

COMPRESSION FOR MACHINE VISION AND BEYOND

A Dissertation
in
Electrical and Electronics Engineering
and
Computer Science

Presented to the Faculty of the University
of Missouri–Kansas City in partial fulfillment of
the requirements for the degree

DOCTOR OF PHILOSOPHY

by
ZHAOBIN ZHANG

M.S., Huazhong University of Science and Technology, Wuhan, Hubei, China, 2015
B.S., Huazhong University of Science and Technology, Wuhan, Hubei, China, 2012

Kansas City, Missouri
2020

© 2020
ZHAOBIN ZHANG
ALL RIGHTS RESERVED

COMPRESSION FOR MACHINE VISION AND BEYOND

Zhaobin Zhang, Candidate for the Doctor of Philosophy Degree

University of Missouri–Kansas City, 2020

ABSTRACT

Compression has been one of the most fundamental and elusive challenges in both academia and industry. With the sheer increase of high-definition video content over the internet, developing improved compression algorithms becomes an urgent necessity. This thesis tackles the problem of visual content compression: how to reduce the transmitted data volume under specific application scenarios. One of the core steps is how to remove the redundancy to achieve a compact latent representation. We approach the problem from two directions: prediction and transform. While a typical prediction process targets at removing the statistical redundancy between the reference and current image blocks, and transform further removes the inter-pixel redundancy between residual samples. We will address compression for machine vision and related topics.

In compression for machine vision, machines will communicate amongst themselves to perform tasks without a human in the mix, which requires a separate pipeline

to achieve optimal coding performance. We aim to investigate how to efficiently transmit image features in low latency scenario and focus on developing a multiple-transform solution to achieve a more compact data representation for image retrieval task. Multiple-transform solution is proven to be more efficient to preserve more distinguishable properties for a large-scale dataset. However, over-sized transform candidate list burdens the bit-rate constraint. We develop a merge scheme to search for the optimal transforms from available transform candidates.

We will also present our efforts at contributing the development of next-generation video coding standard: Versatile Video Coding (VVC), and exploring improved intra prediction schemes beyond the High Efficiency Video Coding (HEVC) standard. 1) Based on observations on the properties of DST-7 and DCT-8, a dual-implementation support solution is developed to reduce the software run-time complexity. The (anti-)symmetric features are leveraged to reduce the number of arithmetic operations involved in deriving the transformed coefficients from the residual block. The scheme has been adopted by MPEG VVC standardization development group and was integrated into VVC reference software. 2) In prediction-relevant attempts, we explore both traditional and Convolutional Neural Network (CNN)-based schemes. Multiple Linear Regression is utilized to further explore spatial correlation with reference pixels and existing intra prediction. A CNN-based scheme is developed by combining local and non-local information for intra prediction. We demonstrate the effectiveness of these approaches.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled “Compression for Machine Vision and Beyond,” presented by Zhaobin Zhang, candidate for the Doctor of Philosophy degree, and hereby certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Zhu Li, Ph.D., Committee Chair
Department of Computer Science & Electrical Engineering

Yugyung Lee, Ph.D.
Department of Computer Science & Electrical Engineering

Reza Derakhshani, Ph.D.
Department of Computer Science & Electrical Engineering

Cory Beard, Ph.D.
Department of Computer Science & Electrical Engineering

Sejun Song, Ph.D.
Department of Computer Science & Electrical Engineering

CONTENTS

| | |
|--|------|
| ABSTRACT | iii |
| ILLUSTRATIONS | viii |
| TABLES | x |
| ACKNOWLEDGEMENTS | xii |
| Chapter | |
| 1 INTRODUCTION | 1 |
| 1.1 Background | 1 |
| 1.2 Motivation and Scope | 4 |
| 1.3 Thesis Outline | 9 |
| 1.4 Contributions | 10 |
| 2 MOBILE VISUAL SEARCH COMPRESSION | 12 |
| 2.1 An Overview of MVS | 13 |
| 2.2 The Framework | 17 |
| 2.3 SIFT Manifold Partition Tree | 20 |
| 2.4 Weighted Grassmann Pruning | 23 |
| 2.5 Experiments | 31 |
| 3 FAST TRANSFORM FOR VVC | 39 |
| 3.1 History of MTS | 40 |
| 3.2 Discrete Sinusoidal Transform Family | 43 |

| | | |
|-----|---|-----|
| 3.3 | Fast Multiple Transform Selection | 45 |
| 3.4 | Theoretical Proof | 59 |
| 3.5 | Complexity Analysis | 65 |
| 3.6 | Experiments | 72 |
| 4 | IMPROVED INTRA PREDICTION BEYOND HEVC | 83 |
| 4.1 | Overview of ANN-based Approaches | 84 |
| 4.2 | Performance Comparison | 88 |
| 4.3 | MLR for Intra Prediction | 95 |
| 4.4 | CNN for Intra Prediction | 106 |
| 4.5 | Discussion | 116 |
| 5 | CONCLUSION | 119 |
| | REFERENCE LIST | 123 |
| | VITA | 140 |

ILLUSTRATIONS

| Figure | | Page |
|--------|---|------|
| 1 | Timeline of video coding standards evolution. | 3 |
| 2 | Framework of MVS compression. | 18 |
| 3 | Building the SIFT Manifold Partition Tree (SMPT). | 22 |
| 4 | Grassmann distance and principle angles. | 24 |
| 5 | Illustration of Grassmann pruning process. | 27 |
| 6 | SIFT descriptor energy preservation comparison. | 34 |
| 7 | Results of SIFT pair-wise matching. | 36 |
| 8 | Results of image retrieval. | 37 |
| 9 | The first four basis plots of DCT-II, DST-VII and DCT-VIII. | 44 |
| 10 | Fast 16-point DST-VII forward transform algorithm. | 47 |
| 11 | Feature #2 shows a mirror-(anti)symmetric pattern. | 51 |
| 12 | Sinusoidal graphs of the tuned transform basis. | 58 |
| 13 | Software execution time of the Luma component. | 70 |
| 14 | Illustration of color space conversion. | 91 |
| 15 | PSNR results of LBC methods comparing with HEVC/VVC. | 93 |
| 16 | MS-SSIM results of LBC methods comparing with HEVC/VVC.. . . . | 94 |
| 17 | HEVC angular intra prediction modes. | 99 |
| 18 | Framework of the MIP model. | 102 |

| | | |
|----|--|-----|
| 19 | Illustration of integrating MIP model into HEVC. | 103 |
| 20 | Framework of CIP model. | 109 |
| 21 | Illustration of integrating CIP model into HEVC. | 112 |

TABLES

| Tables | | Page |
|--------|--|------|
| 1 | Overview of existing MVS descriptors | 17 |
| 2 | Comparison of with and without pseudo-inverse projection | 29 |
| 3 | Overview of MPEG CDVS dataset. | 32 |
| 4 | Comparison by using different Grassmann metrics | 35 |
| 5 | Basis functions of DCT-II, DST-VII and DCT-VIII for N -point input. . . | 43 |
| 6 | Feature #1 applicable equation groups of 16-point DST-VII. | 54 |
| 7 | Comparison of the tuned transform matrices and the original transform matrices. | 57 |
| 8 | The number of arithmetic operations for a 1D forward/inverse transform. . | 67 |
| 9 | Comparisons on software execution speed (seconds). | 69 |
| 10 | Additional metrics related to the proposed fast method. | 72 |
| 11 | Run-time performance compared with VTM-3.0 under CTC. | 74 |
| 12 | BD-Rate performance using tuned transform matrix under CTC. | 76 |
| 13 | Run-time performance compared with VTM-3.0 under Low QP. | 81 |
| 14 | BD-Rate performance of using tuned transform matrix compared with VTM-3.0 under Low QP. | 82 |
| 15 | Comparison with related methods. | 82 |
| 16 | The implementations of each LBC method. | 90 |

| | | |
|----|--|-----|
| 17 | Decoding time in seconds on VVC CTC test sequences. | 96 |
| 18 | Specification of intra prediction modes and associated names | 100 |
| 19 | The BD-Rate results of MLR for Intra prediction. | 105 |
| 20 | The BD-Rate results of CIP model. | 115 |

ACKNOWLEDGEMENTS

The past four years at UMKC have been an invaluable and memorable experience to me. When I first started my PhD in 2016, I knew little about this country and everything was completely new to me. Conceptually, I had some knowledge of “compression”, but I had never heard of the term “video coding”. It is unbelievable that over the following years I have been doing research about compression and participating in the development of video coding standards. I had been very fortunate to witness the standardization process of next-generation video coding standard, as well as the take off of applying ANN to compression. I felt quite excited, and occasionally panicked, from the beginning of this journey to be a part of this trend. I would have not been able to make this journey without the help and support of many people, to whom I feel deeply indebted.

First and foremost, my greatest thanks go to my advisor Zhu Li. He has a wide range of knowledge and always has a very insightful, high-level view about the field. The wealth of knowledge enables him viewing the problems systematically, that helps me out understanding erroneous zone timely. More importantly, Zhu is an extremely kind, open-minded and supportive advisor that I could not have asked for more. He always believes in me and gives me affirmative response even though I am not always that confident about myself. His courage, optimism and humor inspire me to keep going. I would also like to express my gratitude to him for introducing Li Li to me.

I would like to thank Li Li — a knowledgeable mentor as well as a good friend to me. Li is an extremely charming and enthusiastic person for both work and life. There is

always such an aura of calm around him, and such an acute sense of research, that he is always “The Man Who Had the Answers”. I always feel my passion getting ignited after talking with him. He can always keep everything in good order and well arranged. I want to thank Li Li for managing the UMKC MCC Lab and sharing his ideas with us. I will always be proud to be a part of this family.

During my PhD, I have done internships with Tencent Media Lab. I would like to thank my mentors Shan Liu, Xin Zhao, Xiang Li and Xiang Zhang. My internship project of Fast DST-7/DCT-8 leads to the adopted proposal as a part of this dissertation. I was impressed by their professionalism, proactiveness and initiative. I have learned important lessons from them for a successful future career.

Collaboration is a big lesson that I have learned. It has been a great honor to cooperate with Zhangyang Wang — a talented researcher at TAMU. He has a very clear sense about how to define an impactful research topic. He is always very passionate and energetic in research. I can barely understand how one can work so efficiently and orderly. I also want to thank Yiting Shao, with whom we had a very good experience working on point cloud compression.

I thank the whole UMKC MCC Lab members, especially Yangfan Sun, Hongcheng Jiang and Wei Jia, who are all very helpful and they gave me a lot of support at various times.

Outside my research life, I have been extremely lucky being supported by many great friends. Just to name a few (and forgive me for not being able to list all of them): Xiaoliang Liu, my good friend since my elementary school. Although we are following

different career paths, he is always so supportive and thoughtful. He is the person that I could always rely on whenever I need a hand. Yue Li, a brilliant video coding PhD at USTC. He has very solid background and always has very insightful perspective towards research problems. I keep learning from him and getting inspired from every discussion. Jieyang Li, my like-minded friend. He has a lot of interesting ideas, for life and work. I share a lot of joyous and stressful moments with him. Mouqing Jin, a trusted and helpful “brother”. I am glad spending the time with him in UMKC. Kobe Bryant, my spiritual idol. My condolences on the passing of this basketball genius. His life is so inspiring and watching him on the court always reinvigorated me whenever I feel frustrated. I would like to thank him for his company in the years of my youth.

I want to thank my parents: Jifa Zhang and Xiurui Ma. I would like to express my gratitude for the love and care they gave me from childhood. They made me who I am today and I never know how to make myself worthy of their care and upbringing. I hope that they are at least a little proud of me for what I have been through so far. I would like to thank my parents-in-law: Quanfen Pei and Gaiwei Han for their care and support. I would not have been able to make it without their help, especially in the first year of my PhD.

Lastly, I would like to thank my wife Pengpeng Pei and my son Ethan Zhang. Pengpeng sacrificed a lot for me and this family. She is not only my partner, but also my best friend and the person that I admire most, for her courage and dedication. I am so appreciative to her unconditional care and love. She also brought our precious little baby to me in the winter of 2017. It was an unforgettable moment when I first held such

a little one in my hands. I felt excited, happy and a little bit nervous. A new baby means responsibility to me, but it is also a beginning of all things – wonder, hope, a dream of possibilities. There are no words that can describe the euphoria I feel when he recognizes me for the first time and smiles. I want to thank Chad Wright and April Wright, who treated us as a part of their family. I am so grateful for what they have done for us, especially when we were preparing for the newborn.

CHAPTER 1

INTRODUCTION

Benefit from the rapid development in video capture and video streaming devices, video content has been increasing at an astonishing speed. Video content is estimated to occupy up to 82% IP traffic over the internet by the year 2021 [1]. Transmitting over the internet challenges existing visual compression solutions. It is estimated that current compression scheme struggles to process the large volume data even 5G technology is deployed. Advanced techniques need to be explored for the rapidly increasing media content. MPEG standardization group has been working on compression for machine vision and the next-generation video coding standard.

1.1 Background

1.1.1 Video Coding for Machines

Traditional coding methods aim for the best video/image under certain bit-rate constraint for human consumption. However, with the rise of machine learning applications, along with the abundance of sensors, many intelligent platforms have been implemented with massive data requirements including scenarios such as connected vehicles, video surveillance, and smart city.

The sheer quantity of data being produced constantly leads previous methods with a human in the pipeline to be inefficient, and unrealistic in terms of latency and scale.

There are additional concerns in transmission and archive systems which require a more compact data representation and low latency solution. This led to the introduction of Video Coding for Machines. A typical use-case is mobile visual search (MVS) application.

With high-resolution cameras, powerful CPUs and pervasive wireless connections, mobile devices can use image as search queries for objects observed by user [2]. Mobile Visual Search (MVS) applications make use of image processing technologies to recognize captured images by camera-equipped mobile devices and then retrieve relevant information. Emerging applications include military aerial re-identification, scene retrieval, landmark recognition and product identification. Instead of directly transmitting the captured images to the server which might results in unacceptable time delay over the wireless network, the applicable MVS first performs the feature extraction and then send the features to the server. Efficient compression algorithms are necessary to deal with the emerging visual query applications.

Although tons of images are uploaded to the Internet and the necessity of retrieving and analyzing image content is becoming urgent, the MVS applications are still lack of proficiency for generalization. Mobile visual search tasks require visual features to be transmitted over a network. One of the key challenges is the rich information hidden in images makes the feature size exceeds the transmission capability of current wireless networks. To capture the variety of information in the image, existing algorithms tend to generate high-dimension features. Nevertheless, the whole mobile visual search process should be performed in an efficient fashion, since small delay, typically of the order of tens

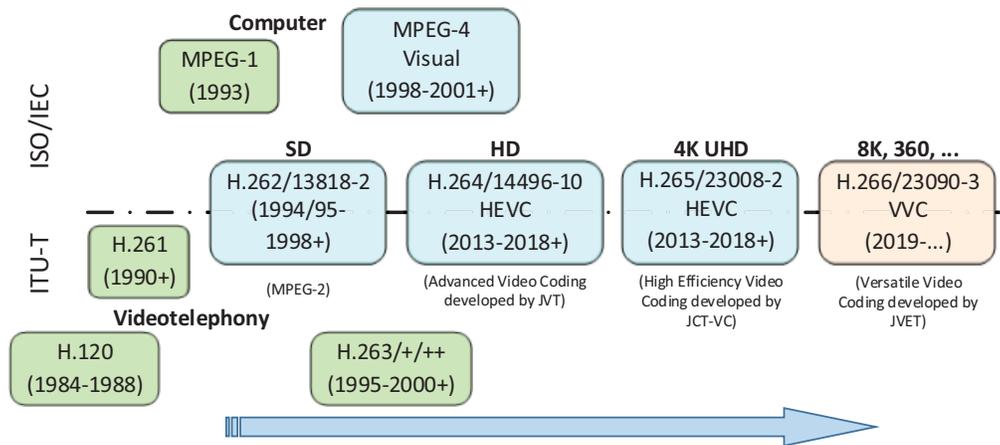


Figure 1: Timeline of video coding standards evolution.

or hundreds of milliseconds, and high frame rates are required. To this end, feature compactness is key since it allows a very large amount of visual information to be efficiently stored and queried. Moreover, concise feature-based representations can be efficiently transmitted in bandwidth-constrained scenarios such as congested mobile networks.

1.1.2 Video Coding Standards

Two standardization organizations, the International Standards Organization (ISO) and the International Telecommunications Union (ITU), have developed a series of standards that have shaped the development of the media industry. Popular ISO coding standards include MPEG-1, MPEG-2 and MPEG-4. ITU-T has published the H.26x line of coding standards including H.261, H.262, H.263 and H.264/AVC. The timeline of major standards video formats over the last 35 years is shown in Figure 1.

Most of the basic concepts of video coding such as motion estimation and compensation, transform coding and entropy coding were developed early in the 1970s and

1980s. MPEG-1 was released in the early 1990s, and MPEG-4 was developed in the late 1990s. After that, H.263 was standardized afterwards, and H.264/AVC was published in 2003. In 2013, High Efficiency Video Coding (HEVC) was released.

The explosion of video content requires video coding technology with compression capabilities that significantly exceed those of the HEVC standard. Since October 2015, ISO/IEC JTC1 SC29/WG11 MPEG and ITU-T SG16/Q6 VCEG have been working together as the Joint Video Exploration Team (JVET) to explore state-of-the-art algorithms and prepare for the next generation video standards beyond HEVC. A new generation of video compression technology that has substantially higher compression capability than the existing HEVC standard is targeted.

1.2 Motivation and Scope

With the rise of machine learning applications and availability of smart devices equipped with advanced image/video acquisition modules, along with the abundance of sensors, many intelligent platforms have been implemented with massive data requirements including scenarios such as connected vehicles, mobile visual search (MVS), video surveillance and smart city. Recently, a MVS dataset and benchmark were released from MPEG, named CDVS [2]. It is of great significance to improve the performance to cope with the drastic increment of social media applications.

Compression for machine vision heavily relies on classical video coding framework. The overall architecture is built with the techniques from both traditional video codecs and artificial neural networks. Investigating classical video coding standards is

vital to achieve substantial coding gain for machine vision compression. In addition, according to the definition from MPEG, the bit stream of machine vision compression can not only be used for machines, but also for human consumption, which shares the common goal as classical video codecs. Compression for human consumption has a loss function on pixels, while compression for machine consumption has task specific loss functions, but they both can share the same compression backbone. Therefore, this thesis will not only focus on directly compressing visual features for machine vision, but also relevant techniques to improve traditional video standards.

Video coding standards have been evolving for decades to adapt to the rapidly developing media content. The typical pipeline of prediction, transform and entropy coding was proposed back to 1980s and has proven its effectiveness over these years. On one hand, in the era of Internet media content is increasing at an unprecedented speed and the last generation video coding standard HEVC has been serving for almost a decade, video coding algorithm needs a renovation to handle the large volume media content. It should, however, not only be able to deal with existing video traffic over the Internet, but also be well-prepared for the increasing video traffic load in the next decade.

On the other hand, artificial neural networks (ANN) have achieved great success in various vision tasks. A great number of researchers in video coding community also shift their attention to using ANN for image/video compression tasks. Instead of optimizing each part separately in classical video coding pipeline, the ANN methods make the joint optimization possible. From this perspective of view, traditional codec has a higher probability to obtain local optimal performance. While the ANN methods hold

great potentials to solve this dilemma.

1.2.1 Compression for Machines

In this thesis, our compression for machine vision attempt will focus on feature compression for Mobile Visual Search (MVS) applications. ISO/IEC Moving Pictures Expert Group (MPEG) launched an MVS standard named Compact Descriptor for Visual Search (CDVS) [2]. The basic idea in CDVS is to apply a linear projection followed by ternary scalar quantization. Only a subset of empirically selected transformed descriptor elements is included in the bitstream [3]. However, it is not enough using one single transformation for all the observations to capture the rich matching information in projection space. Based on our preliminary observations [4, 5], the matching performance could be remarkably improved if multiple transformations are adopted instead of using one single transformation. The following key elements will be investigated to approach the problem.

- *Multiple transform solution*: Multiple transforms are expected to capture more distinguishable information for large-scale dataset. But it is a trade-off between the number of transform candidates and the encoding performance. Therefore, our efforts include searching the optimal transform candidates from all the available transforms.
- *Grassmann manifold*: The Grassmann manifold is a homogeneous space and a subspace is determined by its basis vectors. It provides a criterion to measure the “similarity” or “orthogonality” between two subspaces. This concept will serve as the primary theory foundation to devise our optimal transform candidates searching

algorithm.

1.2.2 Video Coding Standards

Traditional video codecs are optimized under rate-distortion optimization objective, which target for human consumption. Modern hybrid video coding framework is a complex ensemble, which typically consists of the following major steps: prediction, transform, quantization and entropy coding. We focus on two vital steps: prediction and transform, which are very crucial to achieve substantial coding gain. Prediction aims to remove spatial or temporal correlations, while linear transform is typically used to project the residual data into frequency domain to achieve compact representation, which is ideal for entropy coding.

Traditional methods to improve HEVC intra prediction include using multiple reference pixel lines, extended prediction angle, intra block copy prediction, combining chroma prediction with luma prediction, etc. The transform scheme in HEVC includes DST and DCT, which DST only can be applied to 4×4 luma component and DCT is applied to all the remaining cases. In VVC, DST-7 and DCT-8 are adopted as the primary transform scheme to better characterize the varied patterns in residual blocks. However, traversing all the combinations for horizontal and vertical transform is very time consuming. The software run-time complexity remains a problem to be resolved. The key elements that will be investigated in this thesis are summarized as follows.

- *Intra prediction*: Leveraging the spatial correlation, neighboring reference pixels are utilized to predict current block. Typically, reference pixels are subject to linear

filtering and current block is predicted by interpolation.

- *Intra block copy*: Instead of predicting current block using neighboring reference pixels, intra block copy prediction extends the reference pixels in the available decoded area within a certain range. A matching block is directly copied as current block prediction.
- *Transform*: A fast algorithm will be targeted to reduce the time complexity for DST-7 and DCT-8 transforms in VVC.

1.2.3 Artificial Neural Networks

In addition to traditional video codecs, Artificial Neural Network (ANN) has demonstrated its effectiveness in vision tasks. To explore the potentials for future video coding, the following ANN-based research aspects will be investigated.

- *Performance analysis*: Although there exists a rich literature of ANN-based image compression methods, the absence of Common Test Condition (CTC) makes it difficult to evaluate the performance. Performing performance analysis between existing ANN-based methods and state-of-the-art MPEG codecs helps to achieve a better understanding towards ANNs for compression in real applications.
- *End-to-End image compression*: The advantage of ANN-based video coding solution lies in its unique capability of joint optimization. In traditional video coding frame, each stage is optimized separately under Rate-Distortion (RD) objective, which might lead to sub-optimal solution. In contrast, the ANN-based solution

enables and end-to-end framework where all the internal components can be optimized jointly under a single objective function.

1.3 Thesis Outline

Following the two central themes that we just presented, this thesis consists of two parts.

- Part I: Video coding for machines.
- Part II: Video coding standard contributions and improvements.

Part I focuses on video coding for machines, *i.e.*, a task-driven pipeline with an emphasis on compressing the latent features so that machines can recognize between different images.

In Chapter 2, we first give an overview of the history and recent development of the field of mobile visual search. Next we formally define the the problem and describe the framework to perform mobile visual search task. Followed by a brief analysis of limitations using existing methods. We then introduce a compression scheme targeted for mobile visual search that we proposed. We present experimental results on CDVS dataset with two kinds of tasks: feature-level pair-wise matching and image-level image retrieval.

Part II focuses on video coding standard development and ANN for image compression approaches.

In Chapter 3, we first introduce the development status of next-generation video coding codec VVC and point out the major changes in contrast with HEVC. Then we

focus on analyzing the run-time complexity problem by traversing all combinations of all transform candidates for vertical and horizontal direction. Next we present the useful observations on DST-7 and DCT-8 to devise a fast transform solution. A partial butterfly-like fast algorithm is elaborated based on these observations. Theoretical proof is also provided to validate the correctness. We conduct an in-depth analysis of the fast transform algorithm to understand from which the time savings are.

In Chapter 4, we present several improved intra prediction schemes for HEVC. We first introduce the ANN-based image compression methods. Then we provide the performance analysis between ANN-based image compression algorithms and state-of-the-art MPEG codecs intra prediction schemes. A Multiple Linear Regression (MLR)-based intra prediction scheme is introduced to improve HEVC intra prediction scheme. In ANN-based schemes, we combine the local and non-local information using a Convolutional Neural Network (CNN). Experiments in HEVC reference software demonstrate their effectiveness.

We finally conclude the thesis in Chapter 5.

1.4 Contributions

The contributions of this thesis are summarized as follows:

- We design a multiple-transform solution for mobile visual search to improve the performance on top of CDVS. We were among the first to propose using Grassmann for exploring the optimal transform candidates. Noticeable improvements have been observed in the experimental results.

- We contributed to reducing the run-time complexity for VVC, the next-generation video coding standard. Theoretical identification is provided to validate its correctness. Extensive experiments have been made to understand where the benefits come from. The proposed fast DST-7/DCT-8 algorithm with dual-implementation support has been recognized by the JVET members and has been adopted by the VVC standard.
- We made the effort to performing the performance comparison between the ANN-based image compression methods and state-of-the-art MPEG codec intra prediction schemes. We are among the first to include VVC, the latest state-of-the-art MPEG video coding standard in this performance comparison. We shed light on the advantages and disadvantages by using ANN-based solution and explain relative merits.

CHAPTER 2

MOBILE VISUAL SEARCH COMPRESSION

With the popularity of mobile phones and tablets, the explosive growth of query-by-capture applications calls for a compact representation of query image feature. Compact descriptors for visual search (CDVS) is a recently released standard from the ISO/IEC moving pictures experts group (MPEG) which achieves state-of-the-art performance in the context of image retrieval applications. However, they did not consider the matching characteristics in local space in a large-scale database which might deteriorate the performance. In this chapter, we propose a more compact representation with SIFT descriptors for the visual query based on Grassmann manifold. Due to the drastic variations in image content, it is not sufficient to capture all the information using a single transform. To achieve more efficient representations, a SIFT Manifold Partition Tree (SMPT) is initially constructed to divide the large dataset into small groups at multiple scales which aims at capturing more discriminative information. Grassmann manifold is then applied to prune the SMPT and search for the most distinctive transforms. The experimental results demonstrate the proposed framework achieves state of the art performance on the standard benchmark CDVS dataset.

The remainder of this chapter is organized as follows. Section 2.1 gives an overview of MVS. Section 2.2 introduces the framework of proposed method. The proposed SIFT

Manifold Partition Tree (SMPT) is presented in Section 2.3. Weighted Grassmann pruning of SMPT is detailed in Section 2.4. Experimental setups and results are discussed in Section 2.5.

2.1 An Overview of MVS

2.1.1 Background

There are millions of images and videos added to the servers daily. For example, every second 777 photos are posted on Instagram [6] and Snapchat now achieved over 10 billion video views per day during the past year [7]. All these benefit from the rapid development of the technology of mobile devices [8]. Hand-held mobile devices, such as camera-phone, PADs are expected to become ubiquitous platforms for visual search and mobile augmented reality applications [9–11]. They have evolved into powerful image and video processing devices equipped with high-resolution cameras, color displays, hardware-accelerated graphics, Global Position System (GPS) and connected to broadband wireless networks [12]. All these functionalities enable a new class of applications which use the camera phone to initiate search queries about objects in visual proximity to users [13].

Mobile Visual Search (MVS) can be used for identifying interesting products, landmark search, comparison shopping, searching information about movies, CDs, shops, real estate, print media or artworks [14]. First commercial deployments of such systems include Google Goggles [15], Ricoh iCandy [16], Amazon Snaptell [17] and Layar [18]. Recently, Pinterest [19] also moves to leverage visual search technologies in order to

connect consumers in the e-commerce business.

In traditional text-based search, users can get the accurate retrieval results once the exact words are given. However, it is far more difficult to describe an image if people want to search for some relevant information. In the past quite a long time, people have been working on extracting image features and describing the image compactly and accurately [20–22]. Different from text-based content which is very simple and concise, the image contains much more information and is much more challenging to generate concise representations. Therefore, to fully characterize the information of an image, the existing algorithms tend to generate high-dimensional features which are usually oversized to be transmitted over the limited-bandwidth wireless networks. Meanwhile, the requirements for mobile visual search (MVS) such as lower latency, better user experience, higher accuracy pose a unique set of challenges in practical applications. Therefore, an applicable strategy is feature extraction [23] and feature compression are performed at the client end while matching and retrieval is carried out on the server.

2.1.2 Previous Approaches

There are two aspects researchers are working on, i.e., generating compact feature descriptors and compressing the feature descriptors. Developing compact feature descriptors is an effective solution to reduce the transmission data size. Initial research on the topic [12, 24–30] demonstrated that the transmission data can be reduced by at least an order of magnitude via extracting compact visual features.

In order to find compact feature descriptions thus reducing transmission bits, various of feature descriptors have been proposed to achieve robust visual content identification under rate constraints. The early-stage keypoint description algorithms assign to each detected keypoint a compact signature consisting of a set of real-valued elements. In [31], an image retrieval system is proposed, based on Harris corner detector and local grayvalue invariants. Such an approach is invariant with respect to image rotation. The work in [32] proposes Shape Context, a feature extraction algorithm that captures the local shape of a patch. Edge detection is firstly performed over a patch surrounding the point (x, y) followed by the radial grid, finally, a histogram centered in (x, y) counts the number of edge points falling in a given spatial bin.

David Lowe introduces Scale Invariant Feature Transform (SIFT) [33] which is the first to achieve scale invariance. SIFT computes for each keypoint a real-valued descriptor, based on the content of the surrounding patch in terms of local intensity gradients. The final SIFT descriptor consists of 128 elements. Given its remarkable performance, SIFT has been often used as starting point for the creation of other descriptors. Inspired by SIFT, Mikolajczyk and Schmid propose Gradient Location and Orientation Histogram (GLOH) [34]. In the context of pedestrian detection, Dalal and Triggs propose Histogram of Oriented Gradients (HOG) [35], a descriptor based on spatial pooling of local gradients. SURF [36] includes a fast gradient-based descriptor. Fan *et al.* propose MROGH [37], a 192-dimensional local descriptor. Along the same line, Girod and co-workers propose rotation invariant features based on the Radial Gradient Transform [38].

To further reduce transmission bits over the wireless network, binary descriptors

are proposed. Calonder *et al.* introduce Binary Robust Independent Elementary Features (BRIEF) [39], a local binary keypoint description algorithm. Leutenegger *et al.* propose Binary Robust Invariant Scalable Keypoint (BRISK) [40], a binary intensity-based descriptor inspired by BRISK. Differently from BRIEF, BRISK is able to produce scale- and rotation-invariant descriptors. Similarly to the case of BRISK, Fast REtinA Keypoints (FREAK) [41] uses a novel sampling pattern of points inspired by the human visual cortex, whereas Oriented and Rotated BRIEF (ORB) [42] adapts the BRIEF descriptor, so that it achieves rotation invariance.

However, these descriptors are not compact enough to transmit between remote server and client directly due to their large size [43, 44]. Table 1 shows the sizes of the descriptors. Take the SIFT as an example, the uncompressed SIFT descriptor is conventionally stored as 1024 bits per descriptor (128 dimensions and 1 byte per dimension). Even a small number of uncompressed SIFT results in tens of KBs. Hence, local feature compression of these raw features is critical for reducing the feature size.

2.1.3 A Brief Introduction to CDVS

Inspired by these recent developments, CDVS [46] tries to compress SIFT features as well as provides a standardized bitstream syntax to enable interoperability in the context of image retrieval applications and achieves state-of-the-art performance. The local SIFT compression scheme is proposed in [47]. The main idea is to group SIFT descriptors into two groups according to their relative locations and perform linear projection

Table 1: Overview of existing MVS descriptors

| Descriptor | Year | Default size (bytes) |
|----------------------|------|-------------------------|
| Schmid and Mohr [31] | 1999 | 32 |
| Shape context [32] | 2002 | 144 |
| SIFT [33] | 2004 | 128 |
| GLOH [34] | 2005 | 512 |
| HoG [35] | 2005 | 124 |
| SURF [36] | 2006 | 256 |
| DAISY [45] | 2010 | 400 |
| MROGH [37] | 2010 | 192 |
| BRIEF [39] | 2011 | 64 |
| BRISK [40] | 2011 | 64 |
| ORB [42] | 2011 | 32 |
| FREAK [41] | 2012 | 64 |

accordingly. Only a subset of descriptors is empirically selected to achieve different coding bit rates. However several drawbacks need to be addressed. First, it is not effective to perform retrieval task in a large-scale dataset by applying the same transform for all the feature descriptors. Second, with the ear of high definition broadcasting and the improvement of hardware, tons of high-resolution images/videos are generated around us. This requires more efficient algorithms to further compress these content.

2.2 The Framework

The framework of the proposed method is introduced in this section. As illustrated in Figure 2, CDVS dataset is firstly divided into training and test dataset and each with

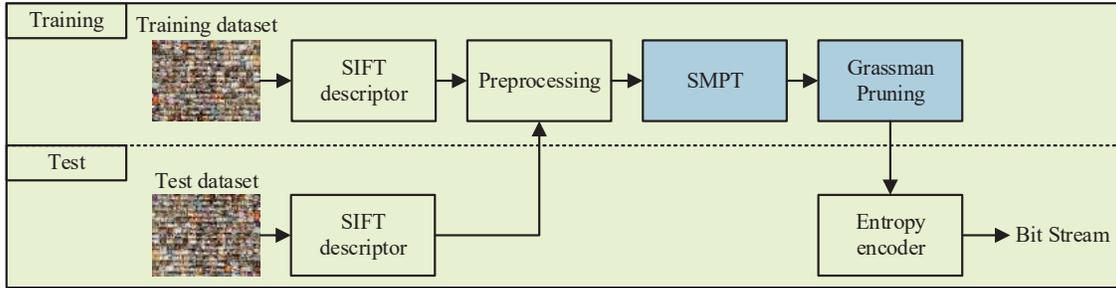


Figure 2: Framework of MVS compression.

half the number of images, including both paired images and non-paired images. Each subset in training or test is constructed so that the number of images belonging to each category is proportional to the total amount of images in each category present in the entire database. This paper addresses two innovations in blue background, *e.g.*, SIFT manifold partition tree (SMPT) and Grassmann pruning.

SIFT Manifold Partition Tree (SMPT) is constructed to divide the training SIFT descriptors into small groups wherein each, a transform will be learned with all the SIFT descriptors in that group, resulting in a set of local transforms in multiple scales. Grassmann metric is introduced to measure the similarity of two transforms and remove redundant ones. Those remaining optimal transforms are utilized for compression. Conventional entropy encoder generates the bit stream.

2.2.1 Training

All the SIFT descriptors from the training dataset are used for training. A hierarchical multi-level SIFT manifold partition tree is constructed to divide the training samples into small *groups*, *a.k.a.* *cluster* or *node*. In each group, a transform is learned

with all the SIFT descriptors in the current group. In the following section without special explanation, we will refer to *global space* as the transform space learned with the whole dataset without partition, and *local space* as the transform space learned with samples in small groups after SMPT.

Since the total number of local transforms might be very large, in addition, not all these local transforms are optimal transforms, e.g., in extreme cases, the training samples in each local space might be only one which definitely will not be able to train a satisfactory transform. Therefore, there is no need to encode all the local transforms. Grassmann metric is introduced after SMPT to prune all these available local transform candidates. In essence, Grassmann manifold provides a criterion to measure the similarity of two subspaces. In practice, it is ideal to have a group of orthogonal transform bases such that each basis captures the latent characteristics in a unique direction. While in visual query feature compression task, it is also desirable to have a bunch of transforms that they are distinctive to each other. Therefore, under Grassmann metric, we will remove those transforms which are closer to each other and only preserve those have larger Grassmann distances thus they are more likely to capture latent features in a more efficient manner.

The final optimal local transforms after Grassmann pruning will be utilized for compression. Conventional entropy encoder is utilized for encoding to obtain bitstream.

2.2.2 Test

As discussed above, half of the randomly-sampled images in CDVS dataset are used for test. Two main experiments have been devised to validate the proposed method,

e.g., pairwise matching and large-scale image retrieval experiments. For the former one, given a query SIFT descriptor from the test dataset, it will be assigned to one of those optimal local transforms obtained from the training process. Corresponding local transform will be applied on the query SIFT. It should be noted that the optimal local transforms might be in different level on the SMPT in order to capture hidden characteristics in different scales. That is where the significance of the proposed SMPT structure. Fisher vector [48, 49] aggregation has been applied to the compressed SIFT descriptors to generate the image-level representation for image retrieval experiments.

2.3 SIFT Manifold Partition Tree

In a large-scale dataset, it is usually not sufficient to capture the intrinsic characteristics in feature space if only one linear transform is applied. Also, due to the variety of dataset sizes, it is difficult to determine the appropriate size of local subspaces. Therefore, it is necessary to design an effective data partition scheme to divide the whole dataset into different levels of small groups. In this work, SIFT Manifold Partition Tree (SMPT) is proposed to divide the global dataset into small groups at different scales.

The SMPT grows in a top-down manner and the number of groups is hierarchically increasing. As the total number of training samples is fixed, different numbers of groups lead to different numbers of samples in each group. This makes it possible to exploit latent discriminative property at different scales. To assign each sample into a group, conventional aggregation methods are considered, which could be roughly divided into two categories: soft assignment and hard assignment. In our case, it is preferable to use

hard assignment as we need to assign a unique local space for each training sample. k -means is adopted in this work as it is able to preserve the geometric structure of the global dataset as well as its nature of simplicity and effectiveness.

In essence, SMPT is constructed by partitioning the large-scale SIFT dataset into small patches followed by proper design of connection relationship. The core of k -means algorithm is an easily-understood optimization problem: given a set of data points (in some vector space), try to position k other points at locations that minimize the (squared) distance between each point and its closest center. We denote the training dataset containing M SIFTs as $X = \{x_m\}$, $m = 1 \dots M$ which has to be partitioned into k clusters. K-means clustering solves

$$\arg \min_c \sum_{i=1}^k \sum_{x \in c_i} d(x, \mu_i) = \arg \min_c \sum_{i=1}^k \sum_{x \in c_i} \|x - \mu_i\|_2^2 \quad (2.1)$$

where c_i is the set of points that belong to cluster i . The standard methods for solving the k -means optimization problem are Lloyd's [50] algorithm (a batch algorithm, also known as Lloyd-Forgy [51]).

Since the algorithm stops at a local minimum, the initial position of the clusters is very important. Some common methods to initialize the centroids:

1. Forgy: set the positions of the k clusters to k observations chosen randomly from the dataset.
2. Random partition: assign a cluster randomly to each observation and compute means.

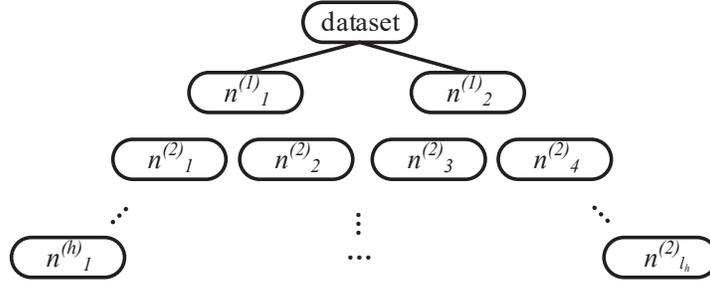


Figure 3: Building the SIFT Manifold Partition Tree (SMPT).

Due to the huge size of MVS datasets, we use the *Forgy* initialization to accelerate convergence. Before partitioning, PCA is applied on the whole SIFT dataset to reduce the dimensionality. There are two reasons for this dimensional reduction operation. Firstly, the lower dimensional local features help to produce compact final image representation as the current in image retrieval. Secondly, applying PCA could help to remove noise and redundancy; hence, enhancing the discrimination [52].

As illustrated in Fig. 3, in each level, the dimension-reduced SIFT descriptors of the whole dataset are divided into small groups via *k*-means. $n_j^{(i)}$ indicates the j -th node on the i -th level, where $i = 1, \dots, h$. It should be noted that all small groups from certain level are directly from the whole dataset, not from its parent level, e.g., combining all SIFT samples from any level would constitute the whole dataset. The association is specified in a bottom-up manner. Euclidean distance is calculated between current child-cluster centroid and its parent-cluster centroid. Current child-cluster is associated with the nearest parent-cluster.

At each node, a transform (PCA in this paper) is trained using all full-dimension (128-d) SIFT descriptors. So far, all the transform candidates are obtained.

2.4 Weighted Grassmann Pruning

As we discussed above, the partition on the global space serves as the first stage of the proposed work. However, leaf nodes may not be the best choice for retrieval because of the following reasons. First, with the increase of the SMPT level, the number of samples associated with each leaf node will decrease. Thus there exist a scenario where the number of samples is not sufficient to train a reliable transform. Take the extreme case as an instance, there will be only one sample in each leaf node when the number of nodes equals the total number of samples in the whole dataset. Second, in practice, it is expensive to encode all available transform candidates. Therefore, it is desirable to devise a scheme to search for a handful of optimal transforms.

We introduce Grassmann manifold [53,54] into the indexing model for manipulating the leaf nodes derived from the data partition tree. Each point on Grassmann manifold is a subspace by the columns of an orthonormal matrix which is invariant to any basis. The notion of principle angle and Grassmann distance allow us to evaluate the homogeneity of the SIFT feature space. In this section, we first briefly review the Grassmannian metric and related concepts, i.e., principal angles. Then we introduce the details of applying the Grassmann metric to the SMPT.

2.4.1 Grassmann Manifold

The Grassmann manifold $G(d, D)$ is the set of d -dimensional linear subspaces of the \mathbb{R}^D [53]. Consider the space $\mathbb{R}_{D,d}^{(0)}$ of all $D \times d$ matrices, i.e., $A \in \mathbb{R}^{D \times d}$. The group of transformation $A = AS$, where S is a $d \times d$ full-rank square matrix, defines an equivalence

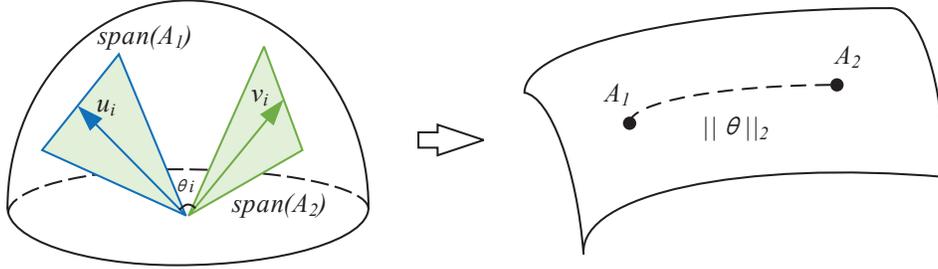


Figure 4: Grassmann distance and principle angles.

relation in $\mathbb{R}_{D,d}^{(0)}$.

$$A_1 = A_2 \text{ if } \text{span}(A_1) = \text{span}(A_2) \quad (2.2)$$

where $A_1, A_2 \in \mathbb{R}_{D,d}^{(0)}$

Therefore, the equivalence classes of $\mathbb{R}_{D,d}^{(0)}$ are one-to-one correspondence with the points on the Grassmann manifold $G(d, D)$, i.e., each point on the manifold represents a subspace.

According to the definition, each point on Grassmann manifold is a subspace. Therefore, to measure the distance between two points on the Grassmann manifold is equivalent to measure the similarities between two subspaces. Principle angle [53–55] is a geometrical measure between two subspaces. Fig. 4 has shown the relationship between principal angle and Grassmann manifold. Suppose A_1 and A_2 are two orthonormal matrices $A_1, A_2 \in \mathbb{R}^{D \times d}$ on the Grassmann manifold, the principal angles $0 \leq \theta_1 \leq \dots \leq \theta_d \leq \pi/2$ between two subspaces $\text{span}(A_1)$ and $\text{span}(A_2)$, are defined recursively by:

$$\begin{aligned} \cos \theta_k &= \max_{u_k \in \text{span}(A_1)} \max_{v_k \in \text{span}(A_2)} u_k' v_k, \\ \text{s.t. } u_k' u_k &= 1, v_k' v_k = 1, \end{aligned} \quad (2.3)$$

$$u_k' u_i = 0, v_k' v_i = 0, (i = 1, \dots, k-1)$$

The vectors (u_1, u_2, \dots, u_d) and (v_1, v_2, \dots, v_d) are principal vectors of the two subspaces. θ_k is the k th smallest angle between two principal vectors u_k and v_k .

In literature, there are a variety of methods to compute the principal angles between two subspaces. One numerically stable way is to apply Singular Value Decomposition (SVD) on the product of the two matrices $A_1' A_2$, i.e.,

$$A_1' A_2 = USV' \quad (2.4)$$

where $U = [u_1, u_2, \dots, u_d]$, $V = [v_1, v_2, \dots, v_d]$ and $S = \text{diag}(\cos \theta_1, \dots, \cos \theta_d)$. The cosine values of principal angles $\cos \theta_1, \dots, \cos \theta_d$ are known as canonical correlations [55].

2.4.2 Subspace Optimization with Grassmann Metric

The distance on Grassmann manifold is defined as follows. A distance is referred to as Grassmann distance if it is invariant under different basis representations. Grassmannian distances between two linear subspaces $\text{span}(A_1)$ and $\text{span}(A_2)$ can be described by principal angles. The smaller principal angles are, the more similar two subspaces are i.e., the closer they are on the Grassmann manifold.

In literature, various Grassmann distance metrics based on principal angle have been developed for different purposes, e.g., projection, Binet-Cauchy, max correlation, min correlation, Procrustes metric [53]. Since the distance metrics are defined with a particular combination of the principal angles, the best distance depends highly on the probability distribution of the principal angles of the given data. Among all these metrics,

max correlation and min correlation only use the maximum and minimum principal angle, respectively, thus may perform less stable when the noise in the data varies. Another criterion for choosing the distance is the degree of structure in the distance. Without any structure, a distance can be used only with a single K-Nearest Neighbor (KNN) algorithm. When a distance having an extra structure such as triangle inequality, for example, we can speed up the nearest neighbor search by estimating lower and upper limits of unknown distances. From this point of view, only Binet-Cauchy metric and projection metric are the most structured metrics as they are induced from a positive definite kernel [53]. In application, they are also the most commonly used Grassmann metrics. Therefore, in the final experimental section, both projection distance and Binet-Cauchy Grassmann distance will be evaluated. The projection Grassmann metric and Binet-Cauchy Grassmann metric can be computed as follows, respectively:

$$d_P(A_1, A_2) = \left(\sum_{i=1}^m \sin^2 \theta_i \right)^{1/2} \quad (2.5)$$

$$d_{BC}(A_1, A_2) = \left(1 - \prod_i \cos^2 \theta_i \right)^{1/2} \quad (2.6)$$

We denote the number of nodes in each level as $L = \{l_i\}$, where $i = 1, \dots, h$.

Hence, the total number of available transforms is

$$S = \sum_{i=1}^h l_i \quad (2.7)$$

Grassmann metric is applied to measure the similarities between the candidates. The similarity between every two candidates will be measured. The two transforms with

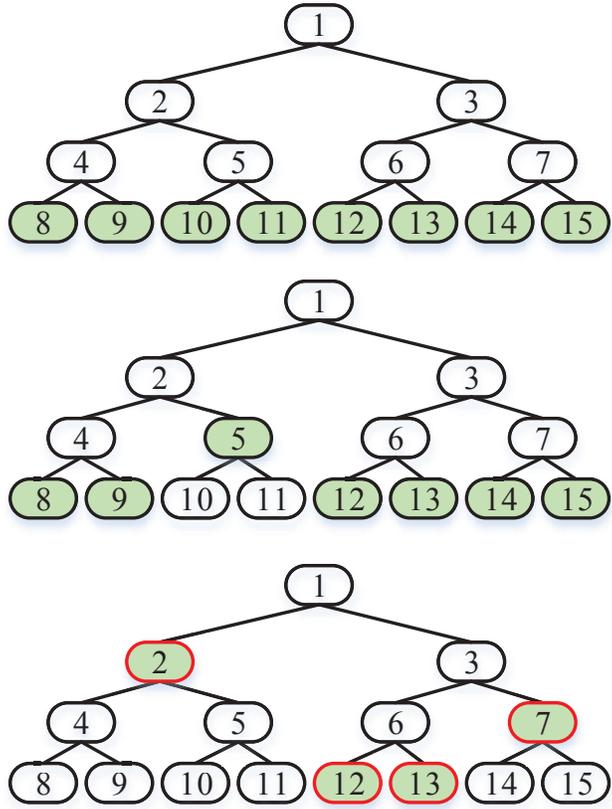


Figure 5: Illustration of Grassmann pruning process.

the shortest Grassmann distance indicates they are the most similar, thus should be merged according to the principle of maximizing distinctiveness. Let us denote all the S available transforms as $\{A_1, \dots, A_S\}$, each of which is trained with all the SIFT descriptors in that node. The number of training SIFT descriptors in each node is $W = \{w_1, \dots, w_S\}$.

Before the merge, suppose the query SIFT descriptors are assigned to R leaf nodes. A merge cost \mathcal{L} is calculated before each merge operation. Children nodes of which LCA node with lowest merge cost will be merged. First, we need to find out the

Lowest Common Ancestor (LCA) of any two existing nodes. As we have R nodes currently, if each two share an LCA node, there would be C_R^2 LCA nodes in total. Let us use G represent C_R^2 for simplicity. The LCA of Node i and Node j can be expressed as follows.

$$\tilde{A}_{ij} = LCA(A_i, A_j), \quad i \neq j, \quad \text{and } i, j = 1, \dots, S \quad (2.8)$$

It should be noted that not all LCA node has only two children. For example in Figure 5, $\tilde{A}_{8,9} = Node_4$ which only has two children 8 and 9, while $\tilde{A}_{5,8} = Node_2$ has three children 8, 9, and 5. We denote the number of children each LCA node has as $\{t_1, \dots, t_G\}$. The merge cost of the g -th LCA \tilde{A}_g is calculated as

$$\mathcal{L}_g = \sum_{i=1}^{t_g} w^{(A_i^{(g)})} \times d_{GSM}(\tilde{A}_g, A_i^{(g)}), \quad (g = 1, \dots, G) \quad (2.9)$$

where $A_i^{(g)}$ is the i -th child of \tilde{A}_g , $w^{(A_i^{(g)})}$ is the number of SIFT descriptors in node $A_i^{(g)}$, $d_{GSM}(a, b)$ is the Grassmann distance between transform a and b . All the children of the LCA which has the lowest cost among all the G LCA nodes will be removed and current LCA node will be a new node.

Fig. 5 is an example of SMPT showing the procedure of merge similar transforms to achieve the final most distinctive local transforms. At initial status, there are 8 leaf nodes on a 3-level SMPT. The objective is to search for 4 most representative transforms. The first step is to find the corresponding LCAs according to Eq. 2.8. Then calculate all the merge cost according to Eq. 2.9. Finally, find the LCA with minimum merge cost and all the children corresponding to that LCA will be merged. This process iterates until the

Table 2: Comparison of with and without pseudo-inverse projection

| test # | preserved dimensionality kd | | | | | | | |
|--------|-------------------------------|------|------|------|-------------------|------|------|------|
| | w/o pseudo-inverse | | | | w/ pseudo-inverse | | | |
| | 4 | 8 | 16 | 32 | 4 | 8 | 16 | 32 |
| 1 | 0.23 | 0.54 | 0.69 | 0.73 | 0.41 | 0.71 | 0.75 | 0.76 |
| 2 | 0.27 | 0.57 | 0.65 | 0.74 | 0.39 | 0.65 | 0.72 | 0.80 |
| 3 | 0.19 | 0.49 | 0.62 | 0.72 | 0.41 | 0.63 | 0.73 | 0.77 |
| 4 | 0.28 | 0.47 | 0.59 | 0.75 | 0.38 | 0.69 | 0.74 | 0.78 |
| 5 | 0.32 | 0.51 | 0.63 | 0.74 | 0.42 | 0.65 | 0.77 | 0.75 |
| 6 | 0.24 | 0.48 | 0.64 | 0.76 | 0.37 | 0.75 | 0.73 | 0.76 |
| 7 | 0.26 | 0.52 | 0.66 | 0.70 | 0.39 | 0.68 | 0.70 | 0.74 |
| 8 | 0.28 | 0.51 | 0.62 | 0.73 | 0.39 | 0.68 | 0.73 | 0.76 |
| 9 | 0.30 | 0.55 | 0.70 | 0.75 | 0.40 | 0.65 | 0.75 | 0.79 |
| 10 | 0.29 | 0.54 | 0.67 | 0.71 | 0.40 | 0.74 | 0.73 | 0.78 |
| Avg. | 0.27 | 0.52 | 0.65 | 0.73 | 0.40 | 0.68 | 0.74 | 0.77 |

number of remaining nodes is 4. According the merge cost, 10 and 11 are merged to 5; 14 and 15 are merged to 7; 8, 9 and 5 are merged to 2. By incorporating Grassmann metric, the most distinctive transformations can be achieved.

2.4.3 Projection and Matching

Given N query SIFT features $\mathcal{F} = \{f_1, \dots, f_N\}$ and K optimal local transforms $\{A_1, \dots, A_K\}$, where $f_n \in \mathbb{R}^{128}$ is a SIFT descriptor. Each query feature in \mathcal{F} is assigned to one of the K nodes. Suppose feature f_n is assigned to transform A_k , the projection is applied as follows.

$$f_n^{(t)} = (f_n - \mu_k) \times A_k \quad (2.10)$$

where $\mu_k \in \mathbb{R}^{128}$ is the mean of all the training samples in node A_k , $f_n^{(t)}$ is the representation of feature f_n in transform domain.

In transform domain, given a descriptor, its nearest neighbor is searched across the transform domain. The descriptor which has the smallest distance will be marked as its matching pair. Given two projected features $f_{n_1}^{(t)}$ and $f_{n_2}^{(t)}$, different schemes are applied in matching procedure if they are associated with different optimal transforms.

It is not fair if computing the distance between two projected features directly. Because they are in different local spaces, their projections are based on different centers, i.e., centroids are different. Directly computing their distance may exaggerate their real distance. As we do not have the original feature information, so using *pseudo inverse* is an appropriate way to reduce the error introduced in this procedure.

Suppose f_{n_1} is associated with transform A_{k_1} and f_{n_2} is associated with transform A_{k_2} . If they are in the same local space, i.e. $k_1 = k_2$, the distance between these two features is Euclidean distance. If they are in different clusters, i.e. $k_1 \neq k_2$, they have to be converted into a uniform local space via *pseudo inverse*. Empirically, the number of samples in each cluster while training is the factor to determine in which local space to convert to, i.e., the node with more training samples will be selected. Let us use w_{k_1} and w_{k_2} representing the number of training samples used to train A_{k_1} and A_{k_2} , respectively, and $w_{k_1} > w_{k_2}$, the distance is calculated as follows.

$$d_{ij} = \begin{cases} \|f_{n_1}^{(t)} - f_{n_2}^{(t)}\|_{L_2} & \text{if } k_1 = k_2 \\ \|f_{n_1}^{(t)} - f_{n_2}^{(inv)}\|_{L_2} & \text{otherwise} \end{cases} \quad (2.11)$$

$$f_{n_2}^{(inv)} = (f_{n_2}^{(t)} \times pinv(A_{k_2}) + \mu_{k_2} - \mu_{k_1}) \times A_{k_1} \quad (2.12)$$

where *pinv* is *pseudo inverse*.

To validate the pseudo-inverse hypothesis, we randomly select 10k matching SIFT pairs from CDVS dataset and check the matching performance with or without pseudo-inverse. The 10k matching SIFT pairs will be assigned to one of 16 leaf nodes of SMPT. Projection with and without pseudo-inverse will be applied to calculate the distance of each SIFT pair in projection domain. Multiple numbers of preserved dimensions are considered. The top-3 accuracy is used to measure the pairwise matching performance. This process is repeated 10 times and different SIFT pairs are used for each time. The accuracy has been listed in Table 2.

As can be observed in Table 2, the pairwise matching performance of the proposed method with pseudo-inverse is better than that without pseudo-inverse. The superiority in low-dimension cases is more obvious than in high-dimension cases. This might be caused by that in low-dimension circumstances, the distance is more sensitive to noise as more information loss has been induced due to dimensionality reduction.

2.5 Experiments

Extensive tests have been conducted to evaluate the performance of the proposed method. There are three parts in this chapter: 1) Evaluation framework description. 2) Energy compaction validation and analysis. 3) The final results of pairwise matching and image retrieval experiments along with the analysis of computational complexity.

Table 3: Overview of MPEG CDVS dataset.

| Dataset | Category | # images | # matching pairs | # non-matching pairs | # retrieval queries |
|---------|------------------|----------|------------------|----------------------|---------------------|
| 1 | Graphics | 2500 | 3000 | 30000 | 1500 |
| 2 | Museum Paintings | 455 | 364 | 3640 | 364 |
| 3 | Video Frames | 500 | 400 | 4000 | 400 |
| 4 | Buildings | 14935 | 4005 | 48675 | 3499 |
| 5 | Common Objects | 10200 | 2550 | 25500 | 2550 |

2.5.1 Evaluation Framework

The experiments are performed over CDVS [46] dataset which consists 10,115 matching image pairs and 112,175 non-matching image pairs. The dataset contains images of 5 categories: *graphics*, *paintings*, *video frames*, *buildings* and *common objects*. They were captured with a variety of camera phones and under widely varying lighting conditions. A brief summary is shown in Table ???. As stated in Section II, half the number of randomly-sampled images are used for training and the other half for test, including both paired images and non-paired images. The number of images in each category is proportional to the percentage of the number of images in current category to that of the total number of images in the dataset.

Before constructing the SMPT for pairwise matching and image retrieval experiments, we preprocess the data with PCA to reduce the dimensionality. The lower dimensional SIFT features help to produce compact representation. In addition, applying PCA could help to remove noise and redundancy; hence, enhancing the discrimination [52]. A 7-level SMPT is constructed with the number of nodes in each level of

$L = \{2, 4, 8, 16, 32, 64, 128\}$ e.g., the first level has 2 nodes, the second level has 4 nodes, etc. The connection relationship is processed in a bottom-up manner as illustrated in Fig. 3. In each node on the SMPT, a local PCA transform will be achieved using full-dimensionality (128-d) SIFT descriptors in that node thus resulting in a total number of $\sum_i^7 l_i = 254$ local transforms. The Grassmann pruning is applied to obtain the final $K = \{4, 8, 16\}$ optimal local transforms.

The energy compaction validation experiment provides empirical evidence from the information theory perspective. The objective is to demonstrate the necessity of partitioning the whole dataset into small groups in order to obtain local transforms. The True Positive Rate (TPR) at less than 1% False Positive Rate (FPR) is reported in the pairwise matching experiment. In compliance with CDVS anchor, average bits per descriptor is used to describe the bit stream. SIFT feature extraction and selection is performed in CDVS software, resulting in about 300 SIFT descriptors for each image. Euclidean distance is used to measure the distance in the transform domain. The mean Average Precision (mAP) is used to evaluate the image retrieval performance. The results of each subset are reported for both pairwise matching and image retrieval experiments.

2.5.2 Energy Compaction Validation

To study the effects of SMPT, we need to verify the coding efficiency which can be measured by probability distribution histogram. If local transforms are more efficient, the data distribution in transform domain should be more compact, i.e., the probability of a value closer to zero is larger.

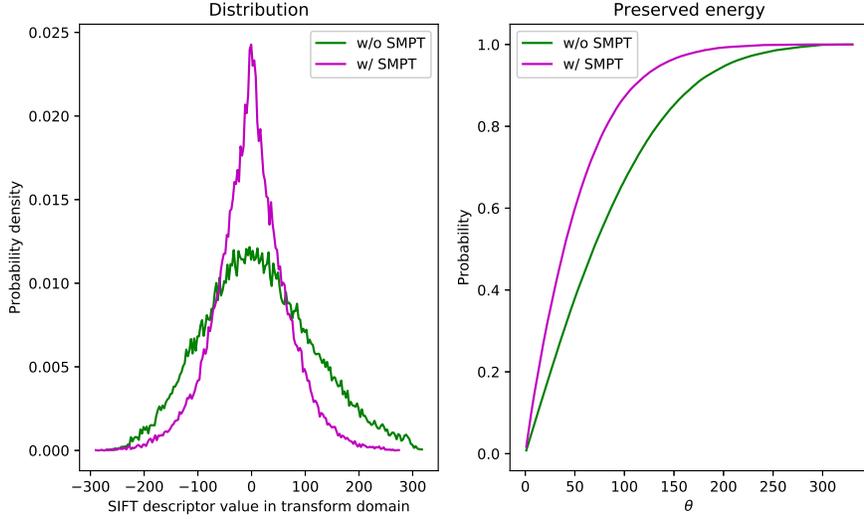


Figure 6: SIFT descriptor energy preservation comparison.

We randomly select 60 images from each subset and extract corresponding SIFT descriptors, thus comprises a total of 90k SIFT descriptors. Training samples are achieved by randomly selected by 70%, and the other 30% is used for the test. Before constructing the SMPT, dimensionality reduction technique (e.g., PCA) is applied to reduce the dimension. The dimensionality-reduced SIFTs are used to build a *5-level* SMPT with the number of nodes in each level $L = \{4, 8, 16, 32, 64\}$. A PCA local transform is trained using all the full-dimensionality (128-d) SIFT descriptors in each node. To remove redundant local transforms, they will be pruned on the Grassmann manifold to achieve final 8 local transforms from $\sum_{i=1}^5 l_i = 124$ available candidates.

After the SMPT model is obtained, the test SIFT descriptors are assigned to one of the 8 optimal nodes according to the Euclidean distance. The test SIFT descriptors in each node will be projected separately using the associated local transform.

Table 4: Comparison by using different Grassmann metrics

| CDVS | | Proposed Projection | | Proposed Binet-Cauchy | |
|---------|---------------|---------------------|---------------|-----------------------|---------------|
| bitrate | repeatability | bitrate | repeatability | bitrate | repeatability |
| 32 | 51.00% | 29 | 52.37% | 26 | 47.48% |
| - | - | 58 | 68.25% | 56 | 67.92% |
| 65 | 67.83% | 84 | 72.99% | 82 | 73.72% |
| - | - | 105 | 76.09% | 108 | 76.68% |
| 103 | 72.70% | 137 | 78.32% | 134 | 78.20% |
| - | - | 159 | 79.27% | 152 | 79.54% |
| 129 | 74.14% | 182 | 80.03% | 176 | 79.54% |
| - | - | 202 | 80.44% | 199 | 80.76% |
| 205 | 76.13% | 228 | 80.76% | 222 | 80.92% |
| - | - | 248 | 80.93% | 250 | 81.17% |

Fig. 6 shows the probability distribution and preserved energy in the transform domain. It is observed that with SMPT more data value aggregates around zero which is definitely beneficial for compression. The preserved energy is calculated by cumulatively summing the probability within range τ away from the origin. When we set the $\tau = 200$, 4.21% more information will be preserved with SMPT than without SMPT.

2.5.3 Experimental Results

To compare the difference of Grassmann projection distance and Grassmann Binet-Cauchy distance, repeatability experiments are conducted using SIFT descriptors from all 5 categories. Table 4 shows the repeatability results of CDVS, proposed method with Grassmann projection distance and the proposed method with Binet-Cauchy distance, respectively. The bitrate variation of the proposed method is achieved by adjusting the number of the preserved dimensionality of SIFT descriptors. It is observed that the proposed

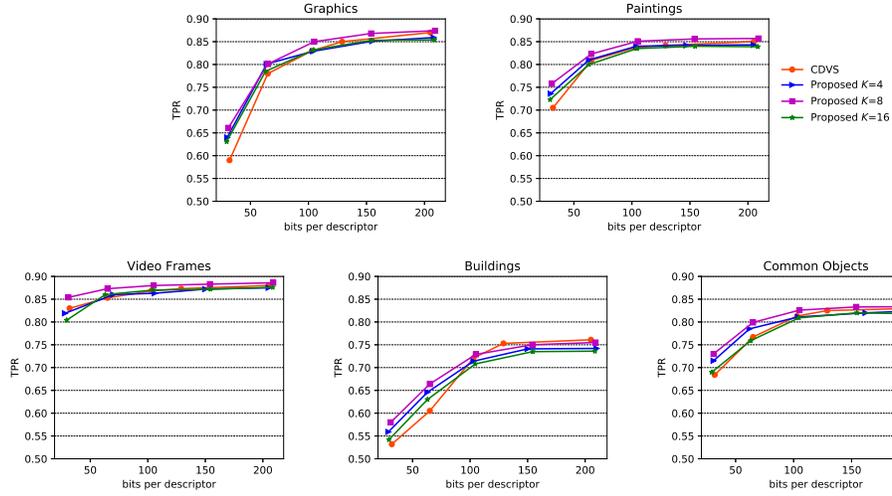


Figure 7: Results of SIFT pair-wise matching.

methods perform better than CDVS. Compared with the best repeatability of CDVS, the proposed SMPT with Grassmann projection and Binet-Cauchy achieves about 4.31% and 4.63% improvement with comparable bitrate, respectively. No significant difference has been observed between two Grassmann distance metrics but Binet-Cauchy is slightly better than projection.

Binet-Cauchy is adopted for pairwise matching and image retrieval experiments. Figure 7 and Figure 8 show the pairwise matching and image retrieval results for each subset, respectively. It can be observed that the proposed method achieves the best performance when $K = 8$, where K is the number of optimal transforms after Grassmann pruning. The performance increases when K decreases from 16 to 8, but deteriorates when continue decreasing from 8 to 4. This phenomenon demonstrates that there exists an optimal solution by adjusting the number of transforms. It is a trade-off between the

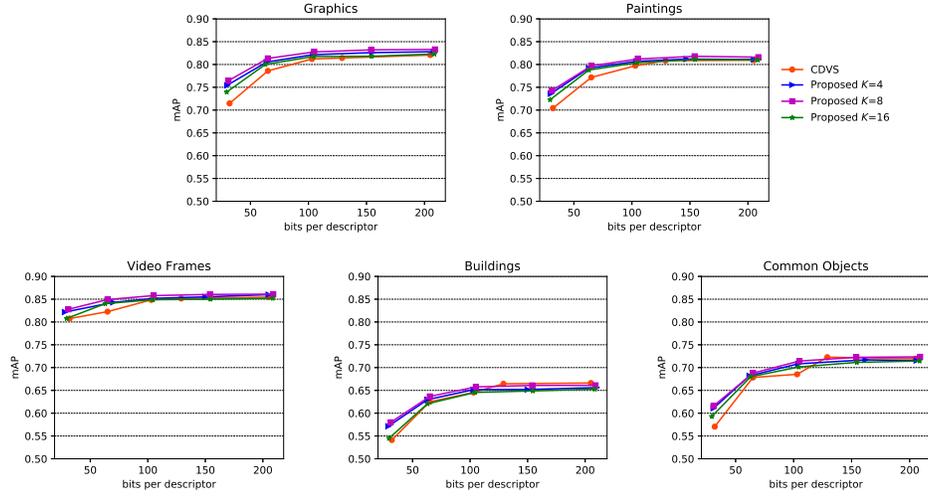


Figure 8: Results of image retrieval.

number of training samples to train a transform and the number of transforms utilized to perform projection.

At best performance, it can be seen that the proposed methods perform better than CDVS with only a few exceptions in *buildings* and *Common Objects*. As the proposed method controls bitrate by adjusting the reduced feature dimensionality, it provides more flexibility in practice. In pairwise matching experiments, the proposed method achieves an improvement of 7.1%, 5.3%, 2.4%, 4.8% and 4.6% at lowest bitrate for *Graphics*, *Paintings*, *Video Frames*, *Buildings* and *Common Objects*, respectively. At highest bitrate, an improvement of 0.4%, 0.7%, 0.6% and 0.35% has been observed for *Graphics*, *Paintings*, *Video Frames* and *Common Objects*, respectively. While in *Buildings*, the proposed method is 0.6% worse than CDVS.

Similar patterns can be witnessed in image retrieval results. The improvement at

lowest bitrate are 4.98%, 3.84%, 2.03%, 3.87% and 4.54% for *Graphics*, *Paintings*, *Video Frames*, *Buildings* and *Common Objects*, respectively. At highest bitrate, the proposed method achieves 1.23%, 0.66%, 0.65% and 0.40% improvement for *Graphics*, *Video Frames*, *Buildings* and *Common Objects*, respectively. A tiny drop of 0.51% exists in *Buildings*. *Buildings* and *Common Objects* contain more images and the content vary more substantially in illumination and deformation and contain more undistinguishable distractors than the other three categories. That might be the reason causing relatively lower performance in these two categories.

The experiments are conducted on a Windows PC with Intel Core CPU i7-7700HQ 2.80GHz. The proposed method requires an average of 1.37 milliseconds per query in pairwise matching experiments for highest performance. In image retrieval experiments, an average of 2.49 seconds is required per query at highest performance. The CDVS has been tested using the same dataset and achieves an average of 0.43 milliseconds and 1.84 seconds per query for pairwise matching and image retrieval experiments, respectively. Future work will focus on reducing the computational complexity.

CHAPTER 3

FAST TRANSFORM FOR VVC

The Joint Video Exploration Team (JVET) recently launched the standardization of the next-generation video coding named Versatile Video Coding (VVC) with the inherited technical framework from its predecessor High-Efficiency Video Coding (HEVC). The simplified Enhanced Multiple Transform (EMT) has been adopted as the primary residual coding transform solution, termed Multiple Transform Selection (MTS). In MTS, only the transform set consisting of DST-VII and DCT-VIII remains, excluding the other transform sets and the dependency on intra prediction modes. Significant coding gains are achieved by introducing new DST/DCT transforms, but the full matrix implementation is relatively costly compared to partial butterfly in terms of both software run-time and operation counts.

In this chapter, we exploit the inherent features existing in DST-VII and DCT-VIII. Instead of repeating the element-wise additions and multiplications in full matrix operation, these features can be leveraged to achieve more efficient implementations which only use partial elements to derive the identical results. Existing transform matrices are further tuned to utilize these (anti-)symmetric features. A partial butterfly-type fast algorithm with dual-implementation support is proposed for DST-VII/DCT-VIII transform in VVC. Complexity analysis including operation counts and software run-time are conducted to validate the effectiveness. In addition, we prove the features are perfectly supported by

theory.

This chapter is organized as follows. Section 3.1 reviews the history of Multiple Transform Selection (MTS). Section 3.2 introduces the sinusoidal transforms involved in this paper and reveals corresponding characteristics. Section 3.3 elaborates the technical details of the proposed fast method. We prove the features in Section 3.4. Section 3.5 performs the complexity analysis, in terms of both the number of arithmetic operations and software execution time. Section 3.6 shows the experimental results.

3.1 History of MTS

3.1.1 Versatile Video Coding

Although deep learning-based methods for image/video coding [56–61] have achieved remarkable progress, there are still many issues need to be solved to be widely used in real applications. Conventional codecs are still playing an indispensable role in industrial application scenarios. Since October 2015, ISO/IEC MPEG and ITU-T VCEG have been working together as the Joint Video Exploration Team (JVET) to explore the state-of-the-art techniques and prepare for the next-generation video coding standards [62] with capability beyond HEVC, termed Versatile Video Coding (VVC). VVC inherits the block-based hybrid video coding framework from its predecessors H.265/HEVC [63] and H.264/AVC [64], but introduces new block partitioning schemes. It supports up to 128×128 Coding Tree Units (CTU) with recursive quadtree (QT) and nested recursive multi-type tree (MTT) partitioning. Various techniques have been incorporated to improve the compression efficiency in the under-development VVC Test Model (VTM). The

primary intra prediction tools include up to 65 prediction angles, prediction filtering using neighboring reference samples and cross-component linear model (CCLM) prediction (Y to Cb, Cr). The primary inter prediction tools include affine motion compensation (4×4 subblocks), improved temporal merge motion vector (MV) predictors (8×8 subblocks) and Adaptive Motion Vector Resolution switches between 1/4, 1, 4 sample accuracy for MVD.

In hybrid video coding frameworks, two techniques of crucial importance to achieve efficient compression efficiency are prediction and residual transform coding. The prediction process focuses on removing the statistical redundancy between the current block and the reference block and transform coding deals with the inter-pixel correlations which is typically done with linear transforms. A variety of transform schemes have been developed in the literature among which the DCT type II (DCT-II) [65] becomes the most popular solution due to its superior capability of balancing the coding efficiency and the time complexity [66].

3.1.2 VVC Transform Solution

In regards to energy compaction performance, it is theoretically proven that DCT-II can efficiently approximate the optimal signal-dependent Karhunen-Loève transform (KLT) [67–70] under the first-order stationary Markov assumption. However, the drastic dynamics existing in natural image content are not always following the first-order Markov condition [71].

To better capture the dynamic characteristics of image data content, numerous of

transform schemes [72–80] have been proposed in the past decades. But those methods suffer from either limited coding efficiency or impractical complexity that hinders the application on video coding codecs. The noteworthy milestone comes when the Enhanced Multiple Transform (EMT), *i.e.*, Adaptive Multiple Transform [67] is proposed. In EMT, four additional transforms including DST-VII, DST-I, DCT-V, and DCT-VIII are introduced. It is reported to achieve -3.1% BD-Rate reduction for AI configuration, and up to -3.6% and -4.0% BD-Rate reduction on 2K and 4K content, respectively [66] which makes it one of the rarest techniques that can achieve more than 2% coding gains since HEVC.

EMT serves as the foundation and prototype of developing the VVC transform solutions due to its superior capability. Since there are five kinds of transforms need to be evaluated to select the optimal category, the encoder run-time complexity is very expensive. In the recent VVC working draft [81], the simplified EMT, named Multiple Transform Selection (MTS) is adopted as the primary residual transform solution after comprehensive consideration about relative merits. In MTS, only the transform set consisting of DST-VII and DCT-VIII remains, excluding the other transform sets and the dependency on intra prediction modes [81, 82].

In VVC, there are two types of MTS, including explicit MTS (with signaling) and implicit MTS (without signaling). The implicit MTS applies DST-VII/DCT-VIII based on block size information (small blocks always use DST-VII for intra), and there is no transform type signaling. The switching between explicit and implicit MTS is done by high-level syntax flags, such that the encoder could choose either explicit MTS or implicit

Table 5: Basis functions of DCT-II, DST-VII and DCT-VIII for N -point input.

| Transform Type | Basis Function $T_i(j)$, $i, j = 0, 1, \dots, N - 1$ |
|----------------|--|
| DCT-II | $T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right),$ $\text{where } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$ |
| DST-VII | $T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$ |
| DCT-VIII | $T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$ |

MTS, based on the different trade-off between coding performance and encoder complexity. More detailed design regarding implicit MTS can be found in [83]. It should be noted that our fast method applies to both explicit MTS and implicit MTS, as long as DST-VII or DCT-VIII is used.

3.2 Discrete Sinusoidal Transform Family

The discrete sinusoidal transform family [84] covers the well-known discrete Fourier transform, cosine transform, sine transform and the Karhunen-Loève transform. Among all the members, there are eight kinds of transforms based on cosine functions and another eight kinds of transforms based on sine functions, namely DCT-I, DCT-II, ..., DCT-VIII and DST-I, DST-II, ..., DST-VIII, respectively. Variants of discrete cosine and sine transforms are derived from different symmetry of their symmetric-periodic sequences [85]. The transform basis functions of selected types of DCT and DST as used in this paper, *i.e.*, DCT-II, DST-VII, and DCT-VIII are formulated in Table 5.

To better understand the sinusoidal families involved in this paper, the first four

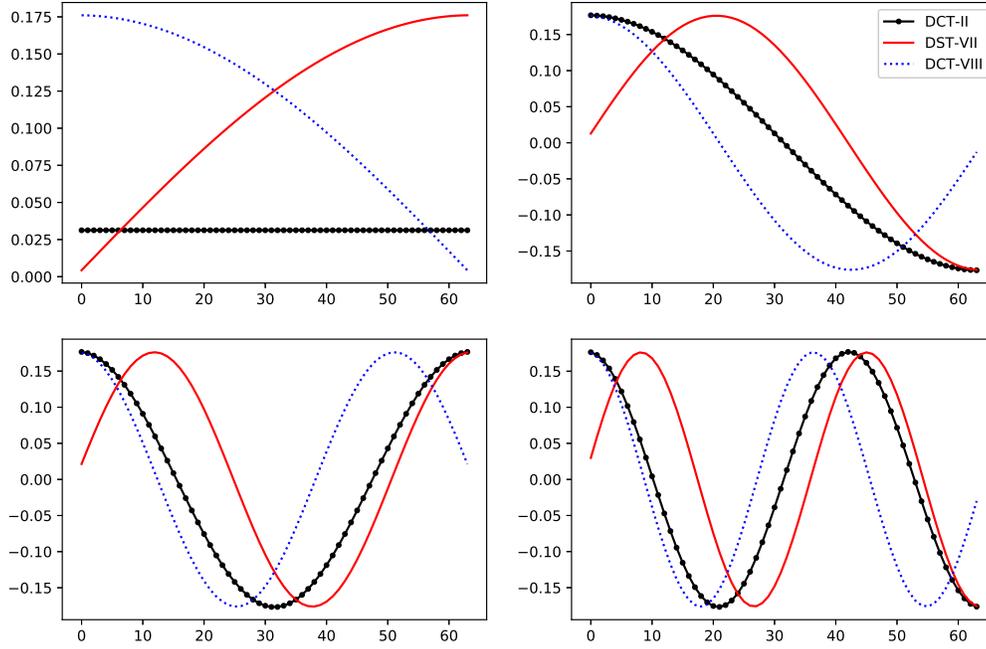


Figure 9: The first four basis plots of DCT-II, DST-VII and DCT-VIII.

most principle basis functions of DCT-II, DST-VII, and DCT-VIII are visualized in Figure 9 for a 64-point input, *i.e.*, $N = 64$. It can be seen that the most principal DCT-II basis, *i.e.*, T_0 shows a constant magnitude distribution, while the DST-VII and DCT-VIII characterize a gradually increasing and gradually decreasing magnitude distribution of the data samples, respectively. Across the plots, we can observe that non-overlapping distributions can be characterized by DST-VII and DCT-VIII under different phases and periods.

The characteristics of the transform basis functions intuitively reveal that the advantages by applying DST-VII/DCT-VIII for intra prediction residuals along the intra

prediction direction since the residual magnitude generally increases along the intra prediction directions. As is observed in Figure 9, inter-prediction residual generally shows a larger residual magnitude for residues closer to PU boundary, thus joint utilization of DST-VII and DCT-VIII would be beneficial to better de-correlate the residual blocks.

3.3 Fast Multiple Transform Selection

In this section, the proposed fast method for DST-VII and DCT-VIII is presented in detail. First, the proposed fast transform scheme is described, including the (anti-)symmetric properties which are considered beneficial for reducing arithmetic operations. Second, an example is provided to showcase how these features are leveraged to devise the fast algorithm. Finally, the transform matrix tuning is presented to compensate for the deviation induced by the rounding error. Multiple measures have been taken to achieve the orthogonality and efficiency of the tuned transform matrices.

In the block-based hybrid video coding scheme, linear transforms are typically applied to the residual blocks obtained from inter- or intra-frame prediction. VVC supports up to 64×64 transform block and introduces the non-square transform block partition scheme. To achieve efficient implementation, existing HEVC/VVC reference software deploys a two-dimensional transform as a consecutive combination of two one-dimensional transforms with each for horizontal and vertical, respectively. The resulting coefficients are further processed by quantization and entropy coding. Typically, the forward and inverse transform matrices are transposed matrix of each other.

In the remaining sections, we present the proposed method by using DST-VII

forward transform as an example unless otherwise specified. Unless otherwise stated, description to the algorithm is based on the 8-bit transform matrix representation. The proposed scheme is applicable to both 8-bit and 10-bit transform matrix representation. This paper only focuses on 16-, 32- and 64-point transform sizes for the following reasons: 1) There is already a similar fast algorithm implemented for 4-point transform. 2) No similar (anti-)symmetric features are observed to the best of our knowledge. 3) The benefits are quite limited even 4-point and 8-point fast implementation is achieved due to their relative small transform sizes. It should be noted that the inverse transforms are the transposed matrices hence the similar deployment can be realized. In addition, the same philosophy can be seamlessly migrated to DCT-VIII.

3.3.1 Fast Transform

In the following chapters, unless other stated, we denote the residual coefficients vector $\{x_0, \dots, x_{15}\}$ as *input vector*, the transformed output vector $\{y_0, \dots, y_{15}\}$ as *output vector*, the transform matrix derived from Table 5 as *transform matrix*, a vector from the transform matrix as *transform vector*, each element in the transform matrix as *element*. The first feature that is noteworthy to mention is that there exist only N distinct possible values except for 0 in an N -point transform matrix. Typically, the first basis vector contains all the values while others only contain partial values with or without sign changes and a possible 0. We denote the transform vector which contains N unique values (usually the first transform vector) as *basis transform vector*, and each element in the basis transform vector as a *member*.

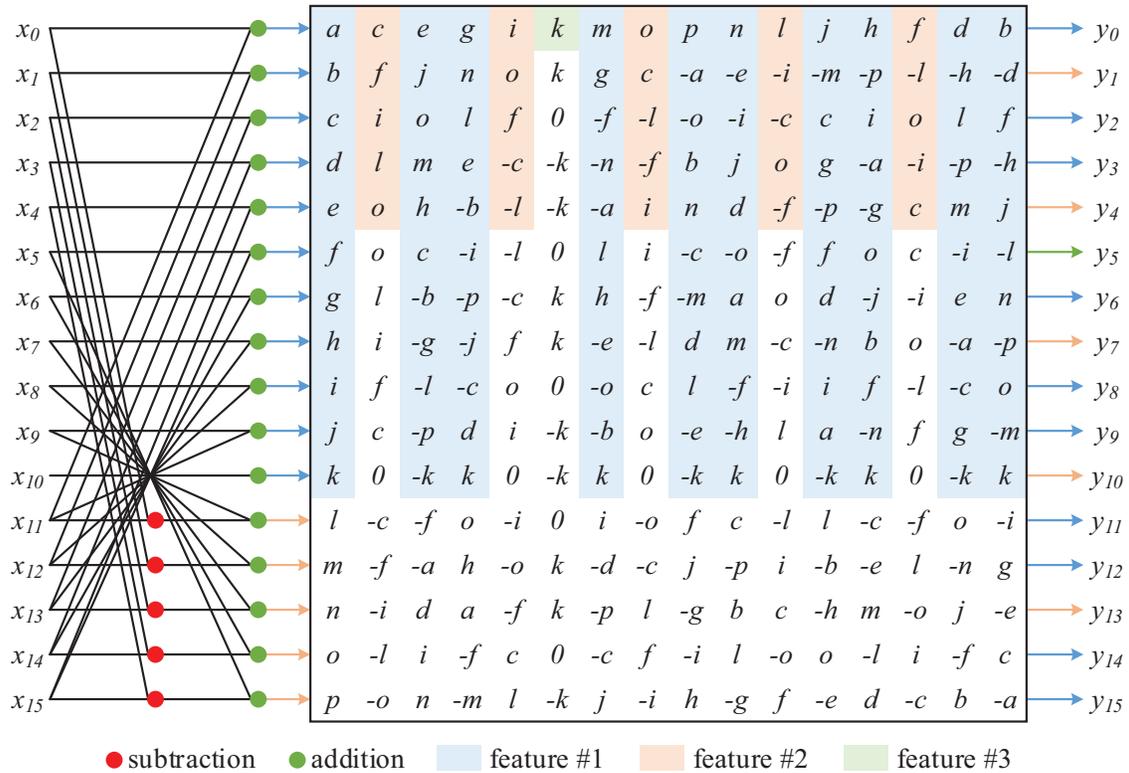


Figure 10: Fast 16-point DST-VII forward transform algorithm.

For example, we use $\{a, b, \dots, p\}$ to represent the *basis transform vector* of a 16-point DST-VII transform matrix that is derived from the equations defined in Table 5. To make it simple, we use T to represent the whole transform matrix. It should be understood that the actual values might be different for a transform matrix of different sizes. For instance, a might be a different number in a 32-point transform matrix as it is in a 16-point transform matrix. However, it is assured that the same *member* represents the identical number within a transform matrix. Similar to the preprocessing applied on DCT-II in HEVC, the transform matrix elements in VVC are also scaled by a scale factor *e.g.*, $64 \cdot \sqrt{N}$ or $256 \cdot \sqrt{N}$, and then rounded to the closest integer. Further tuning by an

offset might also be applied in the application.

Figure 10 sketches the framework of the proposed fast method on a 16-point DST-VII forward transform. The input vector $\{x_0, \dots, x_{15}\}$ multiplies the transform matrix T to obtain the output vector $\{y_0, \dots, y_{15}\}$. The filled red circle and green circle represent the subtraction and addition operation, respectively. In the conventional full matrix multiplication, the transformed results are calculated by repeating multiplying each input with the transform matrix element and adding them together. By leveraging the innate partial butterfly-type features, the proposed fast method accelerates this process through the simplification process and the intermediate results re-use mechanism. The proposed scheme supports both direct matrix multiplication and partial butterfly-type fast method. Parallel processing is supported when selecting direct matrix multiplication.

As shown with different colors in Figure 10, the DST-VII transform matrix consists of three features that are considered useful for a more efficient implementation. These features are summarized as follows. It should be noted that these features are non-overlapping, *i.e.*, only one feature is applicable for a given transform vector.

1. **Feature #1:** N members are included without considering the sign changes. These elements can be grouped into several groups with a fixed number of elements. An equation exists by manipulating additions in each group.
2. **Feature #2:** Only a subset of the N members are included without considering the sign changes. They can be divided into several groups with a fixed number of consecutive elements such that every two consecutive groups are spatially symmetric or anti-symmetric, *i.e.*, symmetric by applying a negative sign.

3. **Feature #3:** Except for zero, some transform vector(s) only contain(s) a single *member* when neglecting the sign changes.

As mentioned in Feature #1, there exists an equation in each group. We observe that the relationships can be expressed using the following equations, *i.e.*, three elements form a group and five groups are derived in each two added elements equals to another element.

$$\begin{aligned}
 a + j &= l \\
 b + i &= m \\
 c + h &= n \\
 d + g &= o \\
 e + f &= p
 \end{aligned} \tag{3.1}$$

Take deriving y_0 using the first transform vector (colored in Ceruleanblue) as an example, conventional matrix multiplication would directly calculate the following equation which requires 16 multiplications and 15 additions.

$$\begin{aligned}
 y_0 &= a \cdot x_0 + b \cdot x_1 + c \cdot x_2 + d \cdot x_3 + e \cdot x_4 + f \cdot x_5 \\
 &+ g \cdot x_6 + h \cdot x_7 + i \cdot x_8 + j \cdot x_9 + k \cdot x_{10} \\
 &+ l \cdot x_{11} + m \cdot x_{12} + n \cdot x_{13} + o \cdot x_{14} + p \cdot x_{15}
 \end{aligned} \tag{3.2}$$

Benefit from Feature #1 (3.1), when calculating $(a \cdot x_0 + j \cdot x_9 + l \cdot x_{11})$ which requires three multiplications and two additions, we can calculate its equivalent form $a \cdot (x_0 + x_{11}) + j \cdot (x_9 + x_{11})$ by combing the common items thereby only two multiplications are needed. Although the addition operations are increased but some intermediate results

can be re-used to eliminate the additional cost. We will introduce this in the following chapters. Similar simplification can be applied to $(b \cdot x_1 + i \cdot x_8 + m \cdot x_{12})$, $(c \cdot x_2 + h \cdot x_7 + n \cdot x_{13})$, $(d \cdot x_3 + g \cdot x_6 + o \cdot x_{14})$ and $(e \cdot x_4 + f \cdot x_5 + p \cdot x_{15})$. Therefore, to calculate y_0 , instead of doing the element-wise multiplication which requires 16 multiplications and 15 additions, the following equivalent formulation can be utilized to derive the identical results.

$$\begin{aligned}
y_0 = & a \cdot (x_0 + x_{11}) + b \cdot (x_1 + x_{12}) + c \cdot (x_2 + x_{13}) \\
& + d \cdot (x_3 + x_{14}) + e \cdot (x_4 + x_{15}) + f \cdot (x_5 + x_{15}) \\
& + g \cdot (x_6 + x_{14}) + h \cdot (x_7 + x_{13}) + i \cdot (x_8 + x_{12}) \\
& + j \cdot (x_9 + x_{11}) + k \cdot x_{10}
\end{aligned} \tag{3.3}$$

which requires 11 multiplications and 20 additions. So we can save 5 multiplications but need 5 additional additions. Typically, it is faster performing addition instruction than multiplication instruction on modern CPUs. Thus time saving can be achieved by the simplification process. In addition, when calculating $y_2, y_3, y_6, y_8, y_9, y_{11}, y_{12}, y_{14}, y_{15}$, similar simplification can be achieved and what really matters is the intermediate results of $(x_0 + x_{11})$, $(x_1 + x_{12})$, $(x_2 + x_{13})$, $(x_3 + x_{14})$, $(x_4 + x_{15})$, $(x_5 + x_{15})$, $(x_6 + x_{14})$, $(x_7 + x_{13})$, $(x_8 + x_{12})$, $(x_9 + x_{11})$ and $k \cdot x_{10}$ can be re-used. This could save a lot more on top of that. In summary, the time complexity reduction from Feature #1 comes from the simplification process and the intermediate results re-use mechanism.

We take the second transform vector (colored in Apricotorange) from T to show-case how Feature #2 is leveraged to achieve a better implementation.

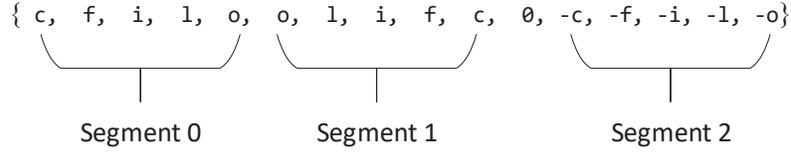


Figure 11: Feature #2 shows a mirror-(anti)symmetric pattern.

As shown in Figure 11, the second transform vector can be classified into three segments. Each segment consists of the same five consecutive elements when neglecting the sign changes. They are replicate with sign changes, or flipped version of each other. Therefore, benefit from Feature #2, when computing the transformed result y_1 , instead of doing the following element-wise operation

$$\begin{aligned}
 y_1 = & c \cdot x_0 + f \cdot x_1 + i \cdot x_2 + l \cdot x_3 + o \cdot x_4 \\
 & + o \cdot x_5 + l \cdot x_6 + i \cdot x_7 + f \cdot x_8 + c \cdot x_9 \\
 & - c \cdot x_{11} - f \cdot x_{12} - i \cdot x_{13} - l \cdot x_{14} - o \cdot x_{15}
 \end{aligned} \tag{3.4}$$

which requires 15 multiplications and 14 additions, the following simplified form can be utilized to derive the identical outcome with only 5 multiplications and 14 additions. The reduced number of multiplications can lead to time complexity reduction.

$$\begin{aligned}
 y_1 = & c \cdot (x_0 + x_9 - x_{11}) + f \cdot (x_1 + x_8 - x_{12}) \\
 & + i \cdot (x_2 + x_7 - x_{13}) + l \cdot (x_3 + x_6 - x_{14}) \\
 & + o \cdot (x_4 + x_5 - x_{15})
 \end{aligned} \tag{3.5}$$

On top of that, when deriving $y_1, y_4, y_7, y_{10}, y_{13}$, the intermediate results $(x_0 + x_9 - x_{11}), (x_1 + x_8 - x_{12}), (x_2 + x_7 - x_{13}), (x_3 + x_6 - x_{14})$ and $(x_4 + x_5 - x_{15})$ can be re-used to further reduce the operation counts. Therefore, the benefits from Feature #2 come from both the simplification process and the intermediate results re-use mechanism.

We take the sixth transform vector (colored in LimeGreengreen) to explain how Feature #3 is utilized to reduce the computational complexity. This transform vector has very distinct characteristics, *i.e.*, consists of only a single *member* k when neglecting the sign changes. To calculate y_5 , the conventional element-wise matrix multiplication calculates in the following manner, which requires 11 multiplications and 10 additions.

$$\begin{aligned}
y_5 = & k \cdot x_0 + k \cdot x_1 - k \cdot x_3 - k \cdot x_4 + k \cdot x_6 + k \cdot x_7 \\
& - k \cdot x_9 - k \cdot x_{10} + k \cdot x_{12} + k \cdot x_{13} - k \cdot x_{15}
\end{aligned} \tag{3.6}$$

Benefit from Feature #3, we can alternatively derive the identical results through the simplified formulation (3.7) by combining the common terms which requires only 1 multiplication and 10 additions. Therefore, the time reduction achieved from Feature #3 comes from the simplification process.

$$\begin{aligned}
y_5 = & k \cdot (x_0 - x_2 + x_3 - x_5 + x_6 - x_8 \\
& + x_9 - x_{11} + x_{12} - x_{14} + x_{15})
\end{aligned} \tag{3.7}$$

It should be noted that in a given transform matrix, the combination pattern is fixed to implement Feature #1, *i.e.*, the pattern of addition of two elements equals to another element only exists in the 16-point or 64-point transform matrices, the pattern of addition of three elements equals to addition of another two elements only exists in a 32-point transform matrix. We summarize the applicable transform vectors for the mentioned three features of 16, 32, and 64-point DST-VII transform matrices as follows.

- 16-point transform

- Feature #1: T_{3n} and T_{2+3m} , where $n = 0, \dots, 5$, $m = 2, 3, 4$

- Feature #2: T_{1+3n} , where $n = 0, \dots, 4$
- Feature #3: T_5
- 32-point transform
 - Feature #1: T_5 and T_{8+m+5n} , where $m = 0, \dots, 3$, $n = 0, \dots, 4$ but $n \neq 2$ when $m = 1$
 - Feature #2: T_{2+5n} , where $n = 0, \dots, 5$
 - Feature #3: T_6 and T_{19}
- 64-point transform
 - Feature #1: T_{2+3n} and T_{3m} , where $n = 0, \dots, 20$, $m = 0, \dots, 21$ but $m \neq 7$
 - Feature #2: T_{1+3n} , where $n = 0, \dots, 20$
 - Feature #3: T_{21}

For the DST-VII inverse transform, the transform matrix is the transposed version of the forward transform matrix thereby similar deployment can be achieved. The above-mentioned features also exist in 32-point and 64-point transform matrices with a slight difference of how the groups are identified and the number of elements in each group. Therefore, a similar implementation can be applied to 32-point and 64-point transform matrices.

Since DST-VII and DCT-VIII share the same implementation logic, we omit full description to DCT-VIII to avoid redundancy. It would be easier to understand by examining the basis transform vector of DCT-VIII. The basis transform vector of the 16-point

Table 6: Feature #1 applicable equation groups of 16-point DST-VII.

| Output | Equation Groups | | | | |
|----------|-----------------|---------------|---------------|---------------|---------------|
| y_0 | $a + j = l$ | $b + i = m$ | $c + h = n$ | $d + g = o$ | $e + f = p$ |
| y_2 | $e - p = -f$ | $j - l = -a$ | $o - g = d$ | $m - b = i$ | $h + c = n$ |
| y_3 | $g + d = o$ | $n - c = h$ | $l - j = a$ | $e - p = -f$ | $-b - i = -m$ |
| y_6 | $m - b = i$ | $g - o = -d$ | $-f - e = -p$ | $-n + h = -c$ | $-a + l = j$ |
| y_8 | $p - e = f$ | $-a + l = j$ | $-o + d = -g$ | $b - m = -i$ | $n - c = h$ |
| y_9 | $n - h = c$ | $-e - f = -p$ | $-i + m = b$ | $j + a = l$ | $d - o = -g$ |
| y_{11} | $j + a = l$ | $-m + i = -b$ | $c - n = -h$ | $g + d = o$ | $-p + f = -e$ |
| y_{12} | $h - n = -c$ | $-p + f = -e$ | $i + b = m$ | $-a - j = -l$ | $-g + o = d$ |
| y_{14} | $d + g = o$ | $-h - c = -n$ | $l - a = j$ | $-p + e = -f$ | $m - i = b$ |
| y_{15} | $b - m = -i$ | $-d + o = g$ | $f - p = -e$ | $-h + n = c$ | $j - l = -a$ |

DST-VII is

$$T_0^{DST-VII} = \{8, 17, 25, 33, 40, 48, 55, 62, 68, 73, 77, 81, 85, 87, 88, 88\} \quad (3.8)$$

and the basis transform vector of the 16-point DCT-VIII is

$$T_0^{DCT-VIII} = \{88, 88, 87, 85, 81, 77, 73, 68, 62, 55, 48, 40, 33, 25, 17, 8\} \quad (3.9)$$

To leverage the proposed rules, the fast method Feature #1 for 16-point DCT-III can be formulated as

$$T_0(15 - j) + T_0(6 + j) = T_0(4 - j), j = 0, \dots, 4 \quad (3.10)$$

which corresponds to (3.16) of the 16-point DST-VII formulation.

3.3.2 Transform Matrix Tuning

Similar to HEVC, VVC implements the transform matrices in a finite-precision approximation in the reference software. Using integer operations is friendly to hardware implementation. In addition, it might also avoid potential mismatch caused by the platforms from different manufactures. One of the drawbacks is the finite-precision representation is not as accurate as the floating-point representation. This might lead to inferior coding efficiency. Another disadvantage is that the useful features as mentioned in Section ?? are not valid in the rounded transform matrices.

To better explain this point, we take the 32-point DST-VII forward transform matrix as an example. We use $\{a, b, \dots, z, A, B, \dots, F\}$ to denote the *basis transform vector* which is also the first transform vector. In floating-point 32-point DST-VII forward transform matrix, Feature #1 can be expressed with the following equations.

$$\begin{aligned} a + l + A &= n + y \\ b + k + B &= o + x \\ c + j + C &= p + w \\ d + i + D &= q + v \\ e + h + E &= r + u \\ f + g + F &= s + t \end{aligned} \tag{3.11}$$

In VVC test model, the transform matrix elements are multiplied and scaled to the

closest integer resulting in the following actual values.

$$\begin{aligned} \{a, b, \dots, F\} = \{4, 9, 13, 17, 21, 26, 30, 34, 38, 42, \\ 46, 49, 53, 56, 60, 63, 66, 69, 71, 74, 76, \\ 78, 81, 82, 84, 85, 87, 88, 89, 89, 90, 90\} \end{aligned} \quad (3.12)$$

Therefore, for quintuple #1 in (3.11), the left side and the right side can be calculated as

$$\begin{aligned} b + k + B &= 9 + 46 + 88 = 143 \\ o + x &= 60 + 82 = 142 \end{aligned} \quad (3.13)$$

which are not equal any more. To bring this rounded transform matrix back to validity for Feature #1, the basis transform vector has to be tuned. In this example, we can either subtract 1 from b , k or B , or add 1 to o or x .

To make the adjusted transform matrices well-adapted in video coding scenario, we define the following principles that should be strictly followed during the tuning process. First, the N -point transform matrix can be represented using N distinct basis transform vector *members* without considering the sign changes. Second, the orthogonality between any two transform vectors should be optimized as much as possible. Finally, the adjusted basis transform vector *members* should be kept as close as possible to the floating-point basis transform vector *members*.

The following metrics are defined to evaluate the quality of the tuned transform matrices, including orthogonality, accuracy and norm measurement.

1. Orthogonality measure: $o_{ij} = \mathbf{d}_i^T \mathbf{d}_j / \mathbf{d}_0^T \mathbf{d}_0, i \neq j$
2. Closeness measure: $m_{ij} = |\alpha c_{ij} - d_{ij}| / d_{00}$

Table 7: Comparison of the tuned transform matrices and the original transform matrices.

| Bit depth | Metric | 16 point | | 32 point | | 64 point | |
|-----------|---------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| | | Tuned | Original | Tuned | Original | Tuned | Original |
| 8-bit | Orthogonality | $o_{ij} \leq 0.0017$ | $o_{ij} \leq 0.0020$ | $o_{ij} \leq 0.0026$ | $o_{ij} \leq 0.0026$ | $o_{ij} \leq 0.0018$ | $o_{ij} \leq 0.0018$ |
| | Closeness | $m_{ij} \leq 0.1910$ | $m_{ij} \leq 0.1509$ | $m_{ij} \leq 0.1956$ | $m_{ij} \leq 0.1956$ | $m_{ij} \leq 0.9037$ | $m_{ij} \leq 0.7902$ |
| | Norm measure | $n_i \leq 0.0047$ | $n_i \leq 0.0066$ | $n_i \leq 0.0039$ | $n_i \leq 0.0045$ | $n_i \leq 0.0039$ | $n_i \leq 0.0045$ |
| 10-bit | Orthogonality | $o_{ij} \leq 0.0009$ | $o_{ij} \leq 0.0010$ | $o_{ij} \leq 0.0005$ | $o_{ij} \leq 0.0007$ | $o_{ij} \leq 0.0010$ | $o_{ij} \leq 0.0010$ |
| | Closeness | $m_{ij} \leq 0.0189$ | $m_{ij} \leq 0.0156$ | $m_{ij} \leq 0.0437$ | $m_{ij} \leq 0.0382$ | $m_{ij} \leq 0.0833$ | $m_{ij} \leq 0.0751$ |
| | Norm measure | $n_i \leq 0.0019$ | $n_i \leq 0.0024$ | $n_i \leq 0.0004$ | $n_i \leq 0.0006$ | $n_i \leq 0.0012$ | $n_i \leq 0.0017$ |

3. Norm measure: $n_i = |\mathbf{1} - \mathbf{d}_i^T \mathbf{d}_i / \mathbf{d}_0^T \mathbf{d}_0|$

Given the original N -point transform matrix element $c_{ij} = T_i(j)$ as defined in Table ??, the scaled approximated transform matrix element of d_{ij} , which constitutes the scaled approximated transform vector $\mathbf{d}_i = [d_{i0}, \dots, d_{i(N-1)}]^T$, where $i = 0, \dots, N - 1$, the global optimization objective is defined as follows.

$$D = |\alpha^2 \cdot I - T \cdot T^T| \quad (3.14)$$

where $i, j = 0, \dots, N - 1$, and the scale factor $\alpha = 64 \cdot \sqrt{N}$. The tuning process is deployed by trying all possible integer values and evaluate the tuned transform matrix using the above-mentioned three measurements and the optimal setting will be adopted.

In addition, the per-element magnitude difference between the tuned transform matrix T and the floating-point transform matrix T_0 is restrained to be no larger than 1. In such a way, the adjusted transform matrices are kept as close as possible to the original floating-point transform matrices to avoid severe performance deviation. The tuned transform matrices in both 8-bit and 10-bit can be found in [86, 87].

The comparison between the tuned transform matrices and the original transform matrices is tabulated in Table 7. The worst value of o_{ij} , m_{ij} and n_i are used to measure the

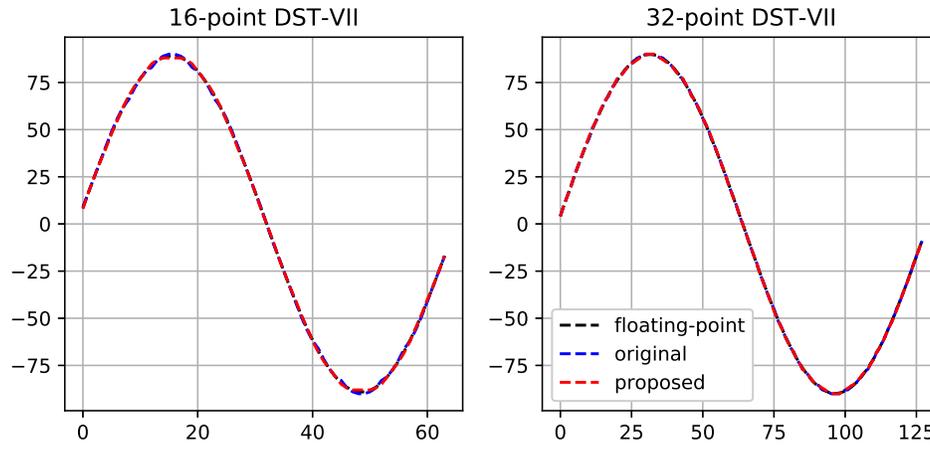


Figure 12: Sinusoidal graphs of the tuned transform basis.

level of approximation. As can be seen from the table, in most cases, although the tuned transform matrices are farther from the orthonormal transform matrices, they provide better orthogonality and norm properties.

The sinusoidal graphs are also provided in Figure 12 to show the closeness of the original transform basis and the proposed transform basis to the floating-point transform basis. Here, the floating-point transform basis is directly derived from the formula listed in Table 5 and multiplied by a scale factor, the original transform basis refers to the existing one in the reference software prior to the proposed scheme was adopted, the proposed transform is the proposed transform basis presented in this paper. We can observe from the graph that, there are only some minor differences on the basis at the peak of wave and the bases look close among the three curves.

3.4 Theoretical Proof

To demonstrate the validity of the proposed features utilized to design the fast methods, we prove them from theory. Since Feature #2 and Feature #3 are very straightforward, we omit the theoretical proof process.

3.4.1 16-point Transform

In the 16-point DST-VII example, Feature #1 (3.1) can be expressed in the form of (3.15). Therefore, Feature #1 can be summarized in a more compact form as in (3.16).

$$\begin{aligned}
 T_0(0) + T_0(9) &= T_0(11) \\
 T_0(1) + T_0(8) &= T_0(12) \\
 T_0(2) + T_0(7) &= T_0(13) \\
 T_0(3) + T_0(6) &= T_0(14) \\
 T_0(4) + T_0(5) &= T_0(15)
 \end{aligned} \tag{3.15}$$

$$T_0(j) + T_0(9 - j) = T_0(11 + j), \quad j = 0, \dots, 4 \tag{3.16}$$

According to the basis function defined in Table ??, they can be re-written in the following format.

$$\begin{aligned}
 T_0(j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(j+1)}{2N+1} \\
 T_0(9-j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(10-j)}{2N+1} \\
 T_0(11+j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(12+j)}{2N+1}
 \end{aligned} \tag{3.17}$$

Therefore, proving (3.16) is equivalent to proving (3.18) by removing the non-zero element $\sqrt{\frac{4}{2N+1}}$.

$$\sin \frac{\pi(j+1)}{2N+1} + \sin \frac{\pi(10-j)}{2N+1} = \sin \frac{\pi(12+j)}{2N+1} \quad (3.18)$$

The proof process is provided as follows.

$$\begin{aligned} & \sin \frac{\pi(12+j)}{2N+1} - \sin \frac{\pi(10-j)}{2N+1} \\ &= 2 \cos \frac{11\pi}{2N+1} \sin \frac{\pi(1+j)}{2N+1} \\ &= 2 \cos \frac{\pi}{3} \sin \frac{\pi(1+j)}{2N+1} \\ &= \sin \frac{\pi(j+1)}{2N+1} \end{aligned} \quad (3.19)$$

Therefore, (3.16) has been proven.

3.4.2 32-point Transform

In the 32-point DST-VII transform matrix, the relationships defined in (3.11) can be expressed in the following form.

$$\begin{aligned} T_0(0) + T_0(11) + T_0(26) &= T_0(13) + T_0(24) \\ T_0(1) + T_0(10) + T_0(27) &= T_0(14) + T_0(23) \\ T_0(2) + T_0(8) + T_0(28) &= T_0(15) + T_0(22) \\ T_0(3) + T_0(8) + T_0(29) &= T_0(16) + T_0(21) \\ T_0(4) + T_0(7) + T_0(30) &= T_0(17) + T_0(20) \\ T_0(5) + T_0(6) + T_0(31) &= T_0(18) + T_0(19) \end{aligned} \quad (3.20)$$

Therefore, we can describe Feature #1 in 32-point transform in a compact manner as defined in (3.21).

$$\begin{aligned}
& T_0(j) + T_0(11 - j) + T_0(26 + j) \\
& = T_0(13 + j) + T_0(24 - j), \quad j = 0, \dots, 5
\end{aligned} \tag{3.21}$$

Based on definitions in Table 5, we have

$$\begin{aligned}
T_0(j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(j+1)}{2N+1} \\
T_0(11-j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(12-j)}{2N+1} \\
T_0(26+j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(27+j)}{2N+1} \\
T_0(13+j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(14+j)}{2N+1} \\
T_0(24-j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(25-j)}{2N+1}
\end{aligned} \tag{3.22}$$

After removing the non-zero element $\sqrt{\frac{4}{2N+1}}$, we can prove the equivalent objective as described in (3.23).

$$\begin{aligned}
& \sin \frac{\pi(j+1)}{2N+1} + \sin \frac{\pi(12-j)}{2N+1} + \sin \frac{\pi(27+j)}{2N+1} \\
& = \sin \frac{\pi(14+j)}{2N+1} + \sin \frac{\pi(25-j)}{2N+1}
\end{aligned} \tag{3.23}$$

On the left side of the equation, when we combine the second and the third term, we achieve the following expression:

$$\begin{aligned}
& \sin \frac{\pi(j+1)}{2N+1} + \sin \frac{\pi(12-j)}{2N+1} + \sin \frac{\pi(27+j)}{2N+1} \\
&= \sin \frac{\pi(j+1)}{2N+1} + 2 \sin \frac{39\pi}{2(2N+1)} \cos \frac{\pi(-15-2j)}{2(2N+1)}
\end{aligned} \tag{3.24}$$

Similar processing can be performed on the right terms, then the right side is transformed to (3.25).

$$\begin{aligned}
& \sin \frac{\pi(14+j)}{2N+1} + \sin \frac{\pi(25-j)}{2N+1} \\
&= 2 \sin \frac{\pi 39}{2(2N+1)} \cos \frac{\pi(-11+2j)}{2(2N+1)}
\end{aligned} \tag{3.25}$$

Equation (3.23) can be re-written as (3.26) by substituting corresponding items with (3.24) and (3.25).

$$\begin{aligned}
& \sin \frac{\pi(j+1)}{2N+1} + 2 \sin \frac{39\pi}{2(2N+1)} \cos \frac{\pi(-15-2j)}{2(2N+1)} \\
&= 2 \sin \frac{39\pi}{2(2N+1)} \cos \frac{\pi(-11+2j)}{2(2N+1)}
\end{aligned} \tag{3.26}$$

After merging similar items, the objective becomes

$$\begin{aligned}
& \sin \frac{\pi(j+1)}{2N+1} \\
&= 2 \sin \frac{39\pi}{2(2N+1)} \left[\cos \frac{\pi(-11+2j)}{2(2N+1)} - \cos \frac{\pi(-15-2j)}{2(2N+1)} \right]
\end{aligned} \tag{3.27}$$

The right-most 2 items can be further simplified as follows.

$$\begin{aligned}
& \cos \frac{\pi(-11+2j)}{2(2N+1)} - \cos \frac{\pi(-15-2j)}{2(2N+1)} \\
&= -2 \sin \frac{-26\pi}{4(2N+1)} \sin \frac{\pi(4+4j)}{4(2N+1)}
\end{aligned} \tag{3.28}$$

Then we substitute corresponding items in (3.27), the objective turns to this form

$$\begin{aligned} & \sin \frac{\pi(j+1)}{2N+1} \\ &= -4 \sin \frac{39\pi}{2(2N+1)} \sin \frac{-26\pi}{4(2N+1)} \sin \frac{\pi(4+4j)}{4(2N+1)} \end{aligned} \quad (3.29)$$

Since $\sin \frac{\pi(j+1)}{2N+1}$ is a non-zero element, we obtain the final simplified objective equation (3.30) which is equivalent to (3.21).

$$\sin \frac{39\pi}{2(2N+1)} \cdot \sin \frac{13\pi}{2(2N+1)} = \frac{1}{4} \quad (3.30)$$

This equation turns to be an identity relation since N is 32. Therefore, the Feature #1 of 32-point DST-VII transform matrix as described in (3.21) has been proved.

3.4.3 64-point Transform

Similar to 16-point and 32-point transform matrices, the 64-point Feature #1 also can be expressed in a compact manner.

$$T_0(j) + T_0(41-j) = T_0(43+j), \quad j = 0, \dots, 20 \quad (3.31)$$

Based on the definitions in Table ??, each item in (3.31) can be expanded to the following form.

$$\begin{aligned} T_0(j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(j+1)}{2N+1} \\ T_0(41-j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(42-j)}{2N+1} \\ T_0(43+j) &= \sqrt{\frac{4}{2N+1}} \cdot \sin \frac{\pi(44+j)}{2N+1} \end{aligned} \quad (3.32)$$

Replacing corresponding items in (3.31) with those in (3.32) and removing the non-zero item $\sqrt{\frac{4}{2N+1}}$, we obtain an equivalent objective equation.

$$\sin \frac{\pi(j+1)}{2N+1} + \sin \frac{\pi(42-j)}{2N+1} = \sin \frac{\pi(44+j)}{2N+1} \quad (3.33)$$

Start from the left side, each term can be expressed as the following form.

$$\sin \frac{\pi(j+1)}{2N+1} = \sin \left[\frac{\pi(44-j)}{2N+1} - \frac{43\pi}{2N+1} \right] \quad (3.34)$$

$$\sin \frac{\pi(42-j)}{2N+1} = \sin \left[\frac{\pi(-44-j)}{2N+1} + \frac{86\pi}{2N+1} \right] \quad (3.35)$$

The terms on the right side can be further expanded to achieve this description.

$$\begin{aligned} & \sin \left[\frac{\pi(44-j)}{2N+1} - \frac{43\pi}{2N+1} \right] \\ &= \sin \frac{\pi(44+j)}{2N+1} \cos \frac{43\pi}{2N+1} - \cos \frac{\pi(44+j)}{2N+1} \sin \frac{43\pi}{2N+1} \end{aligned} \quad (3.36)$$

$$\begin{aligned} & \sin \left[\frac{\pi(-44-j)}{2N+1} + \frac{86\pi}{2N+1} \right] \\ &= \sin \frac{\pi(-44-j)}{2N+1} \cos \frac{86\pi}{2N+1} + \cos \frac{\pi(-44-j)}{2N+1} \sin \frac{86\pi}{2N+1} \end{aligned} \quad (3.37)$$

When we add (3.36) and (3.37) and merge the common terms, we obtain the following equivalent objective.

$$\begin{aligned} & \sin \frac{\pi(j+1)}{2N+1} + \sin \frac{\pi(42-j)}{2N+1} \\ &= \sin \frac{\pi(44+j)}{2N+1} \left[\cos \frac{43\pi}{2N+1} - \cos \frac{86\pi}{2N+1} \right] \\ &+ \cos \frac{\pi(44+j)}{2N+1} \left[\sin \frac{86\pi}{2N+1} - \sin \frac{43\pi}{2N+1} \right] \end{aligned} \quad (3.38)$$

Since $N = 64$, (3.38) is equivalent to (3.39) as follows.

$$\begin{aligned}
& \sin \frac{\pi(j+1)}{2N+1} + \sin \frac{\pi(42-j)}{2N+1} \\
&= \sin \frac{\pi(44+j)}{2N+1} \left[\cos \frac{\pi}{3} - \cos \frac{2\pi}{3} \right] \\
&+ \cos \frac{\pi(44+j)}{2N+1} \left[\sin \frac{2\pi}{3} - \sin \frac{\pi}{3} \right]
\end{aligned} \tag{3.39}$$

It is obvious that

$$\cos \frac{\pi}{3} - \cos \frac{2\pi}{3} = 1 \tag{3.40}$$

$$\sin \frac{2\pi}{3} - \sin \frac{\pi}{3} = 0 \tag{3.41}$$

Finally, (3.39) can be expressed in the following format.

$$\sin \frac{\pi(j+1)}{2N+1} + \sin \frac{\pi(42-j)}{2N+1} = \sin \frac{\pi(44+j)}{2N+1} \tag{3.42}$$

Therefore, (3.31) has been proven.

In summary, these features utilized to design the fast DST-VII/DCT-VIII method are tenable in theory. These inherent properties are perfectly supported and considered useful for a more efficient implementation. It is the deviation caused by the rounding operation in the finite-precision expression that breaks the (anti-)symmetric properties.

3.5 Complexity Analysis

We provide complexity analysis in this section, including both the number of arithmetic operation counts and the actual software execution time. In the software execution

time section, two experiments are devised 1) using a separate test program to execute the transform operation by a large number of repetitions and calculate the average transform time; 2) collecting the actual transform time from the VTM.

3.5.1 Arithmetic Operations

In the element-wise matrix multiplication, N^2 multiplications and $N(N - 1)$ additions are needed to derive a 1-D inverse transformed results. Therefore, the doubled number of operations are needed for calculating a 2-D transform results. We calculate the number of operation counts involved in a 1-D transform process to conduct the comparison.

According to the three features as mentioned above, a reduced number of operation counts can be achieved. To check the practical effects, we tabulate the numbers of arithmetic operations required for a 1-D N -point transform of the full-matrix multiplication and the fast method in Table 8. Overall, 41.8%, 33.1% and 43.8% total number of arithmetic operations are saved for 16-point, 32-point and 64-point DST-VII/DCT-VIII transform, respectively.

To showcase how these numbers of operation counts are obtained, we take the 16-point DST-VII inverse transform as an example to introduce the details. To derive a single output element, 16 multiplications and 15 additions are needed using element-wise matrix multiplication. Therefore, to obtain an output vector, 16 times operation counts are required, *i.e.*, 256 multiplications and 240 additions.

However, if using the proposed fast methods, Feature #1 can be applied to 10

Table 8: The number of arithmetic operations for a 1D forward/inverse transform.

| Transform Size | Matrix Multiplication | | | Fast DST-VII/DCT-VIII | | |
|----------------|-----------------------|------|-------|-----------------------|------|-------|
| | Mult | Add | Shift | Mult | Add | Shift |
| 16 | 256 | 240 | 16 | 127 | 155 | 16 |
| 32 | 1024 | 992 | 32 | 620 | 718 | 32 |
| 64 | 4096 | 4032 | 64 | 2207 | 2331 | 64 |

transform vectors, Feature #2 can be applied to 5 transform vectors and Feature #3 can be applied to the remaining transform vector. In Feature #1, there are 16 shared values to be re-used which occupy 25 additions and another shared value which requires 1 multiplication. To use these shared values derive the transformed results for the 10 transform vectors, an additional of 10×10 multiplications and 10×10 additions are needed. Thus 101 multiplication operations and 125 addition operations are required by applying Feature #1. The number of operation counts can be calculated in a similar way for performing Feature #2, resulting in 25 multiplications and 20 additions. When applying Feature #3, there are 1 multiplication and 10 addition operations. By summing them up, the total numbers of multiplications and additions needed to derive a 1-D 16-point DST-VII transformed vector are 127 and 155, respectively. Therefore, 50.4% and 35.4% operation counts reduction are achieved for multiplication and addition, respectively.

For the 1-D 32-point and 64-point transforms, the numbers of operation counts can be calculated in a similar way. As tabulated in Table 8, 39.5% and 27.6% total number of multiplications and additions are reduced to derive a 1-D 32-point transformed results. In a 1-D 64-point transform case, 46.1% and 42.2% operation counts reduction are achieved for multiplication and addition, respectively.

3.5.2 Software Execution Time

In this subsection, we devise two experiments to compare the software execution time between the proposed fast method and the direct matrix multiplication. Firstly, a standalone test program is used to measure the run-time (in seconds) of individual transform functions by repeating a large number of times. Secondly, the proposed fast method is integrated into VTM-3.0 and executed over all the CTC test sequences, the transform time is collected to provide a statistical summary showing the ratio of the run-time between the proposed fast transform and the anchor. It should be noted that the matrix multiplication implementation is the same no matter which VTM version is used.

The standalone program to measure the run-time of individual functions is written in C, and auto-vectorization is disabled (`#pragma loop(no_vector)`) for compiling to provide a fair comparison. The total number of repeating iterations are set empirically so that the total execution time is within a reasonable range. The detailed configurations of the test environment are available below.

- CPU: i7-6600U CPU @ 2.6 GHz
- Windows 10 Pro, 64-bit
- Memory (RAM): 16 GB
- Compiler: Visual Studio 2017

The results of All Intra (AI) and Random Access (RA) are tabulated in Table 9. By repeating the 16-point DCT-VIII transform function by 2^{23} times, the proposed fast

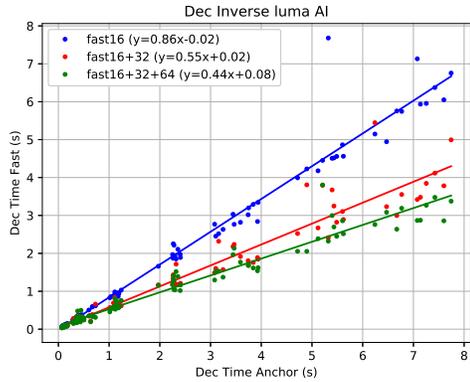
Table 9: Comparisons on software execution speed (seconds).

| Transform Size | No. of Iterations | Matrix Multiplication | | Proposed Fast Method | |
|-------------------|-------------------|-----------------------|---------|----------------------|---------|
| | | Forward | Inverse | Forward | Inverse |
| 16-point DCT-VIII | 2^{23} | 11.3 | 15.2 | 4.8 | 4.9 |
| 32-point DCT-VIII | 2^{20} | 11.1 | 14.8 | 5.7 | 5.5 |
| 64-point DCT-VIII | 2^{17} | 11.0 | 14.2 | 5.2 | 5.1 |

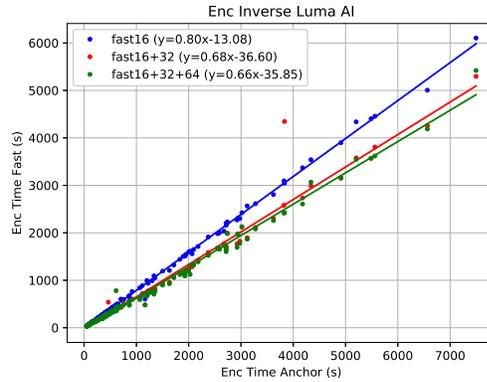
method can save 57.52% and 67.76% software execution time for forward and inverse transform, respectively. In the 32-point DCT-VIII transform, an average of 48.65% and 62.84% time savings are achieved for forward and inverse transform, respectively. Similarly, an average time saving of 52.73% and 64.08% is observed for forward and inverse transform, respectively.

In the second experiment, the 16-point, 32-point, and 64-point DST-VII/DCT-VIII fast methods are integrated into VTM-3.0 reference software. All the test sequences from VVC CTC are utilized for the testing. The software execution time of both forward and inverse transform are collected for the Luma component. Encoding process involves both forward and inverse transforms since the RDO process, while the decoding process only involves the inverse transform. To validate the individual contribution of each block-level, we enable the fast method from the smallest block size and increase gradually to the largest block size.

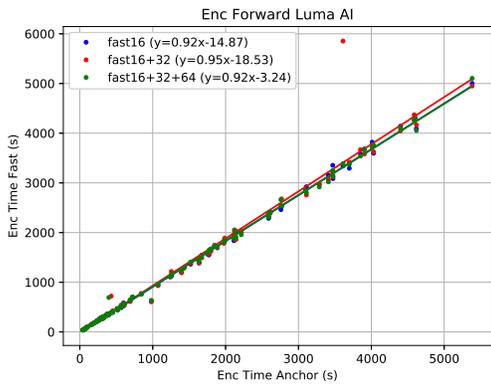
The statistical results are illustrated in Figure 13. *fast16* denotes enabling fast methods of block size 16, *fast16+32* represents enabling fast methods of both block size 16 and 32, and *fast16+32+64* means enabling fast methods of all block levels. The vertical axis is the encoding/decoding time of the proposed fast methods and the horizontal



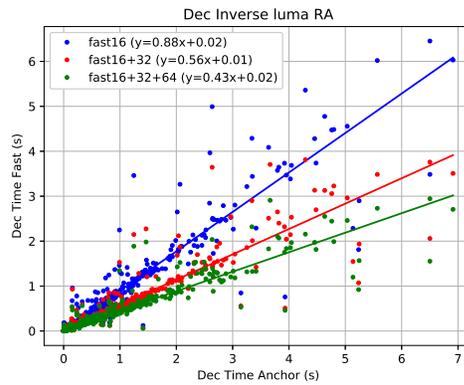
(a) Inverse decoding time of AI.



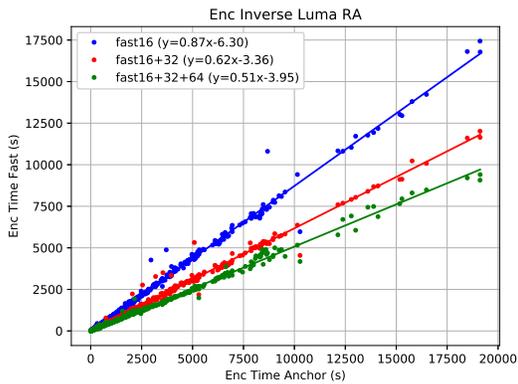
(b) Inverse encoding time of AI.



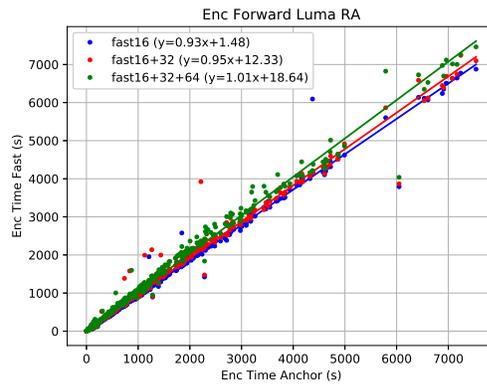
(c) Forward encoding time of AI.



(d) Inverse decoding time of RA.



(e) Inverse encoding time of RA.



(f) Forward encoding time of RA.

Figure 13: Software execution time of the Luma component.

axis is the encoding/decoding time of the anchor. The solid lines are the linear regression approximations with the mean squared loss. The average percentage of time reduction can be approximately represented by $(1 - l) \times 100\%$ where l is the *slope* of the solid line.

Overall, the superiority is quite remarkable, especially during inverse transform process. It is observed in Figure 13 that an average of 56%, 57% inverse decoding time savings are achieved under AI and RA configuration, respectively. In inverse transform of the encoding process, an average of 34% and 49% time savings have been achieved for AI and RA configuration, respectively. In addition, superiority is increased when involving the 32-point and 64-point fast method implementation. Take the decoding time of inverse transform under AI configuration *i.e.*, Figure 13a as an example, 14% average time saving is achieved when only the 16-point fast implementation is enabled. 31% more time saving is achieved when the 32-point fast method is enabled on top of that. An additional 11% time saving is observed when the 64-point fast transform is enabled. Therefore, each size of fast transform has its own contributions to the final performance. The decreased superiority during the forward transform process is caused by the fact that the encoder needs to traverse all possible transforms to perform rate-distortion optimization thereby dilutes the superiority of the fast methods.

3.5.3 Other Metrics

The additional metrics that might help better understanding the proposed method are tabulated in Table 10. The proposed method shares the following merits, no additional memory requirements for storing transform matrices, no additional computations required

Table 10: Additional metrics related to the proposed fast method.

| Metrics | Reported results |
|---|---|
| Additional memory requirements for storing transform matrices | No |
| Minimum bit-precision | Unchanged |
| Specify if a transform requires multiple iterations where transform output is fed back as input to the transform logic, and multiple iterations are required to produce final transform output. | No |
| Other operations and memory requirements relevant to the proposed tool. | No |
| List of all combinations of transforms block size and transform type used for secondary transformation. | Same with VTM-3.0 |
| Provide analysis of implementation and arithmetic commonalities of proposed transforms. | Proposed method supports both partial butterfly and matrix multiplications. |
| If the proposal requires additional computations at the encoder or decoder. | No |

at the encoder/decoder side, the combinations of transform block size and transform type used for secondary transform are consistent with VTM reference software, the minimum bit-precision is unchanged, etc.

3.6 Experiments

3.6.1 Experiment settings

The proposed fast methods are integrated to VVC Test Model VTM-3.0 [88]. Three sets of experiments are conducted, including the CTC Set, the Low QP Set and

comparing with relevant methods. The CTC Set uses the common test condition [89] as defined by Joint Video Experts Team (JVET) for evaluating proposals during the VVC standards development. The quantization parameters (QP) are set to 22, 27, 32, 37. The Low QP set uses low QP values of 2, 7, 12, 17 to test the high-quality compression performance. In each set, we provide the results of both inter MTS is disabled and enabled. By default, the intra MTS is enabled in VVC common test condition. We also compare with similar schemes [90–92] proposed in MPEG meeting and analyze relative merits.

In those experiments, a set of 26 video sequences ranging from 416×240 to 3840×2160 are tested, including four artificial sequences with the computer screen and mixed natural and screen content. It should be noted that Class D and Class F (screen content) are excluded in the overall average performance. The Bjøntegaard delta bitrates (BD-Rate) [93, 94] is used to evaluate the relative coding improvement. The run-time ratio of the proposed method to anchor as defined in (3.43) is used to evaluate the time complexity, with 100% represents no run-time saving. All Intra (AI), Random Access (RA) and Low Delay B (LDB) configurations are covered. Under AI configuration, every picture is coded as Intra while under RA, intra period is set as one second, GOP size is set to 8. The codec is operating in 10-bit mode, RDOQ is enabled. The decoding time and encoding time are measured in seconds.

$$\Delta T = \frac{T_{Proposed}}{T_{Anchor}} \times 100\% \quad (3.43)$$

Table 11: Run-time performance compared with VTM-3.0 under CTC.

| Class | Sequence | All Intra | | Random Access | | | | Low Delay B | | | |
|------------------|---------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| | | ΔT_{Enc} | ΔT_{Dec} | InterMTS off | | InterMTS on | | InterMTS off | | InterMTS on | |
| | | | | ΔT_{Enc} | ΔT_{Dec} |
| Class A1 4k | Tango2 | 93% | 86% | 102% | 101% | 98% | 96% | - | - | - | - |
| | FoodMarket4 | 93% | 86% | 99% | 99% | 97% | 101% | - | - | - | - |
| | CampfireParty2 | 97% | 92% | 98% | 96% | 98% | 99% | - | - | - | - |
| Class A2 4k | CatRobot1 | 95% | 91% | 101% | 101% | 99% | 101% | - | - | - | - |
| | DaylightRoad2 | 98% | 96% | 101% | 101% | 99% | 99% | - | - | - | - |
| | ParkRunning3 | 96% | 89% | 101% | 100% | 96% | 98% | - | - | - | - |
| Class B 1080p | MarketPlace | 98% | 88% | 99% | 97% | 97% | 98% | 100% | 101% | 96% | 83% |
| | RitualDance | 94% | 88% | 98% | 96% | 95% | 96% | 99% | 102% | 97% | 96% |
| | Cactus | 97% | 91% | 100% | 99% | 98% | 99% | 100% | 105% | 97% | 96% |
| | BasketballDrive | 96% | 91% | 99% | 98% | 98% | 96% | 99% | 104% | 98% | 92% |
| | BQTerrace | 96% | 95% | 99% | 99% | 99% | 101% | 100% | 104% | 98% | 92% |
| Class C WVGA | BasketballDrill | 99% | 96% | 101% | 102% | 101% | 104% | 99% | 100% | 98% | 95% |
| | BQMall | 100% | 95% | 100% | 101% | 101% | 102% | 100% | 101% | 98% | 80% |
| | PartyScene | 100% | 99% | 101% | 102% | 98% | 100% | 94% | 93% | 98% | 98% |
| | RaceHorses | 96% | 92% | 100% | 101% | 99% | 97% | 99% | 98% | 97% | 94% |
| Class D WQVGA | BasketballPass | 97% | 90% | 99% | 99% | 99% | 102% | 104% | 103% | 97% | 99% |
| | BQSquare | 98% | 99% | 100% | 101% | 99% | 101% | 99% | 97% | 98% | 98% |
| | BlowingBubbles | 98% | 101% | 99% | 98% | 98% | 100% | 99% | 98% | 98% | 99% |
| | RaceHorses | 97% | 96% | 98% | 99% | 98% | 107% | 100% | 103% | 97% | 99% |
| Class E 720p | FourPeople | 96% | 92% | - | - | - | - | 100% | 100% | 85% | 94% |
| | Johnny | 100% | 95% | - | - | - | - | 99% | 100% | 97% | 93% |
| | KristenAndSara | 96% | 92% | - | - | - | - | 99% | 100% | 97% | 94% |
| Class F | BasketballDrillText | 98% | 95% | 99% | 101% | 98% | 99% | 99% | 99% | 116% | 113% |
| | AOV5 | 96% | 94% | 101% | 100% | 96% | 98% | 100% | 103% | 99% | 98% |
| | SlideEditing | 97% | 99% | 99% | 101% | 99% | 107% | 101% | 103% | 100% | 101% |
| | SlideShow | 96% | 96% | 100% | 101% | 97% | 101% | 105% | 100% | 97% | 101% |
| Average | | 96% | 91% | 100% | 100% | 98% | 100% | 99% | 101% | 97% | 97% |

3.6.2 Common Test Condition

The run-time results under common test condition are shown in Table 11. An average of 9%, 0%, and -1% decoding time savings are achieved for AI, RA and LDB configurations, respectively and 4%, 0%, and 1% overall encoding time savings are achieved for AI, RA, and LDB configurations, respectively. Among AI, RA and LDB configurations, the proposed fast methods achieve the most significant average decoding time reduction under AI configuration with up to 9% decoding time saving. In addition, the proposed fast methods behave consistently comparing with the anchor across all the resolutions. In high-resolution contents, the proposed fast method is slightly better than in low-resolution contents which could probably be caused by more sizes of fast transform matrices are involved. In the encoding process, the most time saving is with Sequence *Tango2* and *FoodMarket4* of 7%; *CampfireParty2*, *RitualDance* and *RaceHorses* of 2%; and *PartyScene* of 6% for AI, RA and LDB configuration, respectively. In the decoding process, the most time saving is with Sequence *Tango2* and *FoodMarket4* of 14%; *RitualDance* of 4%; and *PartyScene* of 7% for AI, RA and LDB, respectively.

An increased superiority is observed when inter MTS is enabled for RA and LDB which is also as expected. Overall, 0% and 3% decoding time savings are achieved for RA and LDB, respectively; an average of 2% and 3% encoding time savings are achieved for RA and LDB, respectively. The overall decoding time saving increases from -1% to 3% for LDB, and the overall encoding time saving increases from 0% to 2% for RA, from 1% to 3% for LDB. This demonstrates the fast method is of significant benefits for inter coding process, especially when inter MTS is enabled when more transform types are

Table 12: BD-Rate performance using tuned transform matrix under CTC.

| Class | All Intra | | | Random Access | | | Low Delay B | | |
|----------------|--------------|---------------|--------------|---------------|---------------|---------------|--------------|--------------|---------------|
| | Y | U | V | Y | U | V | Y | U | V |
| Class A1 | 0.02% | 0.00% | 0.06% | 0.00% | -0.12% | 0.05% | - | - | - |
| Class A2 | 0.01% | 0.01% | 0.03% | -0.04% | 0.05% | 0.01% | - | - | - |
| Class B | 0.00% | -0.01% | -0.01% | -0.02% | 0.04% | -0.08% | 0.00% | -0.26% | -0.19% |
| Class C | 0.01% | -0.01% | -0.15% | 0.01% | -0.09% | -0.01% | -0.01% | 0.09% | 0.12% |
| Class D | 0.01% | 0.15% | -0.05% | -0.03% | -0.26% | -0.17% | 0.01% | 0.25% | 0.50% |
| Class E | 0.02% | -0.03% | 0.17% | - | - | - | 0.02% | 1.16% | -0.73% |
| Class F | 0.00% | -0.06% | 0.12% | -0.04% | -0.01% | -0.06% | 0.05% | -0.04% | 0.07% |
| Average | 0.01% | -0.01% | 0.01% | -0.01% | -0.03% | -0.02% | 0.00% | 0.22% | -0.22% |

involved in the RDO process.

Although some sequences happen to occupy more time than the anchor, *e.g.*, decoding time of *BlowingBubbles* under AI, encoding time of *BasketballDrill* under RA with inter MTS off, the overall time saving is quite encouraging. This phenomenon is probably caused by the CPU perturbation while executing since the decoding time is too short on these sequences.

To validate the coding efficiency by using the tuned transform matrices, we collect the BD-Rate results as shown in Table 12. Overall, the proposed fast method achieves 0.01%, -0.01% and 0.01% Luma component BD-Rate reduction under AI, RA, and LDB respectively compared with the anchor. The Luma component achieves very similar BD-Rate performance in most cases and only trivial difference has been observed compared with anchor. There exist some differences, but they are tolerable by considering the time saving it can bring. In summary, no noticeable side effects have been introduced by

replacing with the tuned transform matrices. The BD-Rate results when inter MTS is enabled are similar to Table 12 thus are omitted here.

3.6.3 Low QP Test Condition

The run-time results using low QP values are tabulated in Table 13. In contrast to common test condition, the proposed fast method achieves decreased superiority for AI and increased superiority for RA and LDB. Under the AI test condition, the decoding time saving decreases from 9% to 1% and the encoding time saving decreases from 4% to 2%. In contrast, both the decoding and encoding saving increases from 0% to 1% under RA test condition. Under the LDB configuration, the decoding time saving decreases from 1% to 0%, and the encoding time increases from -1% to 0%. The changes under the AI test condition are caused by the fact that more smaller block transforms involved in low QP encoding process thus the time saving has been diluted. The essence of the proposed fast method is to accelerate the transform process by reducing the number of operation counts. Therefore, more time saving can be achieved in larger block transform sizes. The changes under RA and LDB test conditions are mostly caused by the fluctuations in terms of both the transform size and the transform counts which heavily depends on the coding parameter settings.

When inter MTS is enabled, an improved time saving performance has also been observed in RA and LDB test conditions. The encoding time saving increases by 1% and 2% for RA and LDB, respectively. The decoding time increases by 1% and 4% for RA and LDB, respectively. This phenomenon is consistent with that of using normal

QP values which reveals that more transform block sizes are involved thereby more time saving is obtained.

To check the effects on the coding efficiency by introducing the new transform matrices under low QP test conditions, we collect the BD-Rate results in Table 14. An average of 0.02%, 0.01%, and 0.00% changes are observed for AI, RA, and LDB, respectively. In the Luma component, a large portion of classes are not affected by introducing the tuned transform matrices, *i.e.*, identical coding efficiency has been achieved. Marginal fluctuations are observed in the other classes which are tolerable since the overall changes are very trivial that can be neglected.

3.6.4 Comparison with Relevant Methods

We compare the proposed methods with relevant schemes that were discussed in the core experiments on complexity reduction of MTS. The DFT-based scheme JVET-M0288 [90], transform adjustment-based method JVET-M0538 [91] and Transform Adjustment Filters (TAF)-based solution JVET-M0080 [92] are included in this comparison.

In JVET-M0288 [90], a DFT-based scheme is proposed by replacing DST-VII and DCT-VIII with corresponding DFT transforms. The existent symmetries can be utilized to devise efficient fast transform implementation. In JVET-M0538 [91], the authors perform a transform “adjustment” in which the DST-VII and DCT-VIII transform matrices are processed vector by vector. The transform vector is decomposed to a combination of an “adjusted” part and a DCT-II coefficients part. The “adjusted” part is achieved by multiplying the 8 lowest frequency coefficients with a pre-defined 8×8 matrix. The remaining

part is copied from the DCT-II transform matrix coefficients. The primary benefits come from the regular patterns in DCT-II transforms which enable faster parallel computation, and the “zero-out” technique used in DCT-II of dimension 64 which reduces the worst-case number of multiplications. In JVET-M0080 [92], another adjustment-based method is proposed called TAF, in which the transform matrices are approximated using a low complexity adjustment stage. TAF is implemented as a sparse matrix, used as a preprocessor towards the partial butterfly DCT-II algorithm.

The results are shown in Table 15, including Luma component BD-Rate reduction, encoding and decoding time ratio over VTM-3.0 with and without inter MTS enabled. It can be observed that though JVET-M0288 achieves better performance in terms of decoding time savings, it requires a two-stage implementation which is not friendly in parallel computation required scenarios. Another set of DFT transform sets need to be stored in the reference software which requires additional memory space. The JVET-M0538 achieves better decoding time savings when inter MTS is on compared with the proposed method, but leads to larger coding degradation. When tested over VTM-3.0 with inter MTS off, it achieves inferior decoding time saving with noticeable encoding time increase. Over VTM-3.0 with inter MTS off, JVET-M0080 performs slightly better in terms of decoding time saving, but with much significant coding performance degradation by comparing with the proposed method. Over VTM-3.0 with inter MTS on, the proposed method performs better than JVET-M0080 in terms of both software run-time savings and coding performance. In addition, none of the counterparts supports dual implementation, *i.e.*, direct matrix multiplication and fast transform deployment.

In summary, the compared methods fail to support the following features simultaneously.

- Noticeable software run-time saving.
- Negligible coding performance degradation.
- Parallel computation supported.
- Dual-implementation supported.

The proposed scheme achieves a superior overall performance by considering the run-time saving, side effects on coding performance and dual-implementation with capability of parallel computation supported.

Table 13: Run-time performance compared with VTM-3.0 under Low QP.

| Class | Sequence | All Intra | | Random Access | | | | Low Delay B | | | |
|------------------|---------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| | | ΔT_{Enc} | ΔT_{Dec} | InterMTS off | | InterMTS on | | InterMTS off | | InterMTS on | |
| | | | | ΔT_{Enc} | ΔT_{Dec} |
| Class A1 4k | Tango2 | 100% | 98% | 98% | 99% | 98% | 99% | - | - | - | - |
| | FoodMarket4 | 98% | 100% | 100% | 100% | 97% | 97% | - | - | - | - |
| | CampfireParty2 | 101% | 101% | 99% | 99% | 108% | 103% | - | - | - | - |
| Class A2 4k | CatRobot1 | 99% | 101% | 98% | 99% | 97% | 98% | - | - | - | - |
| | DaylightRoad2 | 85% | 88% | 98% | 97% | 96% | 96% | - | - | - | - |
| | ParkRunning3 | 96% | 95% | 98% | 96% | 95% | 94% | - | - | - | - |
| Class B 1080p | MarketPlace | 98% | 98% | 100% | 101% | 98% | 98% | 100% | 96% | 98% | 94% |
| | RitualDance | 98% | 100% | 99% | 101% | 98% | 96% | 101% | 98% | 97% | 96% |
| | Cactus | 98% | 99% | 99% | 102% | 99% | 100% | 100% | 101% | 96% | 96% |
| | BasketballDrive | 95% | 101% | 101% | 101% | 100% | 100% | 101% | 102% | 98% | 96% |
| | BQTerrace | 101% | 108% | 100% | 101% | 99% | 99% | 100% | 99% | 100% | 99% |
| Class C WVGA | BasketballDrill | 97% | 106% | 100% | 101% | 97% | 96% | 100% | 100% | 98% | 102% |
| | BQMall | 99% | 103% | 101% | 100% | 97% | 100% | 119% | 114% | 98% | 99% |
| | PartyScene | 99% | 100% | 99% | 100% | 98% | 101% | 101% | 98% | 97% | 101% |
| | RaceHorses | 100% | 101% | 99% | 99% | 98% | 95% | 100% | 99% | 97% | 96% |
| Class D WQVGA | BasketballPass | 100% | 100% | 100% | 102% | 98% | 98% | 99% | 102% | 99% | 99% |
| | BQSquare | 100% | 113% | 98% | 100% | 98% | 106% | 101% | 104% | 98% | 102% |
| | BlowingBubbles | 101% | 102% | 101% | 103% | 100% | 103% | 102% | 106% | 100% | 102% |
| | RaceHorses | 101% | 100% | 102% | 103% | 105% | 108% | 101% | 110% | 98% | 99% |
| Class E 720p | FourPeople | 99% | 102% | - | - | - | - | 98% | 101% | 99% | 99% |
| | Johnny | 97% | 99% | - | - | - | - | 99% | 100% | 98% | 98% |
| | KristenAndSara | 97% | 98% | - | - | - | - | 103% | 99% | 98% | 98% |
| Class F | BasketballDrillText | 100% | 102% | 99% | 100% | 99% | 100% | 99% | 98% | 97% | 94% |
| | AOV5 | 100% | 98% | 100% | 100% | 99% | 105% | 101% | 103% | 99% | 103% |
| | SlideEditing | 99% | 101% | 106% | 107% | 99% | 101% | 101% | 106% | 100% | 103% |
| | SlideShow | 97% | 103% | 102% | 99% | 100% | 102% | 100% | 100% | 101% | 104% |
| Average | | 98% | 99% | 99% | 99% | 98% | 98% | 100% | 100% | 98% | 96% |

Table 14: BD-Rate performance of using tuned transform matrix compared with VTM-3.0 under Low QP.

| Class | All Intra | | | Random Access | | | Low Delay B | | |
|----------------|--------------|--------------|--------------|---------------|--------------|--------------|--------------|---------------|--------------|
| | Y | U | V | Y | U | V | Y | U | V |
| Class A1 | 0.00% | -0.02% | 0.01% | 0.00% | 0.03% | 0.01% | - | - | - |
| Class A2 | 0.07% | -0.02% | -0.01% | 0.02% | -0.01% | -0.01% | - | - | - |
| Class B | 0.00% | 0.01% | 0.00% | 0.01% | 0.00% | 0.01% | 0.00% | -0.01% | 0.02% |
| Class C | 0.01% | 0.00% | 0.01% | 0.00% | 0.00% | 0.02% | 0.00% | -0.01% | -0.02% |
| Class D | 0.01% | -0.05% | -0.01% | 0.00% | 0.03% | 0.00% | 0.00% | 0.01% | -0.02% |
| Class E | 0.00% | 0.00% | 0.00% | - | - | - | 0.00% | -0.01% | -0.01% |
| Class F | 0.00% | 0.01% | -0.02% | -0.11% | -0.11% | -0.05% | -0.06% | 0.02% | -0.04% |
| Average | 0.02% | 0.00% | 0.00% | 0.01% | 0.00% | 0.01% | 0.00% | -0.01% | 0.00% |

Table 15: Comparison with related methods.

| Methods | Configuration | Over VTM-3.0 | | | Over VTM-3.0, InterMTS on | | |
|-----------------|---------------|--------------|------------------|------------------|---------------------------|------------------|------------------|
| | | Y BD-Rate | ΔT_{Enc} | ΔT_{Dec} | Y BD-Rate | ΔT_{Enc} | ΔT_{Dec} |
| JVET-M0288 [90] | All Intra | 0.00% | 96% | 90% | - | - | - |
| | Random Access | -0.01% | 99% | 98% | -0.01% | 97% | 98% |
| | Low Delay B | 0.02% | 100% | 99% | 0.01% | 97% | 96% |
| JVET-M0538 [91] | All Intra | 0.00% | 98% | 94% | - | - | - |
| | Random Access | -0.37% | 119% | 100% | 0.00% | 98% | 98% |
| | Low Delay B | -0.49% | 125% | 100% | 0.05% | 97% | 96% |
| JVET-M0080 [92] | All Intra | 0.09% | 96% | 85% | - | - | - |
| | Random Access | 0.01% | 101% | 98% | -0.04% | 103% | 96% |
| | Low Delay B | 0.07% | 101% | 100% | -0.09% | 106% | 102% |
| Proposed | All Intra | 0.01% | 96% | 91% | - | - | - |
| | Random Access | -0.01% | 100% | 100% | 0.00% | 98% | 100% |
| | Low Delay B | 0.00% | 99% | 101% | 0.00% | 97% | 97% |

CHAPTER 4

IMPROVED INTRA PREDICTION BEYOND HEVC

In this chapter, we will cover improved intra prediction methods, including MLR for intra prediction and Convolutional Neural Network for intra prediction.

Before delving into the details of the models, we give a brief introduction to existing ANN-based image compression approaches in Section 4.1. In particular, we review relative merits of traditional codecs and ANN-based approaches. We hope this will give readers a better sense how these two approaches differ fundamentally.

In Section 4.2, we compare the state-of-the-art LBC methods and traditional MPEG codecs, i.e., VVC and HEVC intra prediction in terms of coding performance and decoding time complexity. Unlike traditional video coding standards with strictly and formally defined Common Test Condition (CTC), it is difficult to compare the performance between LBC methods and traditional codecs. Another hassle is that typically, these two methods use test data in two different color spaces, i.e., YUV and RGB. We provide experiments and analysis and hopefully the observations can serve as the reference basis for the development and perfection of neural network technology for the future video/image coding tasks.

In Section 4.3, we present an intra prediction scheme using Multiple Linear Regression (MLR). Reference pixels and intra prediction block are combined to make a better prediction with a MLR model. The MLR models are trained mode-dependently to

better characterize the varied patterns in natural images.

Next we introduce our CNN for intra prediction (CIP) model. Reference pixels and intra block copy (IBC) prediction are combined to derive the predicted block. The philosophy is to borrow non-local information from IBC and use which to enhance intra prediction with a CNN model. The benefits are twofold: 1) CNN is expected to make a better prediction due to its superior non-linearity capability. 2) The IBC block serving as the additional information is expected beneficial for recurrent patterns.

Finally, we summarize recent advances of LBC for image compression in Section 4.5.

4.1 Overview of ANN-based Approaches

Regarding ANN-based image compression schemes, the previous methods can be divided into two categories: generative models and non-generative models. The generative models could generate realistic images; however, the objective quality is not as good thereby the acceptability of the machine-created image components eventually becomes somewhat application-dependent. The representative non-generative models are proposed by [56, 57, 95–97]. In these approaches, an end-to-end model is devised to exploit the estimation from the latent representations to the actual distribution in order to reduce the number of bits to be transmitted. Note that the actual latent representation distribution and the entropy model are different. The smallest average code length an encoder-decoder pair can achieve, using the entropy model as their shared entropy model, is given by the Shannon cross entropy between the two distributions. And this entropy is

minimized if the model distribution is identical to the marginal. This implies that, when there exist statistical dependencies in the actual distribution of the latent representation, will lead to sub-optimal compression performance.

4.1.1 Typically Architecture

A latent representation is usually achieved through a “analysis” network as introduced by [56] and further be encoded by a entropy model followed by a “synthesis” network to reconstruct the image from the bitstream. The latent representation is typically represented as a joint, or even fully factorized distribution [57]. The inherent advantage of ANN-based image compression is it allows jointly optimizing the learned parameters. In such a way, the framework is optimized in a joint manner instead of performing prediction, transform, quantization separately in traditional codecs.

One of the principle elements in end-to-end optimized image compression is the trainable entropy model used for the latent representations. Since the actual distributions of latent representations are unknown, the entropy models are used to estimate the required bits to encode the latent representations. The actual required bits are composed of the number of bits required by the true distribution and the that of the cross entropy of the difference between true distribution and estimated distribution. In terms of KL-divergence, the bitrate R is minimized when the estimated distribution is perfectly matched with the true distribution. Therefore, the compression performance of the methods essentially depends on the capacity of the entropy model.

4.1.2 Related Approaches

G. Toderici is one of the pioneers working on ANN for image compression. In [95], the authors exploit a small number of latent binary representations to incorporate the compressed information in every step, and each step increasingly stacks the additional latent representations to achieve a progressive improvement in quality of the reconstructed images. The authors claimed this was the first neural network architecture outperforming JPEG across most bitrate ranges on Kodak dataset, with and without the aid of entropy coding.

The milestone representing the ANN-based methods entered the rapidly-developing stage is when the uniform noise quantizer and Rate-Distortion Optimization (RDO) objective were integrated by J. Ballé [56]. A typical “analysis-synthesis” architecture was proposed to learn a latent representation on which to fit a parametric entropy model estimating the actual distribution. The RDO problem was relaxed by replacing the quantization by additive uniform noise. In such a way, the end-to-end training can be implemented with joint optimization of a quantized representation, the conditional entropy model and the base autoencoder. As a successor of the initial prototype, [57] introduces a hierarchical prior to improve the entropy model. They use a Gaussian Scale Mixture (GSM) [98] where the scale parameters are conditioned on a hyperprior. The key insight is the compressed hyperprior can be added to the generated bitstream as *side information*, which allows the decoder to use the conditional entropy model. Another module worth to mention that contributes significantly to the success of [56] [57] is GDN which was studied in

J. Ballé’s previous work [99] [100]. In [100], the authors investigated the effects of different activation functions on the coding performance, including leaky ReLU and GDN.

The autoregressive work [101] is developed on top of [57]. It is the first learning-based method to outperform the top standard image codec (BPG) in terms of both the PSNR and MS-SSIM distortion metrics. The authors extend the GSM-based entropy model in two ways: 1) by generalizing the hierarchical GSM model to a Gaussian Mixture Model (GMM). 2) by adding an autoregressive component. In essence, this work proposes another form of prior that works more efficiently on image compression task.

The authors of [97] attempt to improve the control of the trade-off between distortion and rate of the latent image representation. The key idea is to directly model the entropy of the latent representation by using a context model: a 3D-CNN which learns a conditional probability model of the latent distribution of the autoencoder. This model yields the state-of-the-art performance in terms of MS-SSIM, but the results using PSNR index are missing.

J. Lee *et al.* [96] extends J. Ballé’s model [57] by incorporating two different types of contexts for either bit-consuming scenario or bit-free scenario. These contexts allow the entropy models to more accurately estimate the distribution of the representations with a more generalized form having both mean and standard deviation parameters. Another difference from [57] is they use the discrete latent representations to train the contexts instead using the noisy representations. This model by far retains the state-of-the-art performance among the ANN-based methods.

4.2 Performance Comparison

Deep learning neural networks demonstrated remarkable capability in a variety of vision and learning tasks in recent years and it is tempting to apply this powerful tool to the compression problem. In recent years a variety of Learning Based Compression (LBC) schemes have been published and attracted quite some attention in the research community. In this work, we present a performance comparison between the latest LBC methods with the SOTA traditional MPEG codecs H.265/HEVC and H.266/VVC on image compression tasks. Unlike traditional video coding standards with canonical evaluation standards, lack of Common Test Condition (CTC) makes it difficult to evaluate the LBC schemes. This work aims at providing a fair comparison between the neural network technology on image coding and the latest video codecs. Beyond that, we analyze the relative merits. This work serves as the reference basis for the development and perfection of neural network technology for the future video/image coding tasks and explores the future potentials to apply deep learning for image/video compression in real applications.

4.2.1 Introduction

The development of hardware as well as the latest transmission technology over the Internet promotes the production of extensive videos and images. According to the latest Cisco visual networking index white paper [1] that the IP video traffic video traffic will be 82 percent of all IP traffic and the traffic from wireless and mobile devices will account for 71 percent of total IP traffic by 2022. The rapid growth of high-definition video traffic poses new challenges towards existing video coding technology.

The video coding community devoted significant efforts to promote the development of video coding standards. The major milestones among which are the High Efficiency Video Coding (HEVC) [102] standards by the Joint Collaborative Team on Video Coding (JCT-VC). Faced with new challenges from rapidly growing video content, MPEG and VCEG are working together as the Joint Video Exploration Team (JVET) to explore the state-of-the-art algorithms and prepare for the next-generation video coding standards beyond HEVC [103]. The on-going video coding standards, termed Versatile Video Coding (VVC) supports larger CTU blocks with more flexible partition/transform shapes. However, the traditional pipelines optimize each process separately, which might lead to suboptimal solutions.

In parallel with the development of traditional video coding standards, many researchers shift their interests to use neural networks for image compression [56, 57, 95–97, 100, 101, 104–107]. In LBC methods for image compression, the entropy model aims at learning a latent representation distribution which is as close to the actual marginal distribution such that the additional bits transmitted to complement the difference are minimal. The well-recognized advantage of LBC solutions is joint optimization. Significant improvements have been achieved in the literature, but comparison with HEVC and VVC intra coding is still missing. In addition, the results achieved under different evaluation environments are not comparable thereby hinders people’s real understanding of the compression efficiency.

To benchmark learning-based methods in comparison with the world-class codecs

as well, this paper presents experimental results to compare this rising star with the state-of-the-art, industrial level video coding codecs HEVC and VVC. Hopefully, the observations can serve as the reference basis for the development and perfection for the real application of the neural network technology, based on which to achieve a better understanding towards relative merits.

4.2.2 Models for Comparison

Intra encoders of HEVC [102] and VVC [108] are included as the traditional line-ups. Six LBC models tabulated in Table 16 are adopted in this work.

Table 16: The implementations of each LBC method.

| Method | Implementations |
|---------------------------------------|-----------------------------------|
| G. Toderici <i>et al.</i> , 2017 [95] | rnn-compression [109] |
| J. Ballé, 2018 [100] | Tensorflow Data Compression [110] |
| J. Ballé <i>et al.</i> , 2018 [111] | |
| D. Minnen <i>et al.</i> , 2018 [101] | |
| F. Mentzer <i>et al.</i> , 2018 [97] | imgcomp-cvpr [112] |
| J. Lee <i>et al.</i> , 2019 [96] | CA_Entropy_Model [96] |

LBC Models

In [95], an RNN-based architecture is proposed. J. Ballé [56] solved the back-prorogation problem by replacing the quantization by additive uniform noise. In [57], a hierarchical prior is introduced to further improve the compression efficiency. The autoregressive work [101] is the first learning-based method to outperform BPG in terms of both the PSNR and MS-SSIM. The authors of [97] attempt to improve the control of the trade-off between distortion and rate of the latent image representation. J. Lee *et al.* [96]

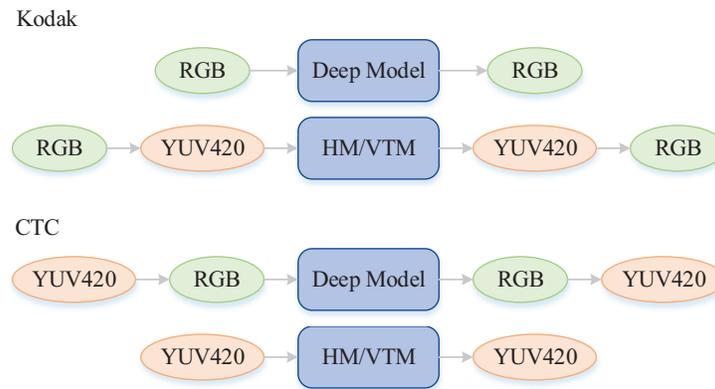


Figure 14: Illustration of color space conversion.

extends J. Ballé’s model [57] by incorporating two different types of contexts. This model is reportedly by far achieves the state-of-the-art performance among the LBC methods.

HEVC and VVC

Both of them are based on the Coding Tree Units (CTU). The intra coding is based on spatial interpolation of samples from previously decoded image blocks. HEVC supports 35 intra prediction modes, including planar, DC and 33 angular prediction modes [113]. The VVC standards have not yet been officially finalized, but major contributions related to intra-frame coding have been incorporated, including larger CTU and transform sizes, more flexible partition schemes, more intra prediction directions, intra block copy, more transform cores, adaptive loop filter, etc. It is reportedly able to achieve significant improvements on top of HEVC.

4.2.3 Evaluation Setup

Dataset and Evaluation Metrics

We include two datasets to evaluate the selected models: Kodak¹ and VVC CTC test sequences [114]. The Kodak dataset is the commonly-used in LBC methods for performance evaluation, which consists of 24 lossless true color images. The latest VVC CTC test sequences which are defined in the 14th JVET meeting consists of 26 test sequences in YUV420 format. The resolution spans from 416×240 to 4096×2160. Class A1 and A2 include 4k content and Class F includes the screen content sequences. To be compatible with LBC methods, sequences in 10-bit are converted to 8-bit. As illustrated in Figure 14, Kodak images are converted to YUV420 format before HM/VTM model and the reconstructed YUV420 files are converted back to RGB domain before computing PSNR and MS-SSIM. Similar conversion is performed on CTC sequences. We use OpenCV-Python package for all color space conversion operations involved in this paper.

We use PSNR and MS-SSIM to evaluate the performance. Eq. (4.1) and (4.2) are used for computing PSNR in RGB and YUV color space, respectively.

$$PSNR_{RGB} = 10 \log_{10} \frac{MAX^2}{MSE} \quad (4.1)$$

$$MSE_{YUV} = \frac{4 \cdot MSE_Y + MSE_U + MSE_V}{6}$$

$$PSNR_{YUV} = 10 \log_{10} \frac{MAX^2}{MSE_{YUV}} \quad (4.2)$$

Models Configurations

All the LBC models are implemented in Tensorflow to avoid the performance discrepancy caused by different platforms. The Github repositories for each model are

¹Downloaded from: <http://www.cipr.rpi.edu/resource/stills/kodak.html>

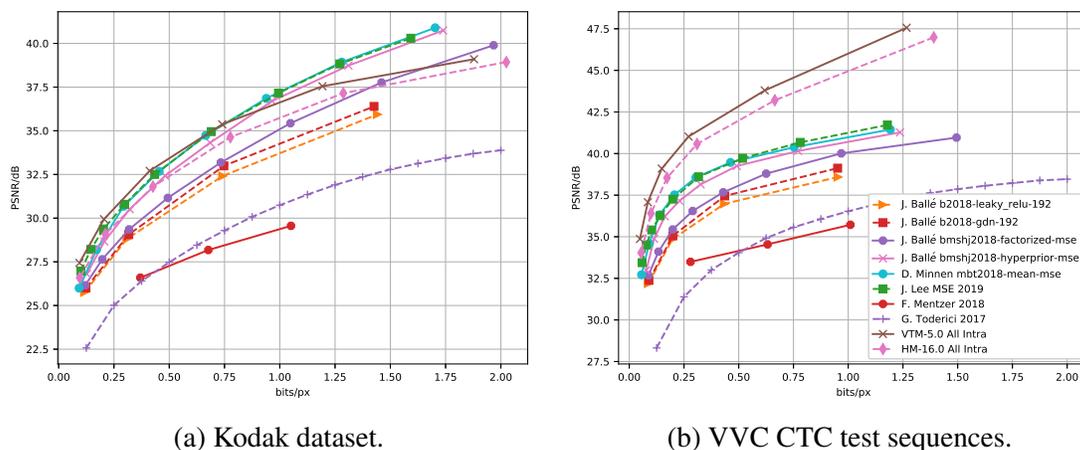


Figure 15: PSNR results of LBC methods comparing with HEVC/VVC.

listed in Table 16.

We use the reference software HM-16.0² and VTM-5.0³ for HEVC and VVC, respectively. Two additional quantization parameter values of [17, 42] are included except for the commonly-used [22, 27, 32, 37]. The parameters *-OutputBitDepth* and *-OutputBitDepthC* are explicitly set to 8 in VTM. The rest configurations follow the All Intra (AI) configurations as defined in CTC in the 14th JVET meeting.

4.2.4 Experiments

In this section, we present the experimental results in terms of PSNR and MS-SSIM. Then we provide the time complexity analysis results.

Experimental Results

The PSNR performance is plotted in Figure 15. On Kodak dataset, [96, 101] achieves the best performance among all the LBC methods. In addition, they also achieve

²<https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-16.0>

³https://vcgit.hhi.fraunhofer.de/jvet/VVCSsoftware_VTM/tree/VTM-5.0

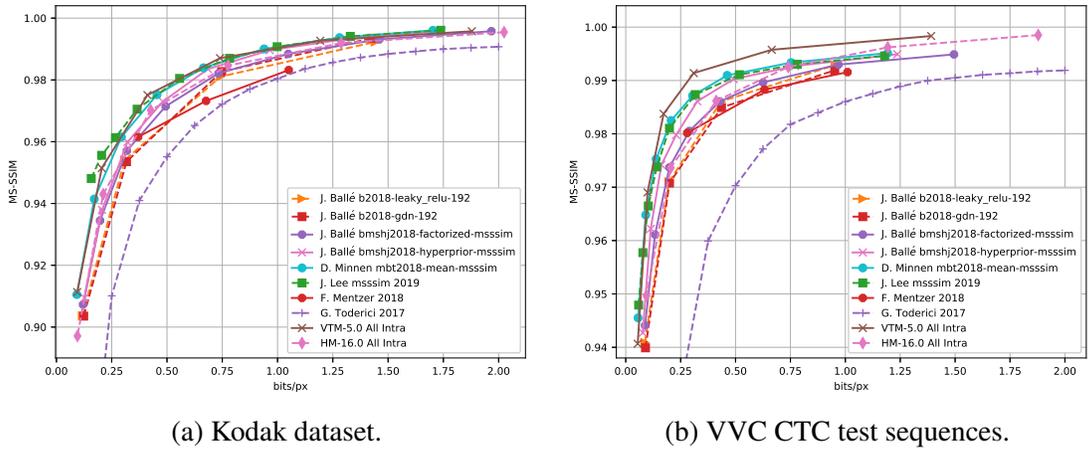


Figure 16: MS-SSIM results of LBC methods comparing with HEVC/VVC..

better PSNR performance in high bitrate range. In contrast, RNN [95] and [97] behave inferiorly both on Kodak and CTC test sequences. It can also be observed that by incorporating GDN [99] and hyperprior [57], the performance has been improved progressively which demonstrates the effectiveness of these two techniques.

However, different pattern is observed on CTC test sequences. VTM and HM show predominant superiority compared with all the other methods across all the bitrate points. At 1.00 bpp, VTM outperforms the best LBC method by about 4.9 dB and the superiority is still increasing at larger bpp regions. The relative ranking of LBC methods keeps consistent to that on Kodak dataset.

The MS-SSIM results are illustrated in Figure 16. Overall, the results are consistent with PSNR results. On Kodak, the best LBC method achieves comparable performance as VTM. The best LBC methods surpass HM by a noticeable margin. But on CTC test sequences, VVC leads the performance with a significant margin. The best LBC methods [96, 101] surpass HM at low bitrate regions but HM achieves better performance

at higher bitrate regions.

4.2.4.1 Time Complexity Analysis

We collect the decoding time for each VVC test sequence and the details are tabulated in Table 17. All the models are configured to decode the bitstream at the closest running point of 0.75 bpp. In J. Ballé [100], *b2018-leaky_relu-192-3* and *b2018-gdn-192-3* differentiated by “LReLU” and “GDN” in Table 17. In J. Ballé, *bmsbj2018-factorized-mse-5* and *bmsbj2018-hyperprior-mse-5* are selected indicated by “factorized” and “hyperprior”, respectively in the Table 17. Since the encoder and decoder are integrated as a whole graph in F. Mentzer [97], we set the timer before and after the *fetch()* function. In the RNN-based model [95], the decoding time of one iteration is achieved by dividing the time by 16 since there are 16 reconstructed images are generated.

The HM-16.0 decoding time is set as the anchor of the decoding time. The fastest codec is VVC with 9% overall decoding time savings when SIMD is enabled. Among the LBC models, J. Lee [96] is the one with the best compression performance, but the time complexity is $783.33\times$ which is intolerant in the application. In contrast, J. Ballé [57] achieves a better trade-off between performance and time complexity. The LBC model with the best time complexity is Model *b2018-leaky_relu-192-3* from J. Ballé [100] with $7.16\times$ average decoding time complexity compared with HM-16.0.

4.3 MLR for Intra Prediction

In video coding frameworks, the essence of intra coding is leveraging the space correlation within a frame to remove redundancy thus achieving compact transmitting

Table 17: Decoding time in seconds on VVC CTC test sequences.

| Class | Sequence Name | J. Ballé [100] | | J. Ballé [111] | | D. Minnen [101] | J. Lee [96] | F. Mentzer [97] | G. Toderici [95] | VTM-5.0 | HM-16.0 |
|-------------------------|---------------------|----------------|--------|----------------|------------|-----------------|-------------|-----------------|------------------|---------------------|---------|
| | | LReLU | GDN | factorized | hyperprior | | | | | | |
| A1 | Tango2 | 6.655 | 6.346 | 9.596 | 9.368 | 14.289 | 873.163 | 36.908 | 14.493 | 0.645 | 0.827 |
| | FoodMarket4 | 6.127 | 5.706 | 8.847 | 8.764 | 13.273 | 786.189 | 27.384 | 9.436 | 0.554 | 0.697 |
| | Campfire | 5.245 | 5.462 | 8.341 | 8.377 | 13.509 | 772.247 | 33.829 | 5.292 | 1.085 | 0.971 |
| A2 | CatRobot | 6.106 | 5.914 | 9.695 | 9.032 | 13.828 | 790.025 | 27.630 | 10.443 | 0.792 | 0.894 |
| | DaylightRoad2 | 6.031 | 5.719 | 8.690 | 8.755 | 13.378 | 809.819 | 27.615 | 16.271 | 0.828 | 0.884 |
| | ParkRunning3 | 6.063 | 6.125 | 9.025 | 9.034 | 13.967 | 830.452 | 27.319 | 6.396 | 1.022 | 1.152 |
| B | MarketPlace | 2.132 | 2.131 | 2.919 | 2.976 | 4.365 | 216.086 | 6.168 | 1.560 | 0.227 | 0.325 |
| | RitualDance | 1.992 | 1.888 | 2.709 | 2.748 | 4.125 | 206.234 | 6.190 | 1.494 | 0.175 | 0.232 |
| | Cactus | 2.041 | 2.050 | 2.771 | 2.877 | 4.287 | 197.209 | 6.123 | 4.885 | 0.353 | 0.280 |
| C | BasketballDrive | 2.113 | 2.012 | 2.819 | 2.868 | 4.269 | 199.110 | 6.198 | 1.726 | 0.246 | 0.133 |
| | BQTerrace | 1.980 | 2.008 | 2.777 | 3.041 | 4.260 | 194.927 | 6.246 | 1.481 | 0.446 | 0.213 |
| | RaceHorses | 0.914 | 0.926 | 1.091 | 1.144 | 1.672 | 46.312 | 0.803 | 0.423 | 0.090 | 0.062 |
| D | BQMall | 0.957 | 0.960 | 1.129 | 1.207 | 1.687 | 43.170 | 1.152 | 0.447 | 0.090 | 0.126 |
| | PartyScene | 0.912 | 0.931 | 1.098 | 1.161 | 1.689 | 41.690 | 0.806 | 0.376 | 0.110 | 0.198 |
| | BasketballDrill | 0.933 | 0.940 | 1.099 | 1.190 | 1.686 | 40.976 | 0.850 | 0.444 | 0.100 | 0.117 |
| E | RaceHorses | 0.729 | 0.755 | 0.802 | 0.912 | 1.196 | 13.251 | 0.173 | 0.249 | 0.028 | 0.037 |
| | BQSquare | 0.729 | 0.762 | 0.805 | 1.107 | 1.211 | 13.268 | 0.294 | 0.248 | 0.036 | 0.072 |
| | BlowingBubbles | 0.731 | 0.735 | 0.773 | 0.841 | 1.200 | 12.582 | 0.167 | 0.249 | 0.035 | 0.042 |
| F | BasketballPass | 0.751 | 0.767 | 0.793 | 0.874 | 1.228 | 12.643 | 0.185 | 0.260 | 0.029 | 0.048 |
| | FourPeople | 1.316 | 1.284 | 1.639 | 1.769 | 2.499 | 102.818 | 2.796 | 0.833 | 0.119 | 0.141 |
| | Johnny | 1.300 | 1.274 | 1.627 | 1.742 | 2.424 | 98.666 | 2.034 | 0.878 | 0.090 | 0.110 |
| F | KristenAndSara | 1.312 | 1.309 | 1.637 | 1.676 | 2.446 | 97.198 | 2.103 | 0.852 | 0.099 | 0.110 |
| | ArenaOfValor | 2.101 | 20.534 | 21.608 | 26.660 | 39.328 | 214.568 | 8.632 | 4.544 | 0.412 | 0.585 |
| | BasketballDrillText | 0.948 | 1.060 | 1.116 | 1.222 | 1.689 | 41.446 | 0.829 | 0.446 | 0.081 | 0.100 |
| F | SlideEditing | 1.229 | 1.279 | 1.606 | 1.685 | 2.541 | 93.765 | 2.042 | 0.720 | 0.126 | 0.160 |
| | SlideShow | 1.230 | 1.237 | 1.590 | 1.616 | 2.461 | 94.557 | 2.046 | 0.738 | 0.098 | 0.219 |
| | Total Time | 62.577 | 80.204 | 106.602 | 112.646 | 168.507 | 6842.371 | 236.519 | 85.184 | 7.956 | 8.735 |
| Time Complexity* | | 7.16× | 9.18× | 12.20× | 12.90× | 19.29× | 783.33× | 27.08× | 9.75× | 0.91 × [†] | 1.00× |

*Compared with HEVC reference software HM-16.0.

[†]SIMD is enabled.

data. With modern video acquisition devices improvement, more high-definition videos emerges into people's lives which has set a new challenge for high efficiency video coding. In this paper, we propose a novel intra video coding scheme based on Multiple Linear Regression (MLR), named Multiple linear regression Intra Prediction (MIP). Instead of predicting pixel values by extrapolating, we try to exploit the potential capability of homogeneous regression method. The proposed method has a very concise and neat design yet achieve better performance compared with High Efficiency Video Coding (HEVC) reference software anchor. The experimental results demonstrate the effectiveness of the proposed method and provide interesting insights for further exploiting the capability of conventional algorithms for video coding when many people favor deep learning-based approaches.

4.3.1 Introduction

A lot of efforts have been made on video compression, including the traditional methods and the ANN-based attempts. However, neither conventional methods nor deep learning-based methods are the ideal solution. Conventional methods often involve complicated arithmetic derivation. Deep learning-based methods are often accompanied by higher time complexity. Current HEVC intra prediction solution is not able to capture the rich texture information by one-time interpolation. In this paper, we propose a concise design based on multiple linear regression which allows for further exploiting the correlation between known pixels and current block. The proposed model takes both the boundary reference pixels and the best intra prediction after Rate-distortion optimization

(RDO) as inputs. It is expected to make a better estimation by a secondary exploration. To refine the model, the prediction mode information is also taken into consideration. The manifold structure of this high dimensional input is hierarchically partitioned by grouping the input into several modes according to the dominant texture orientation. In addition, instead of predicting in the transform domain which could induce information loss during dimensionality reduction, we apply MLR directly in the pixel domain. The proposed method is integrated into HEVC reference software. The experimental results yield interesting coding gain and indicating future possible research directions on conventional algorithms.

4.3.2 Related Work

In this section, we will briefly review HEVC current intra coding schemes as well as giving theoretical description of Multiple Linear Regression.

HEVC Intra Coding

HEVC inherits block-based scheme from previous video coding frameworks with the main difference of Coding Tree Units (CTU) concept. HEVC intra coding is based on spatial interpolation of samples from previously decoded image blocks. It supports up to 35 intra prediction modes named planar, DC and 33 angular prediction modes as shown in Table 18.

The left column and top row neighboring pixels are utilized to predict current block. The planar prediction mode first interpolates the bottom right pixel and the other

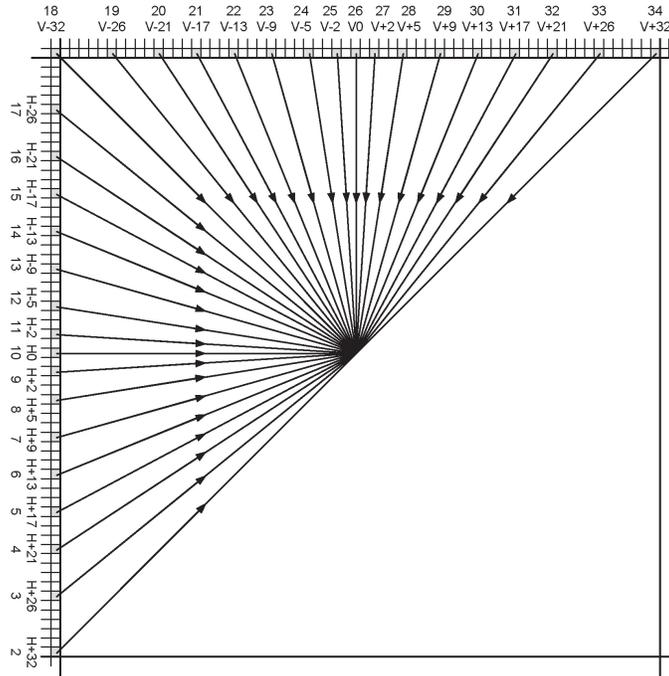


Figure 17: HEVC angular intra prediction modes.

pixels are interpolated by bi-linear method. The DC mode make the prediction by averaging the top row and the left column. Figure 17 depicts the 33 angular prediction modes numbered from 2 to 34. The current block is predicted with the boundary pixels through interpolation. Motivated by the success in [115] by incorporating piece-wise linear projection, this paper aims at investigating the potentiality of combining interpolation and multiple linear regression.

Multiple Linear Regression

Multiple linear regression (MLR) is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression (MLR) is to model the relationship between the explanatory and response variables. The following model is a multiple linear regression model with k

Table 18: Specification of intra prediction modes and associated names

| Intra Prediction Mode | Associated Names |
|-----------------------|------------------------------|
| 0 | Planar |
| 1 | DC |
| 2..34 | Angular (N), $N = 2..34$ |

predictor variables, x_1, \dots, x_k .

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_k + \epsilon \quad (4.3)$$

where ϵ is the error term, β_0 is the intercept, β_1 - β_k are partial regression coefficients, e.g., β_i when $1 \leq i \leq k$ represents the change in the mean response corresponding to a unit change in x_i when the other variables are held constant. The objective of MLR is to solve for the coefficient set $\Theta = \{\beta_0, \beta_1, \dots, \beta_k\}$ given observations X and targets Y .

Least squares is often used to solve the MLR problem. Suppose each predictor variable x_1, x_2, \dots, x_k has n observations. Then x_{ij} represents the i th observation of the j th predictor variable x_j . For example, x_{51} represents the first value of the fifth observation. Specifically, the previous equation can be expressed as:

$$y_j = \beta_0 + \beta_1 x_{j1} + \beta_2 x_{j2} + \dots + \beta_k x_{jk} + \epsilon_j \quad (4.4)$$

where $1 \leq j \leq n$, y_j is the j th target value. The system of n equations can then be represented in matrix notation as follows:

$$y = X\beta + \epsilon \quad (4.5)$$

The matrix is referred to as the design matrix. It contains information about the levels of the predictor variables at which the observations are obtained. The vector β contains all the regression coefficients. To obtain the regression model, β is estimated using the least square estimates as expressed below.

$$\hat{\beta} = (X^T X)^{-1} X^T y \quad (4.6)$$

Then the estimated value of y can be calculated as follows after $\hat{\beta}$ is obtained.

$$\begin{aligned} \hat{y} &= X\hat{\beta} \\ \epsilon &= y - \hat{y} \end{aligned} \quad (4.7)$$

4.3.3 MLR for Intra Prediction

In this section, we first introduce the proposed scheme along with our motivation. Then the way of how the proposed prediction scheme is integrated into HEVC reference software is detailed.

Framework of MIP

The proposed scheme architecture is depicted in Fig. 18. As the best intra prediction block is obtained by Rate Distortion Optimization (RDO), it contains abundant detailed information of current block. However, there is still some information that interpolation is not able to capture. Multiple linear regression is trying to predict current block on top of best intra prediction by combining interpolating and linear regression.

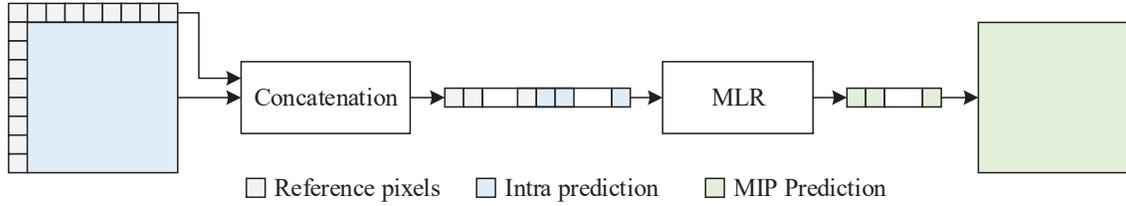


Figure 18: Framework of the MIP model.

In HEVC implementation, a total of $4N + 1$ reference pixels are incorporated to predict current $N \times N$ block, the top row, top right row, left column and left bottom column. Since the top right row and left bottom column might not be available sometimes and filling non-existing pixels will induce additional distortion to the MIP model, we only leverage the top row and the left column, making the reference pixels dimension of $2N + 1$. The reference pixels and the best intra prediction will be flattened and concatenated together and fed into the MLR model. The target is the ground truth block of $N \times N$. We denote the concatenated reference pixels and the intra prediction as X , the target block as Y , the MIP prediction as \hat{Y} . The training process is trying to minimize the following loss function.

$$L = \| Y - \hat{Y} \|_2^2, \quad \hat{Y} = XA + b \quad (4.8)$$

Due to the rich content in natural video sequences, it is not capable of capturing the structure with a single prediction mode. Therefore, HEVC developed multiple intra prediction modes, e.g., Planar, DC and 33 angular modes. Similar idea has been adopted in this work. Separate MIP model is trained for each mode. MIP modes are designed according to intra prediction modes. Other possible classification schemes are left for

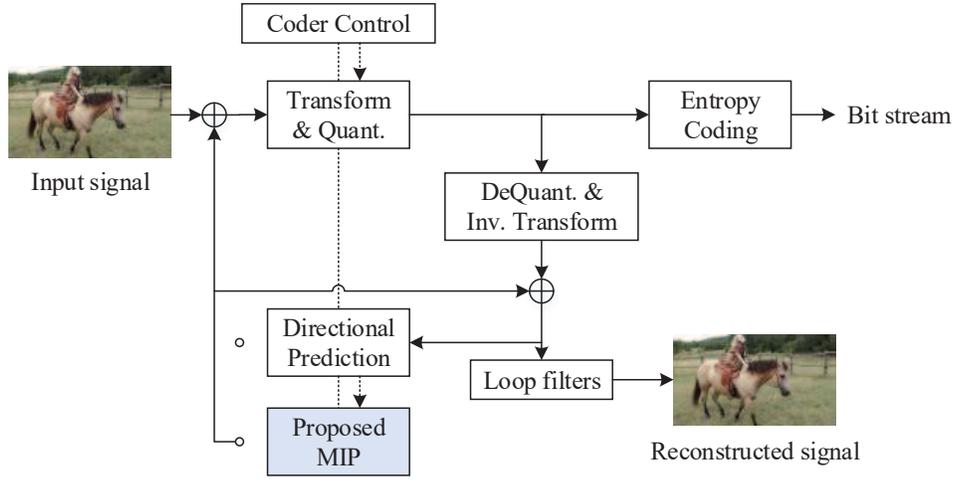


Figure 19: Illustration of integrating MIP model into HEVC.

future investigation. Separate model is trained for Planar and DC mode due to their special texture characteristics. Since the neighboring angular modes share a lot of similarities, we combine 3 adjacent angular modes into a single MIP angular mode to avoid singularity. To be specific, the relationship between MIP prediction mode and HEVC intra prediction mode is expressed as follows.

$$m = \begin{cases} 0, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ \lfloor \text{loor}(\frac{n-2}{3}) + 2, & \text{if } n > 1 \end{cases} \quad (4.9)$$

where n is HEVC intra prediction mode index and m is MIP mode index. Therefore, there are in total 13 MIP modes.

Integration into HEVC

As shown in Fig. 19, input video signal is performed by directional prediction first, e.g., HEVC existing intra prediction scheme which consists of 35 modes. Then

it will be processed by the proposed method and RDO is followed to find the optimal prediction mode. An additional bit flag is encoded to indicate whether MIP is selected.

4.3.4 Experiments

The proposed MIP method is implemented in HEVC reference software 16.0 [116] and its anchor is used for comparison. All the experiments are conducted under HEVC common test condition All Intra (AI) configuration. Quantization Parameter (QP) is set to $\{22, 27, 32, 37\}$. HEVC common test sequences are used to perform the evaluation, consisting of 5 classes with resolutions ranging from 720p to 4k. To save simulation time, only the first frame is used for each sequence.

Training Data Preparation

DIV2K 2K resolution high quality image dataset is used to generate the training dataset. DIV2K contains 800 2K training images, 100 validation images and 100 testing images covering a wide range of contents.

To obtain the best intra prediction, we encode the training samples with HEVC reference software and extract the best intra prediction from the bitstream. The boundary reference pixels are extracted from the reconstructed data. To classify the training samples into local groups according to their directional information, the intra prediction direction is saved. Since we will train separate set of MIP models for each QP and block size combination, and there are 13 transformations in each set. Therefore, there are in total $4 \times 4 \times 13 = 208$ MIP models.

Results and Analysis

Table 19: The BD-Rate results of MLR for Intra prediction.

| Sequence | | BD-Rate | | |
|----------|-----------------|---------|-------|-------|
| | | Y | U | V |
| Class A | Traffic | -0.9% | -0.6% | -1.3% |
| | PeopleOnStreet | -0.5% | 0.0% | 0.1% |
| | Nebuta | -0.9% | -0.8% | -0.7% |
| | SteamLocomotive | -0.6% | -0.3% | -0.3% |
| Class B | Kimono | -0.6% | -1.5% | -1.1% |
| | ParkScene | -0.6% | -1.1% | -1.0% |
| | Cactus | -0.7% | -0.3% | -1.1% |
| | BQTerrace | -0.4% | -1.3% | -0.6% |
| | BasketballDrive | -1.0% | 0.2% | -1.0% |
| Class C | BasketballDrill | -0.1% | -2.0% | -0.8% |
| | BQMall | -0.2% | 0.1% | -0.4% |
| | PartyScene | -0.1% | -0.2% | -0.7% |
| | RaceHorsesC | -0.4% | -0.2% | -0.7% |
| Class D | BasketballPass | 0.1% | 1.1% | -0.8% |
| | BQSquare | 0.1% | 0.1% | -1.1% |
| | BlowingBubbles | 0.4% | -1.3% | -1.4% |
| | RaceHorses | -0.3% | -1.6% | 0.6% |
| Class E | FourPeople | -0.4% | 0.3% | -1.6% |
| | Johnny | -0.3% | -1.1% | -1.3% |
| | KristenAndSara | -0.5% | -0.4% | -0.6% |
| Class A | | -0.7% | -0.5% | -0.5% |
| Class B | | -0.6% | -0.8% | -1.0% |
| Class C | | -0.2% | -0.6% | -0.8% |
| Class D | | 0.1% | -0.4% | -0.7% |
| Class E | | -0.4% | -0.4% | -1.2% |
| Average | | -0.4% | -0.6% | -0.8% |
| Enc Time | | 487% | | |
| Dec Time | | 154% | | |

The BD-Rate reduction of all the test sequences is summarized in Table 19. An average of -0.4%, -0.6%, and -0.8% BD-Rate saving is achieved for Luma and two Chroma components respectively by the proposed method. It is noticeable that MIP performs better on high-resolution sequences than on low-resolution sequences. As high as -0.9% BD-Rate reduction is observed on *Traffic* from Class A while only -0.2% BD-Rate saving is achieved on Class C. It can also be seen that the proposed method performs consistently better on chroma components across different resolution test sequences. As the training is off-line and all the transformations and bias matrices are saved thus not much encoding time is increased. In contrast, encoding time criteria is not as crucial as decoding time. In addition, the intra coding only occupies only a small fraction of the total encoding time in video coding, and the increase of encoding complexity is less significant in the whole video coding system.

4.4 CNN for Intra Prediction

Intra prediction is an essential step to remove spatial redundancy in image coding. How to best predict current block given surrounding pixels is the key efficiency challenge. Inspired by the recent success in applying deep learning to image/video coding systems, we propose an intra prediction method by combining intra block copy and neighboring samples using convolutional neural networks. A novel CNN is developed to further exploit the spatial correlation. Instead of only considering local information, the proposed method can infer the current block via fusing the non-local recurrent features, which is

captured by intra block copy, with the local samples located at the left and above boundaries of current block. We also investigate how the performance is affected by the way of fusing IBC and reference boundary pixels. In addition, training data pre-processing is studied to enable the CNN with a better learning capability. Simulation results yield promising coding gain and indicate great potential ability that CNN can be used for next generation video coding framework.

4.4.1 Introduction

Recently, deep learning has shown superior capability in various tasks, such as image classification [117], image restoration [118]. There are also some recent works applying deep learning on video coding. [119] proposed a down-/up-sampling-based coding scheme using CNN for intra coding. Significant coding gain has been observed. [120] tried to fit a fully connected network on multiple reference pixels and achieved promising results.

Motivated by the recent advances of deep learning in video coding and the efficiency of IBC, we devise a CNN-based Intra Prediction method, termed CIP which tries to learn a model better characterizing the correlation between reconstructed pixels and current block. The proposed CIP is fed with multiple inputs, including both the neighboring reference pixels and the best prediction found by IBC. We also investigated different fusion methods to better leverage extracted features. Data preprocessing is very crucial to train a clean model, therefore a threshold is set to remove distractors from raw IBC blocks.

The experimental results show that the proposed scheme achieves an average of 1.3% bitrate saving on luma component for HEVC test sequences under all intra configuration compared with HEVC 16.0 anchor.

4.4.2 CNN for Intra Prediction

With support of modern advanced mobile devices, multimedia been dramatically increasing over the Internet and video is undoubtedly the dominant Internet traffic through the world. The rich contextual information existing in natural video poses unique challenges on streaming such huge data. In existing HEVC implementations, a large portion of spatial correlation cannot be captured due to the limits of simple interpolation solution. Intra block copy has been demonstrated the effectiveness in estimating recurrent patterns, e.g., screen content coding. In addition, CNN has shown superior power in various tasks. It is natural to try combining them together by CNN. As CNN is adept in exploiting complicated hidden features, we expect it performs a better job in intra prediction. First, the CIP architecture is introduced. Second, the training procedure and hyper-parameter tuning are detailed. Finally, how the proposed method is integrated in HEVC reference software is elaborated.

CIP Architecture

With support of modern GPUs, current CNN has been developed to contain more layers and complicated architectures. However, it is not applicable for video coding which requires higher time efficiency. To balance the trade-off between performance and time

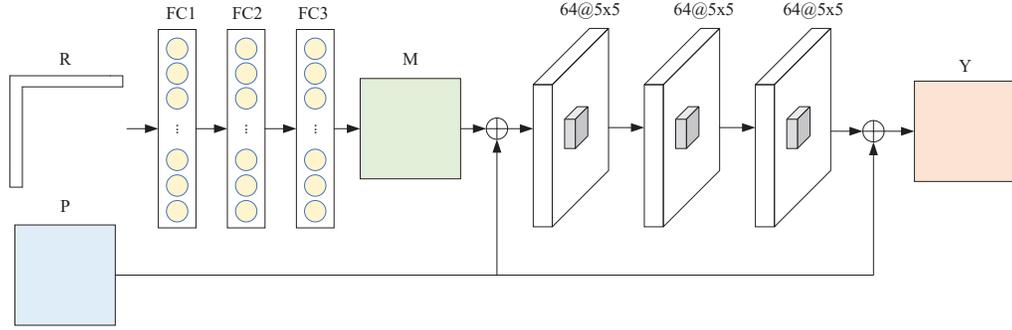


Figure 20: Framework of CIP model.

complexity, we start with a relatively neat design. The proposed CIP architecture is illustrated in Fig. 20. The framework consists of two parts, fully-connected (FC) layers for reference boundary pixels feature extraction and convolutional neural networks for fusion pixels reconstruction.

For current ground truth block Y_0 with size $N \times N$, the proposed CIP aims at learning a projection from $\{R, P\}$ to Y_0 , where R is the $2N + 1$ boundary reference pixels and P is the intra block copy prediction. Intra block copy is obtained through searching in a wider range from the reconstructed regions while the boundary reference pixels can be leveraged to extract local prediction. Intuitively, convolutional neural networks are expected to learn a better projection by combining the local and non-local information. To make the network easier to train, the pixel values are normalized to range $[0, 1]$. Let us denote the depth of the fully-connected layers and that of the convolutional neural networks as d_f and d_c respectively. The input of the FC layers is the $2N + 1$ boundary reference pixels and outputs a vector contains N^2 pixel values which will be reshaped to a $N \times N$ block. For the i th FC layer, its output is a vector in K_i -dimensional, and it is calculated as.

$$F_i^F(x) = W_i^F \cdot x_i^F + b_i^F, \quad 1 \leq i \leq d_f \quad (4.10)$$

where W_i^F and b_i^F represents weights and biases. x_i^F is the input vector of the i th FC layer. When i equals to one, x_1^F is the input reference pixels in $2N + 1$ dimensional. Each FC layer is followed by a non-linear activation layer.

The output of FC layers is reshaped to a $N \times N$ block M . Element-wise summation is applied on M and intra block copy P followed by convolutional neural networks. Each convolutional layer is followed by a non-linear activation layer which is not counted. Given the previous layer's output x_j^C , current layer feature map F_j^C is calculated as follows.

$$F_j^C = W_j^C \cdot x_j^C + b_j^C, \quad 1 \leq j \leq d_c \quad (4.11)$$

where W_j^C and b_j^C are the weights and biases of current layer. x_j^C is the output of previous layer. When $j = 1$, namely the first layer, the input should be the summation of FC output M and intra block copy P . Both FC layers and convolutional layers adopt Rectified linear unit (ReLU) as the activation function. Residual learning is adopted in the proposed method due to its fast convergence property. Intra block copy is added to the end of CNN.

To better control the trade-off between the IBC prediction accuracy and the computational complexity. The process of IBC is parametrized by search range r . Δx and Δy denote the displacements from current block coordinates along x and y axis respectively. The values of Δx and Δy should guarantee the IBC reference block is within current

available pixel regions.

$$r \leq |\Delta x| + |\Delta y| \quad (4.12)$$

Training

The training process is actually estimating the network parameter $\Theta = \{W^F, b^F, W^C, b^C\}$ with predefined loss function. The objective of this network is to learn the end-to-end mapping from (R, P) to Y_0 , i.e.,

$$f(R, P, \Theta) = Y \quad (4.13)$$

This is achieved by minimizing the loss between the predicted block Y and the corresponding ground truth block Y_0 . Euclidean loss is adopted in this work due to its simplicity and popularity. To avoid over-fitting during the training procedure, regularization term is added to the loss function. Suppose there are totally S training pairs, the loss function is formulated as follows.

$$L(\Theta) = \frac{1}{2S} \sum_{s=1}^S \|Y^s - Y_0^s\|_2^2 + \frac{\lambda}{2} \|\Theta\|_2^2 \quad (4.14)$$

where λ is the regularization term and is set to 10^{-4} in implementation. Stochastic gradient descent (SGD) is utilized to minimize the loss while training. SGD updates the parameter set Θ by combining current gradient and previous parameter update. Specifically, update at iteration $t + 1$ can be expressed as:

$$V_{t+1} = uV_t - \alpha \nabla L(\Theta) \quad (4.15)$$

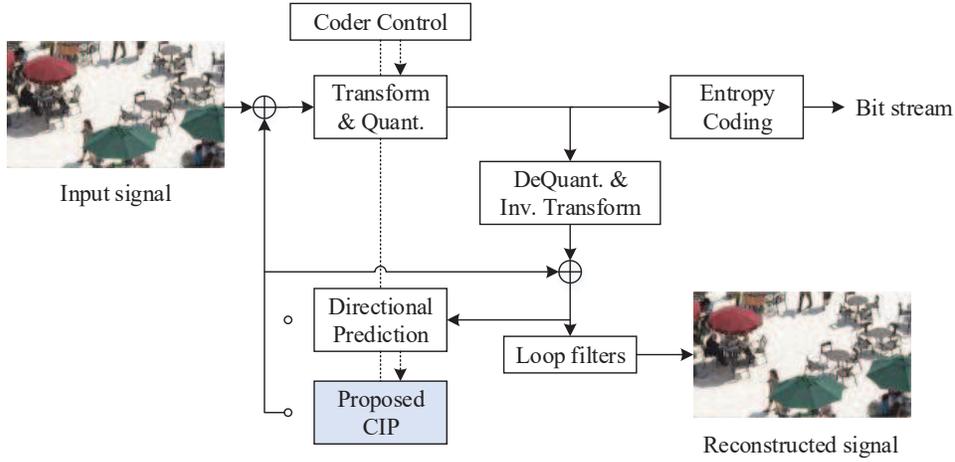


Figure 21: Illustration of integrating CIP model into HEVC.

$$\Theta_{t+1} = \Theta_t + V_{t+1} \quad (4.16)$$

where $\nabla L(\Theta)$ are gradients with respect to the parameters Θ to be updated. V_t is previous parameter update. α is the learning rate and u is the momentum. Θ is initialized by random Gaussian distribution with zero mean and standard deviation of 1. Momentum u is set as 0.9 and the learning rate α is set to decay exponentially from 10^{-4} to 10^{-9} by a factor of 10^{-1} .

Integration into HM

In this work, we design the CIP model cooperate with existing directional intra prediction model in HEVC coding framework, and rate-distortion optimization is used to choose the optimal model. A binary flag will be transmitted to indicate whether CIP is adopted. As shown in Fig. 21, the proposed method is plugged after the existing intra prediction process. An optimal intra prediction mode will be obtained after the 35 intra prediction mode decision. The proposed CIP is performed after that and the optimal prediction mode is updated accordingly. If CIP mode is adopted, corresponding motion

vector will be transmitted to decoder. The decoder will first check whether CIP is used. If CIP is used, motion vector will be decoded followed by the CIP model, otherwise conventional directional intra decoding process is performed.

4.4.3 Experiments

The proposed method is integrated into HM 16.0 [116] and the results are compared with HM-16.0 anchor. The preliminary experiments will only show block size of 16×16 . In this section, experimental settings are introduced first. Training data derivation is detailed afterwards. Finally, the experimental results and analysis are shown.

Experimental Settings

In HEVC, the block-based intra prediction follows a quadtree pattern resulting the CU size ranging from 8 to 64. As block size 64 is too large to find an ideal intra block copy prediction, we constraint the block size ranging from 8 to 32. The preliminary work only implements block size of 16 and the other sizes will be left for future work. All-intra configuration suggested by HEVC common test condition is adopted. Quantization parameter (QP) is set to $\{22, 27, 32, 37\}$. The CIP model is implemented in caffe [121]. The base learning rate is set to 0.0001 and decay exponentially every 100k iterations. The total number of iterations is 1.5M and it takes about 20 hours on a GTX 1080Ti GPU.

Training Data Preparation

DIV2K 2K resolution high quality image dataset [122] is used to generate the training data. There are 1000 high definition high resolution images among which 800 images for training, 100 images for validation and another 100 images for test. The

images content covers a wide range of objects.

Data preparation is to find the input pair $\{R, P\}$. The reference pixels R is extracted from the original image. The IBC search range is empirically set to $r = 128$ to balance the time complexity and matching accuracy. The IBC block is searched pixel by pixel within the available pixels according to Eq. 4.12. The difference is measured by Sum of Absolute Difference (SAD) and the block which has the minimum SAD from the ground truth is selected as current IBC prediction. To refine the IBC training samples, mean square error (MSE) is used to measure the distance between IBC to corresponding ground truth. We only keep those IBC with a MSE smaller than 0.5, in such a way, about 55% training samples are preserved.

Results and Analysis

Three different kinds of fusion methods have been tried in the proposed method, e.g., element-wise addition, 1x1 Convolution and Concatenation. The simulation results indicate that the element-wise addition is the optimal fusion in our method. Therefore, the final results are implemented with element-wise addition.

The BD-Rate reduction of all the test sequences are listed in Table 20. In addition, we also show the average results for each class. An average of -1.3%, -1.4% and -1.6% BD-Rate saving are achieved by the proposed method for luma, Cb and Cr component respectively. The peak performance on luma component is on BQTerrace with -3.6% BD-Rate reduction. Overall, the performance of proposed method on chroma component is consistent with that on luma component. And the best performance on Cb and Cr is on

Table 20: The BD-Rate results of CIP model.

| Sequence | | BD-Rate | | |
|----------|-----------------|---------|-------|-------|
| | | Y | U | V |
| Class A | Traffic | -1.5% | -0.8% | -0.7% |
| | PeopleOnStreet | -2.1% | -2.8% | -2.6% |
| | Nebuta | -0.4% | -0.7% | -0.4% |
| | SteamLocomotive | -0.1% | 0.9% | -0.1% |
| Class B | Kimono | 0.3% | -0.2% | -0.3% |
| | ParkScene | -0.6% | -1.1% | -1.0% |
| | Cactus | -2.0% | -2.7% | -1.8% |
| | BQTerrace | -3.6% | -2.5% | -4.5% |
| | BasketballDrive | -2.8% | -5.7% | -4.2% |
| Class C | BasketballDrill | -2.8% | -4.5% | -3.6% |
| | BQMall | -0.8% | 0.1% | -0.7% |
| | PartyScene | -0.7% | 0.2% | 0.1% |
| | RaceHorsesC | -0.2% | -0.4% | 0.2% |
| Class D | BasketballPass | -0.7% | -0.7% | -3.4% |
| | BQSquare | -0.9% | -0.3% | 0.1% |
| | BlowingBubbles | 0.2% | -1.0% | -1.0% |
| | RaceHorses | 0.3% | 1.3% | 0.9% |
| Class E | FourPeople | -1.3% | 0.5% | -0.3% |
| | Johnny | -2.7% | -2.3% | -3.9% |
| | KristenAndSara | -2.1% | -4.9% | -2.5% |
| Class A | | -1.1% | -0.8% | -1.0% |
| Class B | | -1.8% | -2.3% | -2.4% |
| Class C | | -1.2% | -1.2% | -1.0% |
| Class D | | -0.2% | -0.2% | -0.8% |
| Class E | | -2.0% | -2.2% | -2.6% |
| Average | | -1.3% | -1.4% | -1.6% |
| Enc Time | | 794% | | |
| Dec Time | | 150% | | |

sequence BasketballDrive and BQTerrace with -5.7% and -4.5% BD-Rate reduction respectively. The encoding time is much higher than anchor and decoding time is 50% than anchor. Fortunately, the encoding time does not count that much compared with decoding time complexity in video coding. It is noticeable that the proposed method does not perform very well on Class D which might be caused by the training dataset not covering similar content. It should be noticed that this version is our preliminary benchmark and the same CIP model is shared among different block sizes. In addition, different dataset also has different impacts on the final performance. All possible improvement work will be left for the future work.

4.5 Discussion

Upon the completion of this thesis, the MPEG next-generation video coding standard was just released. In VVC, a variety of advanced techniques are incorporated. The major updates on intra prediction compared to HEVC include

- 67 Intra prediction modes
- Cross-component linear model prediction
- Position dependent intra prediction combination
- Multiple reference line (MRL) intra prediction
- Intra sub-partitions (ISP)
- Matrix weighted intra prediction (MIP)

Starting from [56], the LBC methods have made significant progress on image compression tasks. One of the most fundamental problems that was solved is using additive noise to stimulate the “quantization” process that exists in classical video coding solution. This enables the back-propagation possible in CNN pipeline. With following efforts [57,96] to improve on top of [56], the compression performance has outperformed HEVC intra prediction. The achievements indicate using ANNs for image compression has become much more matured and has great potentials for real applications. However, the LBC methods have not been validated with formally defined test conditions thus still not ready for real applications. The following issues need to be resolved

- Variable bit-rate control
- Decoding time complexity
- Unpredictable output

Although there is Lagrangian Multiplier λ involved in the loss function to enable the Rate-Distortion Optimization (RDO) process, it can only control a relative compression ratio. It fails to support the derivation of input coding parameters that result in a given bit-rate bitstream. From our experiments, the decoding time complexity is another major issue needs to be resolved. Moreover, the LBC model should be able to achieve applicable decoding time complexity with only CPU is available. In addition, LBC methods should be able to provide a lower bound for the outcome, i.e., what might be the worse case given all input parameters.

In summary, the major techniques to boost LBC methods include additive noise quantization [56], GDN [99], hyper-prior [57] and context-adaptive model [96]. Existing state-of-the-art LBC method is able to achieve superior compression performance compared with HEVC intra coding. But Common Test Condition is suggested to formally defined to regulate the development of LBC methods towards real applications. Efforts on making it explainable is also very crucial to measure the risk at using LBC methods for certain unpredictable scenarios. In contrast, the classical video codec still dominates this field even only marginal improvements can be obtained with a single technique. The ensemble of various techniques with solid theory foundations follows its predecessors' architecture and will continue to serve for the society.

CHAPTER 5

CONCLUSION

In this dissertation, we gave readers a thorough overview of compression for machine vision, along with how we contributed to the development of this field.

In Chapter 1, we walked through the history of compression, including video coding standards and video coding for machines (VCM). The classical video coding concepts were defined as early as in 1980s and video coding standards have been evolving for decades serving the human society. Each generation video coding standard includes new techniques than its predecessor with substantial coding gain to adapt to the increment of media content. Benefit from the advanced hardware such as the smart mobile devices with high definition acquisition module equipped, it is time to explore state-of-the-art technologies for processing the massive media content. On one hand, JVET started working on development of the next-generation video coding standard since 2015. On the other hand, artificial neural networks have been applied to video coding and image compression tasks. It is of great significance to contributing to VVC development, as well as exploring the potentials by using ANNs for future video coding.

In Chapter 2, we present an effective framework to produce more discriminative image representations for image retrieval with SIFT feature. We devise a non-linear data partition tree architecture to divide a large-scale training dataset into local patches. The proposed scheme is dedicated to an exploration of capturing the latent characteristics at

multiple scales, different numbers of local spaces are generated in each level. The affinity relationship is defined to enable optimization for effective transformation search. To search for the optimal local transformations, Grassmann manifold is adopted to prune the data partition tree. Instead of traversing all the possibilities on the data partition tree, we incorporate Grassmann metric to measure their distinctiveness thus the most representative candidates will be obtained. A new projection and matching metric is devised which involves the local space each feature belongs to. This actually guarantees the fairness of distance computation. Extensive experiments are performed to evaluate the proposed method, including pairwise matching and large-scale image retrieval. Theoretical analysis, as well as empirical evidence, are provided to validate the necessity of the proposed data partition tree from the information theory perspective.

In Chapter 3, we contributed to the development of next-generation video coding standard, versatile video coding. We present a fast DST-VII/DCT-VIII method with dual-implementation support for Multiple Transform Selection (MTS). The inherent partial butterfly-type features are exploited to reduce the number of arithmetic operations. A fast DST-VII/DCT-VIII method is devised by using these features which achieves an approximate of 50% arithmetic operations with dual-implementation support, *i.e.*, either full matrix multiplication or fast partial butterfly-type implementation. Complexity analysis is performed to validate the effectiveness in terms of the number of operation counts and software run-time. In addition, the theoretical proof is provided to demonstrate that these beneficial features are tenable in theory. The proposed scheme was adopted in the 13th JVET Meeting of ITU-T VCEG and ISO/IEC MPEG in January 2019.

In Chapter 4, we present the efforts on improving HEVC intra prediction scheme. A multiple linear regression approach is proposed by enhancing intra prediction block with the reference pixels. To refine the model, separate models are trained according to their intra prediction directional information. This guarantees that the linear regression model can capture more useful correlations from the directional patterns. It shares a very neat and concise design yet achieves promising performance. In regards of ANN attempts, we first perform performance analysis between the ANN-based approaches and state-of-the-art MPEG codecs. Overall, the ANN methods achieve competitive performance with state-of-the-art codec VTM. But the results also reveal the data-dependent nature of ANN methods. To better facilitate research in ANN area, it is important to establish a common test data set and Common Test Condition (CTC), such that different approaches and schemes can have an objective and common metric to evaluate, and eventually driving the research to some useful ends. Next we propose a convolutional neural network method by combining the local and non-local information to predict current block. The proposed CNN is fed with both the neighboring pixels and the best prediction found by intra block copy. Different methods of combining intra block copy and boundary reference pixels have been investigated. To make the proposed CNN more tractable and efficient, training data is refined by empirically setting a threshold to remove distractors. Overall, in classical transform coding methods, compression researchers have exploited statistical dependency in the latent variables by carefully hand-engineering entropy codes modeling the dependencies in the quantized regime. While ANN approaches rely on a fully factorized entropy model whose parameters are determined through training.

In summary, this thesis contributes to this area in the following aspects.

- Chapter 2 provides an efficient multiple-transform solution for feature compression.
- Chapter 3 presents a fast transform solution for VVC.
- Chapter 4 exploits relevant techniques to improve HEVC intra prediction as well as potentials of using ANN methods for compression.

The remaining issues and future research directions include:

- Feature compression for multi-task learning.
- ANN-based compression complexity reduction and rate control enabled scheme.
- End-to-end video compression with ANN methods.

All together, we are really excited about the progress that has been made during the past 3 years and have been glad to be able to contribute to this field. We believe there are more unknown surprises for classical video coding standards in the future. We also deeply believe that there is still a long way for ANN approaches to go towards real applications. There are still enormous challenges and a lot of open questions that we need to address in the future. The most obvious challenges of ANN approaches include decoder complexity, rate control, data-dependent nature.

We also hope to encourage more researchers to work on this field, or apply video coding technology to new domains or tasks. We believe that it will lead us towards building better compression solutions and hope to see these ideas implemented in industry.

REFERENCE LIST

- [1] Cisco. Cisco visual networking index: Forecast and trends, 2017–2022 white paper, 2019.
- [2] L. Duan, V. Chandrasekhar, J. Chen, J. Lin, Z. Wang, T. Huang, B. Girod, and W. Gao. Overview of the mpeg-cdvs standard. *IEEE Transactions on Image Processing*, 25(1):179–194, Jan 2016.
- [3] S. Paschalakis. Cdvs ce2: Local descriptor compression proposal. *ISO/IEC JTC1/SC29/WG11/M25929*, Jul.
- [4] Z. Zhang, L. Li, Z. Li, and H. Li. Visual query compression with locality preserving projection on grassmann manifold. *IEEE International Conference on Image Processing*, Sep 2017.
- [5] Z. Zhang, L. Li, and Z. Li. Visual query compression with embedded transforms on grassmann manifold. *IEEE International Conference on Multimedia and EXPO Workshop*, Jul 2017.
- [6] Instagram. Instagram photos uploaded in 1 second, <http://www.internetlivestats.com/one-second/instagram-band>, 2017.
- [7] Snapchat, <https://www.bloomberg.com/technology>, 2017.

- [8] W. Tan, B. Yan, and C. Lin. Beyond visual retargeting: a feature retargeting approach for visual recognition and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–1, 2017.
- [9] T. Gabriel, C. Vijay, G. Natasha, X. Yingen, C. Wei-Chao, B. Thanos, G. Radek, P. Kari, and G. Bernd. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, MIR '08, pages 427–434, New York, NY, USA, 2008. ACM.
- [10] T. Sam S., C. David, S. J. Pal, and G. Bernd. Rate-efficient, real-time cd cover recognition on a camera-phone. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 1023–1024, New York, NY, USA, 2008. ACM.
- [11] A. Sarkar, V. Singh, P. Ghosh, B. S. Manjunath, and A. Singh. Efficient and robust detection of duplicate videos in a large database. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(6):870–885, June 2010.
- [12] B. Girod, V. Chandrasekhar, D. M. Chen, N. M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham. Mobile visual search. *IEEE Signal Processing Magazine*, 28(4):61–76, July 2011.
- [13] V. R. Chandrasekhar, D. M. Chen, S. S. Tsai, N. Cheung, H. Chen, G. Takacs, Y. Reznik, R. Vedantham, R. Grzeszczuk, J. Bach, and B. Girod. The stanford mobile visual search data set. In *Proceedings of the Second Annual ACM Conference*

- on Multimedia Systems*, MMSys '11, pages 117–122, New York, NY, USA, 2011. ACM.
- [14] X. Song, X. Peng, J. Xu, G. Shi, and F. Wu. Cloud-based distributed image coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(12):1926–1940, Dec 2015.
- [15] Google. Google-goggles, <http://www.google.com/mobile/goggles/>, 2017.
- [16] B. Erol, E. Antúnez, and J. J. Hull. Hotpaper: Multimedia interaction with paper using mobile phones. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 399–408, New York, NY, USA, 2008. ACM.
- [17] Amazon. Snaptell, <http://www.snaptell.com>, 2007.
- [18] Layar. Layar, <http://www.layar.com>, 2010.
- [19] Y. Jing, D. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel. Visual search at pinterest. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1889–1898, New York, NY, USA, 2015. ACM.
- [20] Q. Liu, J. Fan, H. Song, W. Chen, and K. Zhang. Visual tracking via nonlocal similarity learning. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–1, 2017.

- [21] C. Lee, C. E. Rhee, and H. J. Lee. Complexity reduction by modified scale-space construction in sift generation optimized for a mobile gpu. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(10):2246–2259, Oct 2017.
- [22] J. Jiang, X. Li, and G. Zhang. Sift hardware implementation for real-time image feature extraction. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(7):1209–1220, July 2014.
- [23] F. C. Huang, S. Y. Huang, J. W. Ker, and Y. C. Chen. High-performance sift hardware accelerator for real-time image feature extraction. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(3):340–351, March 2012.
- [24] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, Y. Reznik, R. Grzeszczuk, and B. Girod. Compressed histogram of gradients: A low-bitrate descriptor. *International Journal of Computer Vision*, 96(3):384–399, 2012.
- [25] D. M. Chen and B. Girod. Memory-efficient image databases for mobile visual search. *IEEE MultiMedia*, 21(1):14–23, Jan 2014.
- [26] D. M. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs, J. Singh, and B. Girod. Tree histogram coding for mobile image matching. In *2009 Data Compression Conference*, pages 143–152, March 2009.
- [27] R. Ji, L. Duan, J. Chen, H. Yao, Y. Rui, S. Chang, and W. Gao. Towards low bit rate mobile visual search with multiple-channel coding. In *Proceedings of the*

- 19th ACM International Conference on Multimedia, MM '11*, pages 573–582, New York, NY, USA, 2011. ACM.
- [28] R. Ji, L. Duan, J. Chen, H. Yao, J. Yuan, Y. Rui, and W. Gao. Location discriminative vocabulary coding for mobile landmark search. *International Journal of Computer Vision*, 96(3):290–314, 2012.
- [29] J. Lin, L. Y. Duan, Y. Huang, S. Luo, T. Huang, and W. Gao. Rate-adaptive compact fisher codes for mobile visual search. *IEEE Signal Processing Letters*, 21(2):195–198, Feb 2014.
- [30] J. Chao, R. Huitl, E. Steinbach, and D. Schroeder. A novel rate control framework for sift/surf feature preservation in h.264/avc video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(6):958–972, June 2015.
- [31] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 19(5):530–535, 1997.
- [32] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, Apr 2002.
- [33] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.
- [34] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, October 2005.

- [35] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.
- [36] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.
- [37] B. Fan, F. Wu, and Z. Hu. Aggregating gradient distributions into intensity orders: A novel local image descriptor. In *CVPR 2011*, pages 2377–2384, June 2011.
- [38] G. Takacs, V. Chandrasekhar, S. S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod. Fast computation of rotation-invariant image features by an approximate radial gradient transform. *IEEE Transactions on Image Processing*, 22(8):2970–2982, Aug 2013.
- [39] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. *BRIEF: Binary Robust Independent Elementary Features*, pages 778–792. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [40] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, Nov 2011.

- [41] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, June 2012.
- [42] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.
- [43] A. Araujo and B. Girod. Large-scale video retrieval using image queries. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–1, 2017.
- [44] L. Sun and G. Liu. Visual object tracking based on combination of local description and global representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(4):408–420, April 2011.
- [45] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, May 2010.
- [46] L. Y. Duan, V. Chandrasekhar, J. Chen, J. Lin, Z. Wang, T. Huang, B. Girod, and W. Gao. Overview of the mpeg-cdvs standard. *IEEE Transactions on Image Processing*, 25(1):179–194, Jan 2016.

- [47] S. Paschalakis, K. Wnukowicz, M. Bober, A. Mosca, M. Mattelliano, G. Francini, S. Lepsoy, and M. Balestri. Local descriptor compression proposal. *Inputs: ISO/IEC JTC1/SC29/WG11 MPEG2012/M25929*, Jul 2012.
- [48] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.
- [49] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV, ECCV'10*, pages 143–156, Berlin, Heidelberg, 2010. Springer-Verlag.
- [50] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, Mar 1982.
- [51] E. Forgy. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, 21(3):768–769, 1965.
- [52] T. Hoang, T. Do, D. L. Tan, and N. Cheung. Selective deep convolutional features for image retrieval. In *ACM Multimedia*, 2017.
- [53] J. Hamm and D. Lee. Grassmann discriminant analysis: A unifying view on subspace-based learning. In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, pages 376–383, New York, NY, USA, 2008. ACM.

- [54] J. Hamm. *Subspace-based Learning with Grassmann Kernels*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 2008. AAI3328568.
- [55] T. Wang and P. Shi. Kernel grassmannian distances and discriminant analysis for face recognition from image sets. *Pattern Recognition Letters*, 30(13):1161 – 1165, 2009.
- [56] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. *CoRR*, abs/1611.01704, 2016.
- [57] J. Ballé, D. Minnen, S. Singh, S. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.
- [58] N. Yan, D. Liu, H. Li, T. Xu, F. Wu, and B. Li. Convolutional neural network-based invertible half-pixel interpolation filter for video coding. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 201–205, Oct 2018.
- [59] N. Yan, D. Liu, H. Li, B. Li, L. Li, and F. Wu. Convolutional neural network-based fractional-pixel motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(3):840–853, March 2019.
- [60] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao. DVC: an end-to-end deep video compression framework. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 11006–11015, 2019.

- [61] N. Yan, D. Liu, H. Li, and F. Wu. A convolutional neural network approach for half-pel interpolation in video coding. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, May 2017.
- [62] Z. Zhang, X. Zhao, X. Li, Z. Li, and S. Liu. Fast adaptive multiple transform for versatile video coding. In *2019 Data Compression Conference (DCC)*, pages 63–72, March 2019.
- [63] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, Dec 2012.
- [64] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, July 2003.
- [65] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Trans. Comput.*, 23(1):90–93, January 1974.
- [66] X. Zhao, J. Chen, M. Karczewicz, A. Said, and V. Seregin. Joint separable and non-separable transforms for next-generation video coding. *IEEE Transactions on Image Processing*, 27(5):2514–2525, May 2018.
- [67] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li, and W. Chien. Enhanced multiple transform for video coding. In *2016 Data Compression Conference (DCC)*, pages 73–82, March 2016.

- [68] K. Karhunen. *Ueber lineare Methoden in der Wahrscheinlichkeitsrechnung*. Annales Academiae scientiarum Fennicae. Series A. 1, Mathematica-physica. 1947.
- [69] M. Loeve. *Probability Theory II*. F.W.Gehring P.r.Halmos and C.c.Moore. Springer, 1978.
- [70] R.J. Clarke. Relation between the karhunen loève and cosine transforms. *IEEE Proceedings F (Communications, Radar and Signal Processing)*, 128:359–360(1), November 1981.
- [71] X. Zhao, L. Li, Z. Li, X. Li, and S. Liu. Coupled primary and secondary transform for next generation video coding. In *2018 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, Dec 2018.
- [72] R. H. Bamberger and M. J. T. Smith. A filter bank for the directional decomposition of images: theory and design. *IEEE Transactions on Signal Processing*, 40(4):882–893, April 1992.
- [73] J. C. Emmanuel and D. Donoho. Ridgelets: A key to higher-dimensional intermittency? *R Soc Lond Philos Trans Ser A Math Phys Eng Sci*, 357, 12 2000.
- [74] J. Starck, E. J. Candes, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, June 2002.
- [75] M. N. Do and M. Vetterli. The contourlet transform: An efficient directional multiresolution image representation. *Trans. Img. Proc.*, 14(12):2091–2106, December 2005.

- [76] B. Zeng and J. Fu. Directional discrete cosine transforms—a new framework for image coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18:305 – 313, 04 2008.
- [77] X. Zhao, L. Zhang, S. Ma, and W. Gao. Video coding with rate-distortion optimized transform. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(1):138–151, Jan 2012.
- [78] X. Cao and Y. He. Singular vector decomposition based adaptive transform for motion compensation residuals. *2014 IEEE International Conference on Image Processing, ICIP 2014*, pages 4127–4131, 01 2015.
- [79] Y. Ye and M. Karczewicz. Improved h.264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning. *2008 15th IEEE International Conference on Image Processing*, pages 2116–2119, 2008.
- [80] X. Zhao, L. Zhang, S. Ma, and W. Gao. Rate-distortion optimized transform for intra-frame coding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, 14-19 March 2010, Sheraton Dallas Hotel, Dallas, Texas, USA*, pages 1414–1417, 2010.
- [81] B. Bross, J. Chen, and S. Liu. Versatile video coding (draft 2). *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-K1001(Ljubljana, SI):10â18*, July 2018.

- [82] J. Chen, Y. Ye, and S.H. Kim. Algorithm description for versatile video coding and test model 2 (vtm 2). *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, JVET-K1002-v2(Ljubljana, SI):10â18, July 2018.
- [83] H. Gao, S. Esenlik, B. Wang, A. M. Kotra, J. Chen (Huawei), H. E. Egilmez, A. K. Ramasubramonian, A. Said, N. Hu, V. Seregi, G. Van der Auwera, M. Karczewicz (Qualcomm), S.-C. Lim, J. Kang, J. Lee, H. Lee, and H. Y. Kim (ETRI). Non-ce6: Combined test of jvet-n0172/jvet-n0375/jvet-n0419/jvet-n0420 on unification of implicit transform selection. *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, JVET-N0866(Geneva, CHE):10â18, March 2019.
- [84] A. K. Jain. A sinusoidal family of unitary transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(4):356–365, Oct 1979.
- [85] S. A. Martucci. Symmetric convolution and the discrete sine and cosine transforms. *IEEE Transactions on Signal Processing*, 42(5):1038–1051, May 1994.
- [86] Z. Zhang, X. Zhao, X. Li, and S. Liu. Ce6-related: Fast dst-7/dct-8 with dual implementation support. *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, JVET-K0291(Ljubljana, SI):10â18, July 2018.
- [87] X. Zhao, X. Li, Y. Luo, and S. Liu. Ce6: Fast dst-7/dct-8 with dual implementation support (test 6.2.3). *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 13*, JVET-M0497(Marrakech, MA):10â18, Jan 2019.

- [88] JVET VVC. Versatile video coding (vvc) reference software: Vvc test model (vtm). https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tree/VTM-3.0, May 2019.
- [89] J. Boyce, K. Suehring, X. Li, and V. Seregin. Jvet common test conditions and software reference configurations. *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, JVET-J1010(San Diego, US):10â20, April 2018.
- [90] M. Koo, M. Salehifar, J. Lim, and S. Kim. Ce6: Fast dst-7/dct-8 based on dft (test 6.2.1). *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 13*, JVET-M0288(Marrakech, MA):10â18, Jan 2019.
- [91] A. Said, H.E. Egilmez, Y.-H. Chao, V. Seregin, and M. Karczewicz. Ce6: Efficient implementations of mts with transform adjustments (tests 1.4a-d). *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 13*, JVET-M0538(Marrakech, MA):10â18, Jan 2019.
- [92] P. Philippe. Ce6: Mts simplification with transform adjustment (taf) (tests 1.5a-d). *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 13*, JVET-M0080(Marrakech, MA):10â18, Jan 2019.
- [93] G. Bjontegaard. Calculation of average psnr differences between rd-curves. *ITU-T SG16/Q6*, Doc. VCEG-M33(Austin):10â18, Apr. 2001.

- [94] G. Bjontegaard. Improvement of bd-psnr model. *ITU-T SG16/Q6*, Doc. VCEG-A111(Berlin, Germany):10â18, Jul. 2008.
- [95] G. Toderici, D. Vincent, N. Johnston, S. Hwang, D. Minnen, J. Shor, and M. Covell. Full resolution image compression with recurrent neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [96] J. Lee, S. Cho, and S. Beack. Context-adaptive entropy model for end-to-end optimized image compression. In *International Conference on Learning Representations*, 2019.
- [97] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and V. Luc. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [98] M. Wainwright and E. Simoncelli. Scale mixtures of gaussians and the statistics of natural images. In *Proceedings of the 12th International Conference on Neural Information Processing Systems, NIPS'99*, pages 855–861, Cambridge, MA, USA, 1999. MIT Press.
- [99] J. Ballé, V. Laparra, and E. Simoncelli. Density modeling of images using a generalized normalization transformation. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

- [100] J. Ballé. Efficient Nonlinear Transforms for Lossy Image Compression. *arXiv e-prints*, page arXiv:1802.00847, Jan 2018.
- [101] D. Minnen, J. Ballé, and G. Toderici. Joint autoregressive and hierarchical priors for learned image compression. pages 10794–10803, 2018.
- [102] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, Dec 2012.
- [103] Z. Zhang, X. Zhao, X. Li, Z. Li, and S. Liu. Fast adaptive multiple transform for versatile video coding. In *2019 Data Compression Conference (DCC)*, pages 63–72, March 2019.
- [104] M. Li, W. Zuo, S. Gu, J. You, and D. Zhang. Learning content-weighted deep image compression. *CoRR*, abs/1904.00664, 2019.
- [105] M. Akbari, J. Liang, and J. Han. DSSLIC: deep semantic segmentation-based layered image compression. *CoRR*, abs/1806.03348, 2018.
- [106] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and V. Luc. Generative adversarial networks for extreme learned image compression. *arXiv preprint arXiv:1804.02958*, 2018.
- [107] Mu Li, Kede Ma, Jane You, David Zhang, and Wangmeng Zuo. Efficient and Effective Context-Based Convolutional Entropy Modeling for Image Compression. *arXiv e-prints*, page arXiv:1906.10057, Jun 2019.

- [108] B. Bross, J. Chen, and S. Liu. Versatile video coding (draft 5). *14th Meeting, Geneva, Switzerland, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, pages JVET–N1001–v9, March 2019.
- [109] Raphaël Zumer. Image compression with neural networks, Mar 2018.
- [110] J. Ballé, S. Hwang, and N. Johnston. Data compression in tensorflow, May 2019.
- [111] J. Ballé, D. Minnen, S. Singh, J. Sung, and N. Johnston. Variational image compression with a scale hyperprior. *CoRR*, abs/1802.01436, 2018.
- [112] fab jul. Tensorflow implementation of conditional probability models for deep image compression, Mar 2019.
- [113] Z. Zhang, Y. Li, L. Li, Z. Li, and S. Liu. Multiple linear regression for high efficiency video intra coding. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1832–1836. IEEE, 2019.
- [114] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin. Jvet common test conditions and software reference configurations for sdr video. *14th Meeting, Geneva, Switzerland, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, pages JVET–N1010–v1, March 2019.
- [115] Y. Li, L. Li, Z. Li, and H. Li. Hierarchical piece-wise linear projections for efficient intra-prediction coding. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, Dec 2017.

- [116] HEVC. Hvc reference software 16.0, 2018.
- [117] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.
- [118] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018.
- [119] Y. Li, D. Liu, H. Li, L. Li, F. Wu, H. Zhang, and H. Yang. Convolutional neural network-based block up-sampling for intra frame coding. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2017.
- [120] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao. Fully connected network-based intra prediction for image coding. *IEEE Transactions on Image Processing*, 27(7):3236–3247, July 2018.
- [121] Berkeley AI Research. Caffe, 2018.
- [122] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1122–1131, July 2017.

VITA

Zhaobin Zhang was born on May 17, 1990 in Shijiazhuang, Hebei. He received the B.S. and M.S. degrees in Mechanical and Electronic Engineering from Huazhong University of Science and Technology (HUST), Wuhan, Hubei, China, in 2012 and 2015, respectively.

He is currently pursuing his Ph.D. degree in Electrical and Electronics Engineering, UMKC. He interned with Dolby Laboratories (2020) and Tencent Media Lab (2018, 2019). His research interests include video coding, image compression and machine learning.