been proposed to model the weaker correlations more precisely [9, 28]. These models are global and were proposed to provide a closer fit to the average auto-covariance of the MC residual obtained from different parts of a frame. All these models are global and separable, and cannot adequately capture local anisotropies in images and prediction residuals.

To capture local anisotropies in images and prediction residuals, we use a generalized model, shown in equation (3.3).

$$C_g(\theta, I, J) = \rho_1^{|Icos(\theta)+Jsin(\theta)|}\rho_2^{|-Isin(\theta)+Jcos(\theta)|} \tag{3.3}$$

This model has an additional degree of freedom provided by the parameter θ . The parameter θ allows rotation of the axes of the auto-covariance model and enables capturing local anisotropic features by adjusting to these features. The separable model is a special case of the generalized model. The generalized model with $\theta = 0°$ is the separable model. Figure 3-4 shows both models. Characterization of images with similar generalized auto-covariance models have been made [8]. Characterizations of images and MC residuals with the separable model, or its derivatives, have also been made [1, 9, 28, 16]. However, prediction residuals have not been characterized with a direction-adaptive model.

3.2.2 Estimation of Parameters of Auto-covariance Models

We estimate the parameters ρ_1 and ρ_2 for the separable model, and the parameters ρ_1, ρ_2 and θ for the generalized model from blocks of 8x8-pixels of the images and

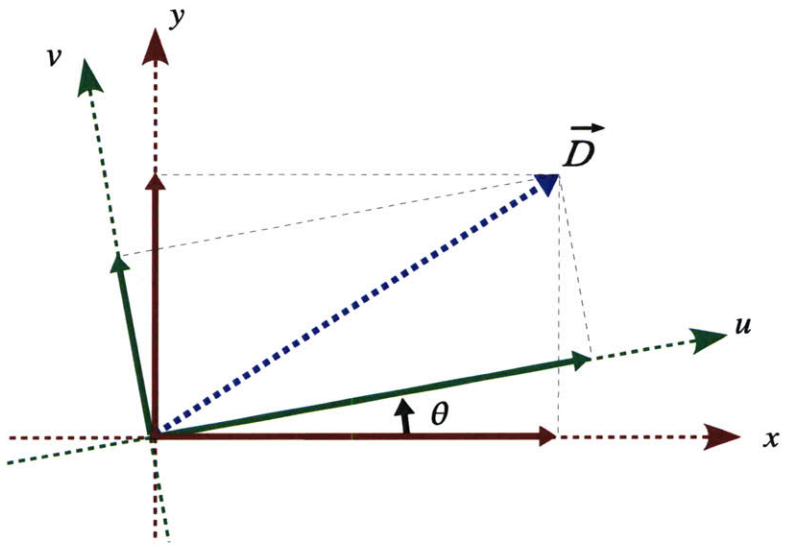
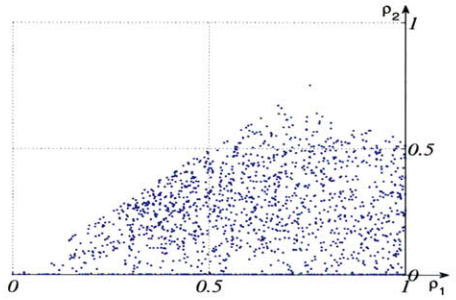
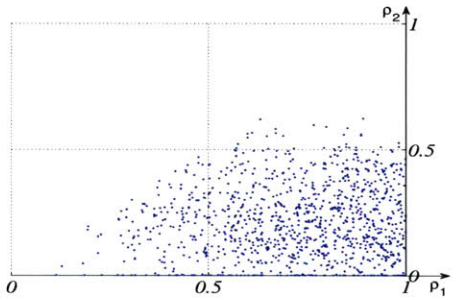


Figure 3-4: Comparison of separable and the generalized auto-covariance models. Use of the separable model corresponds to expanding the distance vector $\vec{D} = I\vec{u_x} + J\vec{u_y}$ in the cartesian coordinate system. Use of the generalized model corresponds to expanding the distance vector $\vec{D}$ in a rotated coordinate system.

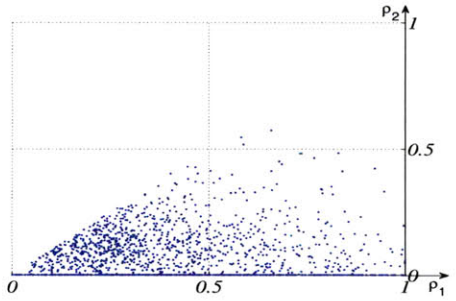
prediction residual shown in Figures 3-1, 3-2 and 3-3. We first use the unbiased estimator to estimate a non-parametric auto-covariance of each block. This is accomplished by removing the mean of the block, correlating the zero mean-block with itself, and dividing each element of the correlation sequence by the number of overlapping points used in the computation of that element. Then we find the parameters ρ_1, ρ_2 and θ so that the models in equations (3.2) and (3.3) best approximate the estimated non-parametric auto-covariance, by minimizing the mean-square-error between the non-parametric auto-covariance estimate and the models. In the minimization, we use lags less than four (i.e. $|I|, |J| < 4$) because at large lags the number of overlapping points becomes less and the estimates become noisy. We use ρ_1 for the larger covariance coefficient and let θ vary between 0° and 180°. The estimation results are shown in Figures 3-5, 3-6 and 3-7 for the images and prediction residuals shown in Figures 3-1, 3-2 and 3-3, respectively. The results are similar in these figures and we focus on Figure 3-5 to simplify the discussion of the results. Each point in the plots represents the estimate from one 8x8-pixel block.



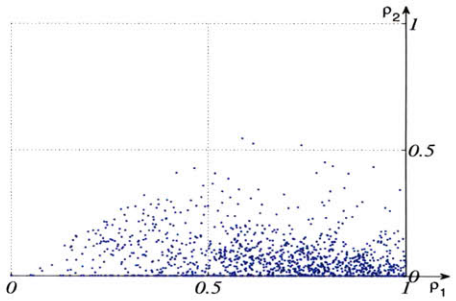
(a) Image, separable model



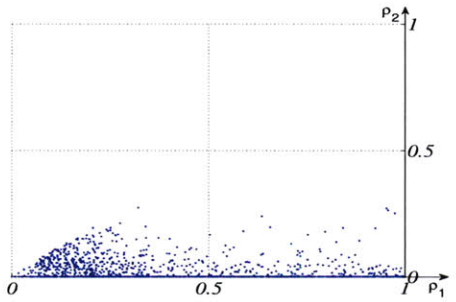
(b) Image, generalized model



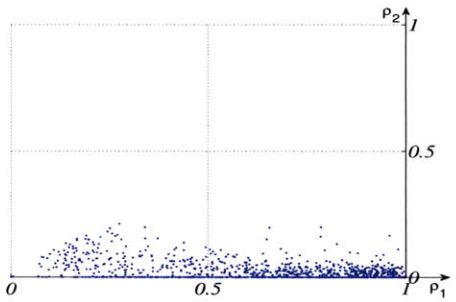
(c) MC residual, separable model



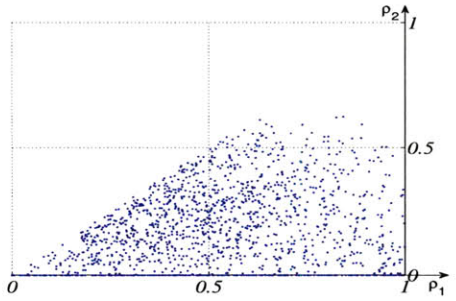
(d) MC residual, generalized model



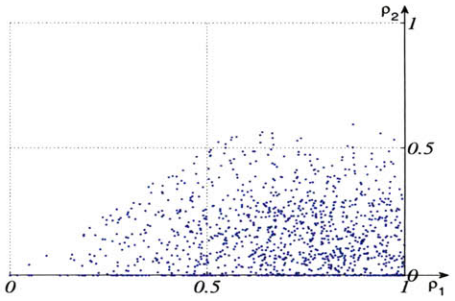
(e) RE residual, separable model



(f) RE residual, generalized model

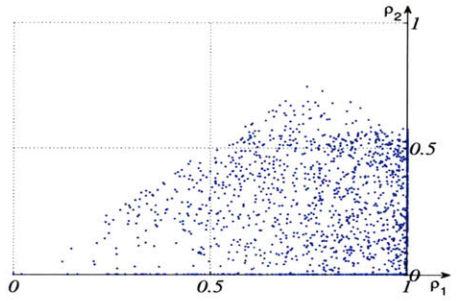


(g) IP residual, separable model

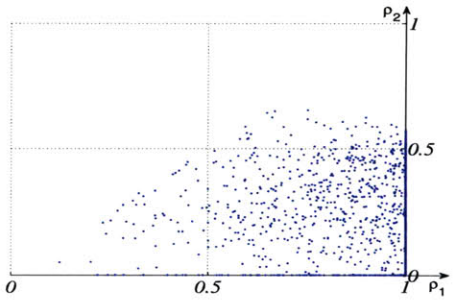


(h) IP residual, generalized model

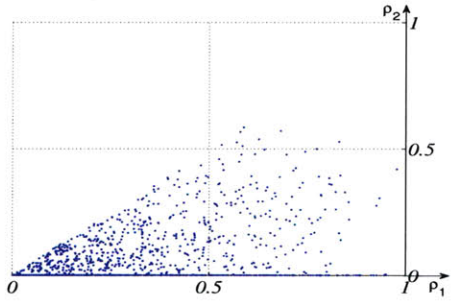
Figure 3-5: Scatter plots of (ρ_1, ρ_2)-tuples estimated using the separable and generalized auto-covariance models from the image, MC residual, RE residual and IP residual shown in Figure 3-1. Plots on the left column show parameters estimated using the separable model and plots on the right column show parameters estimated using the generalized model. Plots on each row were estimated from a different source signal.



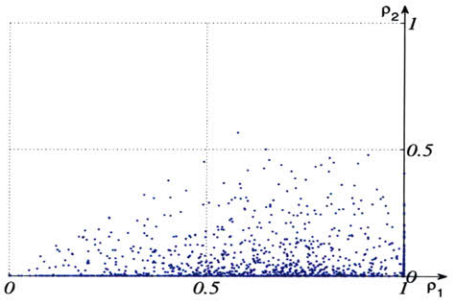
(a) Image, separable model



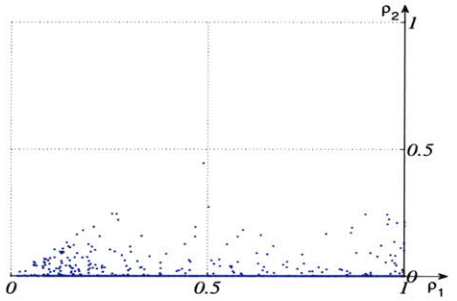
(b) Image, generalized model



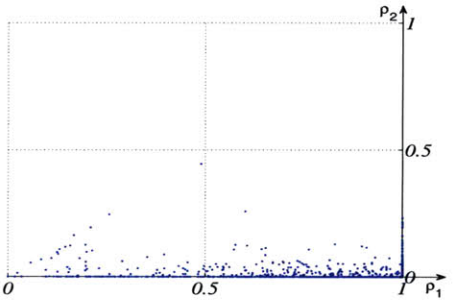
(c) MC residual, separable model



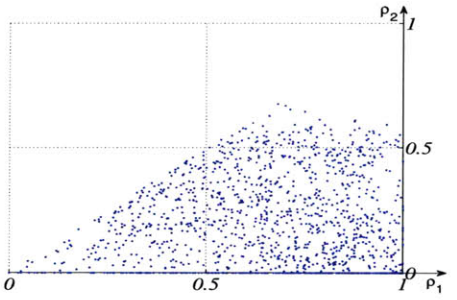
(d) MC residual, generalized model



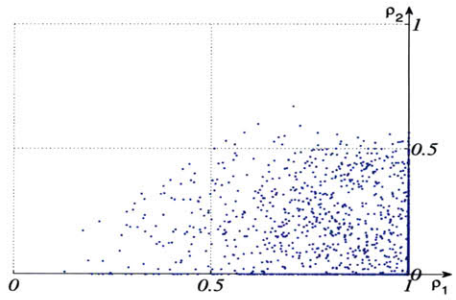
(e) RE residual, separable model



(f) RE residual, generalized model

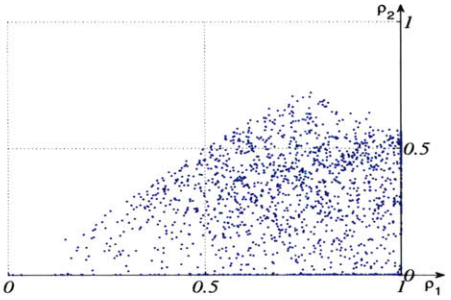


(g) IP residual, separable model

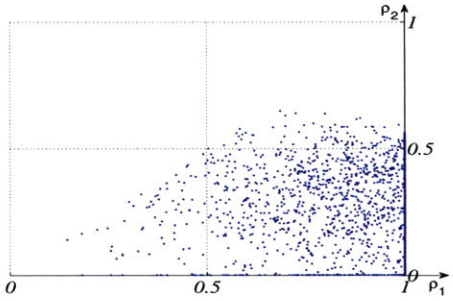


(h) IP residual, generalized model

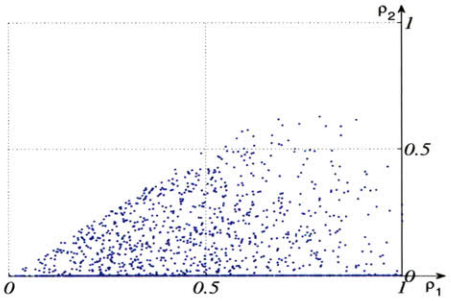
Figure 3-6: Scatter plots of (ρ_1, ρ_2)-tuples estimated using the separable and generalized auto-covariance models from the image, MC residual, RE residual and IP residual shown in Figure 3-2. Plots on the left column show parameters estimated using the separable model and plots on the right column show parameters estimated using the generalized model. Plots on each row were estimated from a different source signal.



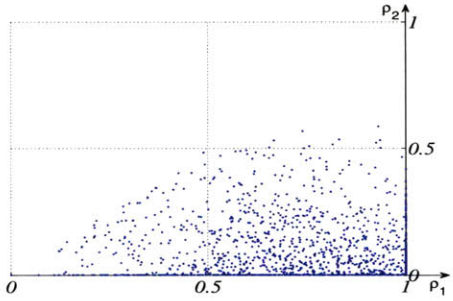
(a) Image, separable model



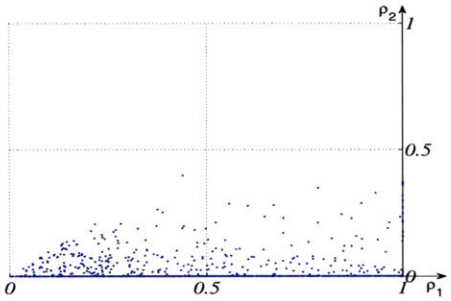
(b) Image, generalized model



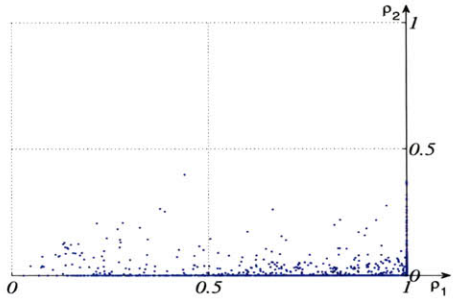
(c) MC residual, separable model



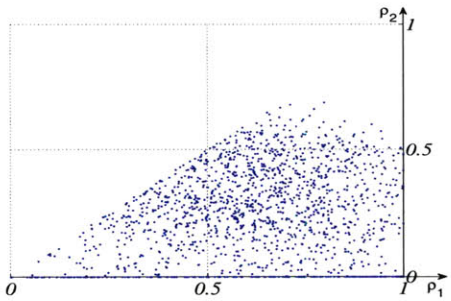
(d) MC residual, generalized model



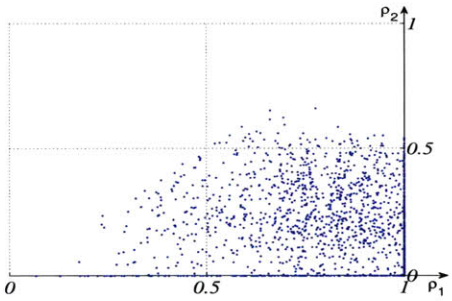
(e) RE residual, separable model



(f) RE residual, generalized model



(g) IP residual, separable model



(h) IP residual, generalized model

Figure 3-7: Scatter plots of (ρ_1, ρ_2)-tuples estimated using the separable and generalized auto-covariance models from the image, MC residual, RE residual and IP residual shown in Figure 3-3. Plots on the left column show parameters estimated using the separable model and plots on the right column show parameters estimated using the generalized model. Plots on each row were estimated from a different source signal.

3.2.3 Estimated Model Parameters for Images

First, consider the scatter plots shown in Figures 3-5 (a) and (b). They were obtained from the image shown in Figure 3-1 (a). In the plot from the separable model (Figure 3-5 (a)), the points fill most regions, except the northeast corner where both ρ_1 and ρ_2 are large. This indicates that the parameters ρ_1 and ρ_2 have large variability when modeled with the separable model. In the plot from the generalized model (Figure 3-5 (b)), the points tend to concentrate in the southeast corner where ρ_1 is typically larger than 0.5 and ρ_2 smaller than 0.5. Significantly fewer points have a ρ_1 less than 0.5 compared to the separable case. This has two implications. First, the variability of parameters ρ_1 and ρ_2 of the auto-covariance is reduced, when modeled with the generalized model. Reduction of variability is important as it can model the source better and may lead to better compression of the source. Second, ρ_1 is typically larger than 0.5 and this means the generalized model can often capture high correlation from the source. The parameter θ adjusts itself such that ρ_1 points along directions with smaller variations than in the separable model. This is consistent with the resampling and lifting methods in [20] and [8], which perform filtering along directions with smaller variations than the predefined horizontal or vertical directions.

3.2.4 Estimated Model Parameters for MC and RE Residuals

Next, consider the scatter plots obtained from the MC and RE residuals shown in Figure 3-5 (c), (d), (e) and (f). The plots obtained using the separable model (Figure 3-5 (c) and (e)) have typically a ρ_1 smaller than 0.5. This is in contrast to the typical ρ_1 in Figure 3-5 (a) which is larger than 0.5. MC and RE residuals usually are more random since they are the parts of images which could not be predicted well, and ρ_1 tends to be smaller.

Even though MC and RE residuals are more random than images, many regions of these types of prediction residuals still have some structure. The separable model can not capture those well and produces a small ρ_1 estimate. Figure 3-5 (d) and (f) show the estimated ρ_1 and ρ_2 when the auto-covariance of MC and RE residuals is modeled with the generalized model. In this case, many more points have a ρ_1 larger than 0.5 compared to the separable case (Figure 3-5 (c) and (e)). The majority of the points

have a large ρ_1 and a small ρ_2.

In summary, if the auto-covariance of MC and RE residuals is modeled with the separable model, estimated ρ_1 (and ρ_2) are both typically small. If the generalized model is used, then typically ρ_1 is large and ρ_2 is small. An estimated large ρ_1 indicates that some structure has been captured from the local region in MC and RE residuals. The combination of a large ρ_1 and a small ρ_2 indicates that the structure exists only along the direction of the ρ_1, indicating a 1-D structure.

3.2.5 Estimated Model Parameters for IP Residuals

The estimated model parameters for the IP residual are shown in Figure 3-5 (g) for the separable model and in Figure 3-5 (h) for the generalized model. Even though the IP residual is also a type of prediction residual, these plots are similar to the plots of the image (Figure 3-5 (a) and (b)) and differ from the plots of the MC and RE residuals. The reason for this can be seen from the picture of the IP residual shown in Figure 3-1 (d). Intra prediction (especially with 8x8 blocks) is typically not as good as motion compensation or interpolation. Intra prediction predicts pixels that are spatially nearby and far away. While nearby pixels can be predicted well, far away pixels typically can not be predicted well. Therefore, the IP residual typically does not have as many 1-D structures as the MC and the RE residuals. However, the mean of the IP residual is typically zero, and often the IP residual looks like a mean-removed image. Since we remove the mean when estimating the non-parametric auto-covariance, the estimated parameters for the image and the IP residual are very similar.

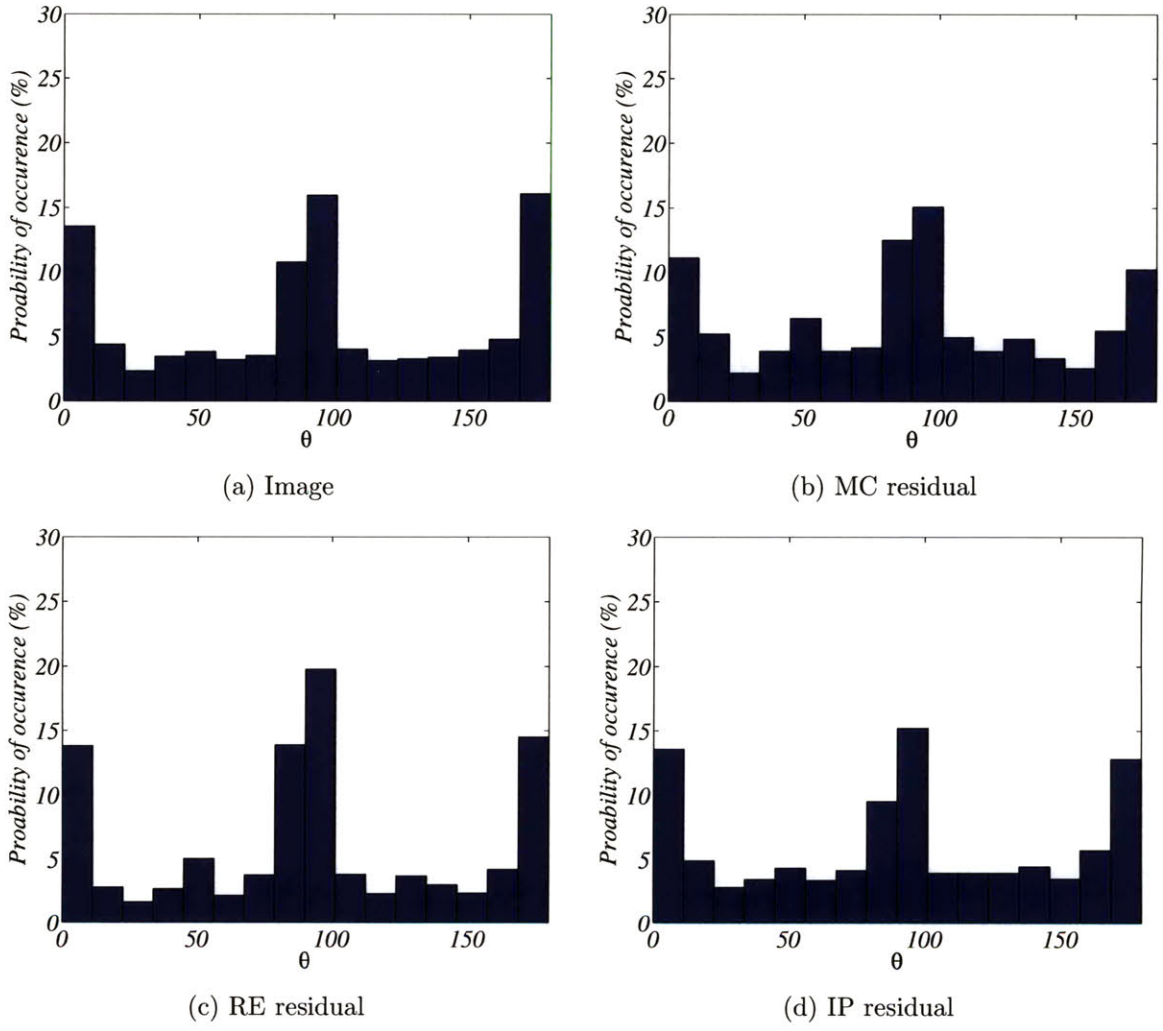
3.2.6 Comparison of Estimated Model Parameters Between Images and MC or RE Residuals

Figure 3-5 also illustrates the difference between the locally anisotropic features of images and MC or RE residuals. Consider the generalized auto-covariance characterization of the image and the MC residual in Figure 3-5 (b) and (d). In both plots, the majority of the points have a ρ_1 larger than 0.5. However, the points in the plot of the MC residual have a smaller ρ_2. In other words, given any (ρ_1, ρ_2)-tuple in the image characterization, the smaller covariance factor becomes even smaller in the MC residual characterization.

The same variation exists also between the plots of the image and RE residual (Figure 3-5 (b) and (f)) and this is a major difference in the statistical characteristics between images and MC or RE residuals. It indicates the difference between the anisotropic characteristics of images and MC or RE residuals; images have 2-D anisotropic characteristics, while MC and RE residuals have 1-D anisotropic characteristics.

3.2.7 Estimated Angles (θ) Using the Generalized Model

We also provide plots of the estimated angles (θ) of the generalized auto-covariance model from the image and prediction residuals of the mobile sequence shown in Figure 3-1. The plots are shown in Figure 3-8. The highest peaks in the plots are at around 0°, 90° and 180°, where peaks at 0° and 180° correspond to horizontally aligned features, and a peak at 90° corresponds to vertically aligned features. This indicates that the image and prediction residual shown in Figure 1-1 have more horizontal and vertical features than features along other directions and this tendency seems to be a common one.



(a) Image

(b) MC residual

(c) RE residual

(d) IP residual

Figure 3-8: Histograms of estimated angles (θ) of the generalized auto-covariance model from the image and prediction residuals of the mobile sequence in Figure 3-1.

Chapter 4

1-D Directional Transforms

Based on the visual inspection of prediction residuals in Section 3.1, a large number of local regions in MC and RE residuals consist of 1-D structures, which follow object boundaries or edges present in the original image. This indicates that using 2-D transforms with basis function that have 2-D support is not the best choice for such regions. We propose to use transforms with basis functions whose support follow the 1-D structures of the MC and RE residuals. Specifically, we propose to use 1-D directional transforms for MC and RE residuals.

The results of the auto-covariance characterization in Section 3.2 support the discussed characteristics and consequently also suggest the use of 1-D directional transforms for MC and RE residuals. The scatter plots of ρ_1 and ρ_2 estimated from MC and RE residuals using the generalized model (Figure 3-5 (d) and (f)) indicate that often one of the two covariance coefficients is significantly larger than the other. If one considers asymptotic cases, where $\rho_1 \rightarrow 1$, $\rho_2 \rightarrow 0$, then these assumptions suggest a transform with decorrelation along the direction of ρ_1 using DCT's. The signal is already close to uncorrelated along the direction of ρ_2, therefore, no decorrelation is necessary along this direction. As a result, the overall transform is 1-D.

Since we compress MC residuals using the H.264/AVC codec in our experiments, we discuss sets of 1-D directional transforms on 8x8-pixel and 4x4-pixel blocks. However, the idea of 1-D transforms for prediction residuals can also be extended to wavelet transforms [17].

The 1-D directional transforms that we use in our experiments are shown in Figures

4-1 and 4-2. Figure 4-1 shows the 1-D block transforms defined on 8x8-pixel blocks and Figure 4-2 shows the 1-D block transforms defined on 4x4-pixel blocks. We have a total of sixteen 1-D block transforms for 8x8-pixel blocks and a total of eight 1-D block transforms for 4x4-pixel blocks.

Each of the 1-D block transforms consists of a number of 1-D patterns which are all roughly directed at the same angle, which corresponds to the direction of the large covariance coefficient. For example, all 1-D patterns in the fifth 1-D block transform defined on 8x8-pixel blocks or the third 1-D block transform defined on 4x4-pixel blocks are directed towards south-east. The angle is different for each of the 1-D block transforms and altogether they cover 180°, for both 8x8-pixel blocks and 4x4-pixel blocks. Each 1-D pattern in any 1-D block transform is shown with arrows in Figures 4-1 and 4-2, and defines a group of pixels over which a 1-D DCT is performed.

Note that 1-D patterns in some block transforms (for example, first and fifth in Figure 4-1) are straight, in others they are not. In the first class of block transforms, lines along the direction of the large covariance coefficient pass through full-pixel locations, and in the second class of block transforms, they do not. For the second class of transforms, we have chosen to use 1-D patterns which approximate the desired straight 1-D patterns. An alternative approach could be to interpolate intensities at sub-pixel locations so that virtual straight 1-D patterns are created. However, this approach has two drawbacks. First, the overall block transform loses orthogonality. Second, and more importantly, the interpolated values do not provide better energy compaction. This is because the interpolated value is obtained by filtering across 1-D patterns. However, the correlation across 1-D patterns is weak as this direction aligns with the direction of the small covariance coefficient. In some preliminary experiments, we observed that this alternative approach provides inferior compression than the block transforms with approximate 1-D patterns.

Even though 1-D directional transforms improve the compression of MC and RE residuals for many regions, the 2-D DCT is essential. There exist regions in these prediction residuals which can be better approximated with 2-D transforms. Therefore, in our experiments, we use both 1-D directional transforms and the 2-D DCT. Encoders with 1-D transforms have access to 2-D DCT and can adaptively choose to use one among the available 1-D transforms and the 2-D DCT.

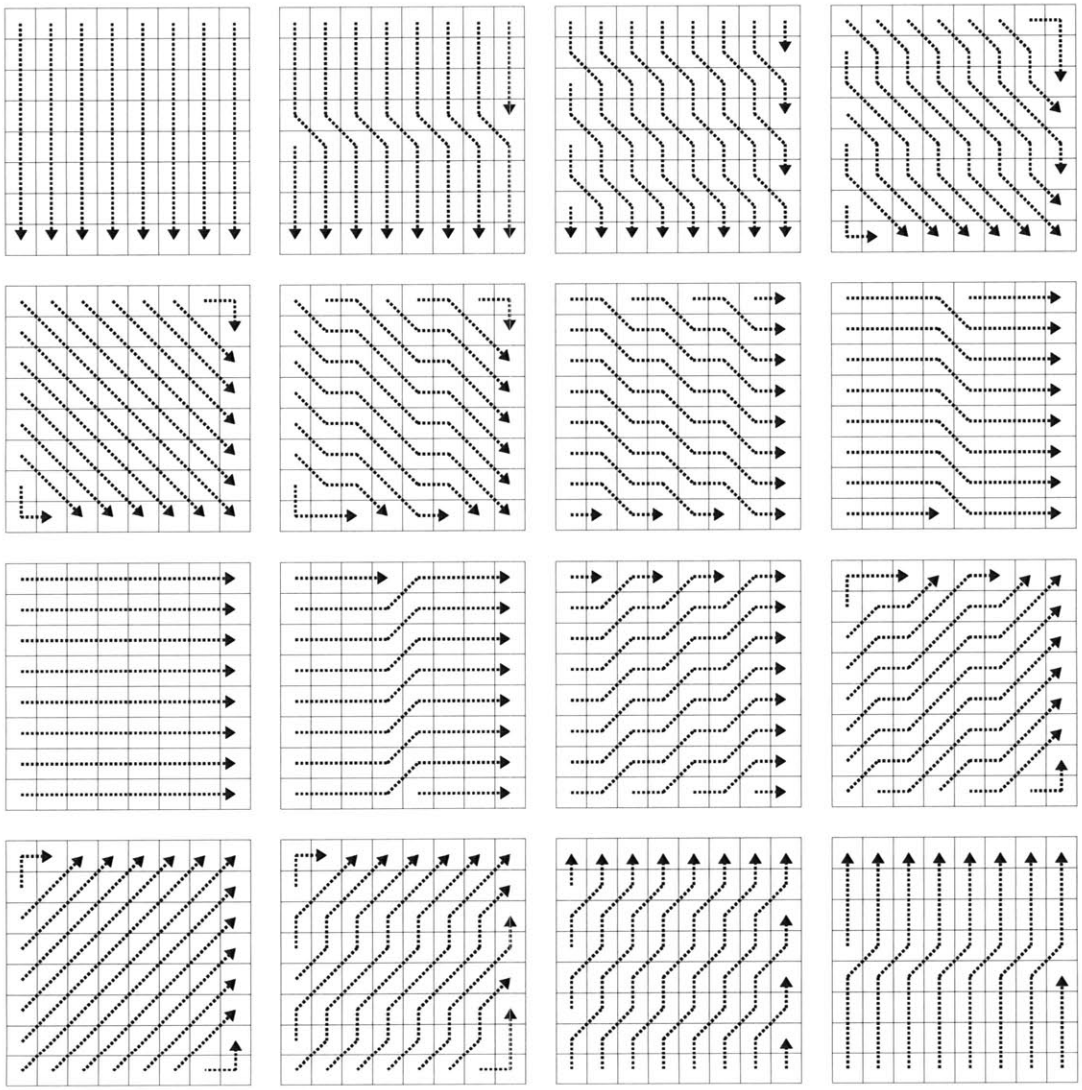


Figure 4-1: Sixteen 1-D directional block transforms defined on 8x8-pixel blocks. Each transform consists of a number of 1-D DCT's defined on groups of pixels shown with arrows. Arrangement of groups of pixels determines the direction of each block transform and the direction of the first block transform (top left block) is the horizontal direction. The direction of the next transform moves a step in the counter clockwise direction from the direction of the previous block transform and directions of all transforms together cover 180°.

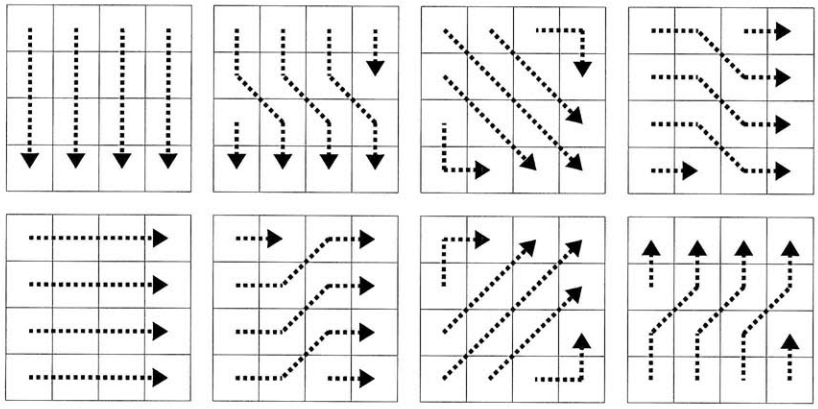


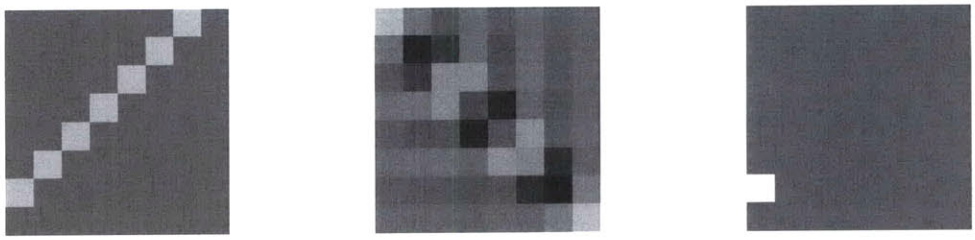
Figure 4-2: Eight 1-D directional block transforms defined on 4x4-pixel blocks. Each transform consists of a number of 1-D DCT's defined on groups of pixels shown with arrows. Arrangement of groups of pixels determines the direction of each block transform and the direction of the first block transform (top left block) is the horizontal direction. The direction of the next transform moves a step in the counter clockwise direction from the direction of the previous block transform and directions of all transforms together cover 180°.

To show the effectiveness of the proposed transforms we present three examples in Figures 4-3, 4-4 and 4-5. Figure 4-3 (a) shows a sample residual block, Figure 4-3 (b) shows the transform coefficients obtained by transforming the block with the 2-D DCT, and Figure 4-3 (c) shows the transform coefficients obtained by transforming the block with a 1-D transform aligned with the structure in the residual (the specific transform used is 1-D Transform #13 in Figure 4-1). The mid-gray level in these figures represents zero, and the residual block consists of an artificially created 1-D structure aligned diagonally. Such a residual block can possibly be obtained from the prediction of a local region which contains an edge separating two smooth regions in the original image block. To represent this residual block, 2-D DCT requires many nonzero transforms coefficients while the 1-D transform requires only one nonzero transform coefficient.

The second example is shown in Figure 4-4. The residual block in this example consists of a vertical 1-D structure. Figure 4-4(c) shows the transform coefficients obtained by transforming the block with a 1-D transform aligned with the vertical structure in the residual (the specific transform used is 1-D Transform #1 in Figure 4-1), and this block can be represented with a single nonzero transform coefficient. The transform coefficients obtained by transforming the block with the 2-D DCT are shown in Figure 4-4(b). We note that the separable 2-D DCT can be performed by first applying 1-D transforms along the vertical dimension and then applying 1-D transforms along the horizontal dimension. The first set of horizontal 1-D transforms is equivalent to the 1-D transform used in Figure 4-4(c). As a result, when performing the separable 2-D DCT, the result of the first set of vertical 1-D transforms provides already a good representation of the block (since only a single nonzero coefficient suffices, as shown in Figure 4-4(c)), and applying the second set of horizontal 1-D transforms results in more nonzero coefficients. In summary, for residual blocks with a 1-D structure, even if the alignment of the structure is consistent with the directions of the 2-D transform, 1-D transforms can represent such blocks better.

The third example is shown in Figure 4-5. The residual block in this example is taken from a motion compensated prediction residual (shown in Figure 3-2 (b)) and has a structure aligned along a direction roughly between south-east and south. The structure in this example is not as obvious and clean as the one in Figure 4-3 because actual prediction residuals contain noise from various sources. The superiority of 1-D transforms for such residual blocks is not as large as for the artificial residual block

in Figure 4-3, but is still significant. Figure 4-5 (b) shows the transform coefficients obtained by transforming the block with the 2-D DCT, and Figure 4-5 (c) shows the transform coefficients obtained by transforming the block with a 1-D transform aligned with the structure in the residual. To represent the residual block, the 1-D transform requires fewer large transform coefficients than the 2-D DCT. The upper left corner of the coefficient block obtained with the 1-D transform contains a bright pixel, which represents a large transform coefficient that alone can capture a significant fraction of the energy of the block. Figure 4-6 shows the fraction of retained energy as a function of the number of retained transform coefficients for both the 2-D DCT and the 1-D transform. The single large coefficient of the 1-D transform can account for more than half of the total energy of the residual block and to retain an equal fraction of energy, the 1-D transform always requires fewer coefficients. To capture 75% of the total energy, the 1-D transform requires about half as many coefficients as the 2-D DCT.



(a) Residual block

(b) Transform coefficients obtained with 2-D DCT

(c) Transform coefficients obtained with 1-D Transform

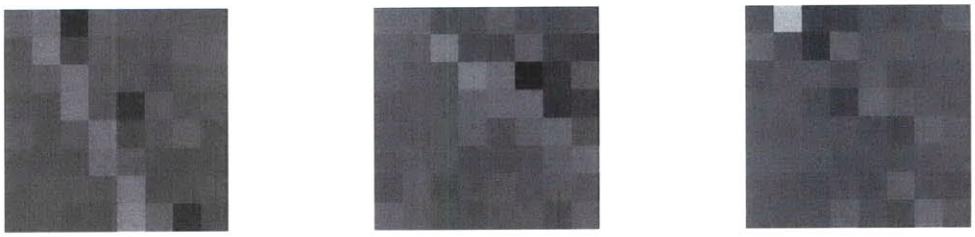
Figure 4-3: Comparison of 2-D DCT and 1-D directional transform on an artificial residual block consisting of a 1-D structure (mid-gray level represents zero). To represent the residual block, 2-D DCT requires many nonzero transform coefficients while the 1-D transform requires only one nonzero transform coefficient.

(a) Residual block

(b) Transform coefficients obtained with 2-D DCT

(c) Transform coefficients obtained with 1-D Transform

Figure 4-4: Comparison of 2-D DCT and 1-D directional transform on an artificial residual block consisting of a vertical 1-D structure (mid-gray level represents zero). To represent the residual block, 2-D DCT requires many nonzero transforms coefficients while the 1-D transform requires only one nonzero transform coefficient.



(a) Residual block

(b) Transform coefficients obtained with 2-D DCT

(c) Transform coefficients obtained with 1-D Transform

Figure 4-5: Comparison of 2-D DCT and 1-D directional transform on a residual block with a 1-D structure taken from the motion compensated prediction residual frame shown in Figure 3-2 (b) (mid-gray level represents zero). To represent the residual block, 1-D transform requires fewer large transform coefficients than the 2-D DCT.

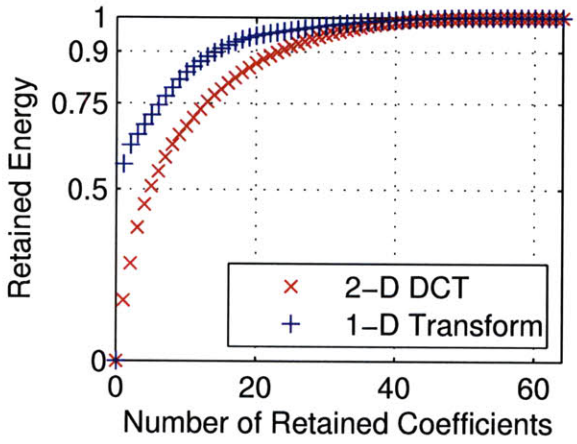


Figure 4-6: Fraction of retained energy as a function of the number of retained transform coefficients for the residual block in Figure 4-5. A single coefficient of the 1-D transform can account for more than half of the total energy of the residual block.

Chapter 5

System Implementation with 1-D Directional Transforms

To examine the performance of the proposed 1-D transforms, a number of related aspects need to be carefully designed. These include the implementation of the transforms, quantization of the transform coefficients, coding of the quantized coefficients, coding of the side information which indicates the selected transform for each block, rate-distortion optimized selection of transforms, and the overall increase in the complexity of the codec. This section discusses these aspects.

Based on the analysis of different types of prediction residuals, we proposed 1-D directional transforms for motion compensation (MC) and resolution enhancement (RE) residuals. In order to estimate a proper performance of the proposed transforms, they need to be implemented in an actual codec and realistic simulations need to be performed. Recent video coding standards are quite complex due to the use of many complex and adaptive coding tools and integrating new transforms into such codecs is a difficult and time consuming task. In order to keep this task reasonable for the simulations in this thesis, we refrained from simulations with RE residuals which require implementation in a scalable codec. We present simulation results with the much more widely coded MC residuals using an H.264/AVC codec (JM reference software 10.2) that is modified according to the discussion in this chapter. Even though IP residuals within H.264/AVC do not have as many 1-D structures as MC or RE residuals, we also present simulation results with IP residuals since these are readily available within our codec

and application of our transforms on these residuals is feasible. The experimental results will be presented in the next chapter, and this chapter discusses the aspects that need to be designed in order to integrate the proposed 1-D transforms in to the used codec.

The system used for the experiments in thesis is based on the H.264/AVC video coding standard because it is the most advanced video coding standard at the time this research was conducted. H.264/AVC makes use of state-of-the-art video coding techniques and has been shown to significantly increase coding efficiency relative to previous standards. By using a system based on H.264/AVC, the results presented in this thesis can more easily be compared against current and future work. An excellent review of new coding tools in H.264/AVC can be found in [52, 37].

5.1 Implementation of Transforms

Discrete cosine transforms can be implemented using fast algorithms [1, 22]. Since our 1-D directional transforms consist of 1-D DCT's, these fast algorithms can be used in the implementation of our 1-D transforms as well. In H.264/AVC, transform and quantization are merged so that these computations can be implemented with integer arithmetic using addition, subtraction and bitshift operations. This has many advantages including the reduction of computational complexity [52, 25]. The computational complexity is not important in this thesis, and we use floating point operations for these computations. This does not change the results. We note that it is possible to merge the transform and quantization for our proposed 1-D transforms so that these computations can also be implemented with integer arithmetic.

5.2 Coding of 1-D Transform Coefficients

After prediction, transformation, and quantization of residual transform coefficients, the compressed video information (quantized residual transform coefficients, motion vectors, coding modes of blocks, etc.) must be converted to a stream of bits with as few bits as possible using entropy coding techniques. The H.264/AVC standard provides two entropy coding options; universal variable length coding (UVLC) and context adaptive binary arithmetic coding (CABAC).

The simpler UVLC approach uses exponential Golomb codes for all syntax elements except for transform coefficients [37]. Each syntax element is assigned a nonnegative integer code number, with more probable outcomes assigned to smaller code numbers. Given a code number the associated codeword can easily be obtained, and given a codeword the associated code number can easily be obtained; there is no need for storing codeword tables or searching in codeword tables. The residual transform coefficients are first scanned into a one-dimensional array and the array is encoded using context-adaptive variable length coding (CAVLC). These variable length codes are similar to Huffman codes, designed according to the characteristics of the residual transform coefficients. They are also adapted both to the local region by using multiple codeword tables (each adapted to smooth and busy regions), and to the context by using multiple codeword tables dependent on the previously coded syntax elements.

The second available entropy coding option in H.264/AVC is based on context-adaptive binary arithmetic coding (CABAC). This approach provides increased efficiency relative to the CAVLC approach at an increased complexity. Arithmetic coding in a sense allows for joint encoding of many syntax elements, including the possibility of using less than one bit per syntax element. In addition, this approach estimates the probabilities of syntax elements from previously coded syntax elements and allows adaptation to the characteristics/probabilities of the particular video sequence being encoded. CABAC encoding is also used on a wide range of syntax elements including residual transform coefficients. Further details on CABAC can be found in [26].

Depending on the chosen entropy coding mode in H.264/AVC, the quantized transform coefficients can be encoded using either context-adaptive variable-length coding (CAVLC mode) or context-adaptive binary arithmetic coding (CABAC mode). In both cases, coding methods are adapted to the characteristics of the coefficients of the 2-D DCT. It is desirable to adapt the coding methods to the proposed 1-D transforms by designing entirely new coefficient coding algorithms that are thoroughly adapted to characteristics of 1-D transforms. For the experiments in this thesis, however, we use a simple adaptation scheme. We use the same coefficient coding method in H.264/AVC in CAVLC mode with the exception of the scan. We use different scans adapted to each of the 1-D transforms.

Figure 5-1 shows the scans that we use when coding the coefficients of 8x8-pixel

block 1-D transforms. For transforms defined on 8x8-pixel blocks, H.264/AVC generates four length-16 scans instead of one length-64 scan when entropy coding is performed in CAVLC mode, and we have four length-16 scans in each block in Figure 5-1. Scans in each block belong to the corresponding transforms in Figure 4-1. These scans were designed based on two considerations. The first is that coefficients less likely to be quantized to zero are closer to the front of the scan and coefficients more likely to be quantized to zero are closer to the end of the scan. The second consideration is that neighboring 1-D patterns are grouped into one scan. The 1-D structures in prediction residuals are typically concentrated in one region of the 8x8-pixel block and the 1-D transform coefficients representing them will therefore be concentrated in a few neighboring 1-D patterns. Hence, grouping neighboring 1-D patterns into one scan enables capturing those 1-D transform coefficients in as few scans as possible. More scans that consist of all zero coefficients can lead to more efficient overall coding of coefficients.

Figure 5-2 shows the scans for the 1-D transforms defined on 4x4-pixel blocks shown in Figure 4-2. Similarly, these scans were designed so that coefficients less likely to be quantized to zero are closer to the front of the scan and coefficients more likely to be quantized to zero are closer to the end of the scan.

5.3 Rate-distortion Optimized Transform Selection

An important question that arises with multiple available transforms is how exactly to choose the best transform for each local region. Similar problems are often encountered in video coding. To increase the coding efficiency, multiple coding options or modes are typically available, each performing particularly well in specific cases, and the best option needs to be determined in order to code each local region most efficiently. For example, in H.264/AVC a number of block sizes are available to perform motion compensated prediction and the best block size needs to be determined for each macroblock.

The selection of the best coding modes can be cast as a budget constrained resource allocation problem. For a given total rate R_T, determine the modes for each block i so that the distortion metric $\sum_i D_i$ is minimized subject to the bitrate constraint $\sum_i R_i \leq R_T$. Here D_i represents the distortion (typically measured using the mean square error metric) of each block i and R_i represents total number of bits used to code

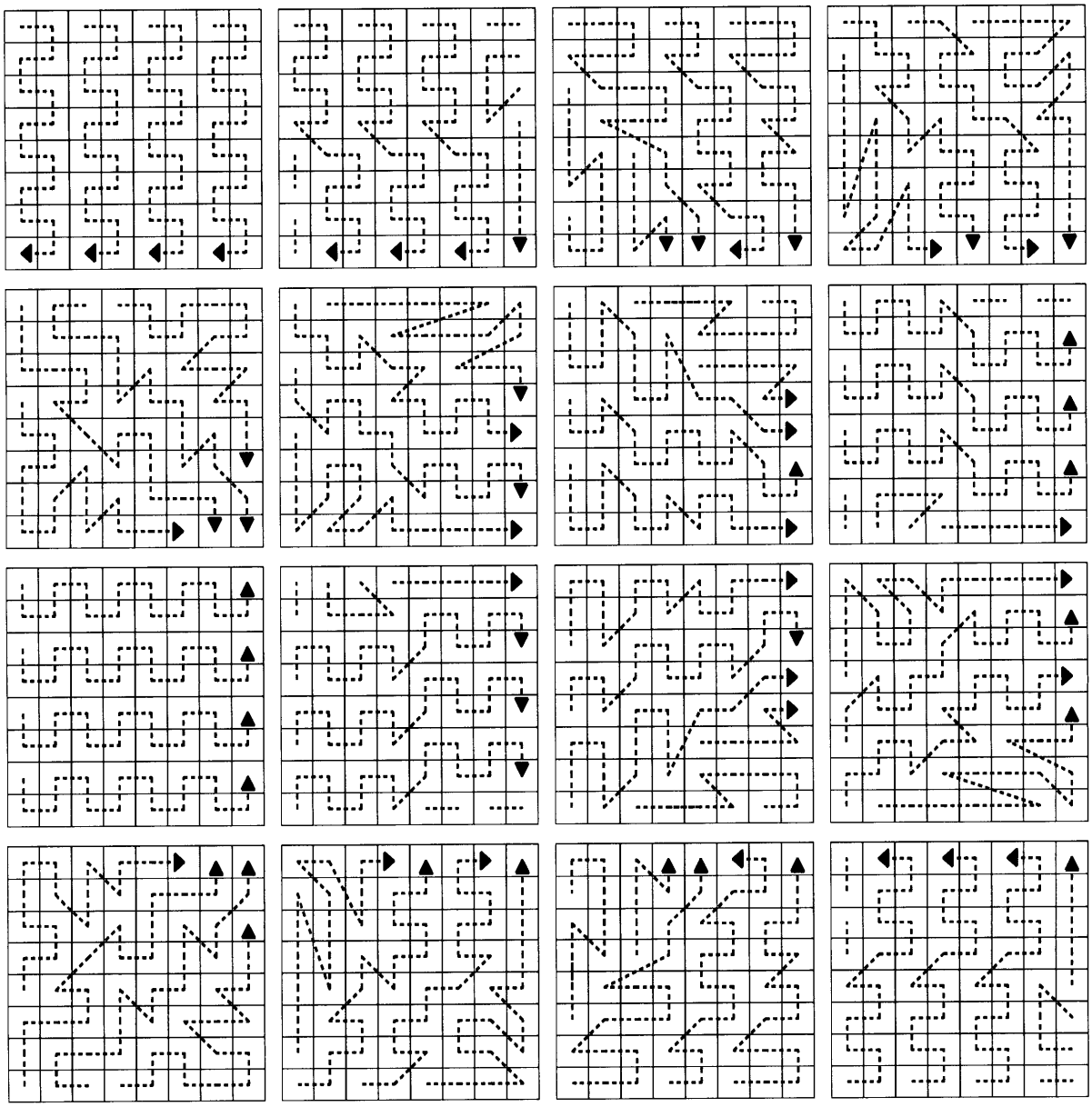


Figure 5-1: Scans used in coding the quantized coefficients of 1-D transforms defined on 8x8-pixel blocks.

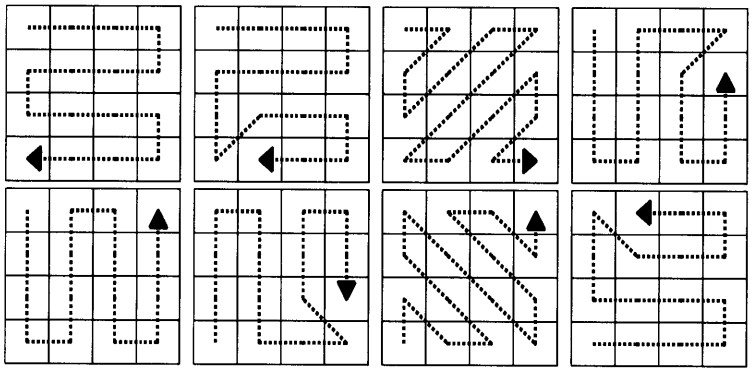


Figure 5-2: Scans used in coding the quantized coefficients of 1-D transforms defined on 4x4-pixel blocks.

each block i. The classic solution to this problem is given by Lagrangian optimization theory [5, 32]. This theory states that if a particular set of modes minimizes the so-called Lagrangian cost given by equation (5.1), then the same set of modes also solve the budget constraint resource allocation problem.

$$J(\lambda) = \sum_i (D_i + \lambda R_i) \tag{5.1}$$

In the problem formulation above, it is ideally desired to include all blocks within a sequence in the sum in equation (5.1) because the distortion and bitrates of blocks are not independent. For example, motion compensated prediction introduces dependencies between frames and intra prediction introduces dependencies between blocks within a frame. However, taking into account such dependencies makes the solution of this problem infeasible from a computational point of view, and typical video encoders assume that each macroblock is independent. This assumption allows the resource allocation problem to be solved separately for each macroblock, and the best coding mode of each macroblock can be determined by minimizing the Lagrangian cost of that individual macroblock, shown in equation (5.2).

$$J_i(\lambda) = D_i + \lambda R_i. \tag{5.2}$$

An important aspect of the Lagrangian based solution is the determination of the parameter λ. This parameter represents a trade-off between the distortion D_i and the

bitrate R_i. Setting λ to zero ignores the rate and results in minimizing the distortion. Setting λ arbitrarily high ignores the distortion and results in minimizing the bitrate. Wiegand et.al. studied Lagrangian based Rate-Distortion optimized selection of coding modes in H.264/AVC and proposed a method to determine the values of λ depending on the used quantization parameter [40, 51]. This method is used in the reference software of H.264/AVC that we use in our experiments.

To select the best transforms in our system, we also employ the Lagrangian-based optimization approach. We note that in our system, there are two resource allocation problems to be solved. One is the selection of the best mode (i.e. best block size for motion compensated prediction) for each macroblock and the other is the selection of the best transforms for each block. We solve these problems jointly. For each possible mode we select the best transform. Then we select the best mode given the best transforms and the best mode-transform combination is determined. The value of λ that we use when selecting the best transform for a given mode is the same value that is used when selecting the best mode, because after selecting the best transform for the given mode, the used distortion and bitrate will be reused when forming the Lagrangian cost to select the best mode.

5.4 Coding of Side Information

The selected transform for each block needs to be transmitted to the decoder so that the decoder can use the correct inverse transform for each block. We refer to this transmitted information as the side information. Since we use CAVLC mode for entropy coding in our experiments, we use variable-length codes (VLC) to code the side information. To use a low bitrate for the side information, probabilities of selection of the transforms are needed. Conditional probabilities, conditioned on useful information available at both the encoder and the decoder, can be more useful but we do not consider them in this thesis. Some preliminary experiments have shown that if the side information bits are neglected, probabilities of selection of 1-D transforms and the 2-D DCT are similar, where the probabilities for the horizontally and vertically aligned 1-D transforms and the 2-D DCT are a bit higher than the others. Codewords based on such probabilities would result in similar length codewords.

Table 5.1: Codewords to indicate selected transforms

Transform	Codeword
2-D DCT	1
1-D Transform #1-16	0XXXX

(a) 8x8-block transforms

Transform	Codeword
2-D DCT	1
1-D Transform #1-8	0XXX

(b) 4x4-block transforms

However, we do not have a simple lossless source coding problem here. While the codewords for each transform should be determined according to the probabilities of selection of the transforms, the probabilities of selection are also dependent on the codewords for each transform. This is because the lengths of the codewords affect the RD cost for choosing the best transform in each block, and thus the probabilities. Our ultimate goal is to improve the RD efficiency of the codec and we have determined that it is useful to use a codeword as short as possible for a transform that performs well for a wide range of local regions in the prediction residual and such a transform is the 2-D DCT. The 1-D directional transforms use longer codewords and are chosen if they perform well enough so as to compensate for their longer codewords. As a result, this approach improves the RD efficiency of the codec, but the probabilities of selection of transforms will be biased.

Based on the discussion above, we code the side information in our experiments as shown in Table 5.1. If a macroblock uses 8x8-pixel transforms, then for each 8x8-pixel block, the 2-D DCT is represented with a 1-bit codeword, and each of the sixteen 1-D transforms is represented with a 5-bit codeword. If a macroblock uses 4x4-pixel transforms, then for each 4x4-pixel block, the 2-D DCT can be presented with a 1-bit codeword and each of the eight 1-D transforms can be represented with a 4-bit codeword. Alternatively, four 4x4-pixel blocks within a single 8x8-pixel block can be forced to use the same transform. This allows to represent the selected transforms for these four 4x4-pixel blocks with a single 4-bit codeword. This reduces the average bitrate for the side information but will also reduce the flexibility of transform choices for 4x4-pixel blocks. In our experiments, we use this alternative method of forcing 4x4-pixel blocks with a single 4-bit codeword because it usually gives slightly better results.

5.5 Complexity Increase

Having a number of transforms to choose from increases the complexity of the codec. An important consideration is the increase in encoding time. This increase depends on many factors of the implementation and can therefore vary considerably. Our discussion of the increase in encoding time is based only on the reference software of H.264/AVC in high complexity encoding mode.

In high-complexity encoding mode, RD-optimized encoding is performed, where each available coding option for a macroblock or smaller blocks is encoded and the option(s) with the smallest RD-cost is chosen. The implementation within the reference software is designed for general purpose processors and executes each command successively, with no parallel processing support. Therefore, each coding option is encoded successively. Within each coding option, each block is encoded with each available transform. Hence, the amount of time spent on transform (T), quantization (Q), entropy coding of quantized coefficients (E), inverse quantization (Q), and inverse transform (T) computations increases linearly with the number of available transforms. The factor of increase would be equal to the number of transforms if the computation of the additional transforms (and inverse transforms) takes the same amount of time as the conventional transform. Because the conventional transform is 2-D while our proposed transforms are 1-D, the factor of increase can be represented with αN_{tr}, where N_{tr} is the number of transforms and α is a scaling constant less than 1. The increase of the overall encoding time is typically equal to the increase in TQEQT computation time because other relevant computations, such as computing the RD-cost of each transform, are negligible.

The TQEQT computation time is a fraction of the overall encoding time and the factor of increase of the overall encoding time depends on this fraction when only the conventional transform is used. In our experiments on P-frames with 8x8-block transforms, about 30% of the encoding time is used on TQEQT computations with the conventional transform. The increase in encoding time with the sixteen additional 1-D transforms is a factor of 5.8 (=17α30%+70% where $\alpha = 1$). The actual increase is expected to be significantly less than 5.8 with a more accurate choice of α and integer-point implementations of transform computations. As mentioned earlier, we used floating point computations without fast algorithms for our 1-D transforms. With the 8x8-block integer transform in H.264/AVC, TQEQT computations take only about 6% of the entire encoding time.

If we assume that our 1-D transforms are similarly implemented in integer arithmetic, then the factor of increase in the overall encoding time would be 1.9 ($=17\alpha 6\% + 94\%$ where $\alpha = 1$). For 4x4-block transforms, TQEQT computations take about 8% of the entire encoding time and the factor of increase in the overall encoding time would be 1.6 ($=9\alpha 8\% + 92\%$ where $\alpha = 1$) if 1-D transforms were implemented in integer arithmetic with fast algorithms.

The decoding time does not increase. The decoder still uses only one transform for each block, which is the transform that was selected and signaled by the encoder. Indeed the decoding time can decrease slightly because the decoder now uses 1-D transforms for some blocks and 1-D transforms require less computations than the 2-D DCT.

Chapter 6

Experimental Results and Analysis

This chapter presents experimental results to illustrate the compression efficiency of the proposed 1-D directional transforms on motion compensation (MC) and intra prediction (IP) residuals using an H.264/AVC codec (JM reference software 10.2) modified according to the discussion in Chapter 5. As discussed in the same chapter, no results are provided for resolution enhancement (RE) residuals as this requires the modification of a different codec. We would like to point out that the results provided here were obtained with one particular set of 1-D transforms and one particular implementation of entropy coding the transform coefficients and the side information. The results are intended to demonstrate that there can be considerable gains from using 1-D directional transforms and it is possible that more optimal realizations of these systems can potentially increase compression efficiency.

6.1 Setup for Experiments

We compare the compression efficiency of the proposed transforms with the compression efficiency of the conventional transform (2-D DCT). We also study the effect of the size of the blocks for the transforms. Each encoder in our experiments has access to a different set of transforms which vary in size and in type. The available sizes are 4x4 and/or 8x8. The available types are *dct* (2-D DCT) or $1D$ (1-D directional transforms). Note that encoders with $1D$ type transforms still have access to the conventional transform, as discussed in Chapter 4. As a result, we have the following encoders.

- 4x4-dct
- 4x4-1D (includes 4x4-dct)
- 8x8-dct
- 8x8-1D (includes 8x8-dct)
- 4x4-and-8x8-dct
- 4x4-and-8x8-1D (includes 4x4 and 8x8-dct)

We use 11 QCIF (176x144) resolution sequences at 30 frames-per-second (fps), 4 CIF (352x288) resolution sequences at 30 fps, and one 720p (1280x720) resolution sequence at 60 fps. The first frame of each of these sequences is shown in Figures 6-1, 6-2 and 6-3. All sequences are encoded at four different picture quality levels (with quantization parameters 24, 28, 32 and 36), which roughly corresponds to a range of 30dB to 40dB. The lower end of this range roughly corresponds to typical low quality video streaming applications often encountered on the Internet and the higher end roughly corresponds to typical broadcast quality. As discussed in Chapter 5, entropy coding is performed with context-adaptive variable length codes (CAVLC) and Rate-distortion (RD) optimization is used to choose the most efficient coding mode and transforms in each local region. Specifically, each macroblock is coded with every possible mode (e.g. 16x16-block MC, 16x8-block MC, 8x16-block MC etc..) and within each mode, each transform is used to determine the best coding mode and transform combination.

For MC residual experiments, we encode the first 20 frames from the 720p sequence and the first 180 frames from all other sequences. The first frame is encoded as an I-frame, and all remaining frames are encoded as P-frames. Since these experiments focus on the MC residual, the intra coded macroblocks use always the 2-D DCT and the inter coded macroblocks choose one of the available transforms for each block. Motion estimation is performed with quarter-pixel accuracy and the full-search algorithm using all available block-sizes. For IP residual experiments, all coded frames are I-frames, and all available transforms can be used for intra coded macroblocks here. We encode 5 frames from each sequence for IP residual experiments.

We evaluate the results with bitrate (in kbit/sec) and PSNR (in dB). The bitrate includes all encoded information including transform coefficients from luminance and

(a) Bridge close

(b) Carphone

(c) Claire

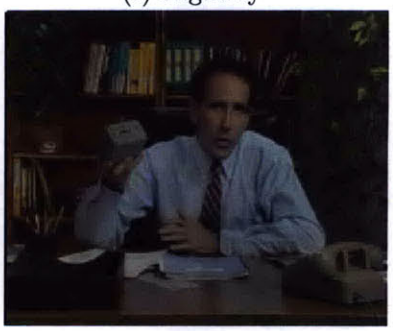
(d) Container

(e) Foreman

(f) Highway

(g) Miss America

(h) Mother daughter

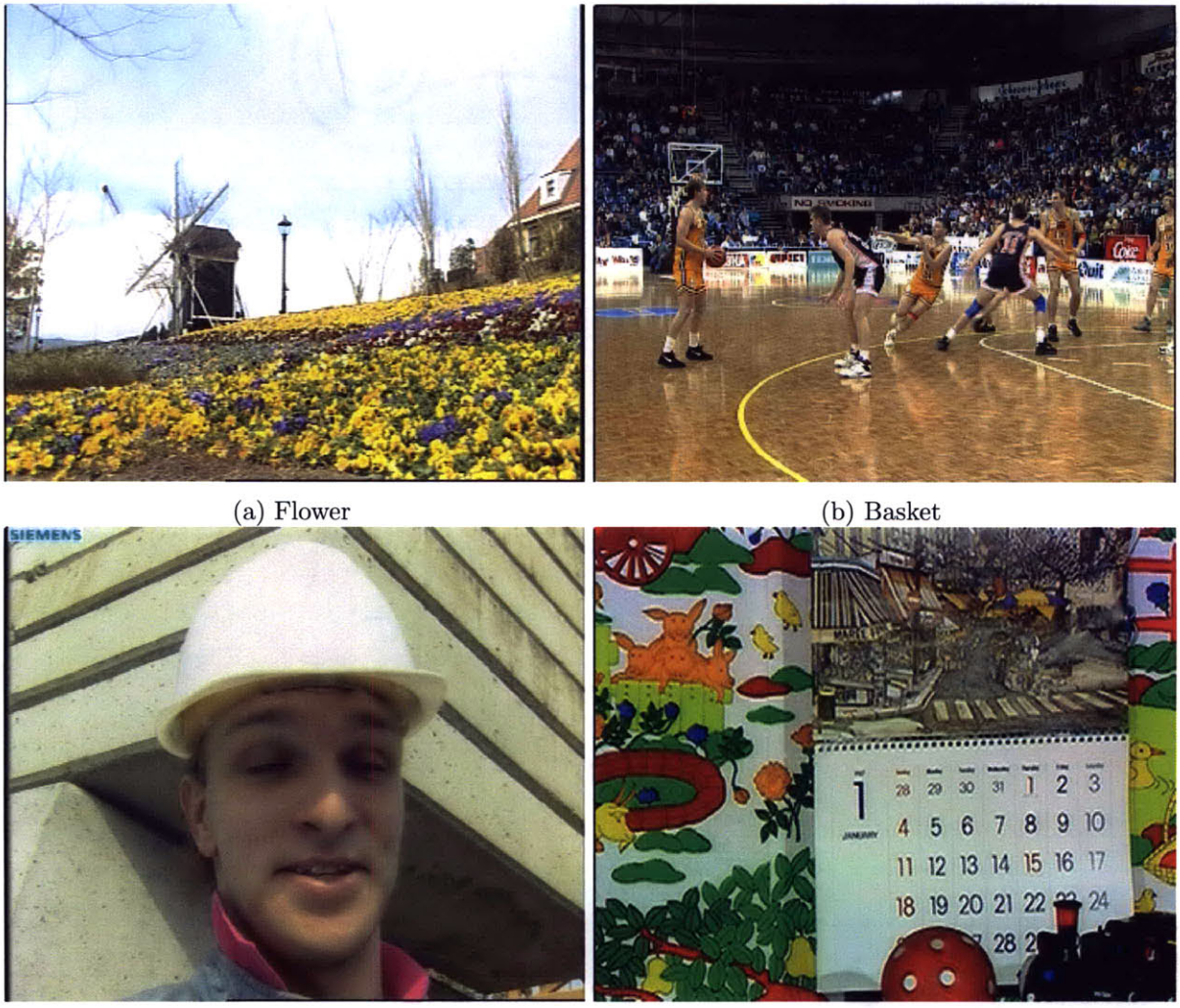


(i) Salesman

(j) Suzie

(k) Trevor

Figure 6-1: QCIF resolution (176x144) sequences used in the experiments.



(a) Flower (b) Basket

(c) Foreman (d) Mobile

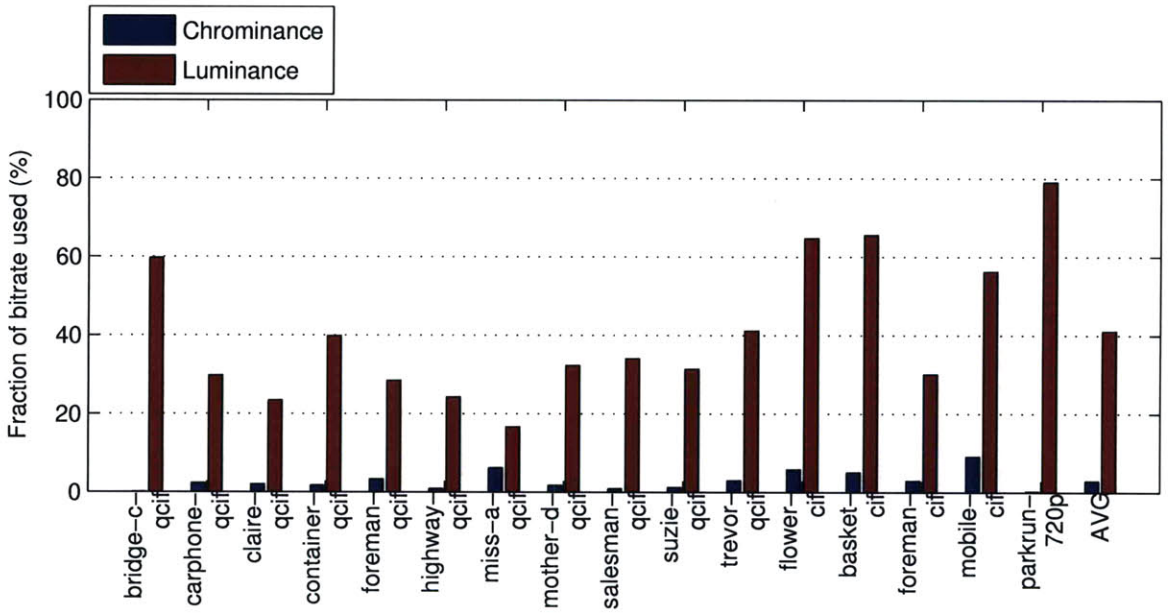
Figure 6-2: CIF resolution (352x288) sequences used in the experiments.

(a) Park run

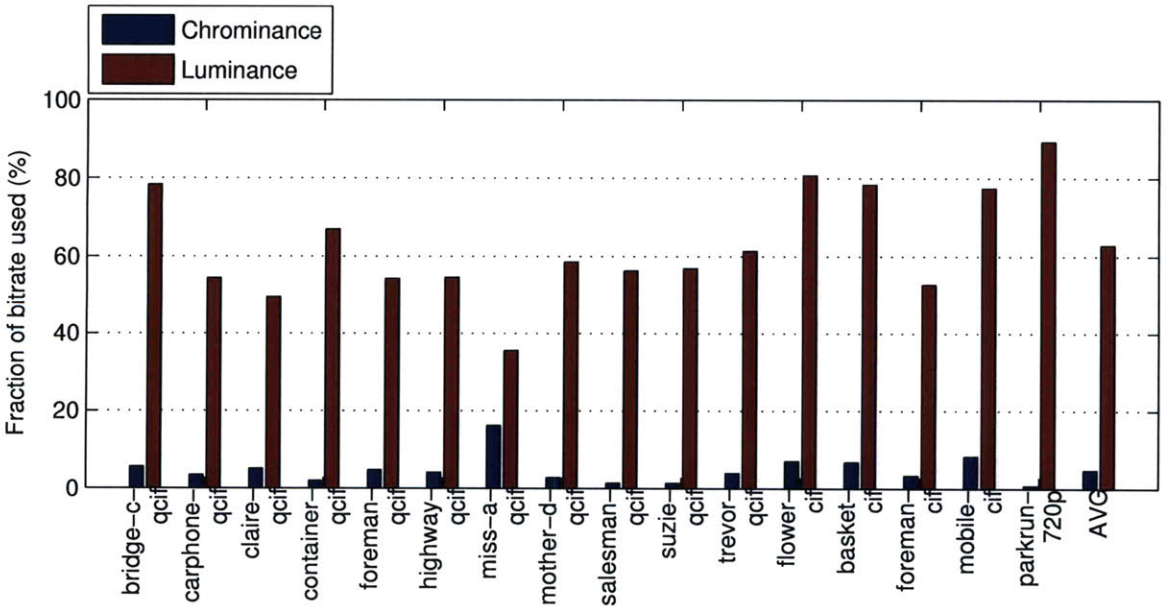
Figure 6-3: HD resolution (1280x720) sequence used in the experiments.

chrominance components, motion vectors, side information for chosen transforms, and all necessary syntax elements and control information. The PSNR, however, is computed from only the luminance component. The proposed transforms are used only for the luminance component, and coding of chrominance components remains unchanged. Figures 6-4 and 6-5 show the fractions of the total bitrate used to code luminance and chrominance residual data using the 4x4-and-8x8-dct encoder for MC and IP residuals. The fractions shown include the bitrate used to code only the transform coefficients of the luminance and chrominance residual data, and any other information such as prediction modes, motion vectors or coded block patterns are not included. It can be seen that the chrominance residual data occupies a small fraction of the entire bitrate and exploration of gains achievable from encoding it with 1-D transforms remains for future research.

Section 6.2 presents experimental results for MC residuals and Section 6.3 presents experimental results for IP residuals. In both sections, the presented results consist of Rate-distortion plots, average bitrate savings achieved, bitrate used to code the side information, probabilities of selection of transforms, and the evaluation of visual quality. Finally, Section 6.4 compares the proposed 1-D directional transforms with a specific type of 2-D directional transforms (originally proposed for image compression) on MC and IP residuals.

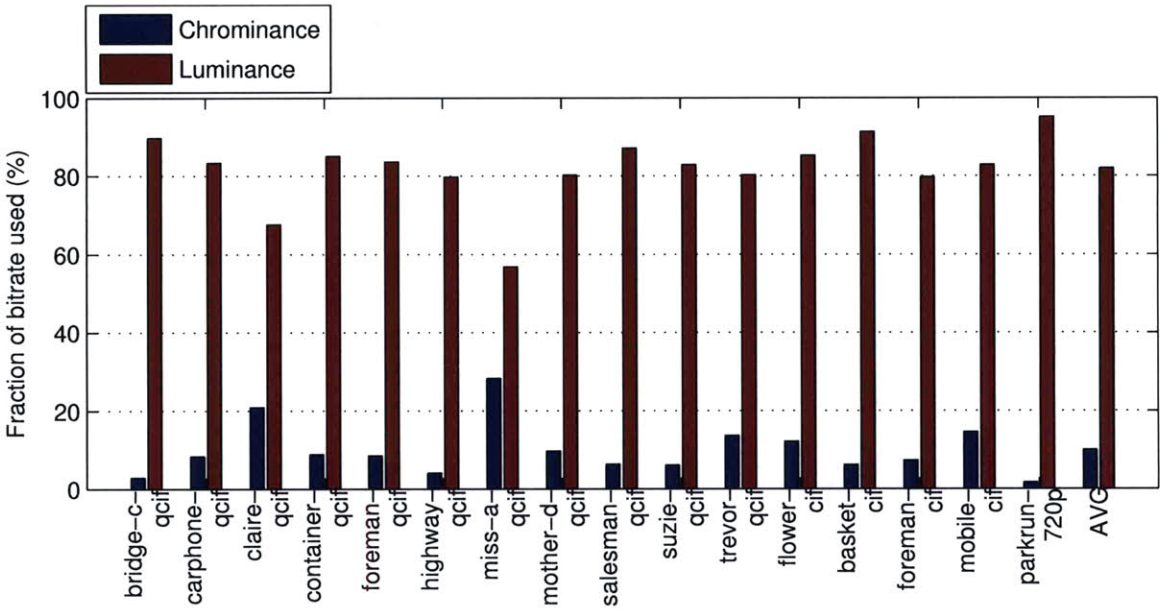


(a) Low picture quality (QP=36)

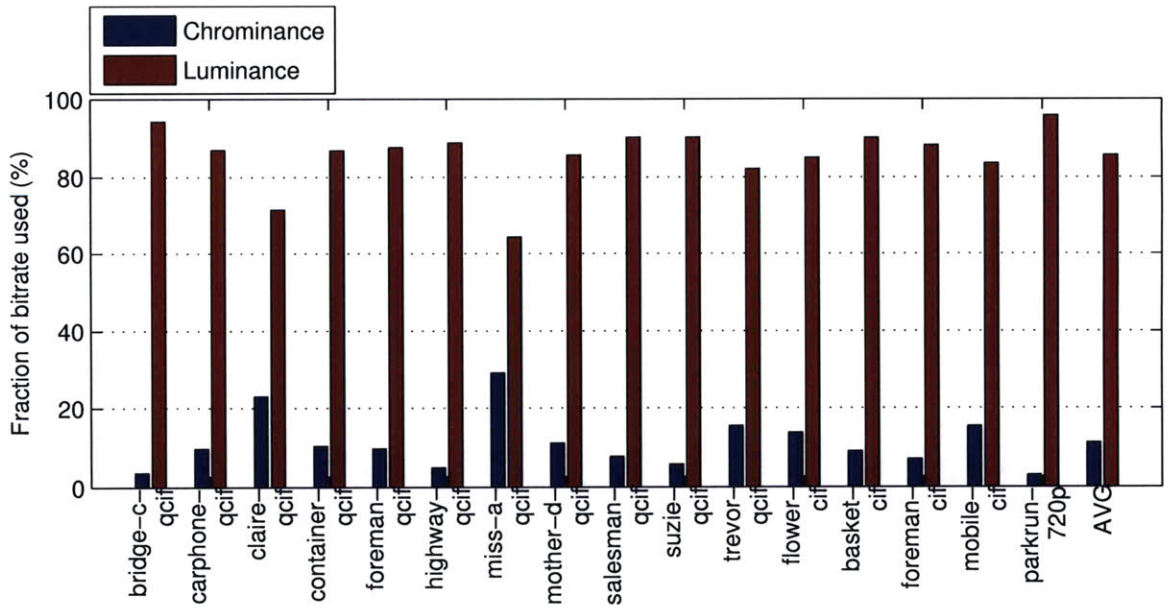


(b) High picture quality (QP=24)

Figure 6-4: Fraction of total bitrate used to code motion compensated luminance and chrominance residual data at low and high picture qualities.



(a) Low picture quality (QP=36)



(b) High picture quality (QP=24)

Figure 6-5: Fraction of total bitrate used to code intra predicted luminance and chrominance residual data at low and high picture qualities.

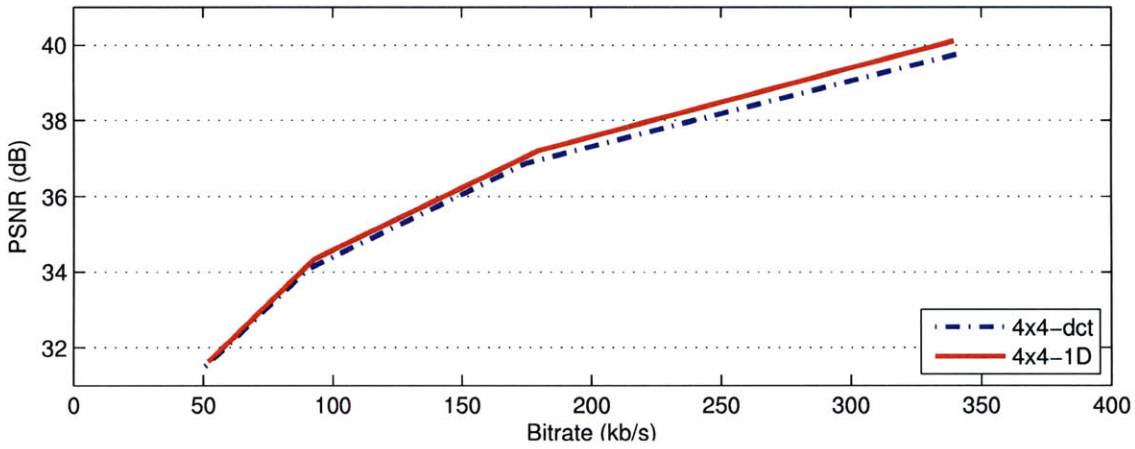
6.2 MC Residual Results

6.2.1 Rate-Distortion Plots

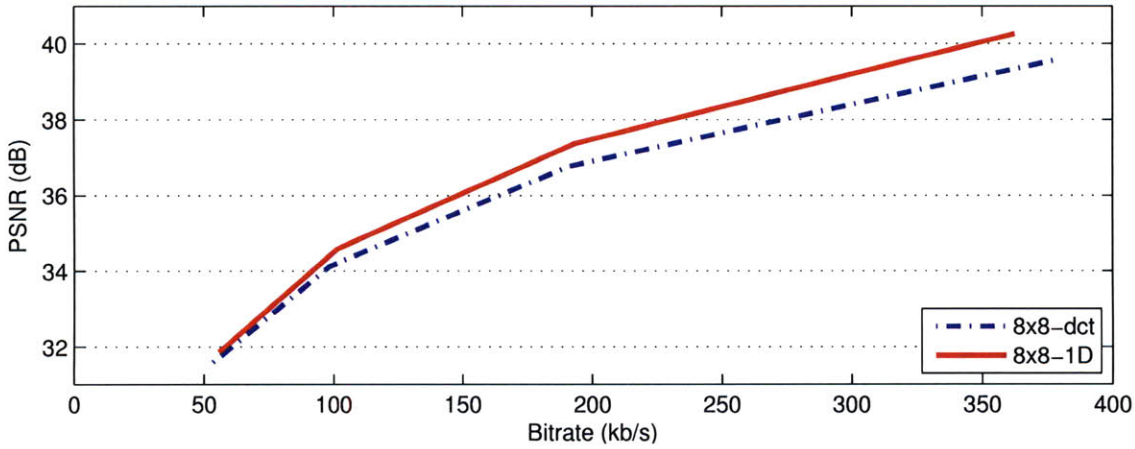
We first present experimental results with Rate-Distortion plots for two sequences. Figures 6-6 and 6-7 show Bitrate-PSNR plots of encoders with access to only *dct* transform(s) as well as encoders with access to both *dct* and $1D$ transforms for Foreman (QCIF resolution) and Basket (CIF resolution) sequences, respectively. Each figure has three plots, each of which provide comparisons using different block sizes for the transforms. Specifically, part (a) of figures compare 4x4-1D to 4x4-dct, part (b) of figures compare 8x8-1D to 8x8-dct, and part (c) of figures compare 4x4-and-8x8-1D to 4x4-and-8x8-dct. It can be observed that encoders with access to both *dct* and $1D$ transforms have better compression efficiency at all encoding bitrates.

The (horizontal or vertical) separation between the Bitrate-PSNR plots of encoders in all figures increases with increasing picture quality. This typically translates to a higher PSNR improvement at higher picture qualities. It also implies a higher percentage bitrate saving at higher picture qualities for many sequences. For example, in Figure 6-6 (c) the PSNR improvement is 0.1dB at 75kb/s and 0.47dB at 325kb/s. Similarly, the percentage bitrate savings are 2.24% at 32dB and 8.15% at 39dB for the same figure.

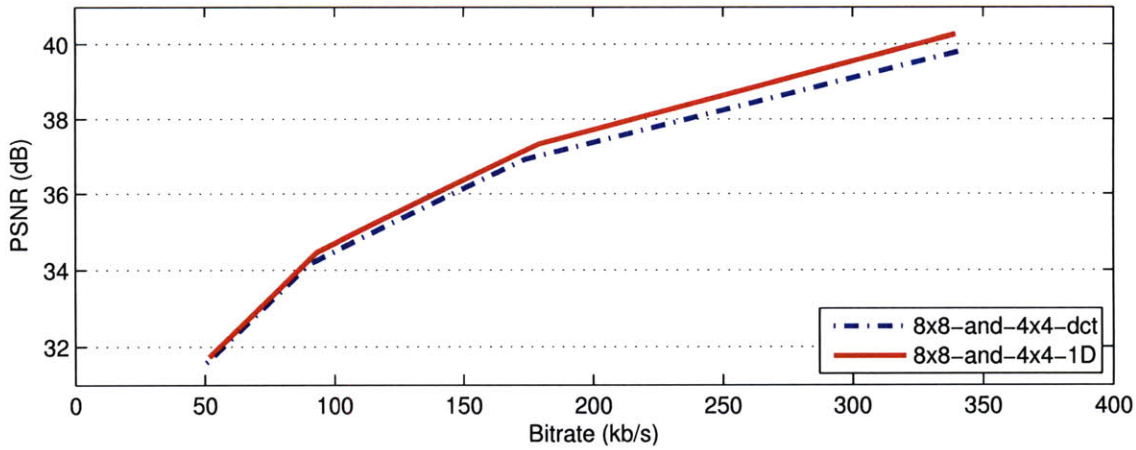
The increase of separation between the plots is in part because at higher picture qualities, the fraction of the total bitrate used to code the transform coefficients of the MC residual data is larger than at lower picture qualities. Figure 6-4 shows that for the 4x4-and-8x8-dct encoder and the Foreman sequence, the fractions are 30% at low picture qualities and 55% at high picture qualities. The lower the fraction is, the lower will be the impact of improved compression efficiency through the use of $1D$ transforms on the overall bitrate saving or PSNR improvement. For the Basket sequence, the fractions are 65% and 80% at low and high picture qualities, and the change from 65% to 80% is not significantly large, and therefore the separation of Bitrate-PSNR plots in Figure 6-7 does not increase as significantly with increasing picture quality as for the Foreman sequence (Figure 6-6). An additional factor that increases the separation between Bitrate-PSNR plots at higher picture qualities is the transmitted side information that indicates the chosen transforms. At lower picture qualities, the side information requires a higher fraction of the entire bitrate and becomes a larger burden.



(a) 4x4-1D vs 4x4-dct

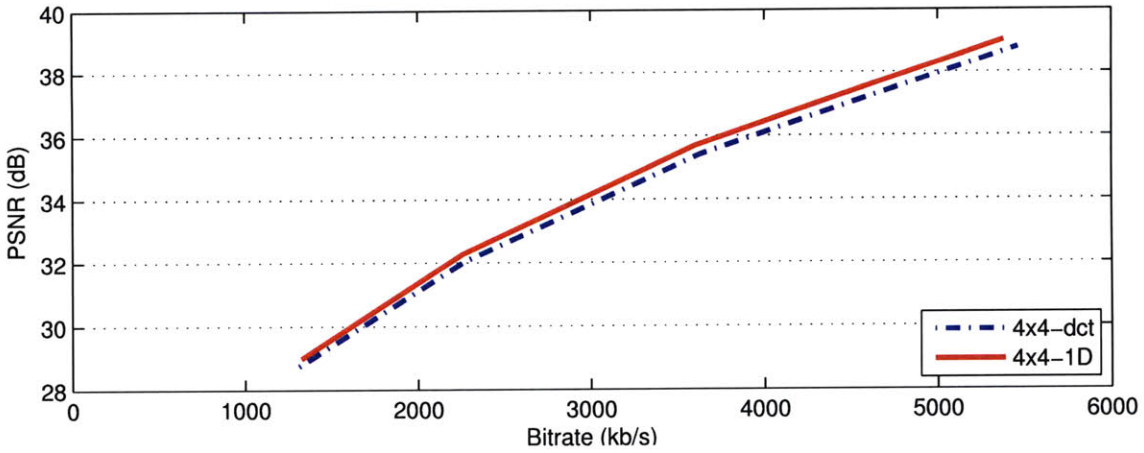


(b) 8x8-1D vs 8x8-dct

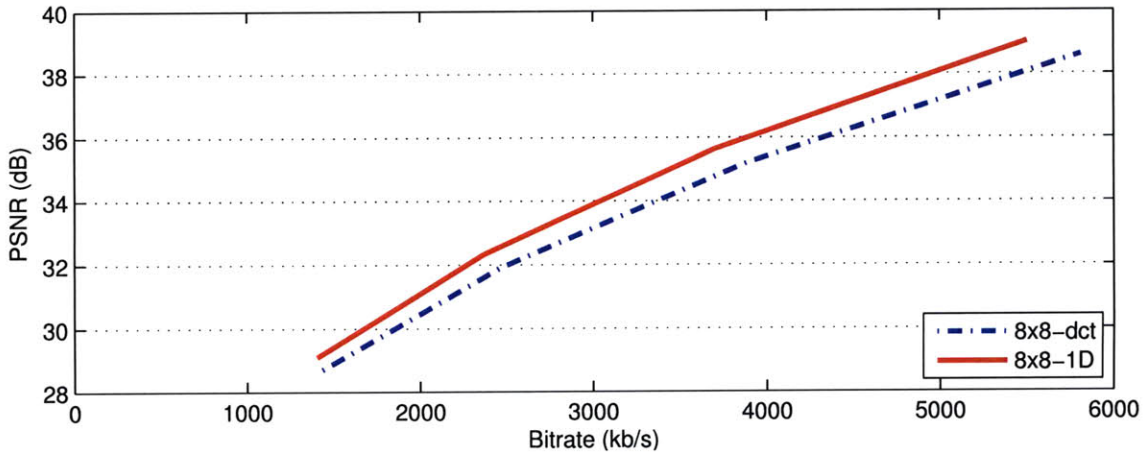


(c) 4x4-and-8x8-1D vs 4x4-and-8x8-dct

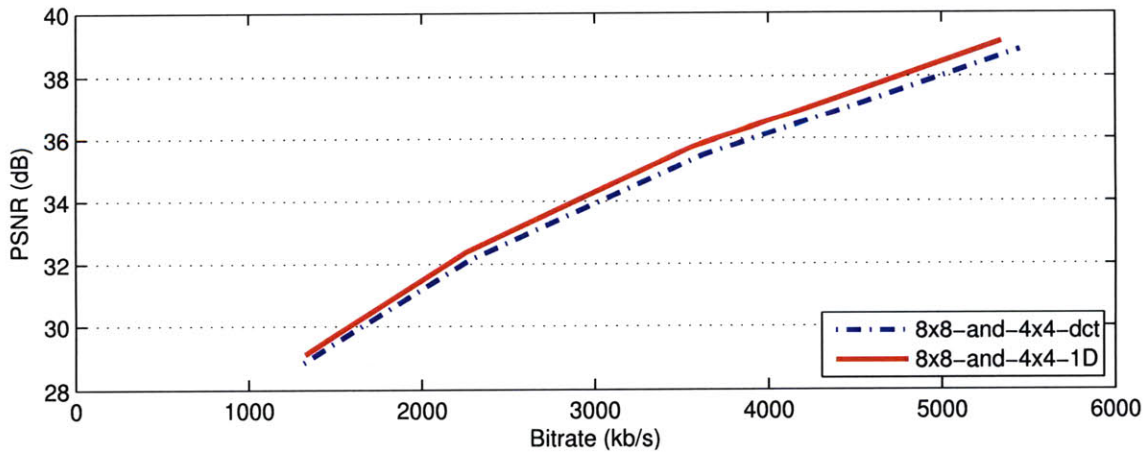
Figure 6-6: Bitrate-PSNR plots for Foreman (QCIF) sequence using encoders with access to different size transfroms.



(a) 4x4-1D vs 4x4-dct



(b) 8x8-1D vs 8x8-dct



(c) 4x4-and-8x8-1D vs 4x4-and-8x8-dct

Figure 6-7: Bitrate-PSNR plots for Basket (CIF) sequence using encoders with access to different size transfroms.

6.2.2 Bjontegaard-Delta Bitrate Results

To present experimental results for a large number of sequences we use the Bjontegaard-Delta (BD) bitrate metric [6]. This metric measures the average horizontal distance between two Bitrate-PSNR plots, giving the average bitrate saving over a range of picture qualities of one encoder with respect to another encoder. Using the BD-bitrate metric, the comparisons of encoders with access to $1D$ transforms to encoders with access to *dct* transform(s) is shown in Figure 6-8. Figure 6-8 (a) compares 4x4-1D to 4x4-dct, Figure 6-8 (b) compares 8x8-1D to 8x8-dct, and Figure 6-8 (c) compares 4x4-and-8x8-1D to 4x4-and-8x8-dct. The average bitrate savings are 4.1%, 11.4% and 4.8% in each of Figures 6-8 (a), (b) and (c).

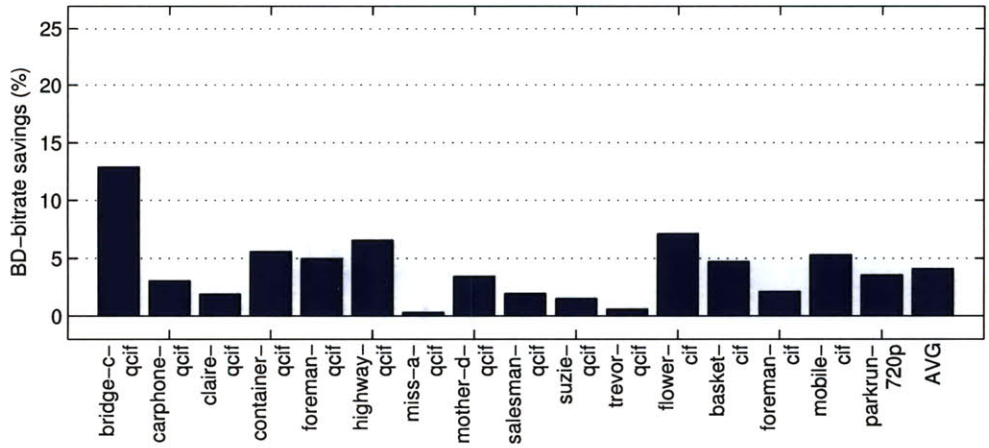
Bitrate savings depend on the block size of the transforms, which is typically also the block size for prediction. Bitrate savings are largest when comparing encoders which have access to only 8x8-pixel block transforms and smallest when comparing encoders which have access to only 4x4-pixel block transforms. This is in part because the distinction between 2-D transforms and 1-D transforms becomes less when the block-size is reduced. For example, for 2x2-pixel blocks, the distinction would be even less, and for the extreme case of 1x1-pixel blocks, there would be no difference at all.

The results also show that the bitrate savings depend on the characteristics of the video sequences. The ranking in performance among different sequences tends to remain unchanged among the three cases. The $bridge-c-qcif$ sequence has the largest savings and the $miss-a-qcif$ sequence has the smallest savings in Figures 6-8 (a), (b) and (c).

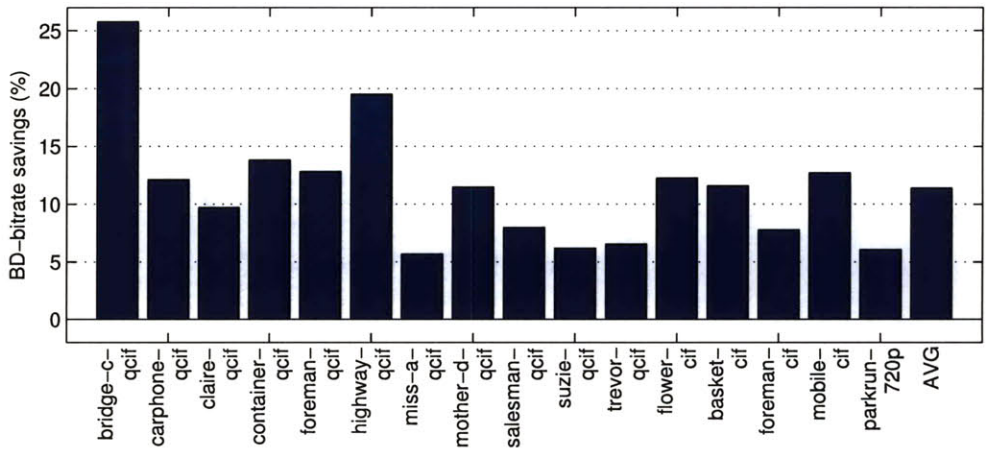
6.2.3 Bitrate for Coding Side Information

The encoder sends side information to indicate the chosen transform for each block. The side information can be a significant fraction of the overall bitrate. Figure 6-9 shows the average percentage of the bitrate used to code the side information in the 4x4-and-8x8-1D encoder for each sequence. These numbers are averages obtained from encoding results at all picture quality levels using quantization parameters 24, 28, 32 and 36. The average percentage bitrate used to code the side information is 4.4%.

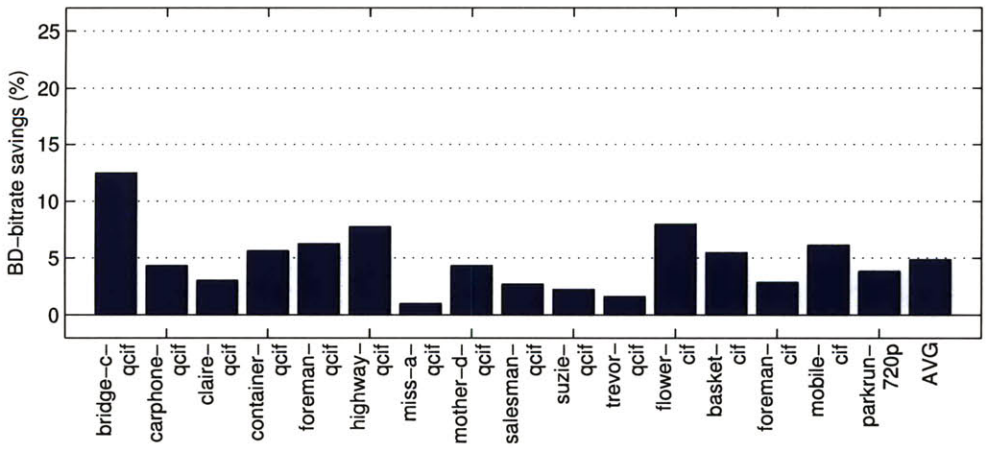
Notice that the percentage of the bitrate used to code the side information for each



(a) 4x4-1D vs 4x4-dct



(b) 8x8-1D vs 8x8-dct



(c) 4x4-and-8x8-1D vs 4x4-and-8x8-dct

Figure 6-8: Average bitrate savings (using BD-bitrate metric [6]) of several encoders with access to $1D$ transforms with respect to encoders with only conventional transform(s). Each plot provides savings when different sized transforms are available.

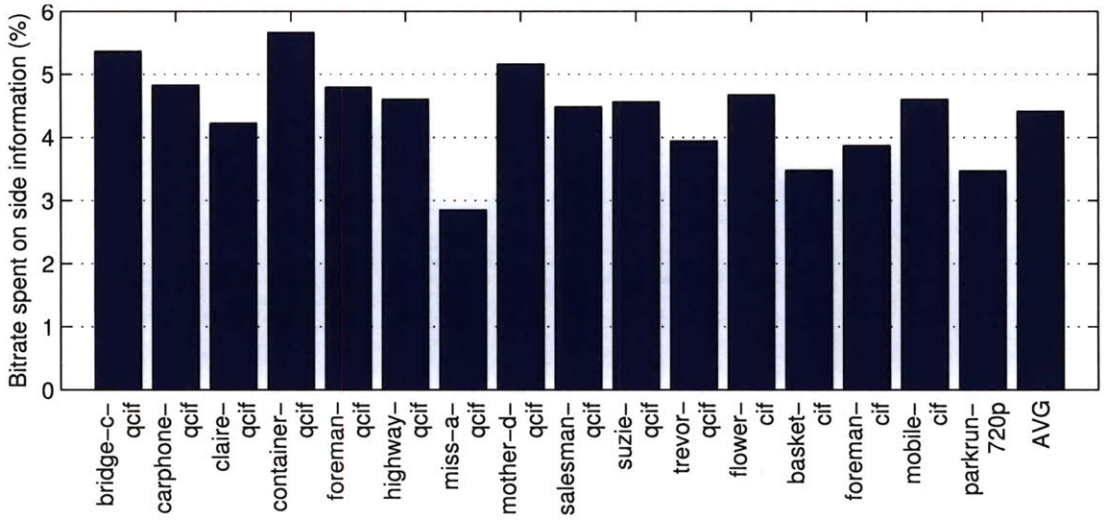


Figure 6-9: Average percentages of total bitrate used to code side information of 4x4-and-8x8-1D for all sequences. Numbers are obtained from all encoded picture qualities.

individual sequence in Figure 6-9 (a) correlates with the average bitrate savings of that sequence shown in Figure 6-8 (c). For example, $miss-a-qcif$ sequence has the smallest bitrate savings in Figure 6-8 (c), and the smallest percentage bitrate to code the side information in Figure 6-9. In general, if sequence A has larger bitrate savings than sequence B, then sequence A also has a larger percentage bitrate for the side information. Bitrate savings typically happen when the prediction residuals of the sequence have more $1D$ structures. This means more frequent use of $1D$ transforms relative to *2-D DCT*, which in turn implies a higher bitrate for the side information.

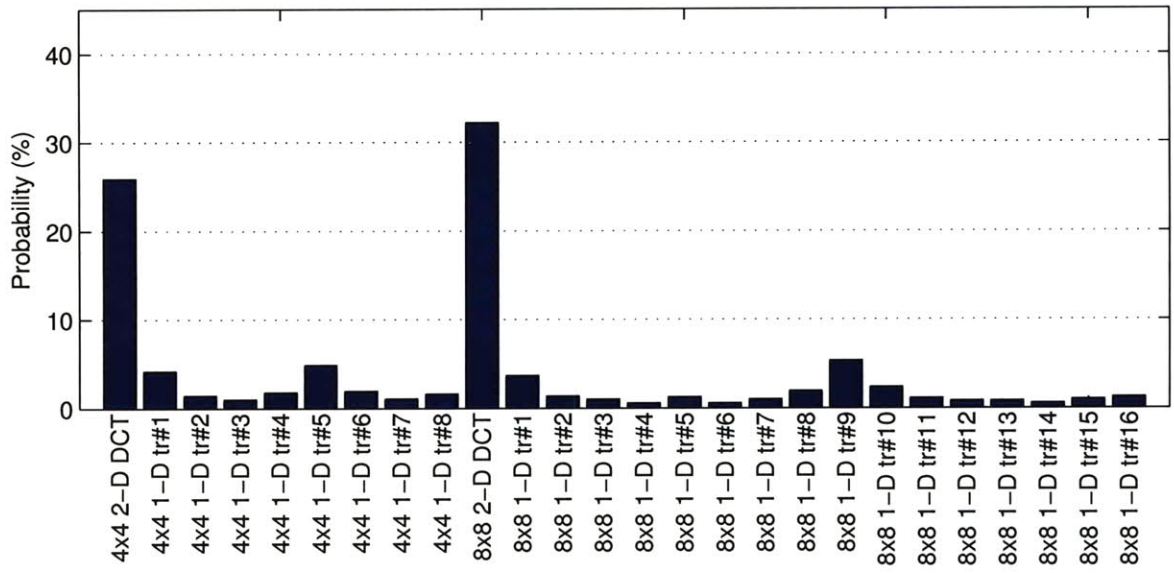
The average percentage of bitrate used to code the side information for different encoders are as follows. Among the encoders with access to $1D$ transforms, the average percentages are 3.6% for 4x4-1D, 5.9% for 8x8-1D and 4.4% for 4x4-and-8x8-1D. These are averages obtained from all sequences at all picture qualities. The lowest fraction is used by 4x4-1D and the highest fraction is used by 8x8-1D. The 4x4-1D uses a 1-bit (2-D DCT) or a 4-bit (1-D transforms) codeword for every four 4x4-pixel blocks with coded coefficients, and the 8x8-1D uses a 1-bit or a 5-bit codeword for every 8x8-pixel block with coded coefficients. In addition, the probability of using a 1-D transform is higher in 8x8-1D than in 4x4-1D.

6.2.4 Probabilities for Selection of Transforms

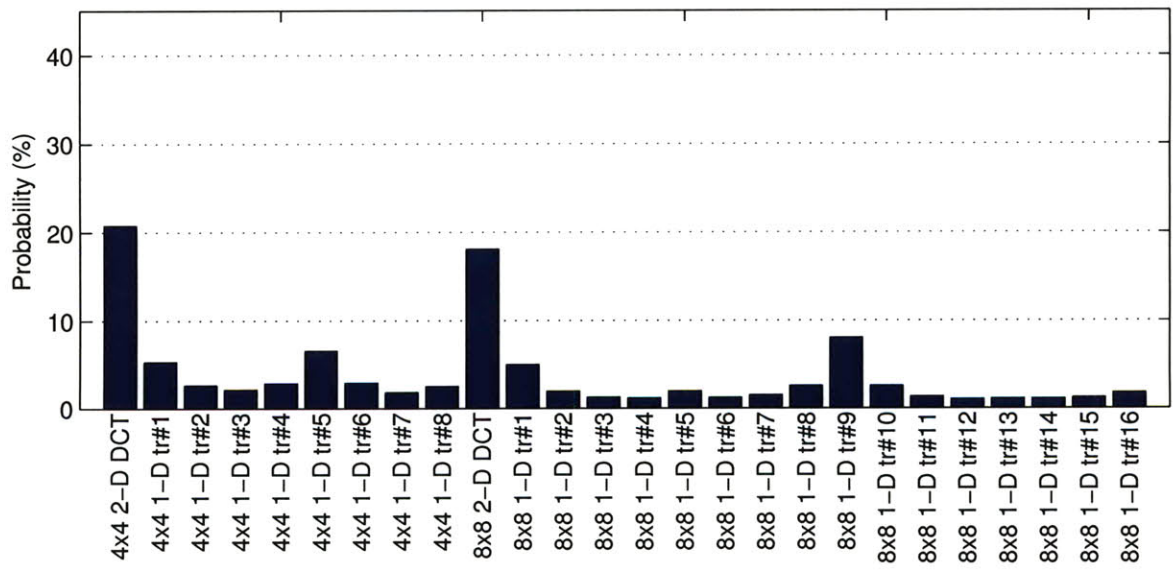
How often each transform is selected is presented in Figure 6-10. These numbers depend on the encoded sequence and picture qualities. Probabilities obtained from all sequences for the 4x4-and-8x8-1D encoder are shown in Figure 6-10 (a) for low picture qualities and in Figure 6-10 (b) for high picture qualities. It can be observed that the 2-D DCT's are chosen more often than the other transforms. A closer inspection reveals that using a 1-bit codeword to represent the 2-D DCT and a 4-bit codeword (5-bit in case of 8x8-pixel transforms) to represent the 1-D transforms is consistent with the numbers presented in these figures.

The 2-D DCT's are chosen much more often than any of the 1-D transforms. One reason for this large difference is the length of the codewords used to indicate the chosen transforms. While the 2-D DCT is indicated using a 1-bit codeword, each of the 1-D transforms are indicated using 4- (in case of 4x4-block transforms) or 5-bit (in case of 8x8-block transforms) codewords. Using a shorter codeword for the 2-D DCT creates an advantage for it and the 1-D transforms need to compress the local region at hand particularly well so as to compensate for this disadvantage. If the 2-D DCT and the 1-D transforms used similar length codewords, a more even distribution of probabilities in Figure 6-10 could be achieved. However such a codeword assignment does not necessarily improve the overall compression efficiency of the codec as we discussed in Chapter 5. Another reason for the large difference is that there are multiple 1-D transforms, each performing well for specific local regions. Their combined probability of selection is similar to that of the 2-D DCT.

At low picture qualities, the probability of selection is 58% for both 2-D DCT's, and 42% for all 1-D transforms. At high picture qualities, the probabilities are 38% for both 2-D DCT's, and 62% for all 1-D transforms. The 1-D transforms are chosen more often at higher picture qualities. Choosing the 2-D DCT costs 1-bit, and any of the 1-D transforms 4-bits (5-bits for 8x8-pixel block transforms). This is a smaller cost for 1-D transforms at high bitrates relative to the available bitrate.



(a) Low picture quality (QP=36)



(b) High picture quality (QP=24)

Figure 6-10: Average probability of selection for each transform at different picture quality levels for 4x4-and-8x8-1D.

6.2.5 Visual Quality

Video sequences coded with 1-D transforms have in general better overall visual quality. Although the improvements are not obvious, they are visible in some regions in the reconstructed frames. Regions with better visual quality typically include sharp edges or object boundaries. Figure 6-11 compares the reconstructed frame 101 of highway sequence (QCIF) coded with 4x4-dct and 4x4-1D at 19.90 kb/s and 20.43 kb/s, respectively. The stripes on the road are cleaner and the poles on the sides of the road are sharper in the frame reconstructed with 4x4-1D. Figure 6-12 shows these regions in more detail for easier comparison. Figure 6-13 compares the reconstructed frame 91 of basket sequence (CIF) coded with 8x8-dct and 8x8-1D at 1438 kb/s and 1407 kb/s, respectively. The arms of the jumping players and the shoulders and faces of the standing players are cleaner in the frame reconstructed with 8x8-1D, and Figure 6-14 shows these regions in more detail.

6.2.6 MC and IP residuals

For the results presented so far in Section 6.2, all inter coded macroblocks were coded using either the 2-D DCT or one of the 1-D transforms, and all intra coded macroblocks were coded using only the 2-D DCT. Specifically, within a P-frame some macroblocks cannot be predicted well using inter prediction, and intra prediction is used. For such intra coded macroblocks, only the 2-D DCT was used, and this choice was made to focus on the achievable gains from using 1-D transforms for MC residuals.

As we show in Section 6.3, however, intra prediction residuals can also be coded more efficiently using 1-D transforms. To show the overall gains that can be obtained from using 1-D transforms for both inter and intra coded macroblocks, we have rerun the MC residual experiments presented in Section 6.2, where both inter and intra macroblocks were coded using either the 2-D DCT or one of the 1-D transforms. As expected, the achievable bitrate saving over a conventional encoder increases. In particular, the average bitrate saving of 4x4-and-8x8-1D with respect to 4x4-and-8x8-dct (shown in Figure 6-8 (c)) increases from 4.8% to 5.6%. Typically few macroblocks in P-frames are coded using intra prediction, and such macroblocks, previously coded using only the 2-D DCT, are now coded using also 1-D transforms and increase the overall bitrate saving.

(a) 4x4-dct

(b) 4x4-1D

Figure 6-11: Comparison of the reconstructed frame 101 of highway sequence (QCIF) coded with 4x4-dct and 4x4-1D at 19.90 kb/s and 20.43 kb/s, respectively. Frame 101 was coded at 33.117 dB PSNR using 680 bits with the 4x4-dct and at 33.317 dB PSNR using 632 bits with the 4x4-1D.

(a) 4x4-dct

(b) 4x4-1D

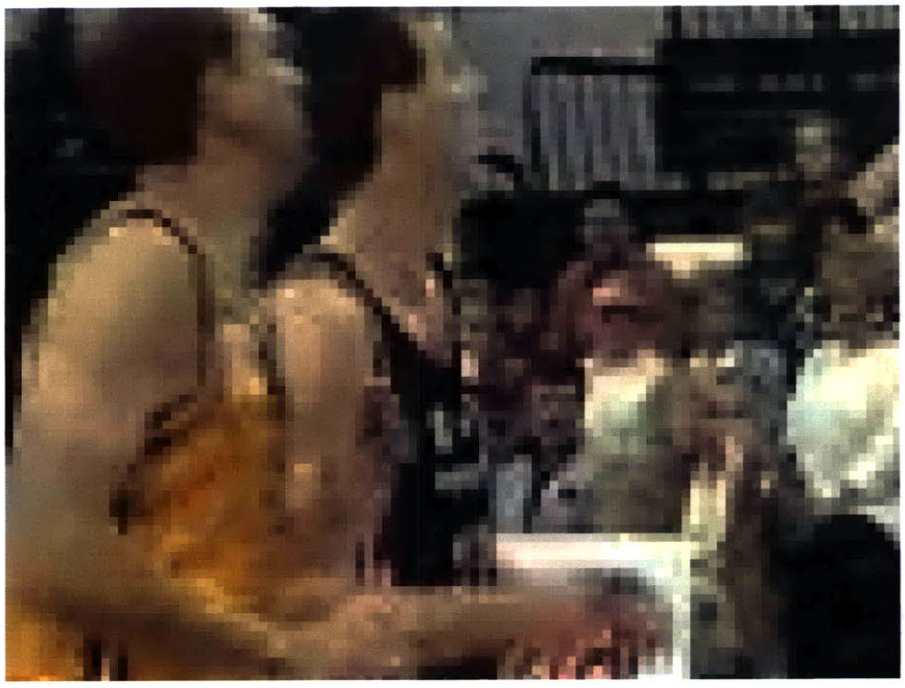
Figure 6-12: Comparison using a region from the frames in Figure 6-11 shown in detail. The stripes on the road are cleaner and the poles on the sides of the road are sharper in the frame reconstructed with 4x4-1D.

(a) 8x8-dct

(b) 8x8-1D

Figure 6-13: Comparison of the reconstructed frame 91 of basket sequence (CIF) coded with 8x8-dct and 8x8-1D at 1438 kb/s and 1407 kb/s, respectively. Frame 91 was coded at 28.834 dB PSNR using 49360 bits with the 8x8-dct and at 29.166 dB PSNR using 47632 bits with the 8x8-1D.

(a) 8x8-dct



(b) 8x8-1D

Figure 6-14: Comparison using a region from the frames in Figure 6-13 shown in detail. The shoulders and faces of the players are cleaner in the frame reconstructed with 8x8-1D.

6.3 IP Residual Results

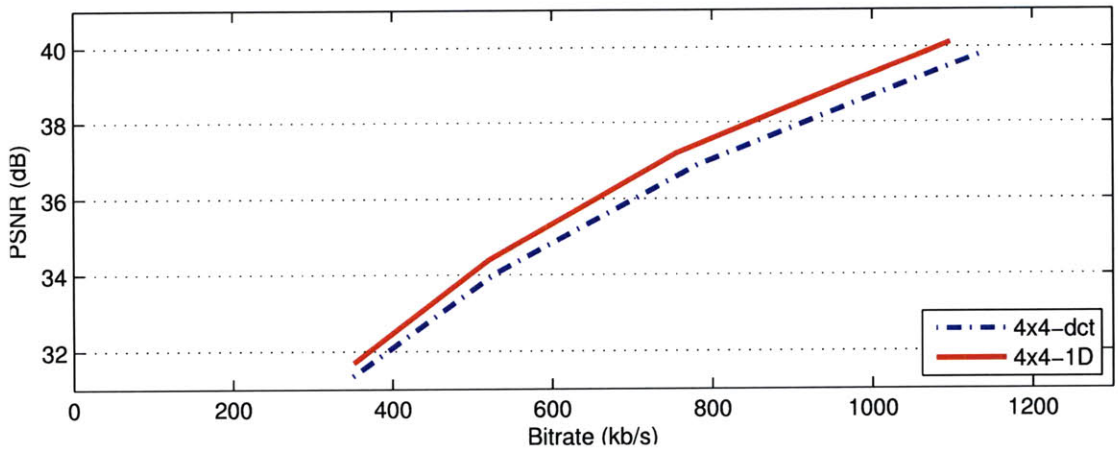
6.3.1 Rate-Distortion Plots

Figures 6-15 and 6-16 show Bitrate-PSNR plots for Foreman (QCIF resolution) and Basket (CIF resolution) sequences, respectively. The plots are provided to compare 4x4-1D to 4x4-dct in part (a) of the figures, 8x8-1D to 8x8-dct in part (b) of the figures, and 4x4-and-8x8-1D to 4x4-and-8x8-dct in part (c) of the figures. It can be observed that encoders with access to both *dct* and $1D$ transforms have better compression efficiency at all encoding bitrates. The (horizontal or vertical) separation between the plots in each figure increases slightly with increasing picture quality and this typically translates to a slightly higher PSNR improvement at higher picture qualities. The increase in the separation with increasing picture quality is not as strong as in MC residuals and therefore, unlike in MC residuals, the percentage bitrate savings achieved with encoders with access to $1D$ transforms decreases slightly or remains roughly similar over the considered range of picture qualities.

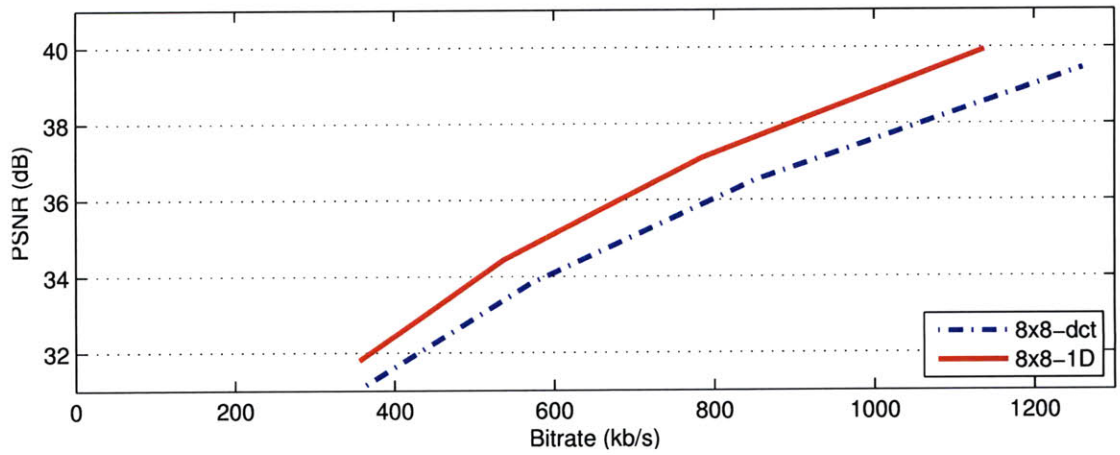
The main reason for the different separation of Bitrate-PSNR plots at low and high picture qualities for MC and IP residuals is the differing fractions of the total bitrate used to code MC and IP residuals. For MC residuals, the average fraction of the total bitrate used to code the residual data is 40% at low picture qualities (QP=36) and 65% at high (QP=24) picture qualities, as shown in Figure 6-4. For IP residuals, about 80% and 85% of the total bitrate are used to code the residual data at low (QP=36) and high (QP=24) picture qualities, as shown in Figure 6-5. The relative increase from 80% to 85% is not as large as the one from 40% to 65% and the separation between the plots for IP residuals in Figures 6-15 and 6-16 do not increase as significantly as the separation for MC residuals.

6.3.2 Bjontegaard-Delta Bitrate Results

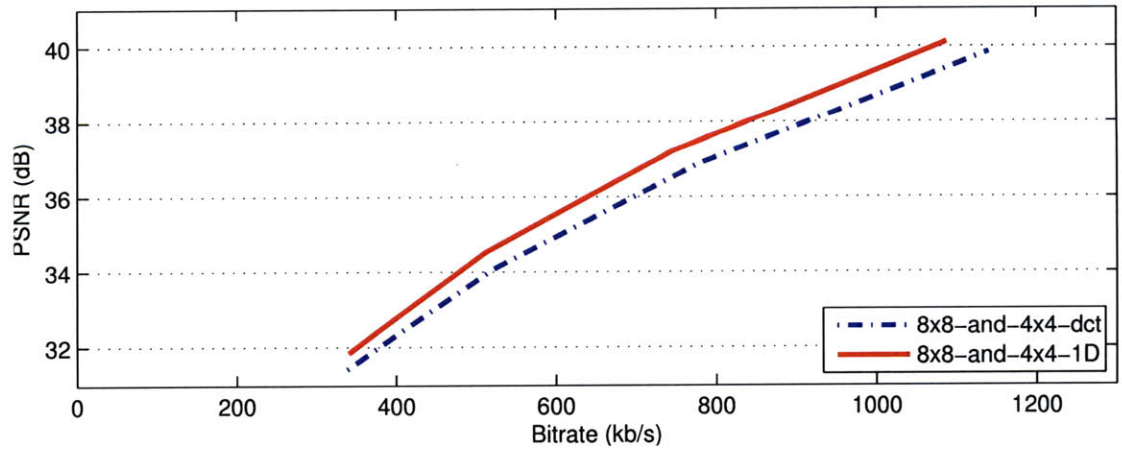
The comparisons of encoders with access to $1D$ transforms to encoders with access to *dct* transform(s) is shown in Figure 6-17 for all sequences using the BD-bitrate savings metric. Figure 6-17 (a) compares 4x4-1D to 4x4-dct, Figure 6-17 (b) compares 8x8-1D to 8x8-dct, and Figure 6-17 (c) compares 4x4-and-8x8-1D to 4x4-and-8x8-dct. The average



(a) 4x4-1D vs 4x4-dct

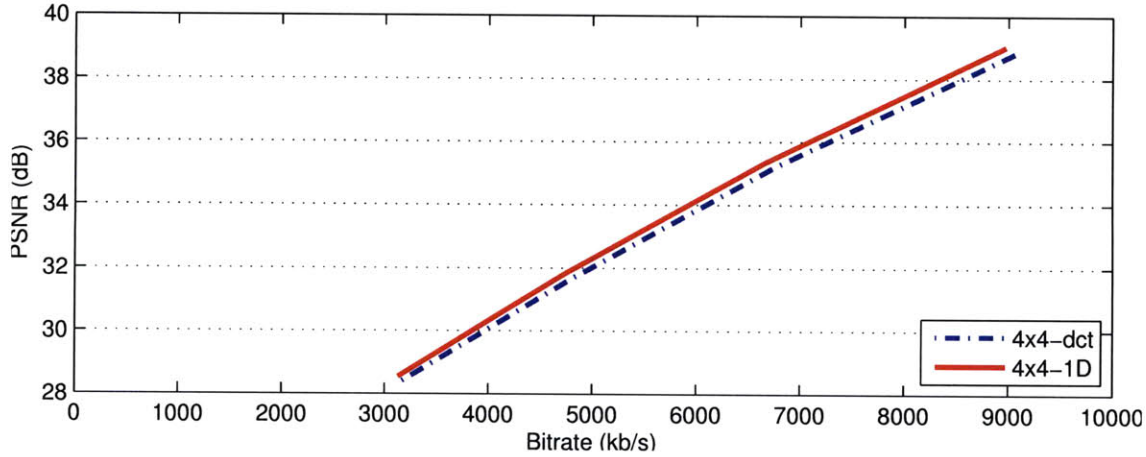


(b) 8x8-1D vs 8x8-dct

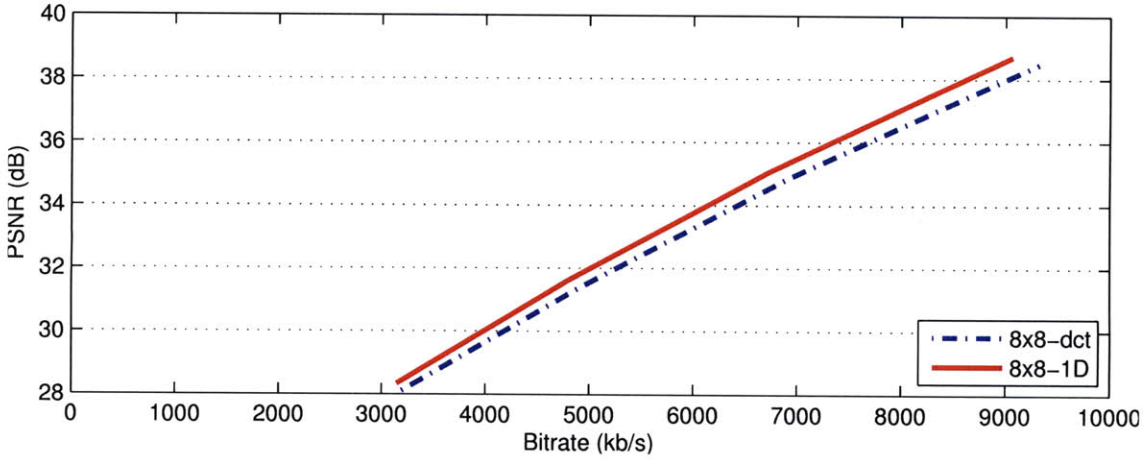


(c) 4x4-and-8x8-1D vs 4x4-and-8x8-dct

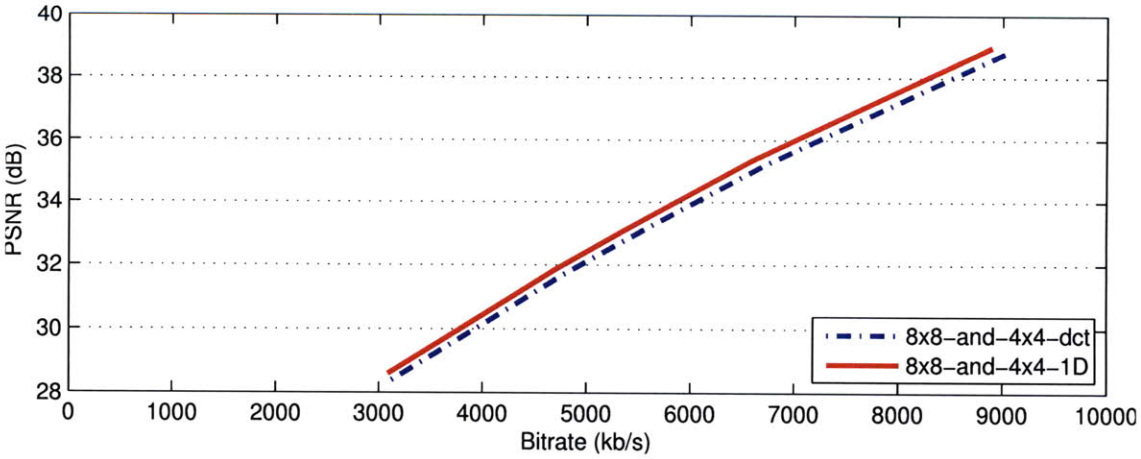
Figure 6-15: Bitrate-PSNR plots for Foreman (QCIF) sequence using encoders with access to different size transfroms.



(a) 4x4-1D vs 4x4-dct



(b) 8x8-1D vs 8x8-dct



(c) 4x4-and-8x8-1D vs 4x4-and-8x8-dct

Figure 6-16: Bitrate-PSNR plots for Basket (CIF) sequence using encoders with access to different size transfroms.

bitrate savings are 4.2%, 10.6% and 5.3% in each of Figures 6-17 (a), (b) and (c). Note that even though we discussed in Chapter 3 that IP residuals do not have as many 1-D structures as MC residuals, the gains reported here are quite similar to the ones reported for MC residuals in Section 6.2.2. The reasons are explained in Section 6.3.4 together with the probabilities of selection of transforms.

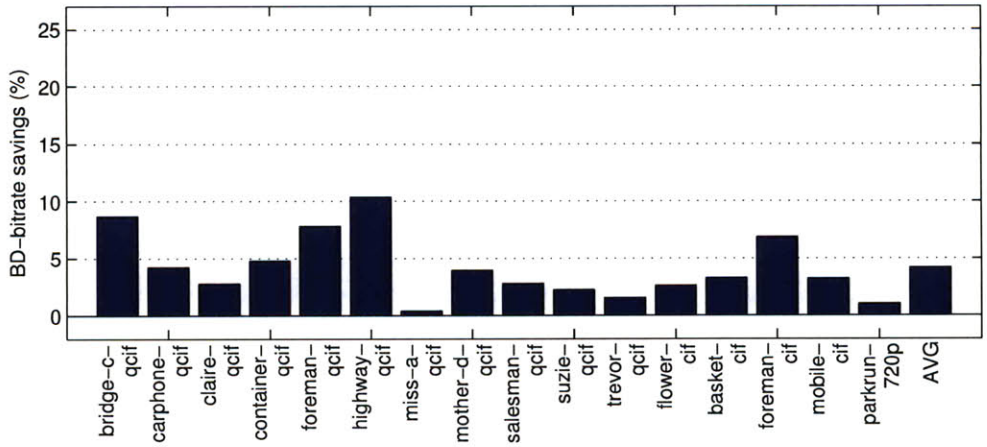
Similar to MC residual results, bitrate savings depend on the block size of the transforms, which is typically also the block size for prediction. Bitrate savings are largest when comparing encoders which have access to only 8x8-pixel block transforms and smallest when comparing encoders which have access to only 4x4-pixel block transforms. Again, the bitrate savings depend on the characteristics of the video sequences. The ranking in performance among different sequences tends to remain similar among the three cases. For example, the $highway - qcif$ sequence has the largest savings in Figures 6-17 (a), (b) and (c).

6.3.3 Bitrate for Coding Side Information

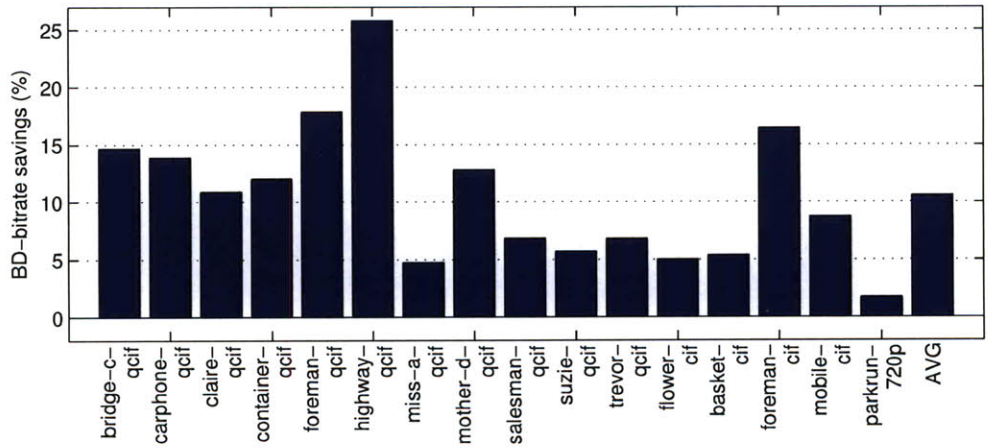
Figure 6-18 shows the average percentage of the total bitrate used to code the side information in the 4x4-and-8x8-1D encoder for each sequence. These numbers are averages obtained from encoding results at all picture quality levels using quantization parameters 24, 28, 32 and 36. The average percentage bitrate used to code the side information is 3.5%, which is smaller than the 4.4% obtained from MC residuals. Intra prediction typically does not work as well as motion compensation and a typical residual block obtained using intra prediction requires a larger number of bits to code than a typical residual obtained using motion compensated prediction. In addition, 1-D transforms are chosen less frequently in IP residuals, as will be shown in the next section. Thus the side information bitrate is typically a smaller fraction of the total bitrate for IP residuals.

Similar to MC residual results, the percentage of the bitrate used to code the side information for each individual sequence in Figure 6-18 (a) correlates with the average bitrate savings of that sequence shown in Figure 6-17 (c). For example, $highway - qcif$ sequence has the largest bitrate savings in Figure 6-17 (c), and the largest percentage bitrate used to code the side information in Figure 6-18.

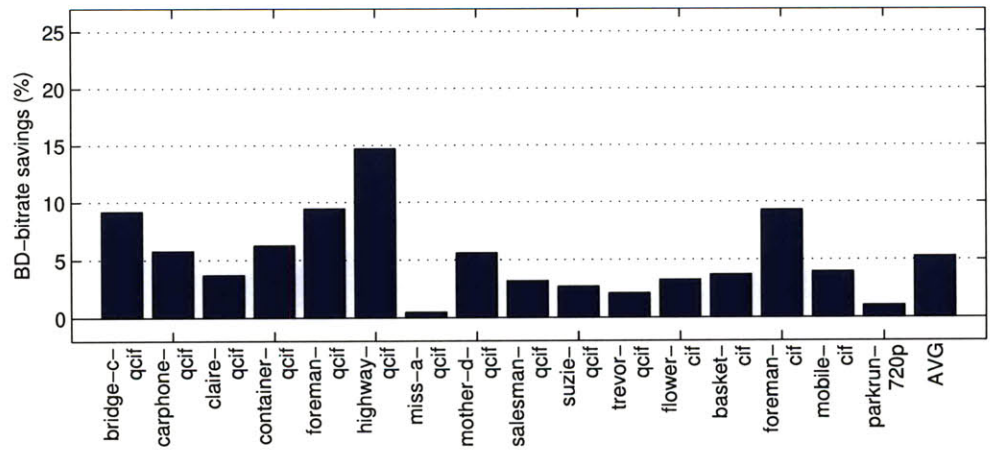
The average percentage of bitrate used to code the side information for different



(a) 4x4-1D vs 4x4-dct



(b) 8x8-1D vs 8x8-dct



(c) 4x4-and-8x8-1D vs 4x4-and-8x8-dct

Figure 6-17: Average bitrate savings (using BD-bitrate metric [6]) of several encoders with access to $1D$ transforms with respect to encoders with only conventional transform(s). Each plot provides savings when different sized transforms are available.

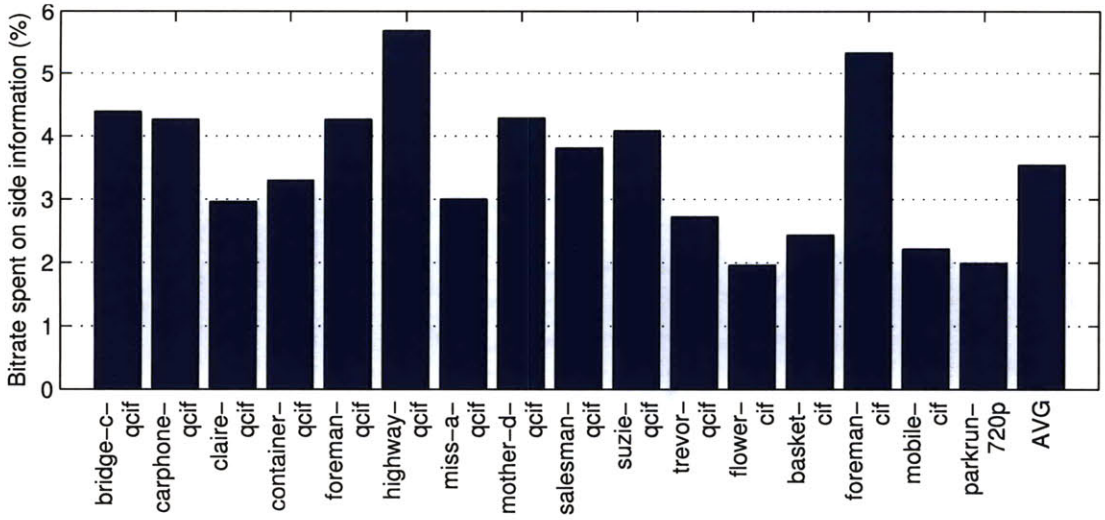


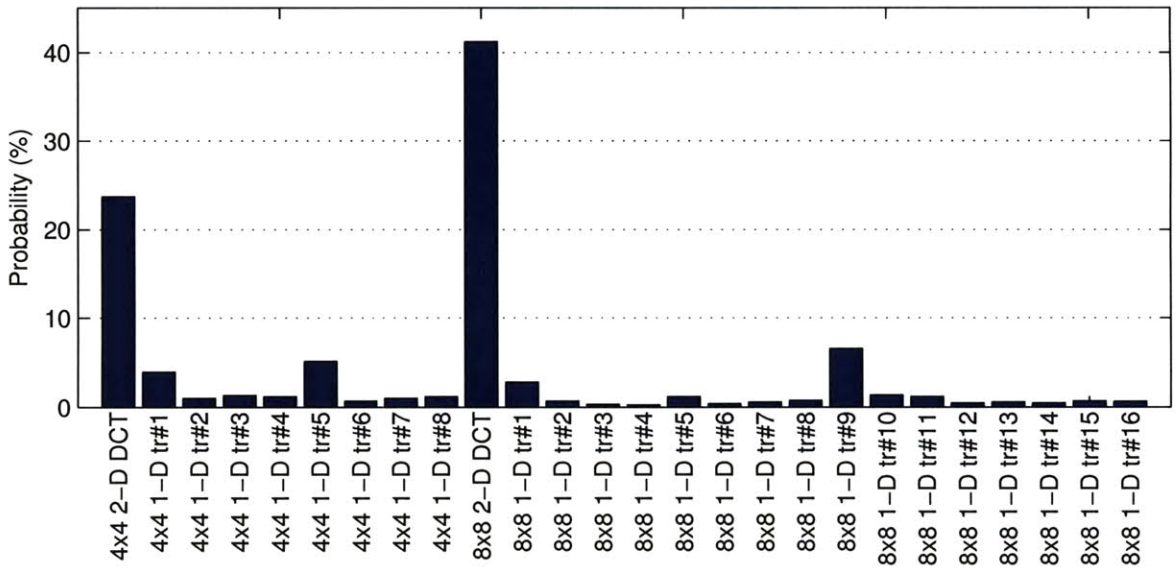
Figure 6-18: Average percentages of total bitrate used to code side information of 4x4-and-8x8-1D for all sequences. Numbers are obtained from all encoded picture qualities.

encoders are as follows. Among the encoders with access to $1D$ transforms, the average percentages are 2.9% for 4x4-1D, 3.8% for 8x8-1D and 3.5% for 4x4-and-8x8-1D. These are averages obtained from all sequences at all picture qualities. The lowest fraction is used by 4x4-1D and the highest fraction is used by 8x8-1D. The 4x4-1D uses a 1-bit (2-D DCT) or a 4-bit (1-D transforms) codeword for every four 4x4-pixel blocks with coded coefficients, and the 8x8-1D uses a 1-bit or a 5-bit codeword for every 8x8-pixel block with coded coefficients.

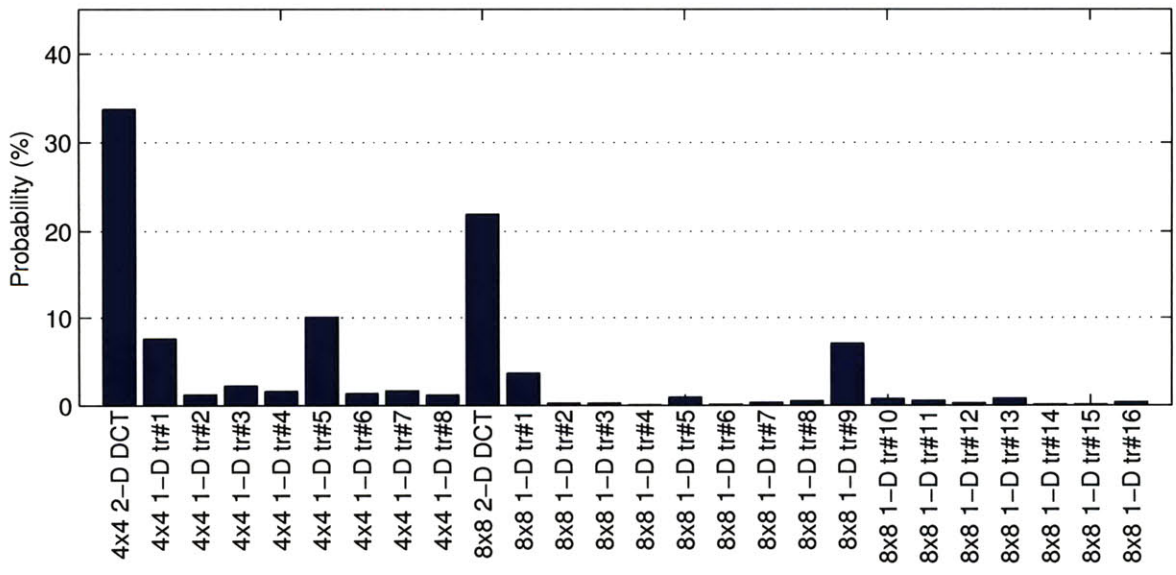
6.3.4 Probabilities for Selection of Transforms

Probabilities indicating how often each transform is selected are shown in Figure 6-19 for the 4x4-and-8x8-1D encoder for all sequences. Figure 6-19 (a) shows probabilities obtained from encoding sequences at low picture qualities and Figure 6-19 (b) shows probabilities obtained from encoding sequences at high picture qualities. It can be observed that the 2-D DCT's are chosen more often than the other transforms.

At low picture qualities, the probability of selection is 65% for both 2-D DCT's, and 35% for all 1-D transforms. At high picture qualities, the probabilities are 55% for both 2-D DCT's, and 45% for all 1-D transforms. The 1-D transforms are chosen more often at higher picture qualities because choosing the 2-D DCT costs 1-bit, and any of the



(a) Low picture quality (QP=36)



(b) High picture quality (QP=24)

Figure 6-19: Average probability of selection for each transform at different picture quality levels for 4x4-and-8x8-1D.

1-D transforms 4-bits (5-bits for 8x8-pixel block transforms). This is a smaller cost for 1-D transforms at high bitrates relative to the available bitrate.

Figure 6-19 also shows that the probability of selection of 2-D transforms is larger in IP residuals than in MC residuals. This indicates that 1-D transforms are more useful for MC residuals and this is consistent with our observations and findings in Chapter 3. The achieved bitrate savings for IP residuals are, however, similar to that of MC residuals and the main reason for this is the differing fractions of the total bitrate used to code the residuals. As shown in Figures 6-4 and 6-5, the average fraction of the total bitrate used to code the residuals is roughly twice as large for IP residuals and coding of IP residuals with 1-D transforms needs to be twice as inefficient so that similar overall bitrate savings are achieved for both types of residuals.

6.3.5 Visual Quality

Similar to MC residual results, video sequences coded with 1-D transforms have in general better overall visual quality and although the improvements are not obvious, they are visible in some regions in the reconstructed frames. Regions with better visual quality typically include sharp edges or object boundaries. Figure 6-20 compares the reconstructed frame 20 of container sequence (QCIF) coded with 4x4-dct and 4x4-1D at 71.71 kb/s and 70.74 kb/s, respectively. The water beneath the ship and the features on the ship are in general sharper in the frame reconstructed with 4x4-1D. Figure 6-21 shows these regions in more detail for easier comparison. Figure 6-22 compares the reconstructed frame 5 of mobile sequence (CIF) coded with 8x8-dct and 8x8-1D at 705.44 kb/s and 683.39 kb/s, respectively. The edges and boundaries are much clearer in the frame reconstructed by 8x8-1D. In particular, in the frame reconstructed with 8x8-dct the edges and boundaries of objects in the background have so called mosquito noise (haziness). Such artifacts are considerably less visible in the frame reconstructed with 8x8-1D. The numbers on the calendar are also clearer in the frame reconstructed with 8x8-1D. Figure 6-23 shows a region in more detail for easier comparison.

(a) 4x4-dct

(b) 4x4-1D

Figure 6-20: Comparison of the reconstructed frame 20 of container sequence (QCIF) coded with 4x4-dct and 4x4-1D at 71.71 kb/s and 70.74 kb/s, respectively. Frame 20 was coded at 31.68 dB PSNR using 11920 bits with the 4x4-dct and at 31.96 dB PSNR using 11784 bits with the 4x4-1D.

(a) 4x4-dct

(b) 4x4-1D

Figure 6-21: Comparison using a region from the frames in Figure 6-20 shown in detail. The water beneath the ship and the features on the ship are in general sharper in the frame reconstructed with 4x4-1D.

(a) 8x8-dct

(b) 8x8-1D

Figure 6-22: Comparison of the reconstructed frame 5 of mobile sequence (CIF) coded with 8x8-dct and 8x8-1D at 705.44 kb/s and 683.39 kb/s, respectively. Frame 5 was coded at 28.76 dB PSNR using 117136 bits with the 8x8-dct and at 29.13 dB PSNR using 113616 bits with the 8x8-1D.

(a) 8x8-dct

(b) 8x8-1D

Figure 6-23: Comparison using a region from the frames in Figure 6-22 shown in detail. Edges and boundaries of objects are cleaner and mosquito noise (haziness) are considerably less visible in the frame reconstructed with 8x8-1D.

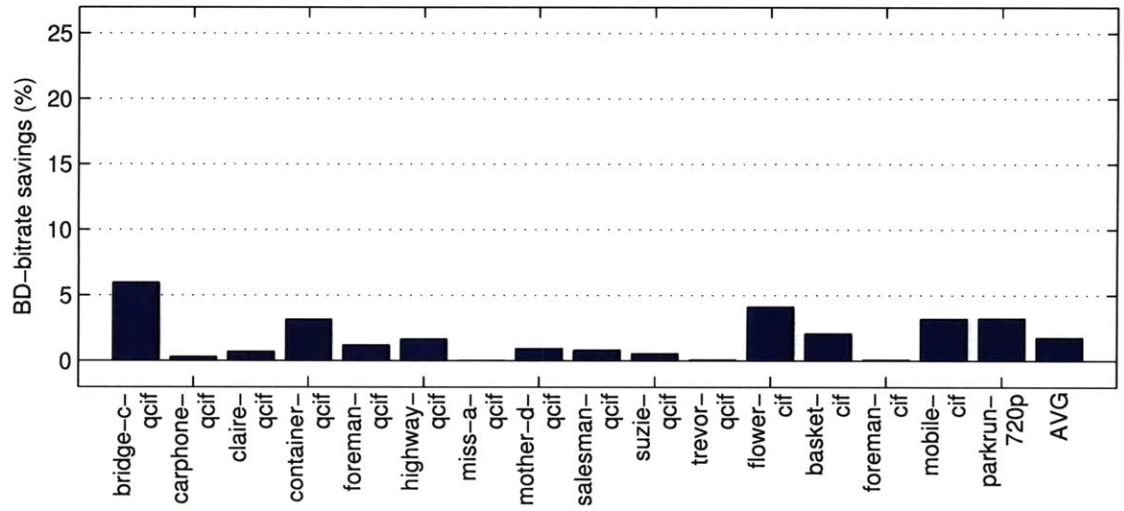
6.4 Comparison with 2-D Directional Transforms

In this section, we compare a specific directional block transform proposed for image compression with our 1-D transforms on MC and IP residuals. These directional block transforms, proposed by Zeng et.al. [55], were discussed in Section 2.4 and are 2-D directional DCT's together with a DC separation and ΔDC correction method borrowed from the shape-adaptive DCT framework in [19]. 2-D directional DCT's are formed by 1-D DCT's along predefined pixels, followed by a second set of 1-D DCT's and DC separation and ΔDC correction computations. DC separation and ΔDC correction are computations introduced to mitigate some undesired properties of the overall transforms.

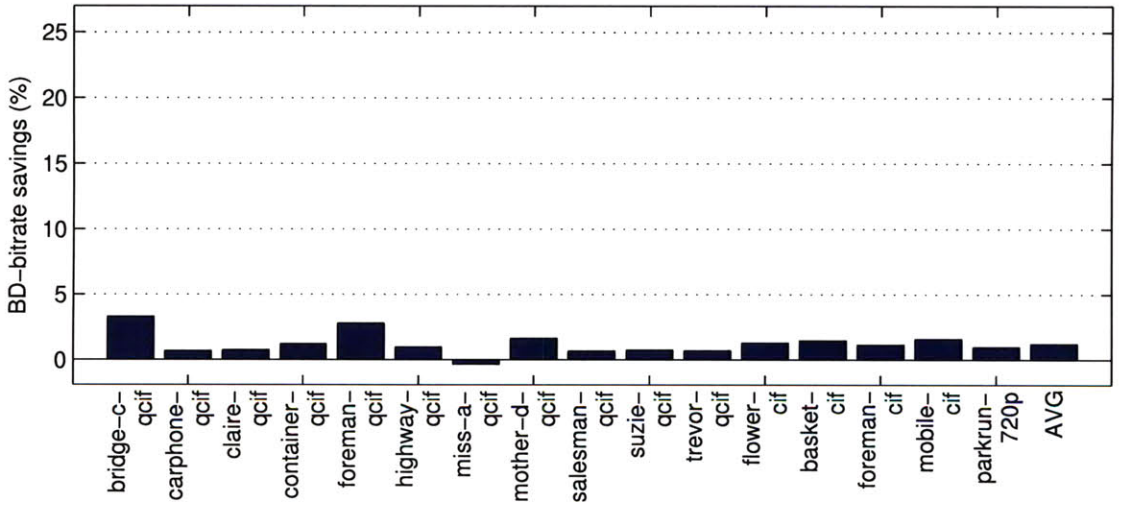
We present experimental results with these transforms from [55] because these transforms are 2-D directional block transforms and were proposed for compressing image intensities and it is typical to use transforms that are originally developed for image compression, to compress prediction residuals. Our intent here is to provide experimental evidence indicating that although 2-D directional transforms can improve compression efficiency for images [55], they are worse than 1-D transforms for improving compression efficiency of MC and IP residuals.

For the experiments, we have complemented the six transforms in [55] with another eight transforms to achieve finer directional adaptivity (which is comparable to the adaptivity of our proposed transforms) in case of 8x8-pixel block transforms. For 4x4-pixel block transforms, we designed six transforms using the techniques provided in [55]. The scanning patterns for the transform coefficients were also taken from [55] and coding of the chosen transform is done similar to the coding of the proposed 1-D directional transforms.

We compare an encoder with $2D$ transforms (including 2-D DCT) to an encoder with *dct* transforms in Figure 6-24. Specifically, we compare 4x4-and-8x8-2D to 4x4-and-8x8-dct on MC residuals in Figure 6-24 (a) and on IP residuals in Figure 6-24 (b). The average bitrate savings are 1.8% for Figure 6-24 (a), 1.2% for Figure 6-24 (b). These averages are considerably lower than the average savings obtained with $1D$ transforms in Figures 6-8 (c) and 6-17 (c), which were 4.8% and 5.3%.



(a) 4x4-and-8x8-2D vs 4x4-and-8x8-dct , MC-residual



(b) 4x4-and-8x8-2D vs 4x4-and-8x8-dct , IP-residual

Figure 6-24: Average bitrate savings of an encoder with access to $2D$ transforms [55] with respect to an encoder with only conventional transforms for MC and IP residuals.

Chapter 7

Conclusions

7.1 Summary

The same transforms are typically used to transform image intensities in image coding and prediction residuals of image intensities in video coding. For example, the 2-D Discrete Cosine Transform (2-D DCT) is used to compress image intensities in the JPEG image compression standard and MC residuals in many video coding standards. However, prediction residuals can have significantly different spatial characteristics from image intensities. It is of interest therefore to study if transforms better than those used for image intensities can be developed for prediction residuals.

It is well known that spatial characteristics of images can vary from one local region to another and adapting the processing according to the variations can yield improved results. Many direction-adaptive transforms have been proposed that can take advantage of the differing anisotropic characteristics in local regions of images. Conventional transforms, such as the 2-D DCT, are carried out as a separable transform by cascading two 1-D transforms in the vertical and horizontal dimensions. This approach favors horizontal or vertical features over others and does not take advantage of locally anisotropic features in images. The direction-adaptive transforms adapt to local anisotropic features in images by performing the filtering along directions where image intensity variations are smaller.

In video coding, prediction residuals of image intensities are coded in addition to

image intensities. Many transforms have been developed to take advantage of local anisotropic characteristics of images, however, local anisotropic characteristics of prediction residuals can be significantly different from the ones of images. This thesis analyzed the differences between the local anisotropic characteristics of images and a number of types of prediction residuals and proposed transforms adapted to the characteristics of some specific types of prediction residuals.

We began by analyzing the local anisotropic characteristics of images and prediction residuals in Chapter 3. A visual inspection showed that anisotropic characteristics of images and prediction residuals can be significantly different in some local regions. Specifically, regions which are difficult to predict include object boundaries and edges, and a significant fraction of large prediction errors concentrate in these regions. In particular, in MC and RE residuals large prediction errors reside on the object boundaries and edges of the original image and since these structures are 1-D, a major fraction of prediction residuals in MC and RE residuals form 1-D structures. Thus while images have 2-D anisotropic characteristics in such regions, MC and RE residuals have 1-D anisotropic characteristics in such regions. This distinction is also shown using an auto-covariance analysis in Chapter 3.

The 2-D DCT can compress smooth regions in images efficiently and the new direction-adaptive transforms improve the compression of anisotropic local regions in images. The analysis in Chapter 3 showed that a significant amount of local regions in MC and RE residuals have 1-D characteristics and this is significantly different from the characteristics of images. Neither of these transforms addresses this difference. Using transforms with 2-D basis functions for such regions is inefficient and we proposed in Chapter 4 transforms with basis functions whose support follow the 1-D structures of MC and RE residuals. We presented a sample set of 1-D block transforms, where each transform in the set is adapted to 1-D structures along a specific direction.

To examine the performance of the proposed 1-D transforms, they were integrated into a codec based on H.264/AVC. To have an efficient system, a number of related aspects needed to be carefully addressed and we discussed these aspects in Chapter 5. Coding of 1-D transform coefficients and coding of the side information to indicate the chosen transforms for each local region were discussed, among other aspects. We used a simple scheme to code the selected transforms. To code the 1-D transform coefficients,

a new method specifically adapted to the characteristics of the 1-D transforms is ideally desirable, however, we used a simplified adaptation here as well; we used scanning patterns adapted to each 1-D transform and the remaining coding algorithm was not changed.

Experimental results were presented in Chapter 6 to evaluate the compression performance of the proposed 1-D transforms on MC and IP residuals. Encoders with access to conventional transforms were compared against encoders with access to both conventional and 1-D transforms. For all sequences that were used in the experiments, encoders with access to 1-D transforms achieved lower bitrates for the same picture quality (PSNR). The achievable bitrate savings depend on the characteristics of the sequences and the block size for the transforms. The average bitrate savings obtained for MC residuals were 4.1%, 11.4% and 4.8% for 4x4-block, 8x8-block and 4x4-and-8x8-block transforms, respectively. For IP residuals, average bitrate savings were 4.2%, 10.6% and 5.3% for 4x4-block, 8x8-block and 4x4-and-8x8-block transforms, respectively.

The experiments in Chapter 6 provided also other useful information that can help understand and improve systems with 1-D transforms. These were bitrate savings at relatively lower and higher picture qualities, average percentage of total bitrate used for coding the selected transforms, probabilities of selection of transforms, and visual quality of reconstructed frames. For MC residuals, typically higher bitrate savings were achieved at higher picture qualities than at lower picture qualities. The average (averaged over all sequences and picture qualities) percentage of the total bitrate used for coding the selected transforms were 4.4% for MC residuals and 3.5% for IP residuals when both 4x4 and 8x8-block transforms were used. The probabilities of selection of transforms were consistent with the particular codeword assignment used to indicate the transforms and the probability to choose the 2-D DCT was about 0.5.

While the results presented in this thesis are specific to the particular set of 1-D transforms and the methods to code the transforms coefficients and the selected transforms, these results demonstrated that 1-D transforms can be useful for MC residuals. Further optimizations of the entropy coding methods are likely to improve these results. Overall, we believe these results are promising and indicate that transforms adapted to characteristics of prediction residuals have the potential to improve compression efficiency of prediction residuals and motivate further research along this path.

7.2 Future Research Directions

Many opportunities exist for future research, including areas of exploration specific to the transforms and systems used in this thesis as well as opportunities with broader perspectives.

The system used in this thesis employed a simplified method to code the selected transforms. More sophisticated methods are likely to reduce the bitrate used to code selected the transforms. For example, the chosen transforms for the previously coded blocks (such as upper and left block) may contain information that can help reduce the uncertainty of the transform for the current block. Another possibility is to explore the correlation of the motion vector and the selected transforms. In particular, blocks which have large motion vectors might be more likely to choose the 2-D DCT since such regions are more likely to contain blurry content and for such regions 1-D transforms do not work well. Yet, another possibility is to adapt the coding of the chosen transforms to the video content being encoded. Some video sequences (or frames) can contain more structures along a particular direction than others and adapting the coding scheme to content is likely to improve the efficiency of coding. The context-adaptive binary arithmetic coding framework in H.264/AVC provides a readily available machinery to develop a coding scheme that can adapt to content, spatial neighborhood or other correlated information.

Another simplified method was used to code the quantized coefficients of the 1-D transforms. Standard codecs code the quantized transform coefficients using methods (and codewords for the syntax elements) that are adapted to the 2-D DCT. To reduce the inefficiency of using such methods for coding the coefficients of 1-D transforms we changed only the scanning pattern of the coefficients and the remaining part of the coding algorithm was not modified. However, the best way to approach this problem is to design an entirely new coefficient coding algorithm that is more thoroughly adapted to characteristics of 1-D transforms.

Research directions with broader perspective include investigations of other prediction residuals. The analysis in Chapter 3 showed that MC and RE residuals can contain a significant amount of 1-D structures and we proposed 1-D transforms for these residuals, yet provided experimental results for only MC residuals. A promising path is to explore how 1-D transforms perform on RE residuals. The visual inspection in Chapter

3 suggests that 1-D transforms can even be more useful for RE residuals since the 1-D structures in RE residuals seemed more prominent.

Intra prediction residuals provide significant potential for further study. Even though IP residuals do not contain as many 1-D structures as MC or RE residuals, we presented experimental results with IP residuals since they were readily available in the software we used for our experiments. The experiments provided considerable gains and we believe that transforms better adapted to IP residuals can even increase these gains. In particular, one significant property of IP residuals is that large prediction errors may not form 1-D structures but tend to concentrate in a region of the block, especially a region furthest away from the prediction pixels. While our 1-D transforms perform transforms on 1-D patterns, the direction-adaptive transforms proposed for images combine these 1-D patterns by applying a second set of transform on these 1-D patterns to obtain 2-D transforms. It seems that an approach midway between these two approaches may better capture the described characteristics of IP residuals. In other words, combining few neighboring 1-D patterns may be a worthwhile path for future study of IP residuals.

Bibliography

[1] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transfom. *Computers, IEEE Transactions on*, C-23(1):90–93, Jan. 1974.

[2] A. Al, B.P. Rao, S.S. Kudva, S. Babu, D. Sumam, and A.V. Rao. Quality and complexity comparison of h.264 intra mode with jpeg2000 and jpeg. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 1, pages 525 – 528 Vol. 1, 24-27 2004.

[3] C. Auyeung, J. J. Kosmach, M. T. Orchard, and T. Kalafatis. Overlapped block motion compensation. In P. Maragos, editor, *Proc. SPIE Vol. 1818, p. 561-572, Visual Communications and Image Processing '92, Petros Maragos; Ed.*, volume 1818 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 561–572, November 1992.

[4] J. Balle and M. Wien. Extended texture prediction for h.264/avc intra coding. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 6:VI –93 –VI –96, 16 2007-Oct. 19 2007.

[5] Dimitri P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1995.

[6] G. Bjontegaard. Calculation of average psnr differences between rd-curves. *VCEG Contribution VCEG-M33*, April 2001.

[7] M.H. Chan, Y.B. Yu, and A.G. Constantinides. Variable size block matching motion compensation with applications to video coding. *Communications, Speech and Vision, IEE Proceedings I*, 137(4):205–212, Aug 1990.

[8] C.-L. Chang and B. Girod. Direction-adaptive discrete wavelet transform for image compression. *Image Processing, IEEE Transactions on*, 16(5):1289–1302, May 2007.

[9] C.-F. Chen and K.K. Pang. The optimal transform of motion-compensated frame difference images in a hybrid coder. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on]*, 40(6):393–397, Jun 1993.

[10] M.D. Flickner and N. Ahmed. A derivation for the discrete cosine transform. *Proceedings of the IEEE*, 70(9):1132–1134, Sept. 1982.

[11] M. Flierl and B. Girod. Multihypothesis motion estimation for video coding. *Data Compression Conference, 2001. Proceedings. DCC 2001.*, pages 341–350, 2001.

[12] B. Girod. Why b-pictures work: a theory of multi-hypothesis motion-compensated prediction. *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, 2:213–217 vol.2, 4-7 Oct 1998.

[13] B. Girod. Motion-compensating prediction with fractional-pel accuracy. *Communications, IEEE Transactions on*, 41(4):604–612, Apr 1993.

[14] B. Girod. The efficiency of motion-compensating prediction for hybrid coding of video sequences. *Selected Areas in Communications, IEEE Journal on*, 5(7):1140–1154, Aug 1987.

[15] B. Girod. Efficiency analysis of multihypothesis motion-compensated prediction for video coding. *Image Processing, IEEE Transactions on*, 9(2):173–183, Feb 2000.

[16] K.-C. Hui and W.-C. Siu. Extended analysis of motion-compensated frame difference for block-based motion prediction error. *Image Processing, IEEE Transactions on*, 16(5):1232–1245, May 2007.

[17] F. Kamisli and J.S. Lim. Directional wavelet transforms for prediction residuals in video coding. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 613 –616, 7-10 2009.

[18] F. Kamisli and J.S. Lim. Transforms for the motion compensation residual. *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 789–792, April 2009.

[19] P. Kauff and K. Schuur. Shape-adaptive dct with block-based dc separation and dc correction. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(3):237–242, Jun 1998.

[20] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *Image Processing, IEEE Transactions on*, 14(4):423–438, April 2005.

[21] Y.L. Lee and H.W. Park. Loop-filtering and post-filtering for low bit-rates moving picture coding. *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, 1:94–98 vol.1, 1999.

[22] Jae S. Lim. *Two-dimensional Signal and Image Processing.* Prentice Hall, 1990.

[23] P. List, A. Joch, J. Lainema, G. Bjntegaard, and M. Karczewicz. Adaptive deblocking filter. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):614–619, July 2003.

[24] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397 –3415, Dec 1993.

[25] H.S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky. Low-complexity transform and quantization in h.264/avc. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):598 – 603, july 2003.

[26] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):620 – 636, july 2003.

[27] R. Neff and A. Zakhor. Very low bit-rate video coding based on matching pursuits. *Circuits and Systems for Video Technology, IEEE Transactions on*, 7(1):158–171, Feb 1997.

[28] W. Niehsen and M. Brunig. Covariance analysis of motion-compensated frame differences. *Circuits and Systems for Video Technology, IEEE Transactions on*, 9(4):536–539, Jun 1999.

[29] S. Nogaki and M. Ohta. An overlapped block motion compensation for high quality motion picture coding. *Circuits and Systems, 1992. ISCAS '92. Proceedings., 1992 IEEE International Symposium on*, 1:184–187 vol.1, 10-13 May 1992.

[30] J. Ohm, M. v.d. Schaar, and J. W. Woods. Interframe wavelet coding - motion picture representation for universal scalability. *EURASIP Signal Processing: Image Communication, Special Issue on Digital Cinema*, 19:877–908, October 2004.

[31] M.T. Orchard and G.J. Sullivan. Overlapped block motion compensation: an estimation-theoretic approach. *Image Processing, IEEE Transactions on*, 3(5):693–699, Sep 1994.

[32] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *Signal Processing Magazine, IEEE*, 15(6):23 –50, nov 1998.

[33] K.K. Pang and T.K. Tan. Optimum loop filter in hybrid coders. *Circuits and Systems for Video Technology, IEEE Transactions on*, 4(2):158–167, Apr 1994.

[34] A. Puri, H.-M. Hang, and D. Schilling. An efficient block-matching algorithm for motion-compensated coding. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.*, volume 12, pages 1063–1066, Apr 1987.

[35] A. Puri, H.-M. Hang, and D. L. Schilling. Interframe coding with variable block-size motion compensation. In *IEEE Global Telecom. Conf. (GLOBECOM)*, pages 65–69, 1987.

[36] K. Ratakonda, Seung Chul Yoon, and N. Ahuja. Coding the displaced frame difference for video compression. *Image Processing, 1997. Proceedings., International Conference on*, 1:353–356 vol.1, 26-29 Oct 1997.

[37] Ian E.G. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia.* Wiley, 2003.

[38] T. Shiodera, A. Tanizawa, and T. Chujoh. Bidirectional intra prediction. *TU-T SG16/Q.6 VCEG, VCEG-AE14*, Jan 2007.

[39] G.J. Sullivan and R.L. Baker. Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks. *Global Telecommunications Conference, 1991. GLOBECOM '91. 'Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services*, pages 85–90 vol.1, 2-5 Dec 1991.

[40] G.J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *Signal Processing Magazine, IEEE*, 15(6):74 –90, nov 1998.

[41] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998.

[42] T.K. Tan, C.S. Boon, and Y. Suzuki. Intra prediction by template matching. *Image Processing, 2006 IEEE International Conference on*, pages 1693 –1696, Oct. 2006.

[43] Bo Tao and M.T. Orchard. Prediction of second-order statistics in motion-compensated video coding. *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pages 910–914 vol.3, 4-7 Oct 1998.

[44] D. Taubman and A. Zakhor. Orientation adaptive subband coding of images. *Image Processing, IEEE Transactions on*, 3(4):421–437, Jul 1994.

[45] V. Velisavljevic, B. Beferull-Lozano, and M. Vetterli. Efficient image compression using directionlets. *Information, Communications and Signal Processing, 2007 6th International Conference on*, pages 1–5, 10-13 Dec. 2007.

[46] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P.L. Dragotti. Directionlets: anisotropic multidirectional representation with separable filtering. *Image Processing, IEEE Transactions on*, 15(7):1916–1933, July 2006.

[47] A. Vetro, W. Matusik, H Pfister, and Jun Xin. Coding approaches for end-to-end 3d tv systems. *Picture Coding Symposium*, pages 1–5, 2004.

[48] S. Li W. Ding, F. Wu. Lifting-based wavelet transform with directionally spatial prediction. *Picture Coding Symp.*, 62:291–294, January 2004.

[49] G.K. Wallace. The jpeg still picture compression standard. *Consumer Electronics, IEEE Transactions on*, 38(1):xviii–xxxiv, Feb 1992.

[50] T. Wedi and H.G. Musmann. Motion- and aliasing-compensated prediction for hybrid video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):577–586, July 2003.

[51] T. Wiegand and B. Girod. Lagrange multiplier selection in hybrid video coder control. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 542 –545 vol.3, 2001.

[52] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):560 –576, july 2003.

[53] Y. Ye and M. Karczewicz. Improved intra coding. *ITU-T Q.6/SG16 VCEG, VCEG-AG11*, Oct 2007.

[54] Y. Ye and M. Karczewicz. Improved h.264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning. *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2116 –2119, Oct. 2008.

[55] Bing Zeng and Jingjing Fu. Directional discrete cosine transforms for image coding. *Multimedia and Expo, 2006 IEEE International Conference on*, pages 721–724, 9-12 July 2006.