Transforms for Prediction Residuals Based on Prediction Inaccuracy Modeling

by

Xun Cai

B.E. University of Science and Technology of China (2010)
M.S. Massachusetts Institute of Technology (2012)

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

February 2017

© Massachusetts Institute of Technology 2017. All rights reserved.

Signature redacted

Author .............................................................
Department of Electrical Engineering and Computer Science

Signature redacted

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

Signature redacted

Accepted by ...........
Leslie A. Kolodziejski
Professor of Electrical Engineering
Chair, Department Committee on Graduate Students
Transforms for Prediction Residuals Based on Prediction Inaccuracy Modeling

by

Xun Cai

Submitted to the Department of Electrical Engineering and Computer Science on December 29, 2016, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

In a typical transform-based image and video compression system, an image or a video frame is predicted from previously encoded information. The prediction residuals are encoded with transforms. With a proper choice of the transform, a large amount of the residual energy compacts into a small number of transform coefficients. This is known as the energy compaction property. Given the covariance function of the signal, the linear transform with the best energy compaction property is the Karhunen Loève transform.

In this thesis, we develop a new set of transforms for prediction residuals. We observe that the prediction process in practical video compression systems is usually not accurate. By studying the inaccuracy of the prediction process, we can derive new covariance functions for prediction residuals. The estimated covariance function is used to generate the Karhunen Loève transform for residual encoding.

In this thesis, we model the prediction inaccuracy for two types of residuals. Specifically, we estimate the covariance function of the directional intra prediction residuals. We show that the covariance function and the optimal transform for directional intra prediction residuals are related with the one-dimensional gradient of boundary predictors. We estimate the covariance function of the motion-compensated prediction residuals. We show that the covariance function and the optimal transform for motion-compensated prediction residuals are related with the two-dimensional gradient of the displaced reference block.

The proposed transforms are evaluated using the energy compaction property and the rate-distortion metric in a practical video coding system. Experimental results indicate that the proposed transforms significantly improve the performance in a typical transform-based compression scenario.

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

First of all, I would like to express my sincere gratitude to my thesis advisor, Professor Jae S. Lim. There are many great researchers who are persistent in pushing the frontier of human knowledge. There are also many pioneering innovators who devote themselves to making life better for human beings. Professor Lim is one of very few professors who excel in both aspects. I feel extremely lucky to be advised by him. Professor Lim constantly shares his insights, knowledge, experience and stories in both academia and industry. These are not only crucial to my entire Ph.D. study including the completion of this thesis, but also immensely valuable to my future career. Being a Ph.D. student in the group led by Professor Lim is also an academic luxury in many other aspects. For example, being free from concerns about funding and project deadlines, able to explore the field as flexibly as possible, abundant of opportunities to give lectures and publish the high-quality research articles in leading journals and conferences. Professor Lim is also a great life mentor. In the years I worked with him, Professor Lim set an excellent example of being extremely creative, professional, meticulous, well-prepared, responsible and calm, all of which are key to one’s success. I learned so much from him.

Next, I feel extremely grateful to Dr. John Apostolopoulos and Professor Vivienne Sze for serving on my thesis committee. During the thesis work, Dr. John Apostolopoulos and Professor Vivienne Sze provided so much constructive feedback that significantly helps improving the quality of the thesis. John visited MIT frequently as a guest lecturer. His unique vision as the CTO of an internationally-renowned large corporation in the industry is a huge boost for the career of a young professional. Vivienne has been a leader in the establishment of the latest state-of-art video compression standard. Her successful experience and insights in developing the HEVC standard are extremely helpful for my thesis work. I feel very lucky to have John and Vivienne on my thesis committee.

My life at ATSP becomes so much more amazing and enjoyable with Cindy, Megan, Lucas, Dr. Gun Bang and Dr. Chang Yoo. I would like to thank all of
them. Cindy is particularly crucial to ATSP. She covers all the logistics and makes our life at ATSP much more efficient. Without Cindy, everything will become unimaginably difficult. Megan and Lucas are great lab mates and friends that I feel lucky to have. The daily discussion with them significantly helped the progress of my research and brought a lot of fun. Dr. Gun Bang and Dr. Chang Yoo visited MIT for a long time. They brought in a lot of interesting insights as senior researchers in the signal processing community.

I was able to join a few other research teams in the past few years. The experience was tremendously helpful for my research vision and career development. These teams include, Information Processing Center at University of Science and Technology of China, Cross-disciplinary Scholars of Science and Technology in University of California, Los Angeles, Google Chrome Media, YouTube and Google Research. I would like to say thank you to all my supervisors and friends in these groups. They are: Professor Rong Zhang, Professor John Miao, Professor Peter S. Pao, Dr. Qunshan Gu, Dr. Yaowu Xu, Dr. Charles Krasic, Professor Peyman Milanfar, Dr. Feng Yang, Dr. Jingning Han and Dr. Hui Su.

Ph.D. life at MIT is not all about research and work. I would like to thank all my friends who helped me, supported me and had fun together. Life without them will be much less colorful. They are: Yongbao Sun, Wenjing Fang, Xu Zhang, Yan Zhao, Xue Feng, James Noraky, Xuwen Zhu, Xin Hong, Jingyi Fang, Xinli Cai, Haoyan Geng and many others. Special thanks to Xiaokui Shu, an excellent photographer who helped create the image library for the group. In addition, I would like to thank the enthusiastic and responsible teammates in the Deadline Oriented band.

Family education is very important in my life. I would like to thank my parents and parents in law for their support of my life, education and career. In my early childhood, my parents exposed latest technologies to me, which stimulated my endless curiosity in science and technology. They also have great vision for my education and career, which was simply not possible among many families in China at that time. These turned out to be extremely important for my career growth. In recent years, my parents and in-laws keep being supportive of the life path that I choose. Without
their support, life will be much harder.

Long time ago, one professor I worked with mentioned how interesting it is to get married with his wife entirely unfamiliar with his field and how helpful she can be on his research. Years later, I got married with Yuna and things turned out to be true. Yuna studied humanity science and language. She does not understand a single equation on my papers. However, her knowledge as an outsider often indirectly brought new perspective to the problems that I have been working on. As a Ph.D. student in engineering, I usually feel difficult to solve a problem in any non-technical approach. Fortunately, she is really good at that. Yuna is also a very caring mother, her support to the family is of great help to my career. I feel particularly grateful to Yuna for her selfless accompany alongside me these years. Having Yuna as my wife, friend and soul mate is definitely a blessing in my life.

Last, I would like to thank my cute son, Lawrence Cai, who brings so much fun for everyone and teaches me how to be extremely patient.
# Contents

1 Thesis overview .................................................. 21

2 Background ......................................................... 25
   2.1 Transform-based image and video compression systems .......... 25
   2.2 Block partition .................................................. 27
   2.3 Prediction ....................................................... 29
      2.3.1 Intra prediction ........................................... 30
      2.3.2 Inter prediction ........................................... 33
   2.4 Transform ....................................................... 37
   2.5 Quantization .................................................... 39
   2.6 Entropy coding ................................................ 40
   2.7 Decoder ........................................................ 42

3 Previous Research ................................................ 45
   3.1 Karhunen Loève transform ...................................... 45
   3.2 Transforms based on covariance modeling ......................... 47
      3.2.1 Discrete cosine transform ................................ 47
      3.2.2 Asymmetric discrete sine transform ....................... 49
      3.2.3 Transforms for motion-compensated residuals ............... 51
   3.3 Transforms based on covariance estimation ....................... 53
      3.3.1 Offline covariance estimation ................................ 53
      3.3.2 On-the-fly covariance estimation ......................... 54
   3.4 Motivation of thesis ........................................... 55
4 Prediction Inaccuracy Modeling

4.1 Residual non-stationarity ........................................... 57
4.2 Residual-gradient correlation ...................................... 59
4.3 Prediction inaccuracy model ....................................... 62
4.4 KLT based on the prediction inaccuracy model ................. 65

5 Directional intra prediction inaccuracy model .................... 69

5.1 Characteristics of directional intra prediction residuals ....... 69
5.2 Model derivation ..................................................... 72
  5.2.1 Model for horizontal prediction ............................. 72
  5.2.2 Model for arbitrary direction ................................ 74
  5.2.3 Alternative derivation of the model ....................... 76
5.3 Model statistics ..................................................... 77
  5.3.1 Mean ............................................................ 77
  5.3.2 Variance ....................................................... 78
  5.3.3 Effectiveness of the proposed model in variance estimation 79
  5.3.4 Covariance ................................................... 81
  5.3.5 Discussion on the R function and the covariance function 82
5.4 Transforms based on the proposed covariance function ......... 84
  5.4.1 Properties of the covariance function and transforms .... 84
  5.4.2 Discussion on the proposed transforms .................... 87

6 Implementation and experimental validation of the directional intra prediction inaccuracy model ........................................... 89

6.1 Gradient computation ................................................. 89
  6.1.1 $n'$ is a positive integer .................................. 90
  6.1.2 $n'$ is not a integer ......................................... 91
  6.1.3 $n'$ is zero (the corner predictor is used) ................ 91
6.2 Example of applying the proposed KLT ........................... 93
6.3 Energy compaction performance .................................... 98
  6.3.1 Experimental setup ......................................... 98
6.3.2 Test sequences ........................................... 99
6.3.3 Results: hybrid with the DCT .......................... 102
6.3.4 Results: hybrid with the ADST ......................... 103
6.3.5 Comparison of the energy compaction performance on different transforms ........................................... 106
6.3.6 Choice of transforms .................................. 107
6.3.7 Discussion of energy compaction performance ....... 109
6.4 H.264/AVC system performance ......................... 110
6.4.1 System implementation ............................... 110
6.4.2 Experimental results .................................. 112
6.4.3 Visual quality .......................................... 117
6.4.4 Discussion on the computational complexity ........ 119

7 Motion compensation inaccuracy model .................. 121
7.1 Characteristics of motion-compensated prediction residuals .......... 121
7.2 Model derivation ........................................ 123
7.3 Model statistics ......................................... 125
  7.3.1 Mean ................................................. 125
  7.3.2 Variance ............................................ 126
  7.3.3 Effectiveness of the proposed model in variance estimation .. 127
  7.3.4 Covariance ........................................... 129
  7.3.5 Discussion on the R function and the covariance function ... 130
7.4 Transforms based on the proposed covariance function .......... 131

8 Implementation and experimental validation of the motion compensation inaccuracy model 133
8.1 Gradient Computation .................................... 133
8.2 Energy compaction performance .......................... 135
  8.2.1 Experimental setup ................................ 136
  8.2.2 Results: hybrid with the DCT ..................... 137
  8.2.3 Choice of transforms ................................ 139
List of Figures

2-1 The diagram of a typical transform-based video compression system . 27

2-2 Block partition strategy in JPEG: in the JPEG system, an image is partitioned into 8x8 blocks. 28

2-3 Block partition strategy in H.264/AVC: in the H.264/AVC system, a frame is first partitioned into 16x16 macroblocks. Each macroblock is processed as an independent unit. Each macroblock can be further partitioned into 8x8 blocks or even 4x4 subblocks. This partition of a macroblock is shown in the upper left macroblock. 28

2-4 Block partition strategy in HEVC: in the HEVC system, a frame is first partitioned into very large Coding Tree Units (CTU). Each CTU is divided into a quadtree of Coding Units (CU). Each CU is partitioned into multiple Prediction Units in the prediction step and a quadtree of Transform Units in the transform step. A large variety of block sizes are available in HEVC. Block partition in HEVC is extremely flexible. 29

2-5 An example of vertical intra prediction: the top boundary above the current block (shaded) is copied along the vertical direction to generate prediction. 31
2-6 Illustration of template matching method: the current block is to be predicted and encoded. To find a correct match, the encoded template (red) is used to find the best match in the encoded region. The encoded block with best matching template is used to generated the prediction for the current block. Since only encoded template is involved in the search process, the decoder can find the same matching block without any side information.

2-7 An example of intra prediction

2-8 An example of block matching motion compensation: the best matching block (yellow) in the first frame is used to predict the current block (blue) in the second frame. The location offset (motion vector) of two blocks is illustrated by the arrow.

2-9 Motion compensated frame and residual frame: in this example, Frame 2 is predicted from Frame 1. For each block in Frame 2, we find the best reference block in Frame 1 that yields the smallest error. We form the prediction frame by displacing the reference blocks. The prediction error, i.e. the motion compensated residual frame is shown in the lower right figure.

2-10 An image and its Fourier transform magnitude image: the energy of the original image spreads across the entire space, while its energy compacts only in a small number of large Fourier transform coefficients.

2-11 Energy compaction plot: the energy compaction plot indicates the performance of the transform. In this example, more than 90% of the energy is preserved with 1% of largest Fourier transform coefficients for a typical image.

2-12 Illustration of synchronization in a compression system: in order to use reconstructed data for prediction, a decoder is included in the encoder side. This ensures the reconstructed data seen by the encoder and the decoder are always identical.
3-1 4x4 2D-DCT basis functions: a total of 16 DCT basis functions. Each subfigure shows a transform basis function of size 4x4.

3-2 4x4 ADST basis functions: a total of 16 ADST functions. Each subfigure shows a transform basis function of size 4x4.

3-3 4x4 one-dimensional transforms

4-1 An example of typical frames and prediction residual frames

4-2 An example of the correlation between residual frame and gradient frame

4-3 An example of the correlation between residual frame and gradient frame

4-4 An example of prediction where the gradient is small: either the vertical prediction or the diagonal down prediction yields an accurate prediction

4-5 An example of prediction where the gradient is large: only a particular prediction direction yields the smallest prediction error. The prediction accuracy is sensitive to the prediction direction.

4-6 The flowchart of the prediction inaccuracy model: general principle

5-1 An example of the intra frame and residual frame

5-2 A residual block that shows non-stationarity

5-3 Illustration of the derivation: horizontal

5-4 Illustration of the derivation: arbitrary

5-5 Comparison between the estimated \( \sigma \) function and residual signal

5-6 Relative energy compaction curves

5-7 An example of the proposed transform basis functions. \( \rho_1 = \rho_2 = 0.99 \)

5-8 An example of the proposed transform basis functions. \( \rho_1 = \rho_2 = 0.1 \)

5-9 The flowchart of the prediction inaccuracy model: intra

6-1 Gradient computation for positive integer \( n' \)

6-2 Gradient computation for non-integer \( n' \)

6-3 Gradient computation for \( n' = 0 \)
6-4 A block of original image (left) and pixel intensities (right). Current block to be encoded is outlined with red grid.

6-5 The vertical prediction block

6-6 The residual block

6-7 The $\sigma$ function

6-8 KLT basis functions

6-9 Magnitude of KLT coefficients and DCT coefficients

6-10 Energy compaction performance of the KLT and the DCT

6-11 Sample sequences in the derf set: QCIF and CIF resolutions

6-12 Sample sequences in the derf set: 720P and 1080P resolutions

6-13 Energy compaction performance of the hybrid transform (vs DCT).
   Sequence: Carphone_QCIF

6-14 Coefficient saving to the frequency of choosing the KLT

6-15 Energy compaction performance of the hybrid transform (vs ADST).
   Sequence: Carphone_QCIF

6-16 Coefficient saving to the frequency of choosing the KLT

6-17 Percentage of preserved coefficients relative to the DCT

6-18 Choice of transform for each block

6-19 H.264/AVC implementation overview

6-20 R-D performance

6-21 BD rate of the proposed method vs ADST: QCIF and CIF resolutions

6-22 BD rate of the proposed method vs ADST: 720P and 1080P resolution

6-23 Frequency of choosing the proposed transform

6-24 Percentage of side information bitrate

6-25 Visual comparison of the proposed method vs ADST

7-1 Video frame and motion-compensated residual frame

7-2 Illustration of the model derivation

7-3 Comparison between the estimated $\sigma$ function and residual magnitude frame
7-4 Relative energy compaction functions .................................. 128
7-5 The flowchart of the prediction inaccuracy model: inter ............ 132

8-1 Gradient computation for the model of motion-compensated residuals 134
8-2 Energy compaction performance of the hybrid transform (vs DCT).
   Sequence: foreman_CIF ...................................................... 137
8-3 Coefficient saving to the frequency of choosing the KLT .......... 139
8-4 Choice of transform for each block .................................... 140
8-5 H.264/AVC implementation overview .................................. 142
8-6 R-D performance: motion compensation with quarter-pixel motion
   vector precision ............................................................. 143
8-7 R-D performance: motion compensation with integer-pixel motion
   vector precision ............................................................. 144
8-8 BD rate of the proposed method vs DCT: QCIF and CIF resolutions 145
8-9 BD rate of the proposed method vs DCT: 720P and 1080P resolutions 146
8-10 Visual comparison of the proposed method vs DCT ............... 148

A-1 Example of transform basis functions with zero variance .......... 165
A-2 Example of transform basis functions under a rectangular variance support ............................................................. 169
List of Tables

6.1 Coefficient saving of the hybrid transform relative to the DCT. Sequence: Carphone_QCIF .......................... 103
6.2 Average coefficient saving of the hybrid transform relative to the DCT. 103
6.3 Coefficient saving of the hybrid transform relative to the ADST. Sequence: Carphone_QCIF .......................... 104
6.4 Average coefficient saving of the hybrid transform relative to the ADST. 105

8.1 Coefficient saving of the hybrid transform relative to the DCT. Sequence: Foreman_CIF .......................... 138
8.2 Average coefficient saving of the hybrid transform relative to the DCT. 138

9.1 Comparison of prediction inaccuracy models for intra and inter residuals ............................................. 150

B.1 Energy compaction performance of proposed transform (Intra): the table shows the coefficient saving (when roughly 5% coefficients are preserved) and the frequency of choosing KLT with DCT and ADST. 172
B.2 Energy compaction performance of proposed transform (continued) 173
B.3 Energy compaction performance of proposed transform (Inter): the table shows the coefficient saving (when roughly 5% coefficients are preserved) and the frequency of choosing KLT with DCT. .......... 174
B.4 Energy compaction performance of proposed transform (continued) 175
Chapter 1

Thesis overview

In transform-based image and video coding, transforms are applied to images and prediction residuals. Transform coefficients are quantized and entropy encoded. Image and residual signals can be represented efficiently in the transform domain. With a proper choice of the transform, a large amount of energy can be preserved with a small number of large transform coefficients. This is known as the energy compaction property of transforms [24, 27]. A transform with better energy compaction allows the image and video signals to be encoded with fewer coefficients, while preserving a certain level of fidelity.

It is well known that for a stochastic signal with a known covariance function, the linear transform with the best energy compaction property is the Karhunen Loève transform (KLT) [23]. The KLT of typical images has been investigated both theoretically and empirically. It has been noted that the KLT basis functions of typical images are very close to the two-dimensional Discrete Cosine Transform (2D-DCT) [3]. The 2D-DCT is also the KLT of a random process characterized by the first-order Markov model for images. Therefore, as a reasonable approximation to the KLT for images, the 2D-DCT is extensively used in many image and video coding systems [4, 6, 36, 40–42].

In various image and video coding systems, prediction is used to reduce the correlation present in image and video signals. The prediction residuals, rather than raw image intensities are encoded by transforms. The characteristics of prediction resid-
uals may be significantly different from those of typical images. As a consequence, the optimal transform for prediction residuals may be different from the 2D-DCT. A substantial amount of effort has been spent on developing transforms for prediction residuals.

In this thesis, we propose the prediction inaccuracy modeling and use it for transform design. We observe that the prediction step in a compression system is inaccurate in many cases. The characteristics of the prediction residuals depend on the inaccuracy in the prediction step. By carefully modeling the prediction step and prediction inaccuracy, we can estimate the residual statistics more accurately. Specifically, we develop new residual covariance functions from the proposed model. Based on the estimated covariance function, we can obtain the KLT for prediction residuals.

In this thesis, we show that the prediction inaccuracy modeling can be applied to a large variety of residual signals. As one example, we observe that the directional intra prediction is most effective for regions of edges with clear directionality. In the ideal case, edges can be predicted fairly accurately if an accurate prediction direction is used. In practice though, an accurate prediction direction is hard to obtain. Based on the inaccuracy of prediction direction, we estimate the residual covariance as a function of the coded boundary gradient. As another example, we observe that motion-compensated prediction residual is small in smooth regions and still regions. In the presence of motion, the motion-compensated prediction residual is large in the regions with edges and large discontinuities in part because of the motion-compensated prediction inaccuracy. The motion-compensated prediction inaccuracy is modeled as a result of inaccuracy within the motion vectors. We relate the residual covariance function to the gradient of the reference block. In both cases, the KLT of the estimated covariance functions is proposed.

We will discuss the prediction inaccuracy modeling for the rest of thesis. This thesis is organized as follows:

- In Chapter 2, we review the concepts and components of transform-based video compression systems.
• In Chapter 3, we review previous literature on transform design for image and video compression applications.

• We discuss the proposed method from Chapter 4 to Chapter 8. In Chapter 4, we analyze the residual non-stationarity present in typical prediction residuals. Based on the empirical observations of the residual non-stationarity, we propose the prediction inaccuracy modeling process from a high-level point of view. The prediction inaccuracy modeling approach is applied to two kinds of residual signals. We discuss the intra case in Chapter 5 and Chapter 6. We discuss the inter case in Chapter 7 and Chapter 8.

• In Chapter 5, we apply the prediction inaccuracy modeling to the directional intra prediction residuals. Specifically, we derive the model for directional intra prediction and relate the covariance function with the one-dimensional gradient of coded boundaries.

• In Chapter 6, we evaluate the performance of the model in Chapter 5 under a typical transform-based coding scenario.

• In Chapter 7, we apply the prediction inaccuracy modeling to the motion-compensated prediction residuals. Specifically, we derive the model for motion compensation and relate the covariance function with the two-dimensional gradient of the displaced reference block in the encoded reference frame used in motion compensation.

• In Chapter 8, we evaluate the performance of the model in Chapter 7 under a typical transform-based coding scenario.

• In Chapter 9, we summarize the contribution of the thesis and discuss possible extensions of the thesis.
Chapter 2

Background

In this chapter, we review the background of transform-based image and video compression systems. In Section 2.1, we discuss the high-level structure of a typical transform-based image and video compression system. From Section 2.2 to Section 2.6, we review the block partition, prediction, transform, quantization and entropy coding components at the encoder side, respectively. In Section 2.7, we discuss the design of the decoder.

2.1 Transform-based image and video compression systems

A typical video sequence in its raw format contains a huge amount of data. Many applications, such as video streaming and video storage, are not possible without video compression. In video compression, we are concerned with representing the video signal with as few bits as possible while preserving a certain level of video quality. Depending on the video quality we would like to preserve in the compression process, image and video compression can be broadly classified into lossless compression and lossy compression. In lossless compression, image and video signals are perfectly preserved. In lossy video compression, we would like to preserve important visual information within image and video signals. Distortion is allowed as long as it does
not significantly affect the quality of image and video signals. In this thesis, we are concerned with the lossy image and video compression. Many video compression architectures and systems have been proposed for lossy compression [4, 32, 35, 36, 40, 42]. Among these systems, the transform-based compression systems attract a lot of research interest among the video coding community, due to its intuitive design and great performance.

Figure 2-1 shows a high-level diagram of a typical transform-based compression system. The encoder side of the system (upper branch) takes a raw input video and produces the compressed bit stream. The compressed bit stream is transmitted or stored. The decoder side of the system (lower branch) receives the compressed bit stream and reconstructs the video. The system is designed in such a way that the compressed bit stream is much smaller in size relative to the raw input data. In addition, the reconstructed video should preserve important visual information within the input video.

To achieve this, a typical transform-based compression system exploits observations such as the limitations of the human visual system and redundancy within typical video signals. In this thesis, we focus on the data redundancy. There are various kinds of data redundancy present in a typical video sequence. For example, for a typical video frame, most regions within that frame are smooth. Neighboring pixels in these regions tend to share similar intensities. This is referred to as the spatial redundancy. In a typical video sequence, different frames close to each other in the temporal direction tend to be similar, except for object motions. This is referred to as the temporal redundancy.

In Figure 2-1, we show the typical structure of a transform-based compression system. The encoder side primarily consists of four modules. Each module is responsible for reducing certain kinds of data redundancy in the video data. We discuss each module in detail for the rest of the chapter.
2.2 Block partition

For a typical transform-based image and video compression system, each frame is encoded one by one. For each frame, the input signal is first partitioned into small blocks. The partition process allows further processing to adapt to local characteristics of images and video frames.

A compression system may choose to use a fixed block size or a variable block size. For example, in the JPEG image compression system [40], each image is segmented into blocks of fixed size 8x8. This is illustrated in Figure 2-2. In the H.264/AVC video compression system [42], each video frame is first segmented into macroblocks of 16x16 pixels. Each macroblock can be adaptively segmented into smaller blocks. The system may choose to use non-square rectangular blocks, such as blocks of 16x8 pixels and blocks of 8x16 pixels. An example is illustrated in Figure 2-3. In more recent video compression systems such as the High Efficiency Video Coding (HEVC) and the VP systems [4, 36], more flexible block partition strategies are used.
Figure 2-2: Block partition strategy in JPEG: in the JPEG system, an image is partitioned into 8x8 blocks.

Figure 2-3: Block partition strategy in H.264/AVC: in the H.264/AVC system, a frame is first partitioned into 16x16 macroblocks. Each macroblock is processed as an independent unit. Each macroblock can be further partitioned into 8x8 blocks or even 4x4 subblocks. This partition of a macroblock is shown in the upper left macroblock.

The system usually starts with very large blocks and recursively segments each block into smaller blocks. The partition strategy can adapt to different steps in the system.
It is very flexible. An example of the HEVC block partition is shown in Figure 2-4. Interested readers are referred to [37] for a complete overview of the sophisticated block partition strategy used in the HEVC. It has been noted that larger block sizes and flexible partition strategy used in HEVC result in a significant improvement of coding performance due to its better adaptivity to the video content, especially for high-definition video content.

![Block partition strategy in HEVC](image)

Figure 2-4: Block partition strategy in HEVC: in the HEVC system, a frame is first partitioned into very large Coding Tree Units (CTU). Each CTU is divided into a quadtree of Coding Units (CU). Each CU is partitioned into multiple Prediction Units in the prediction step and a quadtree of Transform Units in the transform step. A large variety of block sizes are available in HEVC. Block partition in HEVC is extremely flexible.

### 2.3 Prediction

After a frame is partitioned into small blocks, prediction is applied to each block. Each block may choose a different prediction method depending on the type of redundancy to be reduced in that block. In a typical image or video frame, pixel intensities in nearby spatial regions are likely to be close, particularly in smooth regions. This is referred to as the spatial correlation. The intra prediction can be used to reduce the spatial correlation. In the temporal direction, two consecutive video frames may look
similar to each other in most regions. Much of the variation between two frames is
due to object motions. This is referred to as the temporal correlation. The inter
prediction can be used to reduce the temporal correlation.

2.3.1 Intra prediction

The first type of prediction is the intra prediction. It is used to exploit the spatial
correlation within a frame. When processing the current block, intra prediction works
by generating the prediction for the current block from encoded information within
the same frame. The prediction residual signal instead of the raw image signal is
encoded. We hope to generate an accurate prediction with a reasonable choice of the
prediction method so that the amount of energy left to be encoded in the residual
signal can be significantly reduced.

A variety of intra prediction methods have been proposed. In the directional
intra prediction, the current block is predicted by copying a set of boundary pixels
along a certain direction. The boundary pixels are in the row immediately adjacent
to the top of the current block and the column immediately adjacent to the left of
the current block. They are in the blocks that have been previously encoded and
reconstructed, with possible filtering prior to intra prediction. Figure 2-5 shows an
example of the directional intra prediction. In this example, the prediction for the
current block is constructed by copying the upper coded boundary along the vertical
direction. Directional intra prediction of other directions work similarly, possibly with
boundary interpolation. The directional intra prediction is used extensively in video
coding systems. For example, in the H.264/AVC intra coding, 9 different modes are
used [32]. Among these modes, 8 prediction directions are used in directional intra
prediction. In the HEVC intra coding system, as many as 35 modes are used [25].
Among them, 33 directions are used in directional intra prediction. The directional
intra prediction is mostly effective when the current block contains patterns with
clear directionality, such as sharp edges.

In the DC prediction, a block of pixels are predicted from the average of a set of
encoded pixels. For example, in the JPEG compression system, the mean of the cur-
Figure 2-5: An example of vertical intra prediction: the top boundary above the current block (shaded) is copied along the vertical direction to generate prediction

rent block is first subtracted from the mean of the previous block. This is equivalent to generating the current prediction from the DC value of the previous coded block. In the H.264/AVC and HEVC systems, the DC prediction is generated by averaging the boundary predictors. The DC prediction is mostly useful when the current block contains contents that do not have a clear pattern. After the mean removal process, the remaining spatial correlation is handled by transforms.

The DC prediction generates a constant prediction pattern. To extend this idea, the Planar mode in the HEVC and True Motion prediction [4] are proposed to generate non-constant patterns. The prediction is constructed with spatial-coordinate weighted interpolation from boundary predictors.

In the Template Matching method proposed in [38], it is observed that the current block may have repetitive patterns that can be predicted by translating another block in the encoded region of the same frame. This is very similar to the motion-compensated video coding. Different from the motion-compensated prediction, the repetitive pattern within a frame may be located far away from the current block. To prevent the large overhead of transmitting the coordinate offset, a search process is applied at both the encoder and the decoder side. The search process uses a
template that surrounds the current block as a mark. Blocks with similar templates within the encoded region are used to predict the current block. The process is illustrated in Figure 2-6. While the template matching process results in a significant coding performance improvement, the major concern that prevents its extensive use in practice is the computational complexity at the decoder side. As discussed above, the location information of the best matching template is not transmitted. Therefore, the same search process has to be performed at the decoder side. This process is considered to be computationally expensive.

![Encoded region](image)

**Figure 2-6:** Illustration of template matching method: the current block is to be predicted and encoded. To find a correct match, the encoded template (red) is used to find the best match in the encoded region. The encoded block with best matching template is used to generated the prediction for the current block. Since only encoded template is involved in the search process, the decoder can find the same matching block without any side information.

Recently, due to an increasing demand for Screen Content Coding (SCC), intra prediction methods that specifically consider the characteristics of contents from computer displays, web pages and presentation slides have been developed. In screen
content images and videos, repetitive image patches arise more frequently than natural sequences. This observation is considered in the Intra Block Copy (IBC) [7, 43], where the current block is predicted by translating another block in the encoded region of the same frame. The method is also known as intra motion compensation. Different from the Template Matching method, the motion vector is transmitted. The searching process is not performed at the decoder side. Variations of the IBC have also been studied. For example, in the work reported in [46], the current block to be encoded is predicted by the combination of translation and rotation of another block in the encoded region of the same frame. This method allows the efficient prediction of repetitive patterns with possible rotation in the same frame.

To illustrate the efficiency of the intra prediction, we show an example of intra prediction in Figure 2-7. We show the original frame, the intra predicted frame and the intra prediction residual frame in the figure. The intra predicted frame is generated with a fixed block size of 4x4. The 9 prediction modes defined in the H.264/AVC system are used. The residual frame is shown with a constant offset 128. Mid-gray pixels have zero amplitude. Negative values are shown with darker pixels and positive values are shown with brighter pixels. The same convention will be used throughout this thesis. We observe that the intra prediction is very effective for a typical frame. The residual energy is significantly smaller than the energy of the original frame. This implies a smaller amount of data to be encoded in the subsequent processes. We note that intra prediction does not depend on the information in any frame other than the current frame. Therefore, the intra prediction methods can be used for still image compression.

2.3.2 Inter prediction

The second type of prediction is the inter prediction. Frames temporally close to each other usually look similar. Much of the variation between temporally adjacent frames is caused by object motions. Inter prediction is used to exploit such temporal correlation between different frames. The prediction is generated from a different reference frame. The prediction residuals are encoded. Inter prediction works primarily
by exploiting object motions between frames. In the presence of motion, a block in the current frame can usually be predicted fairly accurately by translating another reference block in nearby frames. This is referred to as the motion-compensated prediction.

A common approach to find the best reference block is the block-matching method. In the block-matching method, suppose a current block is being encoded. This block is matched against a set of candidate blocks of the same size in another encoded reference frame. The candidate blocks may be chosen, for example, in a rectangular region of the reference frame centered at the location of the current block. We measure
the prediction error of all candidate blocks and choose the one with smallest error. Two types of information are encoded. First, to indicate the location offset of the best candidate block from the current block, a motion vector is encoded. Second, motion-compensated residual signals are encoded.

This process is sketched in Figure 2-8. We note that in many cases, the true motion vector may not align well with the sampling grid. In other words, motion vectors may have sub-pixel precision. To account for sub-pixel motion, frame interpolation and sub-pixel motion search are used. For example, half-pixel and quarter-pixel motion compensation is used in the H.264/AVC and HEVC systems. Though frame interpolation and sub-pixel motion estimation may increase the computational complexity, it is justified by a significant increase of coding performance.

To illustrate the efficiency of motion compensation, we show an example in Figure 2-9. In this example, we are interested in predicting the second frame from the first frame. The motion-compensated frame is generated by motion compensation with half-pixel accuracy. Motion compensation is able to capture object motions in most regions of a frame. This leads to a residual frame with much smaller energy relative...
to the original frame. As a result, motion compensation can significantly reduce the amount of data to be encoded.

![Frame 1 (Reference)](image1)

![Frame 2 (Current)](image2)

![Motion compensated frame](image3)

![Motion compensated residual](image4)

Figure 2-9: Motion compensated frame and residual frame: in this example, Frame 2 is predicted from Frame 1. For each block in Frame 2, we find the best reference block in Frame 1 that yields the smallest error. We form the prediction frame by displacing the reference blocks. The prediction error, i.e. the motion compensated residual frame is shown in the lower right figure.

Motion compensation depends on previously coded frames. Therefore, it cannot be used for encoding the first frame in a sequence. Motion compensation is not effective when scene change or other irregular frame changes occur. In these cases, the frame should be encoded without referencing other frames and intra prediction can be used instead. The switch between the intra prediction and inter prediction can be signaled at the frame level or the block level.
We note that the motion-compensated prediction is a primary case of inter prediction. The inter prediction, in general, can be used for the prediction between any two similar frames. For example, the VP compression system [4] defines a type of frame called the golden reference frame. The golden reference frame is encoded separately and it can be used as a reference frame for all other frames regardless of their temporal location in the sequence. Other use of inter prediction includes binocular prediction from a stereo pair of images and inter-layer prediction in scalable video coding.

2.4 Transform

While prediction reduces a significant amount of data redundancy, the prediction residuals may still contain much spatial redundancy inside a block. The spatial redundancy within each block is usually handled by transforms.

With a proper choice of the transform, image and residual signals can be represented efficiently in the transform domain. A large amount of energy can be preserved with a small number of large transform coefficients. To see a typical example of the energy compaction phenomenon, Figure 2-10 shows an image and its Fourier transform magnitude image. While the energy of a typical image spreads across the entire spatial domain, its transform representation is quite sparse. Most of the energy compacts into only a small number of large transform coefficients in the low frequency regions.

To quantify the energy compaction property of transforms, the energy compaction plot is used. In Figure 2-11, we show the energy compaction property of the Fourier transform for the image shown in Figure 2-10. We choose different numbers of largest transform coefficients and compute the preserved energy. We plot the preserved energy as a function of the total number of preserved coefficients. The preserved energy is in terms of the percentage relative to the total energy. The total number of chosen coefficients is presented in terms of the percentage relative to the total number of coefficients. From this figure, we can see that for the typical image shown in
Figure 2-10: An image and its Fourier transform magnitude image: the energy of the original image spreads across the entire space, while its energy compacts only in a small number of large Fourier transform coefficients.

Figure 2-11: Energy compaction plot: the energy compaction plot indicates the performance of the transform. In this example, more than 90% of the energy is preserved with 1% of largest Fourier transform coefficients for a typical image.

Figure 2-10, more than 90% of the energy can be preserved with 1% of transform coefficients. The energy compaction phenomenon allows much of the information to be encoded with only a small number of parameters in the transform domain. This
is the major reason why transform-based compression achieves great performance in practice.

In the above example, we show the energy compaction phenomenon of a natural image in the Fourier domain. Most of the energy in a typical natural image concentrates in its low frequency components. This observation, however, is not necessarily true for residual signals. In a compression system where the residual signal is transformed and encoded, obtaining the best transform with good energy compaction capability is a difficult task.

One remarkable work in the development of optimal transform for compression systems is the KLT. Given the covariance function of the signal, the KLT is able to perfectly remove the linear correlation within that signal. In addition, it was shown that the KLT results in the optimal energy compaction performance on average. With the KLT, the expected preserved energy is the largest, among all linear transforms, when a fixed number of coefficients with largest expected magnitudes are used.

To obtain the KLT, the exact knowledge of the residual covariance function is required. In practice, the exact covariance function is usually hard to obtain. Therefore, even though the KLT is of much theoretical interest, it is not used in practice. Many approximations of the KLT have been proposed. Previous research on the residual covariance function and transform design is reviewed in detail in Chapter 3.

2.5 Quantization

Transform coefficients are usually floating point numbers. The transform coefficients are quantized before they are encoded. In the quantization process, each float-point transform coefficient is mapped to an integer. It is known that the optimal scalar quantization method with the smallest Mean Squared Error (MSE) is the Lloyd quantizer [28]. The Lloyd quantization requires the knowledge of the probability distribution of each transform coefficient. In practice, simpler quantization methods are often used. For example, the uniform scalar quantization is used in many practical systems, where a transform coefficient is divided by a fixed quantization step and
rounded to the nearest integers.

We note that in transform-based compression systems, the quantization is a primary step that controls the trade-off between the bit rate and the video quality. It is often interrelated with the energy compaction property of transforms. To see this point, consider choosing a large quantization step. The quantized coefficients with a large quantization step tend to have a smaller possible number of reconstruction levels. Most small coefficients will be quantized to zero. Therefore, a smaller number of bits is necessary to encode these coefficients. By using a large quantization step, however, the quantization error becomes larger. This results in a large distortion.

2.6 Entropy coding

Quantized transform coefficients are converted to bit streams using the entropy coder. The entropy coder is responsible for reducing the statistical redundancy present in transform coefficients and side information. At a conceptual level, the entropy coder exploits the statistics of the input symbols and assign a codeword to each symbol. To reduce the average length of the bit stream, a shorter codeword is assigned to the symbol that is more likely to appear and a longer codeword is assigned to the symbol that is less likely to appear.

In theory, the optimal codewords that can be uniquely decodable and have the shortest average length are the Huffman [18] codewords. The optimality of Huffman codewords relies on two conditions. First, the knowledge of the probability of each possible input symbol is precisely given. Second, multiple independent symbols can be combined and jointly encoded. In practice though, these conditions are hardly satisfied. For the first condition, the probability of a specific block of quantized transform coefficients is hard to estimate. This is because the number of individual symbols may be very large. Consider a simplified case where we would like to encode a 4x4 block, with integer coefficients ranging from -3 to 3. The total number of possible input symbols is as large as $7^{16}$. In this simple example, the number of items in the Huffman code book will be prohibitively large. For the second condition, combining
multiple symbols increases the size of the code book exponentially for a similar reason. This also prohibits a large number of symbols to be jointly encoded with Huffman codewords.

In practice, many empirical observations are used to simplify the design of the entropy coder. In a typical block, for example, due to the energy compaction property, many quantized coefficients are likely to be zero. In addition, large coefficients tend to cluster in a specific region of the coefficient block. These observations are carefully investigated for the design of an entropy coder. For example, one practical implementation of the Huffman coding is the context-adaptive variable-length coding (CAVLC) used in the H.264/AVC system. In this method, the zig-zag running is used to capture the notion that transform coefficients tend to cluster around low-frequency regions of the DCT. Run-length encoding is used to efficiently encode a large chunk of zero coefficients. Context-adaptive codewords are used to account for the dependency between coefficients in different locations and different blocks.

Arithmetic coding [33] is another entropy coding method practically used. In arithmetic coding, multiple symbols are encoded jointly and sequentially. Arithmetic coding conceptually works as follows: for each symbol to be encoded, an interval from 0 to 1 is partitioned into multiple sub-intervals according to the probability mass function of that symbol. The sub-interval corresponding to the actual value of the symbol is chosen. This process continues recursively on the sub-intervals, until all symbols are processed. A number that falls into the final sub-interval is chosen and encoded. The decoding process repeats the interval partition process. It can be shown that under some conditions, arithmetic coding results in the shortest joint average code length. One of the advantages of arithmetic coding is that it is relatively straightforward to combine multiple symbols and jointly encode these symbols. In addition, it can adapt to the changes in the statistics of symbols during the process of encoding. It has been extensively used in many state-of-art image and video compression systems. For example, the context-adaptive binary arithmetic coding (CABAC) is used in the H.264/AVC system and it is the only entropy coder available in the HEVC system [30, 36]. In CABAC, each symbol to be entropy encoded is first
converted into a vector of bits. Each bit is encoded with the binary arithmetic coding engine. The probability distribution of each bit is carefully designed according to the type of the symbol and the context related to that symbol.

2.7 Decoder

The bit stream produced from the encoder side is transmitted or stored. When the decoder receives the bit stream, it inverts each of the operations in the encoder and the reconstructed video is generated. The compressed bit stream is first received and entropy decoded. Then inverse quantization and inverse transform are applied to generate the reconstructed residual signal. The prediction is added back to the reconstructed residual signal to form the reconstructed video.

We note that the synchronization at the encoder and the decoder has to be considered in the design of a compression system. Specifically, when processing the current block, the decoder only has access to previously encoded information. Therefore, the encoder has to use the same encoded information when generating the prediction signal. This implies that a decoder is usually built into the encoder to simulate the information known at the decoder side. Therefore, a practical compression system usually contains complicated structures, such as feedback loops. A high-level diagram of the feedback structure in video compression systems is illustrated in Figure 2-12.
Figure 2-12: Illustration of synchronization in a compression system: in order to use reconstructed data for prediction, a decoder is included in the encoder side. This ensures the reconstructed data seen by the encoder and the decoder are always identical.
Chapter 3

Previous Research

In this chapter, we review previous research on transforms for image and video compression applications. In Section 3.1, we discuss the optimal linear transform, KLT for image and video compression. Based on the principles of KLT, we discuss two commonly used approaches that approximate KLT in practice. In Section 3.2, we review the covariance modeling approach. In Section 3.3, we review the covariance estimation approach. Finally, we summarize previous research and discuss the motivation of this thesis in Section 3.4.

3.1 Karhunen Loève transform

In image and video compression, transforms are used to reduce the local spatial correlation within a block. The correlation is usually characterized by the covariance function of a random process. We consider a block of pixels whose intensities are represented as a zero-mean real random process $x(n)$. The covariance function of this zero-mean random process is given by $\gamma_x(n, m) = E[x(n)x(m)]$. The covariance function of a random process quantifies the amount of linear correlation between two specific random variables in the process. For example, a zero covariance between two pixels indicates that they are not linearly correlated.

In coding applications, we are concerned with encoding the same amount of information with fewer parameters. A representation of signals with fewer components
is usually desired. In the development of KLT, it was shown that given the covariance function of a random process, there exists a linear orthonormal transform that completely removes the linear dependency in the random process. To be specific, for the random process \( x(n) \), there exists a linear orthonormal transform \( T \), such that the covariance function \( \gamma_y(m, n) \) for the transform coefficients \( y(n) = T\{x(k)\} \) is zero whenever \( n \neq m \). The KLT basis functions \( V(n) \) can be obtained by solving the following equation:

\[
\lambda V(n) = \sum_m \gamma_x(m, n)V(m)
\]  

(3.1)

Solving the equation is equivalent to the eigen-decomposition of the covariance matrix.

As a result of perfect decorrelation, the KLT becomes a very efficient representation of the signal. Applying the KLT leads to the optimal energy compaction performance. Specifically, it can be shown that the expected preserved energy by keeping a fixed number of largest KLT coefficients is the largest among all linear orthonormal transforms. In other words, the KLT achieves the optimal energy compaction performance on average.

While the KLT achieves the optimal performance in theory, it suffers from a few difficulties in realistic image and video compression applications. First, obtaining the KLT requires the precise knowledge of the covariance function. It is well known that characteristics of images or prediction residual signals are very complicated. The covariance function may change from one image to another. In many cases, the covariance function may change from one region to another within an image. The underlying covariance function may be hard to obtain in practice. Second, estimating the covariance function usually requires collecting a lot of data. This process may be computationally expensive. In addition, both the decoder and the encoder have to agree on the covariance function or the transform basis functions when using the KLT. Possible transmission of the side information may be expensive as well.

As a result, much effort has been spent on developing transforms based on approx-
imations of KLT based on reasonable assumptions and observations. The methods can be classified broadly into two classes. The first class is referred to as the covariance modeling approach. In this approach, a model is used for the underlying signal. The covariance function of the model is derived. The transform is designed based on the covariance function of the model. The second class is referred to as the covariance estimation approach. In this approach, the covariance is empirically estimated from a carefully chosen subset of available data.

3.2 Transforms based on covariance modeling

3.2.1 Discrete cosine transform

One fundamental breakthrough in transform-based image compression is the derivation of the discrete cosine transform (DCT). The DCT has been extensively used in many transform-based image and video compression systems [4, 36, 40, 42].

The DCT is based on the first-order Markov model [16]. Consider the one-dimensional case. In the first-order Markov model, a one-dimensional signal $x(n)$ is modeled as

$$x(n) = \rho x(n-1) + \sqrt{1-\rho^2} w(n)$$ (3.2)

where $x(n)$ is a zero-mean random variable with unit variance, $w(n)$ is a zero-mean white noise with unit variance and $\rho$ is the correlation parameter that satisfies $0 < \rho < 1$. This model results in a covariance function $\gamma_x(n_1, n_2) = \rho^{|n_1 - n_2|}$. The KLT for the covariance function is obtained in a closed form [16]. In particular, it was shown that the KLT basis functions approach the DCT when $\rho \to 1$.

The first-order Markov model can be extended to a two-dimensional separable model for images. A two-dimensional signal $x(n_1, n_2)$ is modeled as
\[ x(m,n) = \rho_1 x(m-1,n) + \rho_2 x(m,n-1) - \rho_1 \rho_2 x(m-1,n-1) + \sqrt{1 - \rho_1^2} \sqrt{1 - \rho_2^2} w(m,n) \]  
(3.3)

The covariance function of this 2D model, for pixels located at \((m_1, n_1)\) and \((m_2, n_2)\) is
\[ \gamma_x(m_1, n_1, m_2, n_2) = \rho_1^{\left| m_1 - m_2 \right|} \rho_2^{\left| n_1 - n_2 \right|} \].

The 2D-DCT can be derived when \(\rho_1 \to 1\) and \(\rho_2 \to 1\).

Figure 3-1 shows 2D-DCT basis functions of size 4x4. The patterns in the transform basis functions can effectively represent many common structures in typical image blocks. For example, the DC (left bottom) basis function captures the smooth component of an image block. Other complicated textures can be captured by patterns with higher frequencies.

Figure 3-1: 4x4 2D-DCT basis functions: a total of 16 DCT basis functions. Each subfigure shows a transform basis function of size 4x4.

Under the assumption that \(\rho\) is close to one, the first-order Markov model captures a few important characteristics of typical images. First, for typical still image blocks, intensities of nearby pixels tend to be similar. The model implies that pixels nearby are highly correlated. The correlation decays as the distance of pixels increases. Sec-
ond, the Markov model is stationary. The statistics of images are uniform throughout the space. In particular, the variance of each pixel under the model is constant. This implies that each pixel is of equal significance under the first-order Markov model.

In video coding applications, it has been noted that prediction residuals display significantly different characteristics from those of still images. Non-stationarity often arises in residual signals. In this case, the 2D-DCT may no longer be optimal for prediction residuals. Successful systems based on transforms other than the 2D-DCT have been developed for prediction residuals.

### 3.2.2 Asymmetric discrete sine transform

The DCT has been exclusively used in many major image and video compression standards until the recent development of the asymmetrical discrete sine transform (ADST). In the work reported in [17], it is observed that the KLT of the directional intra prediction residuals is close to the ADST. The derivation of the ADST is based on the observation that pixels near the block boundary can be predicted more accurately than pixels far away. This results in a residual signal whose intensities tend to increase as the distance from the boundary increases. The paper proposed a first-order Markov model with a deterministic boundary to characterize this observation. The residual signal \( r(n) \) is modeled as

\[
    r(n) = \begin{cases} 
        \rho r(n - 1) + \sqrt{1 - \rho^2} w(n) & n > 1 \\
        \sqrt{1 - \rho^2} w(1) & n = 1 
    \end{cases}
\]  

(3.4)

It is interesting to study the variance function of this model. In this model, one can find that the variance function of the process is very small near the boundary, due to the deterministic and known boundary predictor. The variance of the process increases when the random noise term propagates along the direction of the prediction direction. The increasing variance function reflects the empirical observation that the residual magnitudes tend to increase in the direction away from the boundary.

The covariance function for the residual \( r(n) \) is derived. It was shown that under
certain reasonable approximations, the KLT for this random process is the ADST. The ADST shows exceptional performance over the DCT for directional intra prediction residuals. A bit rate reduction of a few percent has been reported in [17] when the ADST is used in the H.264/AVC system. A bit rate reduction of around one percent is reported in [34] when the ADST is used in the HEVC system. Due to its superior performance over the DCT, the ADST is used for HEVC intra coding. Specifically, the ADST is used to encode luminance residual blocks of size 4x4 [36]. The DCT is used otherwise.

Figure 3-2 shows ADST basis functions of size 4x4. The transform basis functions, especially those low frequency ones, are significantly different from those of the DCT. For example, the left bottom basis function shows an increasing pattern from the left and bottom boundaries. As the model indicates, this function captures the notion that intensities of residual signals tend to increase in the direction away from the boundary. We note that in a practical compression system, the coordinate system is usually vertically flipped. Therefore, the first ADST basis function shows an increasing pattern from the top and left boundaries.

![Figure 3-2: 4x4 ADST basis functions: a total of 16 ADST functions. Each subfigure shows a transform basis function of size 4x4.](image-url)
We also note that the DCT and the ADST basis function are usually floating-point numbers. A naive implementation of the DCT and the ADST requires matrix multiplication under the floating-point arithmetic. This is usually not very efficient, especially for performance-critical applications and hardware design. Significant amount of effort has been spent on designing transforms that approximate the DCT and the ADST with efficient implementation. In the H.264/AVC compression system, the DCT is approximated with an integer transform [29]. The transform is decomposed into two steps. The first step only uses integer shift and addition as an approximation of the matrix multiplication. The second step scales the transform coefficients from the first step as a normalization process. Note that the second step is jointly designed with the scaling in the quantization step. Look-up tables are used for the combined scaling computation. It has been noted that this particular design of the integer transform significantly improves the computational efficiency of the DCT while not introducing noticeable penalty in rate-distortion performance. In the HEVC compression system, the transform size varies significantly from 4x4 to 32x32. To design a transform implementation that can unify all transform sizes, a 32x32 core integer transform matrix has been proposed [8]. The transform basis function coefficients of other sizes can be obtained by sampling the 32x32 transform matrix. This particular transform design also allows hardware-friendly and parallel-friendly implementation schemes such as the butterfly structure.

3.2.3 Transforms for motion-compensated residuals

Models for motion compensated residuals have been investigated. In the work reported in [20, 22, 45], it is observed that many one-dimensional anisotropic structures arise in motion-compensated residuals. By modeling the motion-compensated residuals with one-dimensional first-order Markov models along a certain direction, the following covariance function is proposed for the motion-compensated residuals

\[
\gamma_g(\theta, I, J) = \rho_1^{\text{1} \text{cos}(\theta) + i \text{sin}(\theta)} \rho_2^{\text{1} i \text{sin}(\theta) + i \text{cos}(\theta)}
\]  

(3.5)
where $I, J$ are distances in the horizontal and vertical directions between two pixels of interest. To model the one-dimensional characteristics of the residual signal, $\rho_1$ is assumed to be close to zero and $\rho_2$ is assumed to be close to one. The parameter $\theta$ is the direction parameter that determines a particular direction of the strong correlation.

To see the anisotropic directionality of this model, consider the case where $\theta = 0$. In this case, the model degenerates to a separable model:

$$
\gamma_g(0, I, J) = \rho_1^{|I|} \rho_2^{|J|}
$$

Under the assumption that $\rho_1$ is close to zero and $\rho_2$ is close to one, the correlation along the $I$ direction can be ignored and the correlation along the $J$ direction is considered to be strong. Consider another case where $\theta = \pi / 2$. In this case, the model degenerates to a separable model:

$$
\gamma_g(\pi / 2, I, J) = \rho_1^{|J|} \rho_2^{|I|}
$$

Notice that $I$ and $J$ switch in this case. It models the strong correlation along the $I$ direction and ignores the correlation along the $J$ direction. Similar argument can be extended to an arbitrary $\theta$ parameter if we consider a change of variable in the exponents with a rotation matrix parameterized by $\theta$.

Based on this model, a set of one-dimensional transforms of many directions are derived for encoding motion compensated residuals. In Figure 3-3, we illustrate these one-dimensional transforms for 4x4 blocks. For each line in the block, a one-dimensional transform of possibly different length is applied. The arrow indicates the pixels each one-dimensional transform is applied to. An average bit rate reduction of over 10 percent was reported when directional 1D transforms are used in addition to the 2D-DCT on the H.264/AVC system. Similar ideas have been applied to lifting wavelet transforms [21].

In the work reported in [14, 19, 31], motion compensated residual signals are modeled with a stationary random process. As opposed to the first-order Markov model, different covariance functions are proposed to account for the notion that the
motion-compensated residual is less correlated than still images. Separable transforms were suggested in these models.

### 3.3 Transforms based on covariance estimation

In the covariance estimation approach, the residual covariance function is approximated by the empirical covariance function of a collection of data. The estimation of the empirical covariance function can be an offline process or on-the-fly during the encoding and decoding processes.

#### 3.3.1 Offline covariance estimation

In the methods based on offline covariance estimation, the covariance function is computed by analyzing a set of typical video sequences in an offline process. The estimated covariance function is used to compute an empirical KLT. Since the transform is computed offline, it does not change throughout the encoding and decoding processes. A variety of transforms based on this approach have been proposed.

In the work reported in [44], it is observed that the patterns of directional intra
prediction residuals depend on the prediction direction. A set of mode-dependent transforms are proposed based on this observation. To be specific, the empirical covariance function of each directional intra prediction mode is estimated from many video signals. The covariance function is used to compute the KLT for each intra prediction mode.

In the work reported in [47], it is observed that multiple mode-dependent patterns arise for each prediction mode. As a result, multiple mode-dependent transforms are proposed for each intra prediction mode. To group the residual signals that share similar characteristics, a method based on the KSVD [2] is used on many video sequences. The proposed method results in more than one covariance function for each mode. Multiple KLTs are proposed for each mode. The same methodology has been applied to motion-compensated residuals in [47].

### 3.3.2 On-the-fly covariance estimation

In the methods based on the online covariance estimation, the covariance function is estimated during the encoding and decoding processes from encoded video data. The KLT from the estimated covariance is obtained on-the-fly and applied to the current block. The estimation process may choose to use different portions of encoded information for better adaptivity. As a result, transforms based on online covariance estimation are usually adaptive. We note that the same piece of coded information is known to both the encoder and the decoder. In addition, the covariance estimation and the KLT computation rules are synchronized at the encoder and the decoder. As a result, the transmission of the transform basis functions is usually not necessary.

As we have discussed in Section 2.3, the template matching method is used in intra prediction. The template matching method indicates that the best prediction for the current block can be obtained from other blocks with a similar template. The best prediction is subtracted from the current block and the residuals are encoded. The subtraction of the prediction from the current block can be regarded as removing the template-dependent estimated mean of the current block. The template-dependent mean can be generalized to other statistics. In the work reported in [26], it is observed
that the covariance functions of intra residual blocks depend on the template. The covariance function is estimated from similar patches with a matching template in encoded video data. Specifically, an empirical covariance function for the current block is estimated by weighted combination of different patches in the encoded region with similar template. The weight is determined by the similarity of the encoded template. The KLT is obtained from the covariance function for the current block.

In the work reported in [39], the second-order statistics of the residual signal are investigated. It is observed that a strong correlation exists between the residual frame and the gradient information in the reference frame. To encode the current residual block, the KLT is obtained from a covariance function

$$\gamma(m_1, n_1, m_2, n_2) = \sigma(m_1, n_1)\sigma(m_2, n_2)\rho_1^{m_1-m_2}\rho_2^{n_1-n_2}$$

(3.8)

for pixels located at \((m_1, n_1)\) and \((m_2, n_2)\). In the model, the pixel-wise variance function \(\sigma(m, n)\) is obtained by applying a non-linear function on the gradient magnitude in the reference frame on a pixel-by-pixel basis. The non-linear function is adaptively estimated on the fly during the encoding process. The non-linear function is transmitted to the decoder side. The optimal \(\rho\) is also estimated and transmitted.

### 3.4 Motivation of thesis

The thesis is motivated as follows:

- It is well known that prediction residuals contain a significant amount of non-stationarity, as opposed to regular image blocks. The method proposed in this thesis attempts to characterize the residual non-stationarity from a new perspective.

- As discussed in earlier sections of this chapter, many other transforms have been developed to handle residual non-stationarity in previous research. Many of previously proposed methods are based on empirical characteristics of residual
data. Very few study the fundamental reasons that lead to the residual non-stationarity. This thesis, along with a few published papers [11–13], is among the first attempts that study residual non-stationarity from a more fundamental point of view. Specifically, the thesis tries to address the question: What leads to the non-stationarity in the prediction residual signals?

- In image and video compression systems, adaptive transforms are used. Depending on the design of the system, the adaptivity of transforms varies. For example, in the systems that exclusively use the DCT [40, 42], the transform is not adaptive. In the systems that use multiple transforms [20, 34, 36, 44], the transforms adapt to a specific mode. In the systems that use fully adaptive transforms [26, 39], the transforms may change from a single block to another. Using more adaptive transforms usually leads to better performance, assuming that the transforms are designed or estimated in a robust way. However, a robust estimation of fully adaptive transforms is a difficult task. This thesis proposes a model that allows relatively robust estimation of transforms from a few data points. This point will be revisited in Chapter 9 after we develop the proposed method in the following chapters.

For the rest of the thesis, a new class of transforms based on the prediction inaccuracy model will be proposed and discussed, based on the above motivations.
Chapter 4

Prediction Inaccuracy Modeling

In this chapter, we discuss the high-level idea of the prediction inaccuracy modeling. In Section 4.1, we study the characteristics of typical residual signals. Specifically, we observe that much non-stationarity arises in typical residual signals. In Section 4.2, we observe that the prediction tends to be less accurate in the regions where strong spatial changes occur. This relates the residual magnitude to the gradient magnitude. In Section 4.3, we explain the phenomenon observed in Section 4.2 as a result of prediction parameter inaccuracy. Based on the prediction inaccuracy modeling, we propose the high-level framework of designing transforms from the prediction inaccuracy models in Section 4.4.

4.1 Residual non-stationarity

The characteristics of typical prediction residual signals are quite different from typical images. Figure 4-1 shows two pairs of typical images and prediction residuals. The first row shows the first frame in a video sequence and its intra prediction residual frame. The intra prediction residual frame is obtained using 9 intra prediction residual modes defined in the H.264/AVC system with a fixed block size of 4x4. The second row shows a video reference frame and motion-compensated residual frame from the reference frame. The motion-compensated residual frame is obtained with half-pixel motion-compensation on 4x4 blocks.
We first observe that most regions in prediction residual frames are very close to zero. This is due to a strong spatial and temporal correlation present in typical video signals. The prediction is very effective in removing such correlation. The prediction, however, may be inaccurate in a small portion of the frame. In these regions where the prediction is not accurate, the prediction residuals may be significant.

More importantly, we observe that the large residuals display a significant amount of non-stationarity. First, the large residuals are highly localized and the energy does not spread uniformly in the entire space. This phenomenon, particularly the one-dimensional pattern, is also observed in [20]. Second, within a small local spatial region around those significant residuals, the residual magnitude changes abruptly.

Figure 4-1: An example of typical frames and prediction residual frames
The residual non-stationarity is mostly reflected by a significant change of magnitude in a local region. These observations are very different from many existing image and residual models, such as the ones in the development of the DCT and the ADST. In those models, the signal is modeled as locally stationary or smoothly changing instead of highly non-stationary. This motivates the study of the underlying reason that leads to the residual non-stationarity and a better model that characterizes the residual non-stationarity.

4.2 Residual-gradient correlation

To see why non-stationarity arises in a typical residual signal, we study the reason that leads to the drastic local change of residual magnitude. Specifically, we are interested in which pixels of the residual frame tend to be more significant than others. Figure 4-2 shows a typical video frame, the residual magnitude and the two-dimensional spatial gradient magnitude of the original video frame. In the figure, we observe that the residual magnitude frame looks very similar to the gradient frame. Specifically, for the pixel where the gradient is large, the residual pixel tends to be significant at the same location as well. In other words, the residual magnitude is correlated with the gradient magnitude on a pixel-by-pixel basis.

To verify this point, we use the relative energy compaction property. Figure 4-3 shows three cumulative energy curves of the residual frame in Figure 4-2. We choose a number of residual pixels and compute the preserved energy from these pixels. We choose the pixels according to different criteria for each of three curves.

In the first curve referred to as the optimal energy curve, we choose the pixels with the largest residual magnitude. The first curve is optimal in the sense that the preserved energy is largest among all different ways of choosing a fixed number pixels. In the second curve referred to as the randomized energy curve, we randomly choose the number of pixels and compute the cumulative energy. In the third curve referred to as the gradient relative energy curve, we rank order the gradient magnitudes. Then, we choose the pixels at the same location of pixels with the largest gradient.
In other words, the gradient relative energy curve computes the cumulative energy of pixels with reference to the rank order information in the gradient frame.

The relative energy compaction plot indicates the strength of the correlation between the residual magnitude and the gradient magnitude. Suppose the gradient is perfectly correlated with the residual magnitude in rank order, choosing the pixels with largest gradient is equivalent to choosing the pixels with largest residual magnitude. In this case, the gradient relative energy curve will be identical to the optimal energy curve. Suppose the gradient does not carry any rank order information of the residual magnitude, choosing the pixels with largest gradient is equivalent to choosing a random set of residual pixels. In this case, the gradient relative energy curve will
be very close to the randomized energy curve. A relative energy curve closer to the optimal curve indicates a stronger correlation between the residual magnitude and gradient magnitude. In Figure 4-3, we can see that the relative energy curve for a typical intra predicted residual frame is very close to the optimal one. This indicates a strong correlation between the residual magnitude and the gradient magnitude on a pixel-by-pixel basis.

![Graph showing correlation between residual frame and gradient frame](image)

Figure 4-3: An example of the correlation between residual frame and gradient frame

The above observation can be summarized as the following: the prediction tends to be less accurate in the region where a significant spatial change occurs. While this statement is correct, it is not the end of the story. First, it does not completely reveal the underlying reason that leads to this phenomenon. Second, it is not useful enough in a practical video coding system. By studying the residual non-stationarity, we would like to obtain information of which residual pixels are more significant than others. We hope to use this information in a practical compression system by encoding more significant pixels with higher priority. However, the gradient information of the current frame is not available before the frame itself has been encoded. In other words, it is generally not possible to use the gradient information in the current block in a practical video compression system.

In the rest of the thesis, we propose the prediction inaccuracy model as a more fundamental interpretation of the above conclusion. In addition, the prediction inac-
curacy model relates the residual magnitude to the gradient information of already coded information. It can be useful in a practical video compression system.

We also note that the relative energy compaction property essentially studies the rank order statistics. Rank order statistics are sufficient for the purpose of roughly estimating the significance of residual pixel based on the gradient information. To exploit the correlation between the gradient information and residual signals and use that in a more practical setting, a functional relation is necessary. In later chapters of this thesis, we will establish a model and derive such functional relations.

4.3 Prediction inaccuracy model

In this section, we study the underlying reason why prediction tends to be less accurate where the spatial change is large. Consider an example of directional intra prediction shown in Figure 4-4 and Figure 4-5. In both figures, the current block of interest is outlined with a red box. The encoded region is shadowed. In Figure 4-4, the prediction is applied to a smooth region, while in Figure 4-5, the prediction is applied to an edge region.

In directional intra prediction, we would like to find a prediction direction that minimizes the error between the original block and the prediction. For the block in Figure 4-4, we consider predicting the current block with two different directions: the vertical direction and a diagonal down direction. They are illustrated on the right side of the figure. The region under consideration is nearly constant. As a result, both prediction directions lead to good predictions. In fact, many other directions lead to small prediction error. In other words, the prediction accuracy is not quite sensitive to the exact choice of the prediction direction.

On the other hand, we repeat the same analysis on the block shown in Figure 4-5. We compare two prediction directions on the right hand side of the figure. Since the region contains a strong edge, only certain choices of the prediction directions lead to a reasonable prediction. In this case, only the particular direction on the top right of the figure leads to a prediction that matches the edge direction in the block. Other
Figure 4-4: An example of prediction where the gradient is small: either the vertical prediction or the diagonal down prediction yields an accurate prediction.

Figure 4-5: An example of prediction where the gradient is large: only a particular prediction direction yields the smallest prediction error. The prediction accuracy is sensitive to the prediction direction.
prediction directions lead to a large prediction error due to the mismatch between the prediction direction and the edge direction. In this case, the correct prediction heavily relies on an exact choice of the prediction direction. A different prediction direction off by even a small amount leads to a large prediction error. In other words, the prediction accuracy in this case is very sensitive to the prediction direction.

In this example, we observe that the sensitivity of the prediction accuracy to the prediction direction depends on the local spatial changes. In a smooth region where pixels share similar intensities, the prediction accuracy is less sensitive to the prediction direction. On the other hand, in the regions where sharp discontinuities exist in the original frame, the prediction is very sensitive to the accuracy of the prediction direction. A small disturbance of the prediction direction away from the correct direction may lead to a large prediction error. In other words, the same amount of prediction parameter inaccuracy leads to a larger prediction error in the case when the local spatial change is larger. This observation relates the residual magnitude with the spatial gradient, as expected from Section 4.2.

In most video compression systems, such prediction parameter inaccuracy universally exists. First, in designing a video compression system, a finite number of prediction methods and prediction parameters are predefined. An arbitrarily accurate prediction requires a very large number of prediction parameters. In the actual prediction process, the prediction is chosen from this finite set. This results in the prediction parameter inaccuracy. This inaccuracy may include, for example, a finite number of intra prediction directions and limited motion vector precision in motion-compensated prediction. Second, the prediction parameter is usually chosen and fixed on the block level. Choosing the best prediction parameter at the pixel level is usually prohibitive due to a large pixel-wise overhead. As a result, the optimal prediction parameter for each block does not necessarily ensure the optimal prediction parameter for each individual pixel. The optimal block level prediction parameter may be a result of a compromise of correct prediction parameters for the pixels within that block. For each individual pixel, the prediction inaccuracy still exists.

The residual magnitude can be related with the spatial gradient only through
encoded information in the view of prediction parameter inaccuracy. In the example above, the gradient of the upper boundary can be used as an indicator of the amount of local spatial change along the prediction direction. The spatial gradient information of the current block can be inferred from the encoded upper boundary. As a result, the proposed method is more practical than the empirical observation in Section 4.2. This point will be discussed in detail in the following chapters.

In this thesis, we refer to the idea of relating the residual non-stationarity to the prediction parameter inaccuracy as the prediction inaccuracy modeling. In the following chapters, we make use of the concept of prediction inaccuracy modeling to study residual statistics. We will derive the prediction inaccuracy models in a more rigorous formulation. Specifically, we characterize the residual non-stationarity with the variance and covariance function of the residual process.

4.4 KLT based on the prediction inaccuracy model

The prediction inaccuracy model interprets the residual non-stationarity. As a result of better estimation of the residual statistics, it is useful in designing optimal transforms for residual encoding.

To briefly see how we can use the information from the prediction inaccuracy model to design transforms, we consider encoding a residual block. The proposed model can be used to estimate which residual pixels are more likely to be significant. Suppose from the prediction inaccuracy model, we determine that only a small portion of the block contains significant residual energy. Then, the optimal transform can be designed in such way that it is supported only in the region with significant residuals. The other regions can be ignored without losing much energy. This will significantly reduce the number of significant transform coefficients, compared to a complete transform representation such as the DCT. As we can see later in the thesis, this idea can be generalized to assigning different priority to each pixel in the form of the variance function from the prediction inaccuracy models. The variance function can be used to construct a covariance function, whose KLT is a better representation
of non-stationary residual blocks.

In the following chapters of this thesis, we derive the prediction inaccuracy model and optimal transforms for two types of residual signals in detail. The first type is the directional intra prediction residual signal. The second type is the motion-compensated prediction residual signal. We will see that with careful formulation, the non-stationarity of the residual signal can be represented as a function of the gradient function in the encoded region.

For the rest of thesis, we will follow general principles outlined as follows:

1. Express the residual signal as the result of the prediction parameter inaccuracy.
2. Establish a statistical model for the prediction parameter inaccuracy.
3. Derive the statistics of the residual signal from the prediction parameter inaccuracy statistics
4. Derive the optimal transform from the residual statistics

More specifically, we will follow the steps outlined in Figure 4-6. We first collect information in the region that have already been encoded prior to the current block. Specifically, we use the reconstructed pixel values in these regions. With the reconstructed pixel values along with a model that we will derive later, we estimate pixel-wise variance function for the current residual block, with the help of prediction inaccuracy model. Intuitively, the estimated variance function indicates our estimation of the significance of each residual pixel. The pixels with greater variance should be considered with higher priority when applying the transform. From the variance function, we construct a covariance function for the current block. This model covariance function is used to derive the KLT for the current block. The KLT is applied to the current block.
Figure 4-6: The flowchart of the prediction inaccuracy model: general principle
Chapter 5

Directional intra prediction inaccuracy model

In this chapter, we derive the prediction inaccuracy model for directional intra prediction residuals. In Section 5.1, we consider the characteristics of typical directional intra prediction residual signals. We focus on the non-stationarity of residual signals. In Section 5.2, we derive the prediction inaccuracy model by relating the residual signal with prediction direction inaccuracy. For simplicity, we first derive the model when the horizontal prediction is used. The model is then extended to an arbitrary prediction direction. In Section 5.3, we use the model derived in Section 5.2 to study the residual statistics. Specifically, we derive the residual mean, variance and covariance function, based on the statistics of prediction direction inaccuracy. In Section 5.4, we discuss the KLT based on the proposed model covariance function.

5.1 Characteristics of directional intra prediction residuals

The characteristics of intra prediction residuals are significantly different from those of still images. Figure 5-1 shows an example of a still frame and its intra prediction residual frame. For a typical still image, we observe that image intensities tend to
be smooth in most regions of the image. For the intra prediction residual frame, we observe that most regions are close to zero, as a consequence of the effective intra prediction in smooth regions. In the regions where sharp edges and busy textures arise, intra prediction becomes less effective, and the residuals become much larger in these regions.

![An intra frame](image1.png) ![Intra prediction residual frame](image2.png)

Figure 5-1: An example of the intra frame and residual frame

To carefully investigate the characteristics of directional intra prediction residuals on a block-by-block basis, we show a 4x4 block of intra prediction residual in Figure 5-2. In this 4x4 block, the vertical prediction is used. We note that intensities of the directional prediction residuals tend to increase along the prediction direction, as the distance from the boundary increases from top to bottom. This observation is typical in many video sequences. It has been investigated in previous research, such as the work reported in [17]. In addition, we note that the residual signal along the direction orthogonal to the prediction direction displays significantly different characteristics. Specifically, the residual intensities change abruptly along the direction orthogonal to the prediction direction, as shown in Figure 5-2. This observation indicates that the residual signal may be highly non-stationary in the direction orthogonal to the prediction direction. The characteristics of the prediction residuals are very sensitive, not only to the prediction direction, but to the local change of the image data as well. In other words, the characteristics of the prediction residuals should not only be mode-dependent, but also data-dependent.
With the principle in Chapter 4 applied to the intra prediction, the non-stationarity of residual signals can be interpreted by the prediction inaccuracy. In those regions where there are sharp discontinuities in the original frame, the prediction tends to be less accurate. Therefore, the residual intensities tend to be large relative to smooth regions. We wish to use this observation to predict the statistics of the residuals. This observation will be useful when we can relate the local change of image data to coded data.

To estimate the residual statistics only from the coded data, we consider the process of directional intra prediction. Specifically, we consider the sensitivity of prediction to the accuracy of the prediction direction parameter. In a smooth region where pixels share similar intensities, the prediction accuracy is less sensitive to the prediction direction. On the other hand, in the regions where sharp discontinuities exist in the original frame, the prediction is very sensitive to the accuracy of the prediction direction. A small disturbance of the prediction direction away from the actual direction may lead to a large prediction error. This observation leads to a model that estimates the residual covariance only from the coded boundary. We discuss the model in detail in the next section.
5.2 Model derivation

5.2.1 Model for horizontal prediction

In this section, we derive the prediction inaccuracy model for the horizontal intra prediction. The derivation of the prediction inaccuracy model for other directions follows in Section 5.2.2.

In the horizontal prediction, the encoded left boundary of the current block is used as the predictor. The pixels from the left boundary are copied along the horizontal direction right to the current block. If the horizontal direction is the optimal prediction direction, one would expect the residual to be close to zero. However, if the optimal prediction direction is slightly different from the horizontal prediction, the prediction will be inaccurate. The prediction error depends on how fast the left boundary changes near the predictor. It also depends on the distance from the boundary and the inaccuracy that occurs. Based on this intuition, we proceed with the detailed derivation.

We first establish the notation for the proposed model. We consider a rectangular block to be encoded, and we use the following notation:

- $f(m, n)$: current block to be encoded
- $\hat{f}(m, n)$: predicted block, obtained by copying the coded left boundary $f(0, n)$ along the horizontal direction.
- $r(m, n)$: residual block, obtained by subtracting $\hat{f}(m, n)$, the predicted block, from $f(m, n)$, the current block.

In the above notation, $m$ is the horizontal coordinate, $m = 0$ corresponds to the coded left boundary that is used for prediction, $n$ corresponds to the vertical coordinate and $m, n \geq 1$ is the area to be encoded. This is illustrated in Figure 5-3.

We first note that the residual is obtained by subtracting the prediction from the current block:

$$r(m, n) = f(m, n) - \hat{f}(m, n)$$ (5.1)
The prediction is obtained by horizontal prediction:

$$
\hat{f}(m, n) = f(0, n)
$$

(5.2)

To quantify prediction direction inaccuracy, we introduce a random variable $\theta(m, n)$ as a function of spatial coordinate. The pixel-wise $\theta$ function characterizes the prediction direction inaccuracy for each pixel. In other words, for each pixel located at position $(m, n)$, $\theta(m, n)$ represents the accurate prediction direction along which the optimal predictor lies in the coded boundary.

We will study the statistics of the $\theta$ variable in the following sections. At this moment, we assume that $\theta$ is small. This assumption is reasonable because different prediction directions in a video compression system are not significantly apart from each other. An accurate prediction direction can usually be found near the chosen prediction direction. Otherwise, it indicates that the directional intra prediction is not effective and will not be used as a prediction method in the first place.

Suppose we denote $n_a$ as the location of the accurate prediction in the coded
boundary. Ignoring the difference between the intensities of the current pixel and the perfect prediction, we obtain:

\[ f(m, n) \approx f(0, n_a) \]  \hspace{1cm} (5.3)

where

\[ n_a = n + m \tan(\theta(m, n)) \approx n + m\theta(m, n) \]  \hspace{1cm} (5.4)

for small \( \theta \). This can be seen from the geometry shown in Figure 5-3.

From equations (5.1), (5.2), (5.3) and (5.4), we obtain:

\[
\begin{align*}
    r(m, n) &= f(m, n) - \hat{f}(m, n) \\
             &\approx f(0, n_a) - f(0, n) \\
             &\approx (n_a - n) \frac{\partial f(0, n)}{\partial n} \\
             &\approx m\theta(m, n) \frac{\partial f(0, n)}{\partial n}
\end{align*}
\]  \hspace{1cm} (5.5)

for small \( \theta \) and therefore small \( n_a - n \).

Equation (5.5) indicates that the residual intensity is proportional to the distance \( m \) and to the boundary gradient. In addition, the residual intensity depends on how inaccurate the prediction direction is away from the actual direction, characterized by a random variable \( \theta \). The equation makes intuitive sense. First, when the current pixel gets away from the boundary, the spatial correlation decreases. As a result, the residual is likely to increase. Second, when the boundary gradient is large, the current pixel is likely to lie in the region where a strong spatial change occurs. In these regions, the residual signal is likely to be large. Third, when the prediction direction inaccuracy increases, the prediction error increases as well. Under all these circumstances, the residual is likely to be more significant.

### 5.2.2 Model for arbitrary direction

When an arbitrary prediction direction is used, the same intuition applies. However, there are two additional considerations. First, the upper boundary may be involved. Second, the derivation follows a slightly different geometry.
For the upper boundary, predicting from the upper boundary can be easily handled with symmetry from the left boundary by switching the horizontal coordinate and the vertical coordinate.

We discuss the different geometry involved in the arbitrary prediction direction. In Figure 5-4, we illustrate the geometry for the derivation of the model. We derive the model when the left boundary is used in prediction.

When an arbitrary prediction is used, the current pixel is predicted from a pixel with a different boundary coordinate. Therefore, \( n \) is replaced by \( n' \). In addition, the displacement from the accurate predictor to the predictor used is related to \( \theta \) in a different way. Consider the geometry shown in Figure 5-4. The arc length resulting from the inaccurate prediction direction is \( d \approx L\theta \). In this relation, \( L \) is the distance from the residual pixel to its boundary predictor. From the geometry shown in Figure 5-4, the displacement becomes \( n_a - n' \approx \frac{d}{\cos \alpha} \), where \( \alpha \) is the angle from the prediction direction to the norm of the boundary.

Combining these results, we obtain the following estimation by analogy to the
horizontal case:

\[ r(m, n) \approx \frac{L}{\cos \alpha} \left( \frac{\partial f(0, n)}{\partial n} \right)_{n'} \theta(m, n) \]  \hspace{1cm} (5.6)

Equation (5.6) indicates that the residual is proportional to the boundary gradient, evaluated at the position of the predictor. In addition, the residual is proportional to the distance from the current pixel to its boundary predictor scaled by a factor related with the prediction direction. We note that the general case is consistent with the horizontal case. When the horizontal prediction is used, \( \alpha = 0 \), \( L = m \) and Equation (5.6) reduces to Equation (5.5). As another example, when the diagonal prediction is used, \( \alpha = \frac{\pi}{4} \). Equation (5.6) will be used to derive the covariance function for the residual signal in Section 5.3.

5.2.3 Alternative derivation of the model

The above derivation based on geometry is simple and intuitive. Alternatively, one can derive the model in a more rigorous way through simple calculus.

From Figure 5-4, for the current pixel located at \((m, n)\), we obtain the following relations:

\[ m = L \cos \alpha \]  \hspace{1cm} (5.7)

\[ n' = n - m \tan \alpha \]  \hspace{1cm} (5.8)

\[ n_a = n - m \tan(\alpha - \theta) \]  \hspace{1cm} (5.9)

From Equations (5.8) and (5.9), we obtain:

\[ n_a - n' = m(\tan \alpha - \tan(\alpha - \theta)) \]  \hspace{1cm} (5.10)

When \( \theta \) is small, we use first-order approximation, and obtain the following:

\[ \tan \alpha - \tan(\alpha - \theta) \approx \theta \frac{d \tan \alpha}{d\alpha} = \frac{\theta}{\cos^2 \alpha} \]  \hspace{1cm} (5.11)
With Equations (5.7), (5.10) and (5.11), we obtain:

\[ n_a - n' = m (\tan \alpha - \tan (\alpha - \theta)) \approx \frac{m\theta}{\cos^2 \alpha} = \frac{L\theta}{\cos \alpha} \tag{5.12} \]

This result is the same as the one derived in Section 5.2.2.

## 5.3 Model statistics

In Equation (5.6), the residual signal is determined by two parts. The first part is a deterministic scalar related to the geometry of the prediction and the current block. The second part \( \theta(m, n) \) is a stochastic process that represents the prediction inaccuracy. The equation indicates that the randomness of residual signal originates from the randomness of the prediction inaccuracy \( \theta \). In this section, we study the mean, variance and covariance of the residual process from the stochastic behavior of \( \theta \).

### 5.3.1 Mean

We first study the mean of the process. By taking the mean of the residual process, we obtain:

\[
E [r(m, n)] = \frac{L}{\cos \alpha} \left. \frac{\partial f(0, n)}{\partial n} \right|_{n'} E [\theta(m, n)]
\tag{5.13}
\]

A practical video compression system usually contains a pretty dense set of the prediction directions. For each chosen prediction direction, the distribution of prediction inaccuracy depends on the local distribution of actual prediction direction close to that chosen prediction direction. The local probability distribution is usually close to uniform. As a result, it is reasonable to assume that the prediction inaccuracy locally would not be biased towards any side. In other words, \( E [\theta(m, n)] = 0 \). This leads to

\[
E [r(m, n)] = 0 \tag{5.14}
\]
We note that this result is consistent with the observation that the mean of a practical prediction residual signal is very close to zero.

5.3.2 Variance

Next, we study the variance of the process. Denote the variance function as $\sigma^2(m, n)$. We take the expectation of $r^2$, with respect to the random variable $\theta$.

$$
\sigma^2(m, n) = E[r^2(m, n)] \approx \left[ \frac{L}{\cos \alpha} \right]^2 \left[ \frac{\partial f(0, n)}{\partial n} \right]^2 \bigg|_{n'} E[\theta(m, n)^2] 
$$

The above equation indicates that there is much non-stationarity within the residual signal. Instead of being uniform throughout the space, the average magnitude of the residual signal is proportional to the squared distance and squared boundary gradient. In other words, the average residual intensity tends to increase when it is going away from the boundary. This is consistent with the observation in [17]. More importantly, the residual intensity tends to be large where the boundary gradient is large. We note that while the image blocks can be reasonably modeled as a stationary process, the gradient is usually highly non-uniform in the space. This relation indicates that the residual non-stationarity is reflected by a drastic change of residual variance, introduced by the non-uniformity of the gradient information.

In addition, as the model indicates, the boundary gradient is an estimation of the amount of local spatial changes along the prediction direction. Therefore, this relation also indicates that the residual is large when the estimated local change at the same location is large. The proposed model successfully established a relation between the residual magnitude of a pixel, the local spatial change at the same location and the boundary gradient along the prediction direction. This is consistent with the intuition discussed in Section 4.2.
5.3.3 Effectiveness of the proposed model in variance estimation

In this section, we discuss the effectiveness of the proposed model, particularly the effectiveness of variance estimation from encoded boundary predictors.

In the proposed method, the non-stationarity of residual signals is reflected by the local change of the estimated variance function. An accurate estimated variance function is very important in the application of the proposed model, as we wish to use this information in the subsequent steps. The estimated variance function should be consistent with the residual magnitude. In particular, the estimated variance function should be large wherever the actual residual magnitude is large.

Figure 5-5 shows the estimated $\sigma$ function from the proposed model and the magnitude of the residual signal, from the intra frame of the sequence “ice.qcif”. We use an intra prediction scheme similar to the H.264/AVC intra prediction with a block size of 4x4. The variance is estimated according to Equation (5.6) based on the chosen prediction direction.

![Estimated $\sigma$ function](image1)

![Magnitude of residuals](image2)

Figure 5-5: Comparison between the estimated $\sigma$ function and residual signal

We first observe that the estimated variance is visually consistent with the magnitude of the residual signal. To quantify the consistency between the magnitude of the residual signal and the estimated variance function, we study the relative energy compaction property. This relative energy compaction property is similar to the analysis in Section 4.2. In Figure 5-6, we show three cumulative energy curves similar to Section 4.2. In the optimal energy curve, we rank order the residual magnitude
and compute the cumulative energy from the largest residual pixels. In the randomized energy curve, we compute the cumulative energy from a randomly chosen set of pixels. In the variance relative energy curve, we rank order the estimated variance and compute the cumulative energy from pixels at the same location of largest estimated variance. The variance relative energy curve computes the cumulative energy of pixels with reference to the rank order information of the variance function. The variance relative energy curve indicates how informative the estimated variance is in preserving the residual energy.

![Energy compaction curves](image)

**Figure 5-6: Relative energy compaction curves**

In the ideal case, suppose the estimated variance is very accurate and precisely reflects the rank order information of the residual magnitude. If we choose the residual pixels from the largest estimated variance, the preserved energy as a function of the number of preserved pixels is the largest. It is represented by the optimal energy curve. On the other hand, suppose the estimated variance is not related to the residual magnitude. In this case, the cumulative energy will be close to a randomized energy curve. The variance relative energy should lie between these two extremes. From Figure 5-6, we see that the cumulative energy from the estimated variance in practice is close to the optimal cumulative energy. This suggests that the estimated variance...
is correlated with the residual pixel magnitude. This observation implies that the estimated variance is informative in predicting the magnitude of the residual signals on a pixel-by-pixel basis. In other words, the estimated variance from the prediction inaccuracy model is effective in estimating the non-stationarity of the residual signal.

### 5.3.4 Covariance

Last, we study the covariance function of the residual signal.

Since the random process is zero-mean, Equation (5.6) and Equation (5.15) directly lead to the following covariance function:

\[
\text{Cov} [r(m_1, n_1)r(m_2, n_2)] = \sigma(m_1, n_1)\sigma(m_2, n_2)R
\]

where \( R \) is the factor that characterizes the normalized covariance of the prediction inaccuracy, defined as

\[
R = \frac{E[\theta(m_1, n_1)\theta(m_2, n_2)]}{\sqrt{E[\theta^2(m_1, n_1)]E[\theta^2(m_2, n_2)]}}
\]

The relationship in Equation (5.16) indicates that the covariance function of the residual signal depends on the estimated residual standard deviation \( \sigma \) and the statistics of the prediction inaccuracy \( R \). Specifically, this equation indicates that the non-stationarity of the residuals is reflected mostly by a drastic change of the residual variance function. The non-stationarity of the residual signal also depends on the normalized covariance function of the prediction inaccuracy. By choosing a reasonable \( R \), we can obtain a reasonable residual covariance function.

In this thesis, we relate the prediction inaccuracy term with the first-order Markov process. Specifically, we choose

\[
E[\theta(m_1, n_1)\theta(m_2, n_2)] = r_{m_1-m_2}^{n_1-n_2}
\]
The choice of this $R$ function is obtained from a few considerations, which we will discuss in detail in Section 5.3.5. With the choice of the function in Equation (5.18), we can see that when $m_1 = m_2$ and $n_1 = n_2$,

$$E[\theta^2(m_1, n_1)] = E[\theta^2(m_2, n_2)] = 1 \quad (5.19)$$

With Equations (5.17), (5.18) and (5.19),

$$R = \rho_1^{m_1-m_2} \rho_2^{n_1-n_2} \quad (5.20)$$

Therefore, the residual covariance function is:

$$\text{Cov} [r(m_1, n_1)r(m_2, n_2)] = \sigma(m_1, n_1)\sigma(m_2, n_2)\rho_1^{m_1-m_2} \rho_2^{n_1-n_2} \quad (5.21)$$

5.3.5 Discussion on the $R$ function and the covariance function

In this section, we discuss in detail why we choose the specific $R$ function in Equation (5.20).

The $R$ function is the covariance function of the prediction direction inaccuracy $\theta$. The $\theta$ function is usually not directly observable. To determine a reasonable statistical model for the $\theta$ function, we consider a typical block where the prediction inaccuracy model applies. In a block where there is a strong edge, the directional intra prediction is usually very effective. The $\theta$ function characterizes the small difference between the chosen prediction direction and the true prediction direction.

There are two important observations of the true prediction direction. First, the true prediction direction for each pixel within a block is usually closely related to the local directionality around that pixel. Second, the local directional features within a small block usually do not change drastically. Geometrically, for a straight edge, the true prediction direction for all pixels are very close to a constant. For a curved edge, the
true prediction direction in a small block changes very slowly and in a continuous fashion. The geometric intuition of the pixel-wise prediction direction suggests that it can be modeled as a stationary Markov process. The \( \theta \) function is just the true prediction direction with a constant offset (the negative chosen prediction). As a result, we model \( \theta \) as the first-order Markov process and obtained a R function in Equation (5.18).

This particular choice of the R function also leads to a few interesting properties of the covariance function and the corresponding KLT. These properties are consistent with the behavior that we expect from the proposed transforms. We briefly outline these properties in this section. For a more detailed and rigorous treatment of this topic, see Appendix A.

- The covariance function in Equation (5.21) leads to the KLT that prioritizes residual pixels with large variance and ignores pixels with zero variance. With this covariance form, the variance function serves as an indicator of the priority of residual pixel in transform coding. Specifically, for the KLT basis functions with non-zero eigenvalues, they will be primarily supported in the region where the variance is large. The region of support will not include these pixels with zero variance, unless the corresponding eigenvalue of that transform basis function is zero.

- The covariance function in (5.21) leads to the KLT whose first transform basis function is equal to the \( \sigma \) function, when \( \rho \) is close to 1. With this property, we can see that the first transform coefficient is obtained by matching the residual signal with the estimated standard deviation function. Since the first transform basis function indicates the residual magnitude according to the priority of residual pixels, much energy will be compacted into the first transform coefficient. In contrast, a typical stationary transform, such as the DCT, will likely treat all residual pixels with equal priority and tries to match the residual signal with a flat pattern in the first transform coefficient. The behavior of the DCT is very inefficient in compacting non-stationary residual energy.
It is also interesting to see how the first KLT basis function compares with that of the ADST. In the ADST, the magnitude of the first transform basis function increases when it moves away from the coded boundary predictor. The magnitude of the first KLT basis function also increases as it moves away from the coded boundary predictor, due to a distance factor in the variance function. The difference between the proposed KLT and the ADST lies in the characterization of residual statistics orthogonal to the prediction direction. In the proposed KLT, the non-stationarity orthogonal to the prediction direction is modeled with a drastically changing variance function, while in the ADST, this observation is not explicitly considered.

- Under various special cases, the covariance function in (5.21) degenerates to the DCT. For example, when the estimated variance function is flat, then the transform basis functions become the DCT basis functions. When the estimated variance function is flat in a rectangular region and zero otherwise, the transform basis functions become the DCT basis functions on that rectangular region. See Appendix A for more examples.

5.4 Transforms based on the proposed covariance function

In this section, we discuss designing transforms based on the covariance function from Equation (5.21).

5.4.1 Properties of the covariance function and transforms

The covariance function in Equation (5.21) characterizes the residual non-stationarity. We can compute the KLT from the model covariance function and use that to encode the current residual block.

It is interesting to see the characteristics of the KLT from the model covariance function. Note that it is difficult to obtain a closed-form solution of the transform ba-
sis functions based on a general covariance function in the form of Equation (5.21). To study the characteristics of the transform basis functions, we use numerical solutions on a few artificial examples.

We first consider a simplified 1-D example. Suppose that a zero-mean signal is denoted as $x(n)$, where $0 \leq n \leq 3$. The variance of this signal is given by $\sigma^2(0) = \sigma^2(1) = 0$ and $\sigma^2(2) = \sigma^2(3) = 1$. A typical transform that ignores the variance information, such as the DCT, will in general result in transform coefficients of length 4. On the other hand, suppose the given variance information is considered. The covariance function proposed in Equation (5.21) will result in a transform with the first two basis functions supported only on $x(2)$ and $x(3)$. If the signal is actually consistent with the variance function, then $x(0)$ and $x(1)$ are very likely to be close to zero. The new transform will compact most energy into at most two transform coefficients. In other words, by considering the variance information, we are effectively adapting the transform to the non-stationarity of the signal. Therefore, the resulting transform tends to achieve much better energy compaction in this example.

As another example, we consider the example shown in Figure 5-7. In this example, we show the variance function in a 4x4 block on the left side. The variance of the brighter pixels is 0.9 while the variance of the darker pixels is 0.1. This variance function is used to construct a covariance function in Equation (5.21), with $\rho_1 = \rho_2 = 0.99$. On the right side, we show the KLT basis functions from this covariance function. The transform basis functions are shown with an offset 0.5 to illustrate negative values. From this figure, we observe that the region of support for the first several basis functions (the bottom row) is mostly within the region where the variance is large. This observation indicates that the proposed transform is adapted well to the non-stationarity of the signal. Specifically, the proposed transform first considers encoding the pixels with large intensities and compacts most of their energy into a small number of transform coefficients.

In Figure 5-8, we show the same variance function and KLT basis functions from the covariance function, but with $\rho_1 = \rho_2 = 0.1$. From this figure, we observe that the characteristics of many transform basis functions do not change significantly even
when $\rho$ parameters drastically change. For the first four transform basis functions (the bottom row), the region of support of these basis functions is mostly within the region of large variance. The patterns of most other transform basis functions do not change, except for possible sign inversion and different orders. This observation indicates
that the characteristics of transform basis functions do not change significantly for
different choice of $\rho$ parameters for this example. This is consistent with how the
non-stationarity is handled by the proposed method discussed in Section 5.3. The
blocks that contain a significant amount of non-stationarity are characterized by the
variance function. The choice of the $\rho$ parameters is less important in the proposed
covariance function.

5.4.2 Discussion on the proposed transforms

The proposed transforms are derived from the proposed covariance function. The co-
variance function models the non-stationarity of each block depending on the context
and changes from block to block. Therefore, the proposed transform is adaptive on
the block-by-block basis.

Despite being highly adaptive, the proposed transform is very robust. In many
transforms proposed in previous research, the transforms are estimated purely from
data [26, 44, 47]. The proposed covariance function in this thesis uses two kinds of
information. It is not estimated purely from encoded data. The first is the coded
boundary information. The second information is prior information that is absorbed
into the modeling processes. This point will be reviewed in more detail in Chapter 9.

The covariance function changes from block to block. Therefore, the transforms
have to be estimated on the fly as well. The derivation of transforms from the covari-
ance function involves eigen decomposition and may be computationally expensive.
We note that in the case of vertical and horizontal prediction, the proposed covariance
function is separable. In both cases, the computational complexity can be reduced
significantly.

Finally, we note that the covariance function is estimated only from the coded
boundaries. Therefore, the same covariance function estimation process can be syn-
chronized at both the encoder and the decoder. We do not have to transmit any side
information associated with the transform basis functions.

To summarize, the algorithm flowchart is illustrated in Figure 5-9.

**Step A:** Start from the left and the upper coded boundaries and estimate the
variance function for the block. For each pixel in the current block, estimate the variance function according to Equation (5.15).

**Step B:** Using the variance function in Step A, construct the covariance function according to Equation (5.21).

**Step C:** Compute the KLT of the covariance function in Step B. Use this KLT to encode the current block.

Figure 5-9: The flowchart of the prediction inaccuracy model: intra
Chapter 6

Implementation and experimental validation of the directional intra prediction inaccuracy model

In this chapter, we investigate the performance of transforms for directional intra prediction residuals based on prediction inaccuracy modeling. In Section 6.1, we discuss computing the gradient function on a discrete sampling grid, specifically in the context of prediction inaccuracy model. We then describe how to apply proposed transforms step by step in Section 6.2. We use an artificial example to illustrate the idea. In Section 6.3, we investigate the energy compaction performance of the proposed transforms. In Section 6.4, we investigate the rate-distortion performance of the proposed transforms based on a pseudo-H.264/AVC intra coding system.

6.1 Gradient computation

In the proposed method, we derive the residual covariance as a function of the boundary gradient. This model requires evaluating the boundary gradient at non-integer coordinates. In the ideal situation, if coded boundary samples are dense enough, the boundary gradient can be computed with very high spatial resolution. In practice, the boundary pixels are only sampled at integer coordinates. This limitation requires
the boundary gradient, at any given location, to be estimated from a small number of boundary pixels. To address this issue, we discuss the gradient computation on a discrete sampling grid. We discuss the gradient evaluation along the left boundary. The upper boundary can be derived with symmetry.

Consider estimating the variance function in Equation (5.15). In this equation, the boundary gradient is evaluated at location \( n' \). The value of \( n' \) can be computed from the location of the current pixel and the given prediction direction. While the coordinates of the current pixel are always integers, \( n' \) may not necessarily be an integer. To compute the gradient for different possible values of \( n' \), we consider three typical cases.

6.1.1 \( n' \) is a positive integer

A positive integer \( n' \) implies that we are interested in evaluating the gradient on the sampling grid. In this case, we consider estimating the gradient from three reference samples. Consider the block shown in Figure 6-1. To evaluate the gradient at location \((0, n')\), we can estimate the gradient as either \( f(0, n') - f(0, n' - 1) \) or \( f(0, n' + 1) - f(0, n') \). From the proposed model, the prediction direction inaccuracy is not biased towards either side of \( n' \). Therefore, the contribution of two estimations is likely to be equal. Since the variance is proportional to the square of the gradient, we can estimate the square of the gradient effectively as the mean square of two estimations. In other words, when \( n' \) is a positive integer:

\[
\left[ \frac{\partial f(0, n')}{\partial n} \right]_{n'}^2 = \frac{1}{2} [f(0, n') - f(0, n' - 1)]^2 + \frac{1}{2} [f(0, n' + 1) - f(0, n')]^2 \tag{6.1}
\]
When \( n' \) is not an integer, we evaluate the gradient in between two boundary pixels \( f(0, \lfloor n' \rfloor) \) and \( f(0, \lceil n' \rceil) \). This is illustrated in Figure 6-2. In this case, the squared gradient is simply given by:

\[
\left[ \frac{\partial f(0, n)}{\partial n} \right]_{n'}^2 = [f(0, \lfloor n' \rfloor) - f(0, \lceil n' \rceil)]^2
\]  

(6.2)

In the case when non-horizontal/vertical prediction is chosen, the upper left corner predictor is used when \( n' = 0 \). This is shown in Figure 6-3. In this case, the gradient can be estimated as \( f(1, 0) - f(0, 0) \) when the accurate prediction is from the upper boundary. On the other hand, the gradient can be estimated as \( f(0, 1) - f(0, 0) \) when
the accurate prediction comes from the left boundary. Both cases are equally likely to happen. As in the case when \( n' \) is a non-zero integer, we wish to estimate the gradient by averaging the two cases.
In Equation (5.15), the variance is scaled by a factor related with the prediction angle $\alpha$. The prediction angle is fixed when only one boundary is used. In the case when $n' = 0$, both the upper boundary and the left boundary are involved in the gradient computation. The prediction angle is different for the upper boundary and for the left boundary. Therefore, we chose to directly estimate the variance in this case. The variance is estimated as:

$$
\sigma^2(m, n) = \frac{1}{2} \left[ \frac{L}{\cos \alpha_U} \right]^2 (f(1, 0) - f(0, 0))^2 \\
+ \frac{1}{2} \left[ \frac{L}{\cos \alpha_L} \right]^2 (f(0, 1) - f(0, 0))^2
$$

(6.3)

where $\alpha_U$ is the prediction angle from the upper boundary and $\alpha_L$ is the prediction angle from the left boundary.

6.2 Example of applying the proposed KLT

To illustrate the intuition of the proposed method, we investigate applying the proposed KLT on an artificial example. In this example, we show a lot of functions on a block. For many of these functions, the dynamic range of different functions may be different. Usually, only relative intensities within a block matter. Therefore, we may normalize these blocks differently for the best visual effect, without explicit statement.

Figure 6-4 shows an image block. The first column and the first row are within the encoded regions of a frame and are used as encoded boundary predictors. The current 4x4 block to be encoded is outlined with the red grid. For all the blocks involved in this example, we also show the pixel intensities on the right.

This block contains a strong edge, very close to the perfect vertical direction. We use the vertical prediction to predict the current block. The vertical prediction in this example results in pretty accurate prediction. We note while other prediction directions are possible, no single prediction direction can lead to the perfect pre-
Figure 6-4: A block of original image (left) and pixel intensities (right). Current block to be encoded is outlined with red grid.

prediction. In this example, the prediction inaccuracy always exists no matter what prediction direction is chosen for the block. The prediction is shown in Figure 6-5. The corresponding residual is shown in Figure 6-6.

Figure 6-5: The vertical prediction block

To use the proposed KLT, we estimate the variance function from Equation (5.15) and the discrete approximation discussed in previous sections. The estimated pixel-wise \( \sigma \) function is shown in Figure 6-7. In previous sections, we mentioned that the variance function indicates the significance of a residual pixel. Notice how the \( \sigma \)
function matches the residual magnitude on a pixel-by-pixel basis.

From the variance function, we obtain the covariance function and the KLT. The KLT basis functions for this example are shown in Figure 6-8. As we have discussed in previous sections, the region of support of significant transform basis functions will focus in the large variance region. In this example, the variance function is supported primarily in the middle two columns of the 4x4 block. The first few transform basis functions have the same region of support.

We can now apply the KLT on the residual block. We show the KLT coefficients
on the left of Figure 6-9. To illustrate the advantage of using the KLT, we also show the DCT coefficients of the same residual block on the right of Figure 6-9. From this figure, it is very clear that the KLT is advantageous over the DCT. Specifically, the first KLT coefficient is significantly larger than other coefficients. On the other hand, the energy is less compact in the DCT domain.

Figure 6-10 shows the energy compaction performance of two transforms, by preserving different numbers of transform coefficients. From the figure, we can see that the energy compaction performance of the KLT is significantly better than the DCT for this example.
Figure 6-9: Magnitude of KLT coefficients and DCT coefficients

Figure 6-10: Energy compaction performance of the KLT and the DCT
6.3 Energy compaction performance

In this section, we investigate the energy compaction property of the proposed method. We discuss the experimental setup in Section 6.3.1. We then compare the performance of the proposed method based on a hybrid transform with the ADST or the DCT. Experimental results indicate that the energy compaction performance significantly increases with the additional use of the proposed transforms.

6.3.1 Experimental setup

In the experiments that we perform, we first obtain the directional intra prediction residuals for each intra frame of interest. The prediction residuals are generated using 9 prediction modes defined in the H.264/AVC system. The block size is fixed to 4x4. For this part, we use the original boundaries for prediction. In practice, we only have access to the reconstructed boundaries. The effect of a distorted boundary will be discussed in Section 6.3.7. The proposed transforms are considered only when a block chooses the directional intra prediction mode. When the DC mode is chosen, a stationary transform is used.

For prediction residual blocks, we apply a hybrid transform. In our study, we observe that the proposed model is effective in modeling blocks with clear directionality and sharp discontinuities. In these cases, the principles of directional prediction apply very effectively and so does the prediction inaccuracy model. In some other cases, textures or other complex features arise. In such situations, intra prediction is not very effective in the first place and a directional mode may be chosen because it happens to minimize the error. In these cases, the prediction model and prediction inaccuracy model do not apply. A stationary transform is usually more robust in these cases. To account for both types of blocks, we choose to use the proposed transforms in hybrid with a stationary transform on a block-by-block basis. The necessity and advantage of a hybrid transform will be discussed again in Section 6.3.6.

For the proposed transforms, we estimate the covariance function as discussed in Chapter 5. The parameter $\rho$ is chosen to be 0.99. We note that in our experiments,
changing $\rho$ within a reasonable range does not significantly affect the results. In the covariance estimation process, the coded boundary gradient may become zero in the boundary and smooth regions. When this happens, the estimated covariance function is ill conditioned. For these cases, we use a stationary transform such as the DCT or the ADST instead.

In our experiments, original samples are used to estimate the covariance function for the proposed transform. In a practical video compression system, we only have access to encoded boundaries subject to certain amount of distortion. In the case when the frame is encoded with reasonable quality, the distortion does not significantly affect the covariance estimation procedure. The performance of the proposed method under distortion will be investigated in Section 6.4.

The energy compaction property of the proposed transforms is investigated. Specifically, we use the proposed transforms in hybrid with the DCT or the ADST. We compare the energy compaction of the hybrid transform with the DCT or the ADST. We compute the preserved energy given the total number of chosen coefficients. Transform coefficients with largest magnitudes within a frame are chosen. In the case of the hybrid transform, the transforms and transform coefficients are selected, for each block, utilizing the algorithms proposed in [9, 10].

We plot the preserved energy as a function of the total number of chosen coefficients. The preserved energy is in terms of the percentage relative to the total energy. The total number of chosen coefficients is presented in terms of the percentage relative to the total number of coefficients. A larger preserved energy value at the same percentage of chosen coefficients indicates a higher performance in energy preservation. It is evident in [9, 10] that the energy compaction capability is a useful measure of performance in coding applications.

### 6.3.2 Test sequences

In the experiments that we perform, we use test sequences in "derf" set [1] with QCIF, CIF, 720P and 1080P resolutions. The sequences in raw formats other than YUV420 have been converted to YUV420 before processing. We show a few examples of test
sequences in Figure 6-11 and Figure 6-12.

Figure 6-11: Sample sequences in the derf set: QCIF and CIF resolutions
Figure 6-12: Sample sequences in the derf set: 720P and 1080P resolutions
6.3.3 Results: hybrid with the DCT

In this section, we compare the energy compaction performance of two transform settings: 1) The 2D-DCT. 2) The proposed transform in hybrid with the 2D-DCT.

Figure 6-13 shows the energy compaction performance of the hybrid transform and the DCT, for the intra frame in the sequence “carphone.QCIF”. From this figure, we see that the same amount of energy can be preserved with a significantly smaller number of transform coefficients by using the KLT in addition to the DCT.

![Figure 6-13: Energy compaction performance of the hybrid transform (vs DCT). Sequence: Carphone.QCIF](image)

To quantify the coefficient saving, we measure the percentage of coefficient reduction when a certain amount of energy is preserved by two transforms. For this example, the additional use of the KLT typically reduces around 20% of coefficients to preserve the same energy as the DCT. The percentage of coefficient saving is summarized in Table 6.1. In this table, the percentage of coefficient saving in the second row is measured when the same amount of energy is preserved by two transforms.
### Percentage of preserved energy (roughly)

<table>
<thead>
<tr>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
</table>

### Percentage of coefficient saving

| 18.6% | 21.9% | 23.1% | 23.5% |

Table 6.1: Coefficient saving of the hybrid transform relative to the DCT. Sequence: Carphone_QCIF

### Percentage of preserved coefficients (roughly)

<table>
<thead>
<tr>
<th>3%</th>
<th>5%</th>
<th>7%</th>
<th>9%</th>
</tr>
</thead>
</table>

### Percentage of coefficient saving

| 15.6% | 16.1% | 16.0% | 15.4% |

Table 6.2: Average coefficient saving of the hybrid transform relative to the DCT.

The percentage of preserved coefficients for the DCT is specified in the first row.

The same experiment was repeated for the test sequences. The average coefficient savings relative to the DCT is summarized in Table 6.2. The detailed experimental data is attached in Appendix B.

In addition, we observe that the proposed model is effective in characterizing a significant portion of residual blocks. To see this, we investigate the frequency of choosing the KLT in our experiments. For the sequences in our tests, the KLT is chosen for 45.67% of the non-zero blocks, when around 5% of the coefficients is preserved. In other words, the proposed transform is more effective than the DCT for these blocks.

In another experiment, we show the scatter plot between the frequency of choosing the KLT and the coefficient saving. This is shown in Figure 6-14. In this figure, each circle represents a test sequence. From this figure, a positive correlation appears between the frequency of choosing the KLT and the coefficient saving. This indicates that more coefficients tend to be saved when the proposed model is effective in capturing more non-stationarity of the residual signal. This is consistent with the insights of the proposed method.

### 6.3.4 Results: hybrid with the ADST

In this section, we compare the energy compaction performance of two transform settings: 1) The ADST. 2) The proposed transform in hybrid with the ADST.
Figure 6-14: Coefficient saving to the frequency of choosing the KLT

Table 6.3: Coefficient saving of the hybrid transform relative to the ADST. Sequence: Carphone.QCIF

<table>
<thead>
<tr>
<th>Percentage of preserved energy (roughly)</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of coefficient saving</td>
<td>13.1%</td>
<td>13.9%</td>
<td>15.1%</td>
<td>16.4%</td>
</tr>
</tbody>
</table>

Figure 6-15 shows the energy compaction performance of two transform settings, for the intra frame in the sequence “carphone.QCIF”. Similar to the case of the DCT, we observe a significant amount of coefficient saving when the proposed method is used in addition to the ADST. The percentage of coefficient saving is summarized in Table 6.3.

The same experiment was repeated for the same sequence set. The average coefficient savings relative to the ADST is summarized in Table 6.4. The detailed experimental results are attached in Appendix B.

We also investigate the frequency of choosing the KLT vs the ADST. For the sequences in our tests, the KLT is chosen for 38.71% of the non-zero blocks, when
Table 6.4: Average coefficient saving of the hybrid transform relative to the ADST.

<table>
<thead>
<tr>
<th>Percentage of preserved coefficients (roughly)</th>
<th>3%</th>
<th>5%</th>
<th>7%</th>
<th>9%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of coefficient saving</td>
<td>12.8%</td>
<td>12.7%</td>
<td>12.8%</td>
<td>12.6%</td>
</tr>
</tbody>
</table>

Figure 6-15: Energy compaction performance of the hybrid transform (vs ADST). Sequence: Carphone_QCIF

around 5% of the coefficients is preserved. Compared to the case when the KLT is used in hybrid with the DCT, the frequency of choosing the KLT is slightly smaller. Still, the frequency of choosing the KLT is significant. This indicates that the proposed transforms are still effective when the ADST replaces the DCT.

The same scatter plot between the frequency of choosing the KLT and the coefficient saving is shown for the case of the ADST in Figure 6-16. A similar positive correlation appears between the two factors.
6.3.5 Comparison of the energy compaction performance on different transforms

In this section, we summarize the energy compaction performance of the proposed transforms. Specifically, we compare the energy compaction performance of the four transforms discussed in previous sections. They are 1) DCT, 2) ADST, 3) KLT hybrid with DCT and 4) KLT hybrid with ADST. We measure the performance in terms of the percentage of coefficients used to preserve the same amount of energy relative to the DCT. The coefficient saving is measured when the same energy is preserved with 5% DCT coefficients, averaged over the sequences that we tested. The result is shown in Figure 6-17.

From the figure, we see that the DCT on average results in the worst performance. Replacing the DCT with the ADST slightly improves the performance as expected from [17]. When the KLT is used in addition to either the DCT or the ADST, the
performance significantly improves. This is because the prediction inaccuracy model is effective in many typical residual blocks. The covariance estimated from this model captures the non-stationarity of residual signals that neither the DCT nor the ADST can capture. In fact, when the KLT is used, whether it is hybrid with DCT or the ADST only makes a small difference. This implies that much non-stationarity in the residual signals is captured by the proposed KLT. The remaining stationary blocks can be encoded with a reasonable stationary transform and the choice of such transform is not as important.

![Percentage of preserved coefficients relative to the DCT](image)

**Figure 6-17:** Percentage of preserved coefficients relative to the DCT

### 6.3.6 Choice of transforms

In our experiments, the KLT is used in hybrid with the ADST or the DCT. We are interested in the choice of transforms for each block.

Figure 6-18 shows the choice of transform for an intra frame "ice_CIF" when the KLT is used in hybrid with the ADST. In this figure, we show the original intra frame, blocks that select the KLT and blocks that select the ADST. For the blocks with very small residual energy, all transform coefficients are below the threshold and they are treated as zero blocks. We only show the intra blocks that have significant residual transform coefficients. We first observe that the KLT is chosen in those regions with
Intra frame

Blocks that choose the KLT

Blocks that choose the ADST

Figure 6-18: Choice of transform for each block

sharp discontinuities and directional structures. We can visually reconstruct the contours of the objects from those blocks that choose the KLT. The proposed model is expected to be effective in these regions. On the other hand, the KLT is less
effective in the regions where the ADST is chosen. From the figure, we see that these blocks are distributed more randomly than the KLT blocks. Most of these blocks are relatively stationary, have complicated textures or discontinuities with less regular directionality. We observe similar patterns for other intra frames in our experiments.

6.3.7 Discussion of energy compaction performance

The energy compaction performance of the proposed method indicates that the additional use of the proposed transforms significantly reduces the number of transform coefficients used to preserve the same energy. This implies that the proposed transforms will potentially be useful in a practical video coding system. In a practical video compression system, the performance of transforms depends on many other factors as well. For example, the magnitude of transform coefficients are important as well as the number of transform coefficients. Quantization and entropy coding methods will affect the coding performance. To indicate which transform to use for each block in a hybrid setting, we need to either infer the transform choice from previously coded information or send a 1-bit side information for each block. These issues are not directly related to the transform itself, but may impact the performance of transforms in a practical system. In the experiments regarding energy compaction performance, we generate the residual signal and estimate the variance function from the original boundaries. In a practical system, we only have access to the reconstructed and distorted boundaries. The distortion present in the reconstructed boundary may also affect the accuracy of the prediction and variance function estimation. Whether the improvement of energy compaction performance can be translated into bit rate reduction in a more practical setting needs to be verified on a practical video compression system.

In the next section, we modify a practical video compression system to include the proposed transforms. The rate-distortion performance is evaluated. We can see that even without much fine tuning of the system, a straightforward implementation of our transforms in a coding system leads to a significant improvement in coding performance.
6.4 H.264/AVC system performance

6.4.1 System implementation

To evaluate the rate-distortion performance of the proposed transform, we implement the proposed transform on a pseudo-H.264/AVC intra coding system. In this section, we discuss the system implementation.

We first obtain the JM 18.6 reference software and make a few minor modifications. First, to compare with the state-of-art transform, we replace the DCT in JM18.6 with the ADST. The implementation is based on the floating-point arithmetic and matrix multiplication. In addition, we change the quantization implementation to floating-point division. The original H.264 implementation couples the transform and the quantization process. This particular modification leads to a better separation between transform and quantization process. This is necessary in our experiments since the original system does not work with normalized coefficients that we are interested in the KLT.

We implement the proposed transforms on the pseudo-H.264/AVC system with high profile. The system diagram is shown in Figure 6-19. The transforms based on prediction inaccuracy modeling are used in hybrid with the ADST. The encoder first makes the mode decision with a full rate-distortion search. For each 4x4 block that uses directional intra prediction, we choose the ADST or the proposed transform whichever results in a better rate-distortion trade-off. A full rate-distortion optimization is applied when choosing the ADST or the proposed transform, for each directional mode of consideration. A one-bit side information for each non-zero block is used to indicate the chosen transform. Due to a roughly equal frequency of choosing the ADST and the proposed KLT, the one-bit side information is encoded with an equal-probability binary model in the CABAC engine. Optimization on the side information context model may further reduce the bit rate for the side information. Note that the proposed transforms are estimated from the same coded boundaries used for H.264/AVC intra prediction. The coded boundaries are known to both the encoder and the decoder. As a result, the encoder and the decoder can always be
synchronized. The components that have been added in addition to the H.264/AVC intra coding system are indicated by green color in the system diagram.

Figure 6-19: H.264/AVC implementation overview

In our implementation, the transform basis functions are scaled properly in accordance with the H.264/AVC quantization and entropy coding modules. After we obtain the transform coefficients, H.264/AVC quantization and entropy coding are applied. In our experiments, the CABAC entropy coding is used. To determine a reasonable scanning order of the transform coefficients, we use adaptive scanning. Specifically, we compute the eigenvalues of the transform coefficients from the estimated covariance function. The eigenvalues indicate the expected variance of transform coefficients. A transform coefficient with larger variance is more likely to be a non-zero significant coefficient. Therefore, the transform coefficients with larger eigenvalues are scanned first. On the decoder side, the covariance function is first computed from the pro-
posed model using the reconstructed boundary. The covariance function is used to
determine the coefficient eigenvalues, whose order is used in determining the location
of transform coefficients, which are decoded afterwards. We do not change other parts
of the quantization and the entropy coding modules. Optimization on these modules
may improve the performance of our proposed method.

We encode a set of intra frames with QCIF, CIF, 720P and 1080P resolutions
among our test sequences. We modify the transforms for 4x4 directional intra pre-
diction residuals in the H.264/AVC system. The transforms for other modes are not
changed. The proposed method is applied to the Y channel only. We compare two
transform settings: 1) the ADST. 2) the proposed transforms used in hybrid with the
ADST. The rate-distortion performance is evaluated.

6.4.2 Experimental results

In Figure 6-20, we show the rate-distortion performance of sequences “ice_qcif” and
“carphone_cif” of the proposed hybrid method and the ADST. We first observe that
the coding performance significantly increases with the proposed transform. For
the sequence “ice_qcif”, the bitrate saving is around 10 %. For the sequence “car-
phone_cif”, the bitrate saving is around 5 %.

To see the performance of the proposed method on a larger set, we show the
Bjontegaard-Delta (BD) rate [5] saving of the proposed method relative to the ADST,
measured from quantization parameters (QP) 24, 28, 32 and 36. The BD rate saving
is measured on the entire intra frame, including 4x4 intra blocks, 8x8 intra blocks
and 16x16 intra blocks. A positive value indicates bitrate saving.

The results for low resolution QCIF and CIF sequences are shown in Figure 6-
21. The results for high resolution 720P and 1080P sequences are shown in Figure
6-22. First, we observe a significant improvement on the coding performance for most
sequences in our experiments, compared to the state-of-art ADST. Second, we observe
that the coding gain varies sequence by sequence. In general, sequences with sharper
edges and more clear local directionality benefit more from the proposed method.
For example, the “ice_QCIF” intra frame mostly contains many sharp edges around
the contour of the skaters. The proposed method has a relatively large coding gain on this frame. On the other hand, the “soccer_CIF” contains many busy textures. This makes the prediction inaccuracy model less applicable. In the high definition sequences, the “life_1080” sequence contains computer generated animation. The computer generated animation contains many very sharp edges. The presence of sharp edges makes the proposed method applicable to the sequence. As a result, the coding gain of the “life_1080” sequence is significantly larger than other high definition sequences. On the other hand, the BD rate for the “blue sky_1080” sequence is
Figure 6-21: BD rate of the proposed method vs ADST: QCIF and CIF resolutions

Figure 6-22: BD rate of the proposed method vs ADST: 720P and 1080P resolution

abnormally negative. We note this sequence is subject to a very strong motion blur due to the camera motion. Therefore, all sharp edges have been significantly blurred, leaving the sequence with few strong discontinuities. This makes the proposed model ineffective. We also observe the coding gain changes with respect to the resolution of
the sequence. The bit rate reduction tends to be smaller for sequences with higher resolution. When the resolution gets larger, the proportion of regions that contain edges typically gets smaller. As a result, the proportion of regions where the proposed method may be effective gets smaller. This reduces the advantage of using the proposed transform and leads to a smaller bit rate reduction. We note all these observations are consistent with the intuition of the proposed model.

In the method we propose, there is a 1-bit/block side information to indicate which transform is used for each 4x4 block. Despite this, the proposed method shows a significant overall bit rate reduction for typical intra frames used in our experiments.

Figure 6-23 shows the frequency of choosing the proposed transform. The results are evaluated at QP = 28. For each sequence, we count the number of blocks that use the proposed transform. We then count the number of blocks that use either the proposed transform or the ADST. The percentage is calculated as a ratio between two.

![Frequency of choosing the proposed transform](image)

Figure 6-23: Frequency of choosing the proposed transform

From this figure, the proposed transform is chosen fairly frequently when used in hybrid with the ADST. In addition, we observe a correlation between the frequency of choosing the proposed transform shown in Figure 6-23 and the coding gain shown in
Figure 6-21. For example, more than 60% of the blocks choose the proposed transform in the “ice_QCIF” sequence. The coding gain is around 10% for this sequence. These are very large among all sequences. We note that this observation is consistent with the analysis shown in Section 6.3.

In the proposed hybrid transform, we use a 1-bit side information for each non-zero 4x4 block to indicate the chosen transform. Figure 6-24 shows the percentage of bitrate for the side information. The results are reported with QP = 28. For a typical sequence, around 2% of the available bitrate is used to transmit the side information. The bitrate for the side information is relatively stationary for all sequences in our experiments. We note that the 1-bit side information for transform choice is encoded independently in each block. In practice, the transform choice of one block may be spatially correlated with that of nearby blocks. The bitrate for the side information could be reduced by exploiting such correlation.

Figure 6-24: Percentage of side information bitrate
6.4.3 Visual quality

Figure 6-25 shows the intra frame of “ice_qcif” and reconstructed frames, encoded by the ADST and the proposed system, respectively. We highlight three image patches where the visual quality is significantly different between two methods. Compared to the ADST, although the proposed method encodes the same frame with much fewer bits (0.846 BPP vs 0.927 BPP), the proposed method shows a significant improvement in visual quality.

Specifically, the reconstructed frame of the ADST is subject to visible compression artifacts in the smooth regions close to edge regions. These artifacts arise primarily because of quantization error. In the blocks with strong edges, many high-frequency components of the residual signal arise in the ADST domain. The support of these transform basis functions extends across the entire block. Therefore, the quantization noise of the high-frequency components extends beyond the edge regions into the smooth regions. This type of artifacts is generally known as ringing artifacts.

With the proposed method, the ringing artifacts are significantly suppressed. In the proposed method, a variance function is estimated. The estimated variance function is supported primarily in the edge region. The variance function is used to derive a covariance function and the KLT. As we have discussed, the significant KLT basis functions are supported in region of large variance. Therefore, quantization noise in these transform basis functions is confined in the edge regions. In other words, the quantization noise is localized around edge regions where the estimated variance is large. The ringing artifacts in the smooth regions are hardly visible. We also note that the localization of quantization error around edge regions helps improve the perceptual quality due to visual masking. Specifically, the same amount of noise is more visible in smooth regions than in edge regions. Visual masking further helps hiding the compression artifacts in the proposed method.
Figure 6-25: Visual comparison of the proposed method vs ADST
6.4.4 Discussion on the computational complexity

The proposed method increases the computational complexity of the system. Specifically, we can analyze the computational complexity by breaking the additional computations into three parts.

(1) Estimation of transform basis functions: To obtain transform basis functions for an $N \times N$ block in Step C discussed in Section 5.4.2, eigen-decomposition of an $N^2 \times N^2$ matrix is involved. This step increases the computational complexity of both the encoder side and the decoder side. In our experiments, we observe that this step introduces most of the additional computations.

We note that the variance function in Equations (5.15) and the covariance function in Equation (5.21) are separable when the prediction direction is vertical and horizontal. In these cases, the eigen-decomposition step can be significantly simplified. Specifically, for an $N \times N$ block, we can use an $N \times N$ matrix instead of an $N^2 \times N^2$ matrix. The resulting transform basis functions are also separable.

(2) Transform and inverse transform: The transform basis functions in the proposed method are in general not separable and a full-matrix multiplication type implementation needs to be involved. This step affects both the encoder side and the decoder side.

We note that in the horizontal and vertical cases, the transform basis functions are separable. In addition, the transform basis functions depend on the boundary gradient only along the direction orthogonal to the prediction direction. The transform for the other direction is fixed and can be hard-coded.

(3) Transform selection: To select the best transform for each block, a full rate-distortion optimization is applied in our implementation. In other words, each block is encoded twice, once with the ADST and once with the proposed transform. This step only increases the computational complexity of the encoder side.
Chapter 7

Motion compensation inaccuracy model

In this chapter, we derive the prediction inaccuracy model for motion-compensated prediction residuals. In Section 7.1, we study the characteristics of motion-compensated residuals. We focus on the residual non-stationarity that arises from motion-compensated prediction inaccuracy. In Section 7.2, we derive the prediction inaccuracy model by relating the residual signal with motion vector inaccuracy. In Section 7.3, we use the model in Section 7.2 and study the residual statistics. Specifically, we derive the residual mean, variance and covariance function, based on the statistics of motion vector inaccuracy. In Section 7.4, we discuss how to design transforms based on the proposed model covariance function.

7.1 Characteristics of motion-compensated prediction residuals

It is well known that the characteristics of motion-compensated residuals are significantly different from those of typical images. In Figure 7-1, we show a typical video frame and the motion-compensated residual frame from a previous frame.

In the motion-compensated residual frame, we first observe that the energy de-
creases due to effective motion compensation in most regions. In addition, the spatial
distribution of the residual energy is significantly different from that of still images.
In still images, the energy distribution is relatively stationary in a small local region.
In the residual frame, however, we observe that the residual intensities are much less
stationary. The residual intensities may change significantly even within a small local
region.

We observe that in the non-stationarity that arises in the residual frame, there is
one particular type of non-stationarity with clear patterns. In many regions of the
residual frame, the residual signal has a strong visual correlation with the gradient
of the original frame. One can even roughly reconstruct the object boundary by
looking at the residual signal. The correlation between motion-compensated residual
frame and the gradient information in the original frame and reference frame was also
reported in [39].

This particular type of non-stationarity of a residual frame can be interpreted in
terms of prediction accuracy. Suppose the variation between two frames consists of
pure object motions. In the ideal case when perfect motion vectors are obtained,
the motion predicted frame will be identical to the original frame. In this case, the
motion-compensated residual frame will be close to zero in most regions unless new
contents appear in the new frame. However, in many video coding systems, the
motion vector is only estimated to a certain accuracy. The use of inaccurate motion
vectors in motion compensation may result in a significant amount of residual energy.

The correlation between the residual frame and gradient information can be interpreted by modeling the motion-compensated inaccuracy as a result of motion vector inaccuracy. To see this point, in smooth regions where the spatial change is very small, the motion-compensated prediction will be pretty accurate for a large number of choices of motion vectors. In this case, using a less accurate motion vector will not significantly affect the motion prediction. However, in regions where strong edges and busy textures arise, the motion prediction is very sensitive to the accuracy of the motion vector. A small disturbance of the true motion vector may result in a large prediction error. The sensitivity of the prediction error to the motion vector depends on the amount of local spatial changes. As a result, the residual non-stationarity can be related to the motion compensation inaccuracy from an inaccurate motion vector and local changes of image data.

This intuition will be used to derive a prediction inaccuracy model in the following sections.

7.2 Model derivation

In this section, we establish the prediction inaccuracy model for motion-compensated residuals. Consider a rectangular block being encoded, as shown in Figure 7-2. We use the following notation:

\( f(m, n) \): a pixel in the current block to be encoded at location \((m, n)\).

\( \hat{f}(m, n) \): a reference pixel in the displaced reference block at the same location \((m, n)\). The reference block is displaced according to the estimated motion vector.

\( r(m, n) \): the motion-compensated residual pixel, obtained by subtracting the prediction \( \hat{f}(m, n) \) from \( f(m, n) \).

\( \sigma^2(m, n) \): the variance function for the residual pixel \( r(m, n) \).

In the above notation, \( m \) is the horizontal coordinate and \( n \) is the vertical coordinate in the block of interest. This is illustrated in Figure 7-2.

Suppose we have very accurate motion vectors. We may, then, find a more accurate
reference pixel at a different location \((m_a, n_a)\), in a small region near the current reference pixel. We denote this pixel as \(\hat{f}(m_a, n_a)\). The difference between \((m, n)\) and \((m_a, n_a)\) is denoted as a random vector \((\Delta m, \Delta n)\). The vector \((\Delta m, \Delta n)\) characterizes the motion vector inaccuracy. The vector \((\Delta m, \Delta n)\) is also a vector function of the spatial coordinate. For notational simplicity, this dependency is omitted. This motion vector inaccuracy universally exists in practical video coding systems in part due to the limited motion vector precision and non-ideal sub-pixel interpolation. We note that \((\Delta m, \Delta n)\) is usually very small. In practice, we may assume, for example, that \((\Delta m, \Delta n)\) is uniformly distributed in a small square region centered around the origin.

Based on these assumptions, we relate the residual signal with the motion vector inaccuracy. We first note that the residual is obtained by subtracting the prediction from the original pixel:

\[
r(m, n) = f(m, n) - \hat{f}(m, n)
\]

(7.1)

Since \(\hat{f}(m_a, n_a)\) is relatively accurate, ignoring the difference between the current pixel and the more accurate reference pixel, we obtain:

\[
f(m, n) \approx \hat{f}(m_a, n_a)
\]

(7.2)

From Equations (7.1) and (7.2), we get:
\[ r(m, n) = f(m, n) - \hat{f}(m, n) \approx \hat{f}(m_a, n_a) - \hat{f}(m, n) \]
\[ \approx (\Delta m, \Delta n) \cdot \nabla \hat{f}(m, n) = \Delta m \frac{\partial \hat{f}}{\partial m} + \Delta n \frac{\partial \hat{f}}{\partial n} \]  
(7.3)

where we use the two-dimensional Taylor expansion for a small \((\Delta m, \Delta n)\).

Equation (7.3) indicates that under the proposed model, the residual signal is related with prediction inaccuracy and gradient of the reference block on a pixel-by-pixel basis. From the form of this equation, the residual magnitude is likely to be large when the gradient magnitude is large. However, different from the prediction inaccuracy model for the intra prediction, the model for inter prediction relies on a two-dimensional inaccuracy term. The exact relation between the residual magnitude and the gradient magnitude requires a careful modeling of this two-dimensional inaccuracy term. In the following section, we will study the statistics of the residual signal from this model.

### 7.3 Model statistics

With Equation (7.3), we study the statistics of the residual signal.

#### 7.3.1 Mean

First, we study the mean of the residual signal. Similar to the argument in the prediction inaccuracy model for intra residuals, in the small region around the accurate motion vector, \(\Delta m\) and \(\Delta n\) will not generally be biased towards any side. Therefore, we obtain:

\[ E[r(m, n)] = \frac{\partial \hat{f}}{\partial m} E[\Delta m] + \frac{\partial \hat{f}}{\partial n} E[\Delta n] = 0 \]  
(7.4)
7.3.2 Variance

Next, we compute the variance function of the residual signal. With a zero-mean residual signal, we compute its variance by:

\[
\sigma^2(m, n) = E[r^2(m, n)] \approx \left( \frac{\partial \hat{f}}{\partial m} \right)^2 E[(\Delta m)^2] + \left( \frac{\partial \hat{f}}{\partial n} \right)^2 E[(\Delta n)^2] + \left( \frac{\partial \hat{f}}{\partial m} \frac{\partial \hat{f}}{\partial n} E[\Delta m \Delta n] \right)
\]

(7.5)

The variance function depends on the variance of inaccuracy from both directions. In addition, it depends on a third cross-correlation term. We note that in practice, the motion vector inaccuracy is usually restricted in a small square region, defined by the quantization granularity of the motion vectors. In this small region, it is reasonable to assume that the distribution of motion vectors is relatively uniform in that small local region. To see a reasonable explanation of this point, we consider the opposite of the assumption. The prediction inaccuracy term being highly non-uniform implies that the distribution of motion vectors is highly irregular with fine structures even in small quantization steps. This is not likely to happen in practice. This leads to the vector \((\Delta m, \Delta n)\) being isotropic with respect to both directions and uncorrelated. In other words, \(E[\Delta m \Delta n] = 0\) and \(E[(\Delta m)^2] = E[(\Delta n)^2] = K\), where \(K\) is a constant that does not depend on the spatial location. We obtain:

\[
\sigma^2(m, n) = E[r^2(m, n)] \approx K \left| \nabla f \right|^2
\]

(7.6)

The above equation implies that the non-stationarity of the residual signal depends on the local changes of the reference frame. Specifically, the variance function of the residual block is spatially variant. The variance function is linearly proportional to the squared gradient magnitude of the reference block. This result is consistent with the intuition discussed in Section 7.1. We note that in previous work reported in [39],
a non-linear relation between the variance function and the gradient magnitude has been proposed.

### 7.3.3 Effectiveness of the proposed model in variance estimation

In this section, we discuss the effectiveness of variance estimation from the proposed model. In the proposed method, the non-stationarity of motion-compensated residuals is reflected by the local change of the estimated variance function. A reliable variance function estimation should be consistent with the residual magnitude. In particular, the estimated variance function should be large wherever the actual residual magnitude is large.

Similar to the intra case, we show the estimated \( \sigma \) function and the magnitude of the residual signal in Figure 7-3 for the sequence “foreman.CIF”.

![Estimated \( \sigma \) function and Magnitude of residuals](image)

**Figure 7-3:** Comparison between the estimated \( \sigma \) function and residual magnitude frame

From the figure, the estimated \( \sigma \) function is visually consistent with the magnitude of the residual frame. We use the relative energy compaction property to quantify the consistency. This relative energy compaction property is similar to the analysis in Section 4.2 and Section 5.3.3. In Figure 7-4, we show three cumulative energy curves. In the optimal curve, we rank order the residual magnitude and compute the cumulative energy from the largest residual pixels. In the randomized curve, we compute the cumulative energy from a randomly chosen set of pixels. In the variance
relative curve, we rank order the estimated variance and compute the cumulative energy from pixels at the same location of the largest estimated variance. The variance relative curve indicates how informative the estimated variance is in preserving the residual energy.

![Relative energy compaction functions](image)

Figure 7-4: Relative energy compaction functions

In the ideal case, suppose the estimated variance is very accurate and precisely reflects the rank order information of the residual magnitude. If we choose the residual pixels from the largest estimated variance, the preserved energy as a function of the number of preserved pixels is the largest. It is represented by the optimal energy curve. On the other hand, suppose the estimated variance is not related to the residual magnitude. In this case, the cumulative energy will be close to a randomized energy curve. The variance relative energy should lie between these two extremes. From Figure 7-4, we see that the cumulative energy from the estimated variance in practice is close to the optimal cumulative energy. This suggests that the estimated variance is correlated with the residual pixel magnitude. This observation implies that the estimated variance is informative in predicting the magnitude of the residual signals on a pixel-by-pixel basis. In other words, the estimated variance from the prediction inaccuracy model is effective in estimating the non-stationarity of the residual signal.
7.3.4 Covariance

Last, we study the residual covariance function. With \( E[r(m, n)] = 0 \), the covariance function is given by:

\[
\text{Cov}[r(m_1, n_1) r(m_2, n_2)] = E[r(m_1, n_1) r(m_2, n_2)]
\]  
(7.7)

Similar to the prediction inaccuracy model for intra residuals, we define

\[
R[r(m_1, n_1) r(m_2, n_2)] = \frac{E[r(m_1, n_1) r(m_2, n_2)]}{\sigma(m_1, n_1) \sigma(m_2, n_2)}
\]  
(7.8)

Then, the covariance function is equal to

\[
\text{Cov}[r(m_1, n_1) r(m_2, n_2)] = \\
\sigma(m_1, n_1) \sigma(m_2, n_2) R[r(m_1, n_1) r(m_2, n_2)]
\]  
(7.9)

The \( R \) function characterizes the pixel-wise correlation of the motion vector inaccuracy in a block. In this thesis, we relate it with the first-order Markov process. Specifically, we use in this thesis

\[
R[r(m_1, n_1) r(m_2, n_2)] = \rho_x^{\vert m_1 - m_2 \vert} \rho_y^{\vert n_1 - n_2 \vert}
\]  
(7.10)

We will discuss why this particular \( R \) function is used in Section 7.3.5.

From Equation (7.9) and Equation (7.10), we obtain the covariance function as:

\[
\text{Cov}[r(m_1, n_1) r(m_2, n_2)] = \\
\sigma(m_1, n_1) \sigma(m_2, n_2) \rho_x^{\vert m_1 - m_2 \vert} \rho_y^{\vert n_1 - n_2 \vert}
\]  
(7.11)

We note that a similar form of residual covariance function has also been proposed in [39] to encode motion-compensated residuals.
7.3.5 Discussion on the R function and the covariance function

The choice of the R function in Equation (7.10) is obtained from similar considerations discussed in Section 5.3.5.

Specifically, the R function for motion-compensated residuals characterizes the motion vector inaccuracy. For a typical block under motion compensation, the true motion vector field of that block is locally highly correlated. The geometric intuition of a motion vector field also indicates that it can be approximated with a Markov process. As a reasonable approximation, we model it as a stationary Markov process. The motion vector inaccuracy term, as the difference between the true motion vector and a constant block-wise motion vector, can be modeled as a stationary Markov process as well. Therefore, we use the first-order Markov process for the motion vector inaccuracy and this leads to the R function in Equation (7.10).

This particular choice of the R function also has a few interesting properties. These properties are consistent with what we expect from the proposed KLT. Briefly, these properties are:

- The covariance function in Equation (7.11) leads to significant transforms basis functions that are mainly supported in the region of large variance.

- The covariance function in Equation (7.11) leads to the KLT whose first transform basis function is equal to the $\sigma$ function, when $\rho$ is close to 1. This is a desirable property since the $\sigma$ function indicates the expected magnitude of the residual block. By matching the residual block with this fast changing function, much non-stationary residual energy can be compacted into the first transform coefficient.

- The covariance function in Equation (7.11) leads to DCT under degenerated cases.

For a more detailed discussion of this topic, interested readers are referred to Appendix A.
7.4 Transforms based on the proposed covariance function

From the covariance function obtained from Equation (7.11) and (7.6), we compute and use the KLT basis functions to encode the current block. The covariance function for motion-compensated prediction residuals in Equation (7.11) is in the same form as that for intra prediction residuals in Equation (5.21). As a result, the transform basis functions share similar characteristics with those for intra prediction residuals. This was discussed in Section 5.4.

Briefly, the proposed transforms are adaptive transforms for residual non-stationarity. The covariance estimation process involves a simple gradient computation procedure. This makes the covariance estimation quite robust. The covariance functions have to be estimated on the fly. For the case of motion-compensated residuals, the covariance function is in general not separable. Since the estimated covariance function only depends on the displaced reference block from the encoded reference frame, it is synchronized at the encoder and the decoder. No side information for transform basis functions needs to be transmitted.

Similar to the intra case, the algorithm flowchart is illustrated in Figure 7-5.

**Step A:** Estimate the variance function for the block from displaced reference block. For each pixel in the current block, estimate the variance function according to Equation (7.6).

**Step B:** Using the variance function in Step A, construct the covariance function according to Equation (7.11).

**Step C:** Compute the KLT of the covariance function in Step B. Use this KLT to encode the current block.
Figure 7-5: The flowchart of the prediction inaccuracy model.
Chapter 8

Implementation and experimental validation of the motion compensation inaccuracy model

In this chapter, we study the performance of the prediction inaccuracy model for motion-compensated prediction residuals. In Section 8.1, we discuss computing the gradient function on a discrete sampling grid. In Section 8.2, we investigate the energy compaction performance of the proposed transforms. In Section 8.3, we investigate the rate-distortion performance of the proposed transforms based on a modified H.264 system.

8.1 Gradient Computation

The proposed model in Equation (7.6) relies on a 2D gradient function defined in the continuous domain. To use the model in practice, the gradient function needs to be evaluated on the discrete sampling grid.

In the model for motion-compensated prediction residuals, the gradient function is computed on the displaced reference block. Even though the displacement of the reference block, determined by the motion vector, may be non-integer, the displaced reference blocks are always resampled on the integer grid with sub-pixel interpolation.
Therefore, the discussion of gradient computation is relatively simple as we only need to consider computing the gradient on integer pixel locations.

We use the notation defined in Section 7.2. To see the gradient computation for the model, we consider Figure 8-1. In this figure, we are interested in computing the squared gradient magnitude of displaced reference block $\hat{f}$ at the current pixel $(x, y)$. In other words, we would like to evaluate $|\nabla \hat{f}|^2$ at $(x, y)$.

![Figure 8-1: Gradient computation for the model of motion-compensated residuals](image)

When a continuous formulation is used, the gradient function is uniquely defined. For a discrete signal, the gradient function can be approximated in multiple ways. Particularly in the case of prediction inaccuracy model, the gradient depends on the quadrant the vector $(\Delta m, \Delta n)$ is in.

Suppose $\Delta m < 0$ and $\Delta n < 0$. The gradient is evaluated in the left upper region of the current pixel. The gradient vector can be approximated as

$$\nabla \hat{f}_{-,-} \approx (\hat{f}(x, y) - \hat{f}(x - 1, y), \hat{f}(x, y) - \hat{f}(x, y - 1)) \quad (8.1)$$

where the subscript denotes the quadrant of the motion vector inaccuracy. The
squared magnitude is given by

\[ |\nabla \hat{f}|_{-,-}^2 \approx (\hat{f}(x, y) - \hat{f}(x - 1, y))^2 + (\hat{f}(x, y) - \hat{f}(x, y - 1))^2 \quad (8.2) \]

Similarly, when \((\Delta m, \Delta n)\) is in other quadrants, we obtain:

\[ |\nabla \hat{f}|_{+,+}^2 \approx (\hat{f}(x, y) - \hat{f}(x + 1, y))^2 + (\hat{f}(x, y) - \hat{f}(x, y + 1))^2 \quad (8.3) \]

\[ |\nabla \hat{f}|_{-,-}^2 \approx (\hat{f}(x, y) - \hat{f}(x - 1, y))^2 + (\hat{f}(x, y) - \hat{f}(x, y + 1))^2 \quad (8.4) \]

\[ |\nabla \hat{f}|_{+,+}^2 \approx (\hat{f}(x, y) - \hat{f}(x + 1, y))^2 + (\hat{f}(x, y) - \hat{f}(x, y + 1))^2 \quad (8.5) \]

Similar to the argument in the intra case, the motion vector inaccuracy is equally likely to be in one of the four quadrants. Therefore, the squared gradient magnitude at \((x, y)\) can be approximated by averaging four cases from Equation (8.2) to (8.5). Therefore, we obtain:

\[ |\nabla \hat{f}|^2 \approx \frac{1}{2}[(\hat{f}(x, y) - \hat{f}(x + 1, y))^2 + (\hat{f}(x, y) - \hat{f}(x, y - 1))^2 + (\hat{f}(x, y) - \hat{f}(x - 1, y))^2 + (\hat{f}(x, y) - \hat{f}(x, y + 1))^2] \quad (8.6) \]

We use this equation for gradient computation throughout the experiments.

### 8.2 Energy compaction performance

In this section, we investigate the energy compaction property of the proposed transforms. We discuss the experimental setup in Section 8.2.1. We then compare the performance of the proposed method based on a hybrid transform and the DCT.
Experimental results indicate that the energy compaction performance of transforms significantly increases with the additional use of the proposed transforms.

8.2.1 Experimental setup

We first obtain the motion-compensated residuals for each sequence in our test. The test sequences that we use are the same as the ones in Section 6.3.2. The motion-compensated residuals are obtained with full search within a 32x32 search region. In the motion estimation process, half-pixel motion vector accuracy is used. We use the same set of interpolation filters defined in the H.264/AVC system for sub-pixel interpolation.

For each 4x4 block in the residual signal, we apply a hybrid transform. Similar to the argument in the intra case, we observe that the proposed model is effective in modeling blocks with non-stationarity in the regions of strong discontinuities. In some other cases, textures or other complex features arise. In this situation, the proposed model is less effective. A stationary transform is used instead. With such considerations, we use the proposed KLT in hybrid with the DCT. We compare the performance of two transforms: the DCT and the proposed KLT used in hybrid with the DCT. In the hybrid transform, the DCT or the proposed KLT is selected with the algorithms proposed in [9].

The parameters $p_x$ and $p_y$ in the model are chosen to be 0.99. We note that in our experiments, changing $p_x$ and $p_y$ within a reasonable range does not significantly affect the results.

We plot the preserved energy as a function of the total number of chosen coefficients. The preserved energy is in terms of the percentage relative to the total energy. The total number of chosen coefficients is presented in terms of the percentage relative to the total number of coefficients. A larger preserved energy value at the same percentage of chosen coefficients indicates a higher performance in energy preservation. It is evident in [9, 10] that the energy compaction capability is a useful measure of performance in coding applications.
8.2.2 Results: hybrid with the DCT

Figure 8-2 shows the energy compaction performance of transforms for the sequence “foreman.CIF”. From this figure, we observe that the same amount of energy can be preserved with a significantly smaller number of transform coefficients with the hybrid transform.

![Energy compaction performance of the hybrid transform (vs DCT). Sequence: foreman.CIF](image)

To quantify the coefficient saving, we measure the percentage of coefficient reduction when a certain amount of energy is preserved by two transforms. For this example, the additional use of the KLT typically reduces around 20% of coefficients to preserve the same energy as the DCT. The percentage of coefficient saving is summarized in Table 8.1. In this table, the percentage of coefficient saving in the second row is measured when the same amount of energy is preserved by two transforms. The percentage of preserved coefficients for the DCT is specified in the first row.

The same experiment was repeated for the test sequences in our test set. The average coefficient saving relative to the DCT is summarized in Table 8.2. The first
<table>
<thead>
<tr>
<th>Percentage of preserved energy (roughly)</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of coefficient saving</td>
<td>19.9%</td>
<td>20.1%</td>
<td>17.7%</td>
</tr>
</tbody>
</table>

Table 8.1: Coefficient saving of the hybrid transform relative to the DCT. Sequence: Foreman_CIF

<table>
<thead>
<tr>
<th>Percentage of preserved coefficients (roughly)</th>
<th>3%</th>
<th>5%</th>
<th>7%</th>
<th>9%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of coefficient saving</td>
<td>13.0%</td>
<td>13.2%</td>
<td>13.1%</td>
<td>12.8%</td>
</tr>
</tbody>
</table>

Table 8.2: Average coefficient saving of the hybrid transform relative to the DCT.

row shows the percentage of preserved coefficients when the DCT is used. The second row shows the average coefficient reduction when the additional KLT is used. More detailed results can be found in Appendix B.

In addition, we observe that the proposed KLT is effective in characterizing a large portion of residual non-stationarity. In our experiments, the KLT is chosen for 43.2% of non-zero blocks when around 5% of the coefficients are preserved. In other words, the proposed method is more effective to characterize the non-stationary features in around half of the blocks for typical residual signals.

In another experiment, we show the scatter plot between the frequency of choosing the KLT and the coefficient saving. This is shown in Figure 8-3. In this figure, each circle represents a test sequence. From this figure, a positive correlation appears between the frequency of choosing the KLT and the coefficient saving. This indicates that more coefficients tend to be saved when the proposed model is effective in capturing more non-stationarity of the residual signal. This is consistent with the insights of the proposed method.
8.2.3 Choice of transforms

To see the effectiveness of the proposed method, we consider an example from the sequence “foreman.CIF” shown in Figure 8-4. In the figure, we show the original frame, the motion-compensated prediction residual frame in the first row. In addition, we show the residual blocks that choose the DCT and the KLT from the hybrid transform in two figures in the second row, respectively. For each 4x4 block, we use a white block to indicate that it chooses the transform. The blocks that choose neither transform are zero blocks.

In the blocks where much motion vector inaccuracy arises, the proposed model is effective in modeling the residual non-stationarity. The proposed KLT is more effective in these blocks. These regions may include, for example, regions with edges and strong discontinuities. In this example, a region that frequently chooses the KLT is the upper right corner with edges of diagonal direction. On the other hand, the residual energy in some other blocks does not arise because of motion vector
inaccuracy. Some of the reasons may include busy textures and non-translational motions. In these blocks, it is hard to predict the residual statistics from the proposed model. We use the DCT for these blocks, for example, in the mouth region of the person in the frame. This observation implies that the proposed model is very effective in characterizing the residual non-stationarity that arises from using an inaccurate motion vector. It also indicates the necessity of using a hybrid transform.

Figure 8-4: Choice of transform for each block
8.3 H.264/AVC system performance

8.3.1 System implementation

To evaluate the rate-distortion performance of the proposed model for motion-compensated prediction residuals, we implement the proposed transform on the H.264/AVC video coding system. The implementation is based on the JM 18.6 reference software. The system diagram is shown in Figure 8-5. The proposed transform is used in hybrid with the DCT. In our implementation, we choose to use the proposed transform or the DCT for each 8x8 block. The 8x8 block is further segmented into 4x4 subblocks and 4x4 transforms are applied. The choice of this implementation conforms with the original H.264/AVC mode decision structure. A one-bit side information for each non-zero 8x8 block is used to indicate the chosen transform. The transforms are derived from the same encoded information from the encoder and the decoder side. No side information of transform basis functions is encoded. The components that have been added in addition to the H.264/AVC coding system are indicated by green color in the system diagram.

The quantization and entropy coding modules are modified similar to the system in Section 6.4. Transform coefficients and basis functions are scaled in accordance with the H.264/AVC structure. Transform coefficients are scanned in the decreasing order of eigenvalues from the model. Quantization and entropy coding are not changed. The proposed transform is only applied to the Y channel. In our experiment, we encode 30 frames per sequence, with the first frame encoded as the I frame and others as P frames. We report the results for all 30 frames.

8.3.2 Rate-distortion results

To see the performance of the proposed method, we show the rate-distortion curve of the hybrid transform and the DCT in Figure 8-6, for the sequences “silent.CIF” and “paris.CIF”. Quarter-pixel motion estimation is used in this experiment. We first observe a significant coding gain with the additional use of the proposed transform.
The coding gain comes from a better characterization of residual non-stationarity from using an inaccurate motion vector in motion compensation. We also observe that the coding gain is larger when the bit-rate is higher. In our implementation, a 1-bit side information is used to indicate the transform choice. The effect of the 1-bit side information is less when the sequence is encoded with a higher bit-rate. We observe similar effects for many other sequences in our experiments.

The proposed method is specifically designed for handling residual non-stationarity from inaccurate motion compensation. We expect to benefit more from the proposed method when motion vectors with coarser precision are used. To see this effect, we disable the sub-pixel motion estimation. The rate-distortion curve for the same sequence is shown in Figure 8-7. The coding gain is significantly larger than that shown in Figure 8-6, due to larger amount of residual non-stationarity handled by the proposed transform.
Figure 8-6: R-D performance: motion compensation with quarter-pixel motion vector precision
Figure 8-7: R-D performance: motion compensation with integer-pixel motion vector precision
To see the performance of the proposed method on many other sequences, we show the BD rate saving of the proposed method relative to the DCT, measured from QPs 20, 24, 28 and 32. A positive value indicates bit-rate saving. We compare the BD rate saving of motion compensation with integer-pixel and quarter-pixel precision, respectively. The results are shown in Figure 8-8 and Figure 8-9. First, we observed that the proposed method results in a significant coding gain for most test sequences. Occasionally, a slight amount of coding loss occurs when very little residual non-stationarity in the sequence is handled by the proposed method. In this case, the 1-bit side information may overwhelm the benefit from the proposed method. These sequences usually include contents with static background and irregular motions. Second, for all the sequences under test, the coding gain is larger when coarse motion vector precision is used. This is consistent the previous discussion. Third, similar to the discussion in Section 6.4.2, the coding gain is generally smaller for sequences with larger resolutions. In the sequences with larger resolutions, the proportion of regions with non-stationarity is smaller. This results in a smaller benefit from the proposed method.

![Figure 8-8: BD rate of the proposed method vs DCT: QCIF and CIF resolutions](image)

We note that the performance of the prediction inaccuracy model for motion-
compensated residuals is overall worse than that for directional intra prediction residuals. The relatively poor performance is due to a smaller amount of significant residual non-stationarity present in motion-compensated residuals. Specifically, due to a very strong temporal correlation in video signals, the motion compensation is usually very effective. The amount of residual energy and residual non-stationarity left to be exploited by the prediction inaccuracy model are in general smaller than in directional intra prediction residuals. This can also be observed by comparing a typical intra prediction residual frame and a motion-compensated residual frame. The intra prediction residual frames usually have much larger energy, implying a bigger potential in performance improvement when a better method is used.

Figure 8-9: BD rate of the proposed method vs DCT: 720P and 1080P resolutions
8.3.3 Visual quality

Figure 8-10 shows a P frame of “paris_CIF” sequence and the reconstructed frame, encoded by the DCT and the proposed method, respectively. Quarter-pixel motion compensation is used. The proposed method shows a significant improvement in visual quality, even though it uses fewer bits to encode the frame.

The visual improvement of the proposed method is similar to the case of intra prediction residuals. Specifically, the proposed method results in an overall less noisy reconstruction. The ringing artifacts from quantizing high frequency DCT coefficients are reduced by using a transform whose basis functions are localized in the regions with significant residuals. The improvement of visual quality is highlighted with three patches in the figure.

8.3.4 Discussion on the computational complexity

Similar to the intra case, the proposed method increases the computational complexity of the system. The computational complexity increases in three parts of the system. The on-the-fly estimation of the basis functions, the transform implemented by matrix multiplication and transform selection in the rate-distortion optimization loop. We note that in the case of the transforms for motion-compensated residuals, the variance function is estimated from the two-dimensional gradient of the reference block. As a result, it is not separable in general. The computational complexity analysis is the same as that of the intra case. Readers are referred to 6.4.4 for detailed analysis of the computational complexity.
Figure 8-10: Visual comparison of the proposed method vs DCT
Chapter 9

Conclusions and future work

9.1 Summary

In this thesis, we propose the prediction inaccuracy model. The prediction inaccuracy model is used to design transforms for residual non-stationarity that arises frequently from inaccurate prediction parameters in practical image and video compression systems. Experimental results indicate that the proposed model effectively characterizes residual non-stationarity. The energy compaction performance and the rate-distortion performance significantly increase with the proposed method.

The prediction inaccuracy model is used to design transforms of the current block following three major steps. First, the pixel-wise variance function is estimated from coded information. Second, the residual covariance function is estimated from the estimated variance function. Third, the KLT of the estimated residual covariance function is computed and applied to the current block.

The prediction inaccuracy modeling approach is applicable to many types of prediction residuals. In this thesis, we provide examples of transforms for directional intra prediction residuals and motion-compensated prediction residuals. We compare and summarize two models in Table 9.1. It is possible to include other types of prediction residuals into the framework. For example, it is possible to apply prediction inaccuracy analysis on interview prediction residuals in multiview video compression and resolution enhancement residuals in spatial-scalable video compression.
Table 9.1: Comparison of prediction inaccuracy models for intra and inter residuals

<table>
<thead>
<tr>
<th>Source of non-stationarity</th>
<th>Intra</th>
<th>Inter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-stationarity modeled as</td>
<td>Directional intra prediction inaccuracy</td>
<td>Motion-compensated inter prediction inaccuracy</td>
</tr>
<tr>
<td>Variance estimated from</td>
<td>Prediction direction inaccuracy</td>
<td>Motion vector inaccuracy</td>
</tr>
<tr>
<td>Separability of transform</td>
<td>One-dimensional gradient of coded boundary predictors</td>
<td>Two-dimensional gradient of displaced reference block</td>
</tr>
<tr>
<td></td>
<td>Separable for horizontal and vertical prediction</td>
<td>Non-separable</td>
</tr>
</tbody>
</table>

The contribution of this thesis is discussed and summarized as follows:

- This thesis proposes the prediction inaccuracy model that explicitly explains prediction residuals from a fundamental point of view. The proposed interpretation originates from a novel perspective different from existing literature.

- In the derivation of the prediction inaccuracy model, we use the idea of perturbation. To study the statistics of a residual signal, we study the behavior of the prediction method under a small perturbation around the accurate but impractical prediction method. By doing so, the residual statistics can be related with predictors in a simple form. Note that the idea of perturbation has been studied extensively in many other fields in mathematics, physics and engineering. To the best of our knowledge, this thesis is the first attempt that tries to apply rigorous perturbation analysis to image and video compression. This thesis also indicates that this methodology may be promising in future image processing and video compression research.

- Most block transform design approaches eventually require obtaining the covariance function. The method proposed in this thesis achieves a well-balanced trade-off between the adaptivity of the covariance function and the robustness of estimation. We discuss this point in detail.

Consider an image or video compression system that only uses one transform.
The optimal transform is computed by obtaining the covariance function for each block, assuming each block is sampled from an underlying unconditional distribution. In other words, if we denote a block of signal as $X$, we are estimating the covariance function $\text{Cov}[X]$ in the marginal sense (not conditioned on anything).

Consider the case when multiple fixed transforms are used in a compression system. For example, with the use of ADST[17], MDDT[44] or 1D-DCT[20], a significant performance improvement has been reported. All these methods, from the perspective of the covariance function, share a common methodology. These methods obtain multiple covariance functions, conditioned on the mode. In other words, these methods obtain $\text{Cov}[X|\text{Mode}]$. For example, in the case of MDDT, one covariance function is estimated for each direction mode. In the case of 1D-DCT, one covariance function is estimated for each residual direction.

Naturally, one will hope that the best transforms can be obtained by extending this idea further. We can obtain one covariance function for each possible configuration of encoded information. In other words, the best transforms can be in theory obtained by obtaining the covariance function conditioned fully on encoded information, namely, $\text{Cov}[X|\text{All encoded information}]$. The encoded information may include, for example, all pixel intensities and mode information in the encoded region.

While this idea sounds interesting in theory, it suffers from a few serious problems. First, the number of possible configurations of encoded information is enormously large, much larger than the number of modes in the case of multiple fixed transforms. This prevents efficient computation and storage of the conditional covariance functions. Second, a robust estimation of the conditional covariance function is prevented by the huge number of possible configurations of encoded information. The total number of samples used to estimate such conditional covariance functions should be at least on the order of the number of possible configurations of encoded information. This is also prohibitively large.
The phenomenon is also referred to as the curse of dimensionality. In practice, a stronger prior that regularizes the relation between the encoded information and the conditional covariance function is required.

The method proposed in this thesis addresses these problems with reasonable assumptions. The prediction inaccuracy model essentially assumes that the conditional covariance function depends on the previous information only through a small subset of predictors. In other words, the prediction inaccuracy model implies the Markov assumption. This leads to a very effective and robust estimation of the conditional covariance function.

9.2 Extensions of thesis work

We propose the prediction inaccuracy modeling in this thesis and show its effectiveness in characterizing residual non-stationarity. The proposed approach can be extended in the future research.

From the research point of view, there are many important questions that arise from this thesis. We list a few of them below.

- The proposed model only considers the prediction inaccuracy. It can be modified to account for other observations for practical residual signals. For example, we can assume that the residual signal is the combination of prediction inaccuracy and a noise term. The noise term can be used to account for other possible reasons that lead to the residual energy. When the noise is white, it can be shown that the KLT basis functions do not change. Whether or not another noise model leads to better performance is an interesting topic.

- The prediction inaccuracy model is effective in the transform-based coding scenario. Whether this concept can be applied to other types and parts of the video compression systems remains unanswered. For example, is it possible to use the prediction inaccuracy model to improve the performance of a system
without transforms? This could be potentially useful for lossless image and video coding.

- In this thesis, we show that the prediction inaccuracy model can be applied to the directional intra prediction residuals and motion-compensated residuals. We also mention that the model can be applied to other types of residual signals. The exact derivation of models for other types of residual signals is very interesting as future research.

- Transforms based on prediction inaccuracy model are designed specifically for handling residual non-stationarity. The transforms are used in hybrid with a stationary transform in the experimental validation. It is interesting to study if it is possible to use a single transform with the benefit of both the prediction inaccuracy model and the stationary transform. If a hybrid transform is necessary, it is interesting to study whether we can get rid of the 1-bit side information. This might be possible with inference of the transform choice from coded information, such as nearby blocks or coded boundaries. As an alternative, it is interesting to investigate the efficient entropy-coding method for the 1-bit side information of transform choice. For example, we can exploit the spatial correlation of the transform choice map and develop a context model for the side information.

- In this thesis, we focus on 4x4 transforms. It is also interesting to see how we can extend the 4x4 transforms to larger sizes. When larger block sizes are considered, a few assumptions in the model may be less valid. For example, when the block size is very large, a small direction inaccuracy will lead to a large offset in the boundary predictors. The variance of the residual pixel may no longer be computed by the gradient of boundary predictors as we have discussed. Developing a robust model for larger block sizes is a very interesting future research topic.

- In this thesis, we only change transform basis functions from the proposed
model and leave the quantization and entropy coding untouched. The prediction inaccuracy model may carry important information for a better design of quantization and entropy coding modules as well. For example, the eigenvalues from the KLT computation indicate the expected energy of each transform coefficient. This might be useful for an improved design of quantization and entropy coding.

- In a few recent video compression systems, such as the HEVC system, the number of prediction modes increases. With more prediction modes, the residual signal tends to be less significant. From another point of view, a better design of transforms can mitigate the consequences of simple and inaccurate prediction. The proposed approach in this thesis is in particular useful when the prediction is not accurate. From this perspective, the proposed approach opens a new dimension in the design of video compression systems by accepting and dealing with prediction inaccuracy, instead of reducing the prediction inaccuracy. This is an interesting topic to study.

In this thesis, we show that the proposed method is effective in a typical video compression scenario. From the engineering point of view, to use the proposed method in a practical video system requires a lot of future engineering work.

- The proposed method fits well into the framework of many transform-based video systems. One major concern is the computational complexity of the KLT basis functions. The KLT basis functions have to be computed both on the encoder side and the decoder side. Reducing the computation in this step is crucial in the successful implementation of the proposed approach. There are many possible solutions. For example, one can design hardware specifically for the KLT basis function computation step. Another solution is to build lookup tables of transform basis functions for common boundary patterns, with boundary predictors as the index.

- The proposed transforms are usually not separable. Obtaining separable approximations may be interesting for practical video compression systems. For
example, the work reported in [15] designs optimal separable approximations of non-separable transforms based on shuffling of transform basis functions.

- In this thesis, we show that the proposed model is effective for typical image and video signals, particularly natural sequences. The proposed model can be applied to other types of sequences as well. For example, screen content coding has recently become a new attention in video compression research. It is well known that the screen content image and video signals display different statistics from the natural sequences. In particular, the screen content videos consist of much sharper edges. It is interesting to study how to apply the proposed method to applications such as compression of snapshots and live videos from computer displays, web pages and presentation slides. This can be potentially useful for web conferencing applications.

- In this thesis, we show the rate-distortion performance of a modified H.264/AVC system, as a proof of concept for the proposed method. Recently, more advanced image and video compression systems are available. For example, the performance of the HEVC system is much better than the H.264/AVC system. The HEVC system features many new coding tools. For example, it includes a larger number of intra prediction directions, a much more flexible block partition strategy and a better motion-compensation engine. We are interested in implementing the transforms on more advanced video compression systems. In this thesis, we did not evaluate the performance of the proposed method on the HEVC system. To clearly show the advantage of the proposed method on the HEVC system, a considerable amount of extra work is required. For example, the proposed model may need to change significantly in order to fit larger block sizes. The advantage of using the proposed method may also be less prominent since the residual non-stationarity might decrease with a better prediction module in the HEVC system.
Bibliography


[8] Madhukar Budagavi, Arild Fuldseth, Gisle Bjøntegaard, Vivienne Sze, and Mangesh Sadafale. Core transform design in the high efficiency video cod-


Appendix A

Properties of the proposed covariance function

Consider the covariance function given by:

\[
\text{Cov} \left[ r(m_1, n_1) r(m_2, n_2) \right] = \sigma(m_1, n_1) \sigma(m_2, n_2) \rho_1^{\text{m}_1 - \text{m}_2} \rho_2^{\text{n}_1 - \text{n}_2} 
\]  \hspace{1cm} (A.1)

In this equation, \((m, n)\) is the spatial coordinate. \(r\) is the residual process. \(\sigma\) is the pixel-wise estimated standard deviation function. \(\rho\) is a model parameter.

In this appendix, we derive a few properties of this covariance function and the KLT from the covariance function.

A.1 Residual priority according to the variance function

In this section, we show that the covariance function proposed in (A.1) leads to KLT basis functions that prioritize residual pixels with large variance and ignore those pixels with zero or small variance.
First, we show that pixels with zero variance are out of region of support of significant KLT basis functions. When the variance function of a certain pixel \((m_1, n_1)\) is equal to zero: \(\sigma(m_1, n_1) = 0\), the covariance function related with that pixel becomes zero: \(\text{Cov} [r(m_1, n_1)r(m_2, n_2)] = 0\), for all \((m_2, n_2)\) by definition in (A.1). With this, we consider an arbitrary KLT basis function \(T(m_1, n_1)\) on \((m_1, n_1)\). By definition of the KLT, we obtain:

\[
\lambda T(m_1, n_1) = \sum_{m_2, n_2} \text{Cov} [r(m_1, n_1)r(m_2, n_2)] T(m_2, n_2) = 0 \quad (A.2)
\]

This implies either \(\lambda = 0\) or \(T(m_1, n_1) = 0\).

**Case 1** When \(T(m_1, n_1) = 0\), the region of support of transform basis functions does not include those pixels with zero variance. In other words, when we apply the KLT, the pixel with zero variance will be ignored.

**Case 2** When \(\lambda = 0\), the transform basis functions may include the zero-variance pixel in its region of support. However, \(\lambda = 0\) implies that the expected energy of this transform coefficient is zero. In other words, this transform coefficient is likely to be very small. The inclusion of such transform basis functions is primarily for the completeness of the KLT and does not correspond to a meaningful pattern in the residual signal. From another perspective, one can also show that these pixels with zero variance form the null space of the operator defined by the covariance matrix. Generally, the null space of an operator usually degenerates and most useful information is preserved in its complement.

To illustrate this point with an example, consider Figure A-1. In this figure, we generate a random 4x4 variance function, except two fixed zeros at two pixels. The \(\sigma\) function is shown on the left. On the right, we show the KLT basis functions from the model. The locations of zero variance are marked with red dots. From this figure, we can see that for all the KLT basis functions except last two, two pixels with zero
variance, marked with red dots, are always zero in the basis functions. The eigenvalue of last two basis functions is zero. The results are expected from the above analysis.

Figure A-1: Example of transform basis functions with zero variance

Next, when the variance function of a residual pixel is non-zero but very small, we can expect a similar effect. We can show that $|\lambda T(m_1, n_1)|$ has a very small upper bound. To see this, suppose the variance function in a block is upper bounded by 1. This assumption does not lose generality. We can scale the variance function within a constant and scale the results accordingly. Consider a particular residual pixel $(m_1, n_1)$ with a very small $\sigma(m_1, n_1) = \epsilon$, where $\epsilon \ll 1$. We estimate $\lambda T(m_1, n_1)$ for that pixel by definition of the KLT:

$$
\lambda T(m_1, n_1) = \sum_{m_2, n_2} Cov [r(m_1, n_1)r(m_2, n_2)] T(m_2, n_2)
$$

(A.3)
\[ |\lambda T(m_1, n_1)| = \left| \sum_{m_2, n_2} \text{Cov} \left[ r(m_1, n_1)r(m_2, n_2) \right] T(m_2, n_2) \right| \quad (A.4) \]
\[ \leq \sum_{m_2, n_2} |\text{Cov} \left[ r(m_1, n_1)r(m_2, n_2) \right]| |T(m_2, n_2)| \quad (A.5) \]
\[ = \sum_{m_2, n_2} \sigma(m_1, n_1)\sigma(m_2, n_2)\rho_1^{m_1-m_2}\rho_2^{n_1-n_2}|T(m_2, n_2)| \quad (A.6) \]
\[ \leq \sum_{m_2, n_2} \epsilon|T(m_2, n_2)| \quad (A.7) \]
\[ = \epsilon \|T\|_{L1} \quad (A.8) \]
\[ \leq \epsilon \sqrt{N} \|T\|_{L2} \quad (A.9) \]

In the above derivation, (A.5) follows from the triangular inequality. (A.6) uses the model from (A.1). (A.7) follows from the assumption of small variance at \((m_1, n_1)\) and bounded variance at all other pixels. (A.9) follows from the bound between the \(L1\) norm and the \(L2\) norm. \(N\) is the number of pixels in the block of interest.

We consider the normalized transform basis functions, \(\|T\|_{L2} = 1\), and obtain:

\[ |\lambda T(m_1, n_1)| \leq \epsilon \sqrt{N} \quad (A.10) \]

From this result, when the variance of a pixel is known to be small, either transform basis functions on the pixel is very small or the expected energy of the transform coefficient is small. In either case, the pixel with a small variance is given a small priority in transform coding.

Equivalently, when the variance function of a residual pixel is significantly larger than other pixels, the region of support of basis functions is not as restricted. This indicates that the region of support of basis functions is likely to focus on the pixels with large variance. These pixels are given higher priority in transform coding.
A.2 First transform basis function when $\rho = 1$

Consider the first KLT basis function from the proposed covariance function, under the assumption that $\rho_1$ and $\rho_2$ are equal to 1. When $\rho_1$ and $\rho_2$ are equal to 1, we can verify that the first KLT basis function is actually $\sigma(m, n)$. By definition of the KLT, we obtain:

\[
\sum_{m_2, n_2} \text{Cov} \left[ r(m_1, n_1)r(m_2, n_2) \right] \sigma(m_2, n_2) = \sum_{m_2, n_2} \sigma(m_1, n_1)\sigma(m_2, n_2)\rho_1^{m_1-m_2}\rho_2^{n_1-n_2}\sigma(m_2, n_2) \tag{A.11}
\]

\[
= \sum_{m_2, n_2} \sigma(m_1, n_1)\sigma(m_2, n_2)\sigma(m_2, n_2) \tag{A.12}
\]

\[
= \lambda_1 \sigma(m_1, n_1) \tag{A.13}
\]

where $\lambda_1 = \sum_{m_2, n_2} \sigma^2(m_2, n_2)$.

This verifies that $\sigma$ is a KLT basis function. To show this basis function is actually the first and most significant transform basis function, we show that when $\rho_1 = \rho_2 = 1$, all other eigenvalues are zero. We consider another transform basis function $T(m, n)$. By the property of the KLT, $T(m, n)$ is orthogonal to the first transform basis function $\sigma(m, n)$. We obtain:

\[
\lambda T(m_1, n_1) = \sum_{m_2, n_2} \text{Cov} \left[ r(m_1, n_1)r(m_2, n_2) \right] T(m_2, n_2) \tag{A.15}
\]

\[
= \sum_{m_2, n_2} \sigma(m_1, n_1)\sigma(m_2, n_2)\rho_1^{m_1-m_2}\rho_2^{n_1-n_2}T(m_2, n_2) \tag{A.16}
\]

\[
= \sum_{m_2, n_2} \sigma(m_1, n_1)\sigma(m_2, n_2)T(m_2, n_2) \tag{A.17}
\]

\[
= 0 \tag{A.18}
\]

where the last equation follows from the orthogonality of $T(m, n)$ and $\sigma(m, n)$. The result is true for all other transform basis functions, indicating that $\lambda$ must be zero.
for all other transform basis functions except the first one.

A.3 Transform basis functions under special cases

The covariance function given by Equation (A.1) leads to transform basis functions that degenerate to the DCT under certain special cases.

A.3.1 Case 1: stationary block

In the special case where the estimated variance function is flat throughout the block given by \( \sigma(m, n) = K \), all pixels within that block are of equal importance. In this case, a reasonable transform should be the DCT. Our model in Equation (A.1) leads to the DCT when \( p \) is close to 1. Given \( \sigma(m, n) = K \), \( \text{Cov} [r(m_1, n_1)r(m_2, n_2)] = K^2 \rho_1^{[m_1-m_2]} \rho_2^{[n_1-n_2]} \). The covariance function is proportional to the covariance function from the first-order Markov model.

A.3.2 Case 2: flat variance function within a rectangular support

In another special case where the estimated variance function is flat in a rectangular sub-block and zero otherwise, a reasonable transform will be the DCT on the sub-block. The proposed model is also consistent in this case. From the analysis in Section A.1, the significant transform basis functions have region of support constraint in the region where the variance is non-zero. In this region, the covariance function is proportional to the covariance function from the first-order Markov model, as shown in Case 1. Therefore, the corresponding transform will be the DCT constraint on the same region with non-zero variance.

This is verified with an example in Figure A-2. In this figure, we show the variance block on the left. The variance block is one in the lower half of the block and zero otherwise. The resulting significant transform basis functions (two bottom rows) are the DCT of size 2x4. The eigenvalue of other transform basis functions is zero.
Figure A-2: Example of transform basis functions under a rectangular variance support
Appendix B

Energy compaction performance

In this appendix, we show the detailed energy compaction performance of the proposed method in comparison with the DCT and the ADST.

In Table B.1 and B.2, we show the energy compaction performance for directional intra prediction residuals. The first column shows the sequence. In the second and third columns, we compare the KLT/DCT hybrid transform with the DCT. We show the percentage of coefficient saving and the frequency of choosing the KLT. In the last two columns, we compare the KLT/ADST hybrid transform with the ADST. We show the percentage of coefficient saving and the frequency of choosing the KLT.

In Table B.3 and B.4, we show the energy compaction performance for motion-compensated prediction residuals. The first column shows the sequence. In the second and third columns, we compare the KLT/DCT hybrid transform with the DCT. We show the percentage of coefficient saving and the frequency of choosing the KLT.
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Gain(DCT)</th>
<th>Freq(DCT)</th>
<th>Gain(ADST)</th>
<th>Freq(ADST)</th>
</tr>
</thead>
<tbody>
<tr>
<td>akiyo.cif.yuv'</td>
<td>16.73%</td>
<td>51.74%</td>
<td>9.70%</td>
<td>35.61%</td>
</tr>
<tr>
<td>bridge-close.cif.yuv'</td>
<td>14.52%</td>
<td>45.92%</td>
<td>12.99%</td>
<td>43.59%</td>
</tr>
<tr>
<td>bus.cif.yuv'</td>
<td>10.47%</td>
<td>36.04%</td>
<td>8.40%</td>
<td>30.26%</td>
</tr>
<tr>
<td>carphone.cif.yuv'</td>
<td>22.89%</td>
<td>58.71%</td>
<td>17.06%</td>
<td>47.61%</td>
</tr>
<tr>
<td>coastguard.cif.yuv'</td>
<td>11.01%</td>
<td>35.95%</td>
<td>7.59%</td>
<td>28.13%</td>
</tr>
<tr>
<td>container.cif.yuv'</td>
<td>9.46%</td>
<td>38.53%</td>
<td>7.38%</td>
<td>33.97%</td>
</tr>
<tr>
<td>flower.cif.yuv'</td>
<td>8.38%</td>
<td>36.90%</td>
<td>8.22%</td>
<td>31.74%</td>
</tr>
<tr>
<td>football.cif.yuv'</td>
<td>12.60%</td>
<td>40.08%</td>
<td>8.41%</td>
<td>29.22%</td>
</tr>
<tr>
<td>foreman.cif.yuv'</td>
<td>22.32%</td>
<td>52.34%</td>
<td>17.44%</td>
<td>42.58%</td>
</tr>
<tr>
<td>hall.cif.yuv'</td>
<td>14.25%</td>
<td>46.69%</td>
<td>11.59%</td>
<td>38.61%</td>
</tr>
<tr>
<td>highway.cif.yuv'</td>
<td>15.23%</td>
<td>40.90%</td>
<td>13.86%</td>
<td>37.27%</td>
</tr>
<tr>
<td>ice.cif.yuv'</td>
<td>27.39%</td>
<td>61.29%</td>
<td>19.56%</td>
<td>53.20%</td>
</tr>
<tr>
<td>miss-america.cif.yuv'</td>
<td>10.01%</td>
<td>34.59%</td>
<td>7.14%</td>
<td>26.44%</td>
</tr>
<tr>
<td>mobile.cif.yuv'</td>
<td>11.72%</td>
<td>41.15%</td>
<td>9.92%</td>
<td>34.35%</td>
</tr>
<tr>
<td>mother-daughter.cif.yuv'</td>
<td>14.36%</td>
<td>41.27%</td>
<td>9.30%</td>
<td>29.34%</td>
</tr>
<tr>
<td>news.cif.yuv'</td>
<td>14.09%</td>
<td>47.26%</td>
<td>10.53%</td>
<td>38.92%</td>
</tr>
<tr>
<td>paris.cif.yuv'</td>
<td>17.92%</td>
<td>51.17%</td>
<td>13.72%</td>
<td>43.27%</td>
</tr>
<tr>
<td>silent.cif.yuv'</td>
<td>10.89%</td>
<td>38.01%</td>
<td>8.87%</td>
<td>30.04%</td>
</tr>
<tr>
<td>soccer.cif.yuv'</td>
<td>13.48%</td>
<td>39.78%</td>
<td>10.77%</td>
<td>34.49%</td>
</tr>
<tr>
<td>stefan.cif.yuv'</td>
<td>9.21%</td>
<td>35.66%</td>
<td>7.34%</td>
<td>28.80%</td>
</tr>
<tr>
<td>tempete.cif.yuv'</td>
<td>10.84%</td>
<td>41.14%</td>
<td>9.28%</td>
<td>33.90%</td>
</tr>
<tr>
<td>tennis.cif.yuv'</td>
<td>15.79%</td>
<td>41.08%</td>
<td>12.75%</td>
<td>38.15%</td>
</tr>
</tbody>
</table>

Table B.1: Energy compaction performance of proposed transform (Intra): the table shows the coefficient saving (when roughly 5 % coefficients are preserved) and the frequency of choosing KLT with DCT and ADST.
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Gain(DCT)</th>
<th>Freq(DCT)</th>
<th>Gain(ADST)</th>
<th>Freq(ADST)</th>
</tr>
</thead>
<tbody>
<tr>
<td>akiyo_qcif.yuv'</td>
<td>21.94%</td>
<td>51.70%</td>
<td>14.02%</td>
<td>40.14%</td>
</tr>
<tr>
<td>bridge-close_qcif.yuv'</td>
<td>17.67%</td>
<td>47.54%</td>
<td>17.13%</td>
<td>43.71%</td>
</tr>
<tr>
<td>bus_qcif.yuv'</td>
<td>14.74%</td>
<td>43.82%</td>
<td>14.99%</td>
<td>43.31%</td>
</tr>
<tr>
<td>carphone_qcif.yuv'</td>
<td>23.12%</td>
<td>56.84%</td>
<td>15.14%</td>
<td>43.24%</td>
</tr>
<tr>
<td>claire_qcif.yuv'</td>
<td>22.76%</td>
<td>51.29%</td>
<td>16.83%</td>
<td>45.98%</td>
</tr>
<tr>
<td>coastguard_qcif.yuv'</td>
<td>11.78%</td>
<td>39.70%</td>
<td>8.76%</td>
<td>31.95%</td>
</tr>
<tr>
<td>container_qcif.yuv'</td>
<td>13.58%</td>
<td>47.89%</td>
<td>10.68%</td>
<td>40.78%</td>
</tr>
<tr>
<td>football_qcif.yuv'</td>
<td>16.23%</td>
<td>48.18%</td>
<td>15.05%</td>
<td>42.03%</td>
</tr>
<tr>
<td>foreman_qcif.yuv'</td>
<td>21.70%</td>
<td>54.21%</td>
<td>15.29%</td>
<td>40.15%</td>
</tr>
<tr>
<td>grandma_qcif.yuv'</td>
<td>13.64%</td>
<td>41.17%</td>
<td>9.36%</td>
<td>33.46%</td>
</tr>
<tr>
<td>hall_qcif.yuv'</td>
<td>16.18%</td>
<td>54.79%</td>
<td>14.74%</td>
<td>46.95%</td>
</tr>
<tr>
<td>highway_qcif.yuv'</td>
<td>26.03%</td>
<td>48.50%</td>
<td>22.12%</td>
<td>41.81%</td>
</tr>
<tr>
<td>ice_qcif.yuv'</td>
<td>38.43%</td>
<td>69.65%</td>
<td>32.62%</td>
<td>63.91%</td>
</tr>
<tr>
<td>miss-america_qcif.yuv'</td>
<td>17.82%</td>
<td>51.65%</td>
<td>13.11%</td>
<td>41.12%</td>
</tr>
<tr>
<td>mobile_qcif.yuv'</td>
<td>16.09%</td>
<td>44.84%</td>
<td>13.59%</td>
<td>37.34%</td>
</tr>
<tr>
<td>mother-daughter_qcif.yuv'</td>
<td>19.56%</td>
<td>47.91%</td>
<td>14.07%</td>
<td>35.03%</td>
</tr>
<tr>
<td>news_qcif.yuv'</td>
<td>14.21%</td>
<td>46.33%</td>
<td>12.42%</td>
<td>42.51%</td>
</tr>
<tr>
<td>salesman_qcif.yuv'</td>
<td>13.55%</td>
<td>41.17%</td>
<td>10.92%</td>
<td>33.67%</td>
</tr>
<tr>
<td>silent_qcif.yuv'</td>
<td>11.01%</td>
<td>36.49%</td>
<td>7.82%</td>
<td>29.52%</td>
</tr>
<tr>
<td>soccer_qcif.yuv'</td>
<td>17.08%</td>
<td>45.75%</td>
<td>11.95%</td>
<td>36.94%</td>
</tr>
<tr>
<td>stefan_qcif.yuv'</td>
<td>12.22%</td>
<td>39.93%</td>
<td>11.54%</td>
<td>38.72%</td>
</tr>
<tr>
<td>suzie_qcif.yuv'</td>
<td>15.50%</td>
<td>41.35%</td>
<td>11.74%</td>
<td>33.05%</td>
</tr>
<tr>
<td>tempete_qcif.yuv'</td>
<td>13.86%</td>
<td>41.77%</td>
<td>12.33%</td>
<td>38.51%</td>
</tr>
<tr>
<td>tennis_qcif.yuv'</td>
<td>16.71%</td>
<td>51.07%</td>
<td>13.96%</td>
<td>38.97%</td>
</tr>
<tr>
<td>trevor_qcif.yuv'</td>
<td>17.84%</td>
<td>46.64%</td>
<td>14.17%</td>
<td>40.75%</td>
</tr>
<tr>
<td>Average</td>
<td>16.10%</td>
<td>45.67%</td>
<td>12.68%</td>
<td>38.15%</td>
</tr>
</tbody>
</table>

Table B.2: Energy compaction performance of proposed transform (continued)
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Gain(DCT)</th>
<th>Freq(DCT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>'akiyo.cif.yuv'</td>
<td>9.84%</td>
<td>38.93%</td>
</tr>
<tr>
<td>'bridge-close.cif.yuv'</td>
<td>10.96%</td>
<td>40.59%</td>
</tr>
<tr>
<td>'bus.cif.yuv'</td>
<td>10.97%</td>
<td>40.75%</td>
</tr>
<tr>
<td>'carphone.cif.yuv'</td>
<td>13.42%</td>
<td>45.78%</td>
</tr>
<tr>
<td>'coastguard.cif.yuv'</td>
<td>7.86%</td>
<td>31.44%</td>
</tr>
<tr>
<td>'container.cif.yuv'</td>
<td>9.04%</td>
<td>35.55%</td>
</tr>
<tr>
<td>'flower.cif.yuv'</td>
<td>13.51%</td>
<td>45.95%</td>
</tr>
<tr>
<td>'football.cif.yuv'</td>
<td>10.37%</td>
<td>40.16%</td>
</tr>
<tr>
<td>'foreman.cif.yuv'</td>
<td>20.11%</td>
<td>46.69%</td>
</tr>
<tr>
<td>'hall.cif.yuv'</td>
<td>11.19%</td>
<td>36.97%</td>
</tr>
<tr>
<td>'highway.cif.yuv'</td>
<td>10.90%</td>
<td>36.57%</td>
</tr>
<tr>
<td>'ice.cif.yuv'</td>
<td>24.52%</td>
<td>64.40%</td>
</tr>
<tr>
<td>'miss-america.cif.yuv'</td>
<td>10.12%</td>
<td>35.52%</td>
</tr>
<tr>
<td>'mobile.cif.yuv'</td>
<td>15.08%</td>
<td>45.78%</td>
</tr>
<tr>
<td>'mother-daughter.cif.yuv'</td>
<td>9.09%</td>
<td>33.14%</td>
</tr>
<tr>
<td>'news.cif.yuv'</td>
<td>10.43%</td>
<td>37.49%</td>
</tr>
<tr>
<td>'paris.cif.yuv'</td>
<td>14.91%</td>
<td>53.10%</td>
</tr>
<tr>
<td>'silent.cif.yuv'</td>
<td>14.01%</td>
<td>40.42%</td>
</tr>
<tr>
<td>'soccer.cif.yuv'</td>
<td>12.65%</td>
<td>46.81%</td>
</tr>
<tr>
<td>'stefan.cif.yuv'</td>
<td>13.05%</td>
<td>43.68%</td>
</tr>
<tr>
<td>'tempete.cif.yuv'</td>
<td>13.70%</td>
<td>46.10%</td>
</tr>
<tr>
<td>'tennis.cif.yuv'</td>
<td>10.21%</td>
<td>36.14%</td>
</tr>
</tbody>
</table>

Table B.3: Energy compaction performance of proposed transform (Inter): the table shows the coefficient saving (when roughly 5% coefficients are preserved) and the frequency of choosing KLT with DCT.
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Gain(DCT)</th>
<th>Freq(DCT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>'akiyo.qcif.yuv'</td>
<td>11.78%</td>
<td>42.92%</td>
</tr>
<tr>
<td>'bridge-close.qcif.yuv'</td>
<td>12.01%</td>
<td>42.75%</td>
</tr>
<tr>
<td>'bus.qcif.yuv'</td>
<td>11.87%</td>
<td>40.71%</td>
</tr>
<tr>
<td>'carphone.qcif.yuv'</td>
<td>18.07%</td>
<td>51.21%</td>
</tr>
<tr>
<td>'claire.qcif.yuv'</td>
<td>18.16%</td>
<td>56.54%</td>
</tr>
<tr>
<td>'coastguard.qcif.yuv'</td>
<td>8.41%</td>
<td>30.89%</td>
</tr>
<tr>
<td>'container.qcif.yuv'</td>
<td>14.13%</td>
<td>42.71%</td>
</tr>
<tr>
<td>'football.qcif.yuv'</td>
<td>12.30%</td>
<td>39.98%</td>
</tr>
<tr>
<td>'foreman.qcif.yuv'</td>
<td>9.86%</td>
<td>34.40%</td>
</tr>
<tr>
<td>'grandma.qcif.yuv'</td>
<td>10.32%</td>
<td>34.23%</td>
</tr>
<tr>
<td>'hall.qcif.yuv'</td>
<td>12.04%</td>
<td>35.22%</td>
</tr>
<tr>
<td>'highway.qcif.yuv'</td>
<td>19.57%</td>
<td>47.41%</td>
</tr>
<tr>
<td>'ice.qcif.yuv'</td>
<td>23.04%</td>
<td>61.45%</td>
</tr>
<tr>
<td>'miss-america.qcif.yuv'</td>
<td>16.54%</td>
<td>49.30%</td>
</tr>
<tr>
<td>'mobile.qcif.yuv'</td>
<td>14.21%</td>
<td>42.41%</td>
</tr>
<tr>
<td>'mother-daughter.qcif.yuv'</td>
<td>12.27%</td>
<td>46.10%</td>
</tr>
<tr>
<td>'news.qcif.yuv'</td>
<td>14.69%</td>
<td>51.57%</td>
</tr>
<tr>
<td>'salesman.qcif.yuv'</td>
<td>11.42%</td>
<td>50.09%</td>
</tr>
<tr>
<td>'silent.qcif.yuv'</td>
<td>14.68%</td>
<td>49.61%</td>
</tr>
<tr>
<td>'soccer.qcif.yuv'</td>
<td>12.83%</td>
<td>38.26%</td>
</tr>
<tr>
<td>'stefan.qcif.yuv'</td>
<td>11.19%</td>
<td>44.44%</td>
</tr>
<tr>
<td>'suzie.qcif.yuv'</td>
<td>16.23%</td>
<td>44.59%</td>
</tr>
<tr>
<td>'tempete.qcif.yuv'</td>
<td>12.17%</td>
<td>42.73%</td>
</tr>
<tr>
<td>'tennis.qcif.yuv'</td>
<td>14.40%</td>
<td>45.66%</td>
</tr>
<tr>
<td>Average</td>
<td>13.22%</td>
<td>43.20%</td>
</tr>
</tbody>
</table>

Table B.4: Energy compaction performance of proposed transform (continued)