The Dissertation Committee for Meixu Chen
certifies that this is the approved version of the following dissertation:

Virtual Reality: Quality and Compression

Committee:

Alan C. Bovik, Supervisor

Haris Vikalo

Wilson S. Geisler

Joydeep Ghosh

Todd Goodall Bell

Virtual Reality: Quality and Compression

by

Meixu Chen

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2022

Dedicated to my grandparents.

Acknowledgments

I wish to thank the multitude of people who have helped and guided me during my PhD.

First of all, I would like to thank my advisor, Prof. Bovik. I still remember the time when I was a first year master student and attended his lecture. His enthusiasm for image and video was very infectious and the lectures were very stimulating and brilliantly taught. He introduced me to the magnificant world of image and video which I later focus my research on. I am lucky to have him as my advisor after my second year and begin my research in image and video from then on. Being an advisor, he is both diligent in his work and considerate in life. On one hand, he believes the potential of his students and sets high bars to help us being the best that we could be. At the same time, he deeply cares about the emotional status of us and ensures that we enjoy the overall journey while accomplishing our goal. He is my role model both academically and in life.

Next I would like to thank all past and present members of LIVE for their support and the fun times that we had: Hamid, Kalpana, Shalini, Michele, Lark, Janice, Zeina, Praful, Deepti, Leo, Christos, Todd, Jerry, Zhen-Qiang, Yize, Sahar, Haoran, Somdyupti, Sungsoo, Li-Heng, Zhengzhong, Dae Yeol, Pavan, Maniratnam, Joshua, Zaixi, Abhinau, Chengyang, and Avinab. I

am honored to be LIVE member where our past members pushed the boundaries in research and industry and I enjoy the vivid discussions we have with the current members.

I am fortunate to have the opportunity to work with our industry partners: Todd, Anjul, Richard and Cheng. They provided me with the great opportunity to do research in a hybrid setting, both exploring new ideas in the academia and considering the impact the research would create in the industry. Thanks to this opportunity, I have learnt to think about ideas deeper both theoretically and pratically.

My boyfriend has supported me throughout this whole journey and I am grateful to have him by my side through my up and downs. This journey is not easy, but he makes it a more enjoyable one with his caring and support.

I would like to dedicate this dissertation to my beloved family for their unconditional love and support throughout my life. My parents work diligently to make sure I grow up in a loving and enriching environment. My grandparents raised me and watched me grow from a toddler to who I am today. I am grateful to have them in my life.

Virtual Reality: Quality and Compression

Publication No. _____

Meixu Chen, Ph.D.
The University of Texas at Austin, 2022

Supervisor: Alan C. Bovik

Virtual Reality (VR) and its applications have attracted significant and increasing attention. However, the requirements of much larger file sizes, different storage formats, and immersive viewing conditions pose significant challenges to the goals of acquiring, transmitting, compressing and displaying high quality VR content. Towards meeting these challenges, it is important to be able to understand the distortions that arise and that can affect the perceived quality of displayed VR content. It is also important to develop ways to automatically predict VR picture quality. Meeting these challenges requires basic tools in the form of large, representative subjective VR quality databases on which VR quality models can be developed and which can be used to benchmark VR quality prediction algorithms. Towards making progress in this direction, here we present the results of an immersive 3D subjective image quality assessment study. In the study, 450 distorted images obtained from 15 pristine 3D VR images modified by 6 types of distortion of varying severities were evaluated by 42 subjects in a controlled VR setting. Both the

subject ratings as well as eye tracking data were recorded and made available as part of the new database, in hopes that the relationships between gaze direction and perceived quality might be better understood. We evaluated several publicly available IQA models on the new database, and also report a statistical evaluation of the performances of the compared IQA models.

Another challenge present in VR is rendering 360 videos within the limited bandwidth. Video has become an increasingly important part of our daily digital communication. With the development of higher resolution contents and displays, its significant volume poses significant challenges to the goals of acquiring, transmitting, compressing and displaying high quality video content. In this direction, we propose a new deep learning video compression architecture that does not require motion estimation, which is the most expensive element of modern hybrid video compression codecs like H.264 and HEVC. Our framework exploits the regularities inherent to video motion, which we capture by using displaced frame differences as video representations to train the neural network. In addition, we propose a new space-time reconstruction network based on both an LSTM model and an UNet model, which we call LSTM-UNet. The combined network is able to efficiently capture both temporal and spatial video information, making it highly amenable for our purposes. Our experimental results show that our compression model, which we call the MOtionless VIdeo Codec (MOVI-Codec), learns how to efficiently compress videos without computing motion. Our experiments show that MOVI-Codec outperforms the Low-Delay P (LDP) *veryfast* setting of the

video coding standard H.264 and exceeds the performance of the modern global standard HEVC codec, using the same setting, as measured by MS-SSIM, especially on higher resolution videos. In addition, our network outperforms the latest H.266 (VVC) codec at higher bitrates, when assessed using MS-SSIM, on high resolution videos.

Because of the high bandwidth requirements of VR, there has also been significant interest in the use of space-variant, foveated compression protocols. We have further integrated these techniques to create another end-to-end deep learning video compression framework in addition to MOVI-Codec. Foveation protocols are desirable since, unlike traditional flat-panel displays, only a small portion of a video viewed in VR may be visible as a user gazes in any given direction. Moreover, even within a current field of view (FOV), the resolution of retinal neurons rapidly decreases with distance (eccentricity) from the projected point of gaze. In our learning based approach, we implement foveation by introducing a Foveation Generator Unit (FGU) that generates foveation masks which direct the allocation of bits, significantly increasing compression efficiency while making it possible to retain an impression of little to no additional visual loss given an appropriate viewing geometry. Our experiment results reveal that our new compression model, which we call the Foveated MOtionless VIdeo Codec (Foveated MOVI-Codec), is able to efficiently compress videos without computing motion, while outperforming foveated version of both H.264 and H.265 on the widely used UVG dataset and on the HEVC Standard Class B Test Sequences.

Table of Contents

Acknowledgments **5**

Abstract **7**

List of Tables **13**

List of Figures **14**

Chapter 1. Introduction **17**
1.1 Problem 17
1.2 Perceptual Quality of VR Content 18
1.3 Deep learning-based Video Compression 19
1.4 Foveation in VR Content 21
1.5 Contributions 23

Chapter 2. Study of 3D Virtual Reality Picture Quality **25**
2.1 Background 26
2.1.1 Subjective Quality Assessment 26
2.1.2 Objective Quality Assessment 28
2.2 Details of the Subjective Study 29
2.2.1 Image Capture 29
2.2.2 Test Images 31
2.2.2.1 Gaussian Noise 32
2.2.2.2 Gaussian Blur 33
2.2.2.3 Downsampling 33
2.2.2.4 Stitching Distortion 33
2.2.2.5 VP9 Compression 34
2.2.2.6 H.265 Compression 36
2.2.3 Subjective Testing Design 36

2.2.4 Subjective Test Display 37
2.2.4.1 Eye tracking 38
2.2.4.2 Viewing and Scoring 39
2.2.5 Subjects and Training 40
2.3 Data Analysis 40
2.4 Objective IQA Model Comparison 46
2.4.1 Performance of Objective Methods 48
2.4.2 Statistical Evaluation 51
2.4.3 Analysis of Eye Tracking Data 51
2.4.4 Discussion of Results 55

Chapter 3. Learning to Compress Videos without Computing Motion 57
3.1 Background 58
3.1.1 Deep Image Compression 58
3.1.2 Deep Video Compression 60
3.1.3 Motion Estimation and Motion Compensation 61
3.2 Proposed Method 63
3.2.1 Framework 63
3.2.2 Displacement Calculation Unit (DCU) 66
3.2.3 Displacement Compression Network (DCN) 67
3.2.3.1 Framework 67
3.2.3.2 Quantizer 68
3.2.3.3 Entropy Coding 69
3.2.4 Frame Reconstruction Network (FRN) 70
3.2.5 Training Strategy 71
3.3 Experiments 73
3.3.1 Settings 73
3.3.2 Results 75
3.3.3 Ablation Studies 79
3.3.3.1 Displaced Frame Difference Combination 80
3.3.3.2 Effectiveness of the LSTM-UNet 81
3.3.4 Motion Vector Analysis 82

3.3.5 Model Analysis 84
3.3.6 Discussion 85

Chapter 4. Foveation-based Deep Video Compression without Motion Search 87

4.1 Background 88
4.1.1 Foveated Video Compression 88
4.1.2 Foveated Video Quality Assessment 90
4.2 Proposed Method 92
4.2.1 Framework 92
4.2.2 Displacement Calculation Unit (DCU) 93
4.2.3 Foveation Generator Unit (FGU) 94
4.2.4 Bit Rate Allocation 97
4.2.5 Training Strategy 98
4.3 Experiments 99
4.3.1 Settings 99
4.3.2 Results 101
4.3.2.1 Rate-Distortion Curve 101
4.3.2.2 Latent Representations 103
4.3.2.3 Bit Allocation 105
4.3.3 Discussion 106

Chapter 5. Conclusion and Future Work 109

Bibliography 113

Vita 135

List of Tables

2.1 Viewing Directions, where ϕ represents the zenith angle, and θ represents the azimuth angle 34

2.2 Min, Max and Median SROCC between randomized subject groups for each distortion category 43

2.3 SROCC of IQA Methods 50

2.4 PLCC of IQA Methods 50

2.5 RMSE of IQA Methods 50

2.6 Statistical Significance Matrix based on IQA-DMOS residuals. All statistical tests are performed at 95% confidence. 52

3.1 Resolutions of different datasets used for evaluation 74

List of Figures

2.1 Plots of Spatial Information (SI) and Colorfulness (CF) of the VR images in the LIVE VR IQA Database 30

2.2 Exemplar VR images in the LIVE VR IQA Database 31

2.3 Insta360 Pro Camera 31

2.4 Example of 14 perspective views that were stitched together . 35

2.5 Different levels of stitching distortion. (a)-(c): Images of level 1, 3 and 5 (higher levels indicate more distortion). (d)-(e): Zoomed-in views of (a)-(c). 35

2.6 HTC Vive integrated with the Tobii Pro Eye Tracking system. 38

2.7 Calibration pattern. 38

2.8 Rating bar used in the subjective study 39

2.9 (a) Histogram of DMOS. (b) SROCC between subject ratings and DMOS. 43

2.10 DMOS of all contents for each level of applied distortion. . . . 44

2.11 Confidence intervals of DMOS over all contents for each applied level of distortion. The blue points indicate the maximum and the minimum DMOS for each distortion type and level. The red points indicate the mean DMOS and the blue bars are the 95% confidence intervals. 45

2.12 Scatter plots of all pairs of objective and subjective IQA scores using different IQA algorithms. 'gb' refers to Gaussian blur, 'gn' refers to Gaussian noise, 'ds' refers to downsampling, and 'st' refers to stitching. 49

2.13 Example gaze maps 53

2.14 Frequency of viewing directions. 53

2.15 Example frequency plots of latitude viewing directions for four exemplar contents. 54

2.16 Example frequency plots of longitude viewing directions for four contents. 54

2.17 Example gaze maps on different distorted versions of a same content. 54

3.1 The overall network architecture of MOVI-Codec, which consists of three components: a Displacement Calculation Unit, a Displacement Compression Network and a Frame Reconstruction Network. 64
3.2 Concept of displaced frame differences, showing a frame t and previous frame $t-1$, and multiple spatially displaced versions of frame $t-1$. 67
3.3 Flow diagram of the Displacement Compression Network. The left side shows the displacement autoencoder architecture, and the right side corresponds to the hyperprior autoencoder architecture. Q represents quantization, and AE, AD represent arithmetic encoder and arithmetic decoder, respectively. Conv(3,64,2) represents the convolution operation with kernel size of 3x3, 64 output channels and a stride of 2. 69
3.4 LSTM-UNet architecture used in Frame Reconstruction Network. 72
3.5 Visual examples of our method as compared with H.264 and HEVC. 76
3.6 MS-SSIM on the VTL dataset (352×288) for different compression codecs. Our method is competitive with the state of the art over varying bit rates on these low-resolution videos. . 78
3.7 PSNR and MS-SSIM on the UVG dataset (1920×1080) for different compression codecs. Our method outperformed all compression methods against the perceptually relevant MS-SSIM, while remaining highly competitive against the non-perceptual PSNR. 78
3.8 PSNR of HEVC test sequences for different compression codecs. The resolution of Class B is 1920×1080, of Class C is 832×480, of Class D is 416×240 and of Class E is 1280×720. Overall, our method is competitive with H.265, and is particularity good at lower bit rates on lower resolution datasets. 79
3.9 MS-SSIM of HEVC test sequences for different compression codecs, where the resolution of Class B is 1920×1080, of Class C is 832×480, of Class D is 416×240 and of Class E is 1280×720. Our method outperformed H.265 and is competitive with other state of the art deep learning models. 80
3.10 Ablation study of displaced frame difference combinations. . . 81
3.11 Ablation study of the effectiveness of the proposed LSTM-UNet. 82
3.12 Distributions of the maximum and minimum motion vector components along the horizontal (left) and vertical (right) axes of the HEVC Class B dataset. 83

3.13 Optical flow along the horizontal direction between two adjacent frames in the Kimono video. 83
3.14 Optical flow along the horizontal directionbetween two adjacent frames in the Basketball Drive video. 84
3.15 Encoding speed of different compression codecs. H.265 refers to the encoding speed of the x265 codec *slower* setting. 86

4.1 Overall network architecture of the Foveated MOVI-Codec, which consists of four components: a Displacement Calculation Unit, a Displacement Compression Network, a Foveation Generation Unit, and a Frame Reconstruction Network. 93
4.2 Concept of displaced frame differences, showing a frame t and a previous frame $t-1$, and multiple spatially displaced versions of frame $t-1$ that can also be differenced with frame t. 95
4.3 Foveation map (left) and quantized foveation map (right), where brighter regions corresponds to larger value. 97
4.4 Quantized contrast sensitivity function. 97
4.5 Training strategy. 99
4.6 Normalized sliced profiles of gaussian foveation masks. 101
4.7 Examplar quantized gaussian foveation masks with different foveation mask space constants σ. 102
4.8 FWQI of the compared models on the UVG dataset and HEVC B test sequences. All video resolutions are 1920×1080. . . . 103
4.9 Visualizations of examplar foveated frames reconstructed by FOV-MOVI-Codec, H.265, and Foveated H.265 (denoted F_265) on the videos (a) Basketball drive and (b) Cactus. 104
4.10 Latent representations generated from four models. The first row correponds to reconstructed frames from each model, the second row shows the cumulative latent representations, and the last row shows the latent representations at each compression level. FOV-MOVI-M1 is Foveated MOVI-Codec with foveation mask space constant $FMSC = H/2$ and FOV-MOV-M2 is Foveated MOVI-Codec with $FMSC = H/4$, where H is the height of the frame. 106
4.11 Sum of latent representations for each channel, where the sum is decreasing in foveated version. 107
4.12 Reconstructed frames, differenced frames and bit-SSIM profiles under different foveation space constants (FMSCs). 108

Chapter 1

Introduction

1.1 Problem

Virtual Reality (VR) and its applications have evolved quickly in recent years since the launches of popular head-mounted consumer displays like the Oculus Rift, HTC Vive and PlayStation VR. Revenues from VR apps, gaming and video reached nearly 4 billion dollars in 2017 and are expected to soar more than fivefold by 2022 [1]. Given the recent availability of cheaper standalone headsets, like the Oculus Quest, and the development of faster and

[1]Meixu Chen, Yize Jin, Todd Goodall, Xiangxu Yu, and Alan C. Bovik. Study of 3D virtual reality picture quality. *IEEE Journal of Selected Topics in Signal Processing,* 14(1):89–102, 2019.
Contributions: Meixu Chen: Writing, Software, Investigation, Formal Analysis; Yize Jin: Software, Investigation, Formal Analysis; Todd Goodall: Conceptualization; Xiangxu Yu: Investigation; Alan C. Bovik: Supervision, Conceptualization, Methodology, Review and Editing.

[2]Meixu Chen, Todd Goodall, Anjul Patney, and Alan C Bovik. Learning to compress videos without computing motion. *Signal Processing: Image Communication*, page 116633, 2022.
Contributions: Meixu Chen: Writing, Software, Investigation, Formal Analysis; Todd Goodall, Anjul Patney: Conceptualization; Alan C. Bovik: Supervision, Conceptualization, Methodology, Review and Editing.

[3]Meixu Chen, Richard Webb, and Alan C. Bovik, Foveation-based Deep Video Compression without Motion Search, *arXiv preprint arXiv 2203.16490*, 2022.
Contributions: Meixu Chen: Writing, Software, Investigation, Formal Analysis; Richard Webb: Conceptualization; Alan C. Bovik: Supervision, Conceptualization, Methodology, Review and Editing.

more reliable 5G wireless networks, the installed base of headsets is expected to grow substantially. VR is being used in an increasing variety of consumer applications, including gaming, 360-degree image and video viewing, and visually immersive education. Websites like Youtube, Facebook and Netflix now support 360 image and video viewing and are offering a variety of online resources, further stimulating more consumer participation in VR.

Unlike traditional viewing conditions where people watch images and videos on flat-panel computer and mobile displays, VR offers a more immersive viewing environment. Since the VR contents can cover the entire viewing space, users are free to view the content in every direction. Usually, only a small portion of the image or video is displayed as they gaze in any given direction, so the content that a user sees is highly dependent on the spatial distribution of content, the object being fixated on, and the spatial distribution of visual attention. The free-viewing of high resolution, immersive VR implies significant data volume, which leads to challenges when storing, transmitting and rendering the content which can affect the viewing quality. Therefore, it is important to be able to analyze and predict the perceptual quality of immersive VR content as well as reducing the size of the immersive content.

1.2 Perceptual Quality of VR Content

Unlike traditional images, VR images are usually captured using a 360 camera equipped with multiple lenses that capture the entire 360 degrees of a scene. For example, the Samsung Gear 360 VR Camera is a portable consumer

VR device with 180° dual lenses that can capture images of resolution up to 5472 × 2736. The recent Insta360 Titan is a professional 360 camera with eight 200° fisheye lenses that can capture both 2D and 3D images of resolution up to 11K. After the images are captured simultaneously by separate lenses, they are stitched together to generate a spherical image. The spherical image is usually stored in equirectangular projection format. Stereoscopic images are usually stored in an over-under equirectangular format, where the left image is on top and the right one is on the bottom. oth subjective and objective tools are needed to understand and assess immersive VR images quality. Subjective VR image quality assessment (VR-IQA) is a process whereby the quality of VR images is rated by human subjects. The collected opinion scores supply the gold standard ground truth on which predictive models can be designed or tested. To our knowledge, there are only a few existing VR databases that include subjective measurements. Most only include traditional distortions such as image compression artifacts, Gaussian noise and Gaussian blur, but fail to capture distortions that are unique to panoramic VR (2D and 3D) images.

1.3 Deep learning-based Video Compression

Video traffic is predicted to reach 82 percent of all consumer Internet traffic by 2021 [2], and to continue this rapid growth even further. The increasing share of video in Internet traffic is being driven by several factors, including the great diversity and extraordinary popularity of streaming and social media services, the rise of video teleconferencing and online video ed-

ucation (accelerated by the Coronavirus Crisis), and significant increases in video resolution. Indeed, it is estimated that by 2023, two-thirds of installed flat-panel television sets will be UHD, up from 33 percent in 2018 [3]. Given significant strains on available bandwidth, it is crucial to continue and greatly accelerate the evolution of video compression systems.

Traditional video compression codecs, like H.264, HEVC and the latest VVC/H.266 process videos through a sequence of hand-designed algorithms and modules, including block motion estimation, and local decorrelating decompositions like the Discrete Cosine Transform (DCT). Although the component modules of modern hybrid codecs have been carefully designed over several generations, the overall codecs have not been globally optimized other than by visual examination or post-facto objective measurement of results, typically by the highly fallible PSNR [4]. Naturally, one could expect the performances of video codecs to be improved by collective, end-to-end optimization. Because of their tremendous ability to learn efficient visual representations, deep learning models are viewed as highly promising vehicles of developing alternative, globally optimal video codecs, and a variety of deep learning based image compression architectures have been proposed [5–20]. These new models have deployed Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), autoencoders, and Generative Adverserial Networks (GAN) yielding rate-distortion efficiencies that are reportedly comparable to those of traditional image compression codecs like JPEG, JPEG 2000, and BPG. Encouraged by these advances, several authors have devised

deep video compression models that suggest the considerable promise of this general approach. Wu *et al.* [21] proposed the first end-to-end trained deep video codec, using a hierarchical frame interpolation scheme. A block-based deep video compression codec was proposed by Chen *et al.* [20]. Lu *et al.* proposed an end-to-end video compression model (DVC) [22] which replaces each component of the traditional hybrid video codec with a deep learning model, which are jointly trained as a global hybrid architecture against a single loss function. Another hierarchical video compression architecture called HLVC (Hierarchical Learned Video Compression) was proposed by Yang *et al.* [23].

1.4 Foveation in VR Content

One advantage of VR is that the two eyes have fixed positions, aside from eye movements, relative to the viewing screen. Because of this, the eye movements, and associated points of gaze on the displays can be measured. This makes it possible to exploit the fact that the density of retinal photosensors is highly non-uniform. The cone cells used in photopic viewing achieve on peak density in the foveal region, which captures a circumscribed FOV of about 2.5° around gaze. This includes only 0.8% of all pixels on a flat panel display when viewed under typical conditions [24], and around 4% of pixels on a VR display [25]. Since the density of photoreceptors falls away quite rapidly with increased eccentricity relative the fovea, much more efficient representations of what is perceived can be obtained by judiciously removing redundant information from peripheral regions.

While foveal processing protocols might be useful for many aspects of VR rendering and viewing, such as enhancement or brightening around the point of gaze, foveated compression may offer the most significant and obvious benefits. While this topic has been studied in the past [26–28], only recently has there been renewed interest in foveating modern codecs [29]. The success of foveation based processing protocols involves several factors, including distribution of retinal ganglion cells [30], cortical magnification [31], and the steep grade of density of the photoreceptors [32]. The spacings of the photoreceptors and the receptive fields of the neurons they feed are smallest in the fovea [33]. The fovea covers an area in the approximate range of 0.8% to 4% of the pixels on a display, depending on the display size, resolution, and the assumed typical viewing distance [24, 25]. Recent advances in eye-tracking technology and their integration into consumer VR headsets have opened the possibility of using them to facilitate gaze-contingent video compression. Indeed, retinal foveation when combined with ballistic saccadic eye movements to direct visual resources, is a form of biological information compression. For example, the density of retinal ganglion cells (RGC) in the fovea is 325,000/mm^2. If the entire retina had this output density, then about 350 million RGCs would be implied. However, the number of axons carrying signals along the optic nerves of each eye is only around 1 million, hence foveation results in a 350-fold compression of data passed along the retino-cortical pathway [34]. In an analogous manner, considerable increases in digital video compression can be obtained by removing visual redundancies (relative to fixation) in the visual

periphery.

1.5 Contributions

We have addressed these two challenges of perceptual quality and compression. Firstly, we have created a more comprehensive database that both includes traditional image distortions as well as VR-specific stitching distortions. We also include eye tracking data that was obtained during the subjective study. The new LIVE 3D VR IQA Database is made publicly available for free to facilitate the development of 2D and 3D VR IQA models by other research groups. Secondly, in the direction of compressing the VR content, we have proposed two deep learning-based video compression models, MOVI-Codec and Foveated MOVI-Codec. MOVI-Codec is a new breed of deep video compression model that are motion computation free, statistically motivated, and have perceptual relevance by capturing displaced frame differences from a large database of videos, and feeding them into a deep space-time coding-decoding network. We further reduces the complexity of compression process by incorporating foveation into a deep video compression model to achieve significant data reductions suitable for eye-tracked VR systems which we call Foveated MOVI-Codec.

The rest of the paper is organized as follows. Chapter 2 introduces my effort towards building a 3D VR image database. Chapter 3 describes details of my motionless compression model, the MOVI-Codec model. Chapter 4 presents the details of my foveated version of the motionless deep learning-

based model, Foveated MOVI-Codec. Chapter 5 concludes the paper with a discussion of future research directions.

Chapter 2

Study of 3D Virtual Reality Picture Quality

Both subjective and objective tools are needed to understand and assess immersive VR images quality. Subjective VR image quality assessment (VR-IQA) is a process whereby the quality of VR images is rated by human subjects. The collected opinion scores supply the gold standard ground truth on which predictive models can be designed or tested. To our knowledge, there are only a few existing VR databases that include subjective measurements. Most only include traditional distortions such as image compression artifacts, Gaussian noise and Gaussian blur, but fail to capture distortions that are unique to panoramic VR (2D and 3D) images. Towards advancing progress in this direction, we have created a more comprehensive database that both includes traditional image distortions as well as VR-specific stitching distortions. We also include eye tracking data that was obtained during the subjective study. The new LIVE 3D VR IQA Database is made publicly

[1]Meixu Chen, Yize Jin, Todd Goodall, Xiangxu Yu, and Alan C. Bovik. Study of 3D virtual reality picture quality. *IEEE Journal of Selected Topics in Signal Processing,* 14(1):89–102, 2019.
Contributions: Meixu Chen: Writing, Software, Investigation, Formal Analysis; Yize Jin: Software, Investigation, Formal Analysis; Todd Goodall: Conceptualization; Xiangxu Yu: Investigation; Alan C. Bovik: Supervision, Conceptualization, Methodology, Review and Editing.

available for free to facilitate the development of 2D and 3D VR IQA models by other research groups. The rest of this chapter is organized as follows. Section 2.1 briefly introduces current progress on objective and subjective VR-IQA research. Section 2.2 describes the details of the subjective study. Section 2.3 discusses data analysis of the subjective study results. Section 2.4 analyzes the performances of a variety of objective IQA models on the new database.

2.1 Background

2.1.1 Subjective Quality Assessment

Although it dates from at least as early as the 1970's, VR has been a topic of considerably renewed interest since the appearance of the Oculus Rift DK1. Viewing images in VR gives a more realistic and immersive viewing experience arising from the large field of view, 360° free navigation and the sense of being within a virtual environment.

However, the immersive environment incurs a significant computational cost. VR images and videos are much larger than traditional planar images displayed on computer or TV screens and require much higher transmission bandwidth and significantly greater computational power. These demands are hard to meet, often at the cost of errors in capture, transmission, coding, processing, synthesis, and display. These errors often degrade the visual quality by introducing blur, blocking, transmission, or stitching artifacts. Therefore, developing algorithms for the automated quality assessment of VR images will help enable the future development of VR technology. Developing these algo-

rithms requires subjective data for design, testing, and benchmarking. There are many widely used image quality databases, such as the LIVE Image Quality Assessment Database [35], the TID2013 database [36], CSIQ [37], and the LIVE In-the-Wild Challenge Database [38]. These databases embody a wide variety of image distortions, but they were built for the purpose of studying traditional "framed" 2D images and are not suitable for building or testing algorithms designed to assess VR images.

Recently, there has been increasing interest in developing VR databases, and progress has been made in this direction. Duan *et al.* [39] developed an immersive video database, containing downsampling and MPEG-4 compression distortions. In [40], Upenik *et al.* introduced a mobile testbed for evaluating immersive images and videos and an immersive image database with JPEG compression. Sun *et al.* [41] constructed the Compression VR Image Quality Database (CVIQD), which consists of 5 reference images and corresponding compressed images created using three coding technologies: JPEG, H.264/AVC and H.265/HEVC. An omnidirectional IQA (OIQA) database established by Duan *et al.* [42] includes four distortion types, JPEG compression, JPEG2000 compression, Gaussian blur and Gaussian noise. This database also includes head and eye tracking data that compliment the objective ratings. Another database that both includes head and eye movement data, VQA-OV, was proposed in [43]. This database includes impairments from both compression and map projection. Xu *et al.* [44] also established a database with viewing direction data on immersive videos. However, most of the available

VR databases only include distortions that occur in planar images, but without distortions that are specific to VR, such as stitching. Moreover, newer compression methods such as VP9 are relevant to the encoding of VR images, and these other compression methods are likely to play a substantive role in the future. Furthermore, amongst all the existing VR databases, to the best of our knowledge there are no 3D VR image quality databases as of yet.

2.1.2 Objective Quality Assessment

Both MSE and PSNR were long used as the basic way to assess image and video quality prior to the appearance of modern image objective quality assessment (IQA) methods. These IQA methods can be classified as: full reference (FR-IQA), reduced reference (RR-IQA) , or no reference (NR-IQA). Full reference IQA is appropriate when an undistorted, pristine reference image is available. Reduced reference IQA models only require partial reference information, and no reference IQA algorithms operate without any reference image information at all.

Popular modern FR picture quality models include the Structural Similarity (SSIM) [45], its multiscale form, MS-SSIM [46], Visually Information Fidelity (VIF) [47], FSIM [48], GMSD [49], VSI [50] and MDSI [51]. NR-IQA models have also been proposed, including BRISQUE [52], NIQE [53], BLIINDS [54], and CORNIA [55].

Several VR-specific IQA models have also been proposed over the years. Yu *et al.* [56] proposed a spherical PSNR model called S-PSNR, which averages

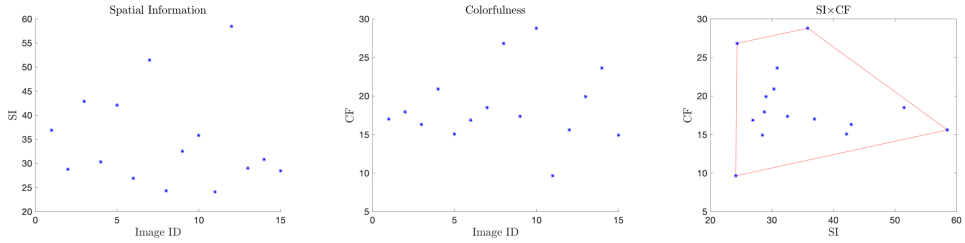
quality over all viewing directions. The authors of [57] introduced a craster parabolic projection based PSNR (CPP-PSNR) VR-IQA model. Xu *et al.* [58] proposed two kinds of perceptual VQA (P-VQA) methods: a non-content-based PSNR (NCP-PSNR) algorithm and a content-based PSNR (CP-PSNR) method. WS-PSNR [59] is yet another PSNR based VR-IQA method, which reweights pixels according to their location in space. SSIM has also been extended in a similar manner, as exemplified by S-SSIM [60]. Yang *et al.* [61] proposed a content-aware algorithm designed specifically to assess stitched VR images, by combining a geometric error metric with a locally-constructed guided IQA method. A NR-IQA method designed to assess stitched panoramic images using convolutional sparse coding and compound feature selection was proposed in [62]. Given the explosive popularity of deep learning, many more recent VR-IQA methods have been learned to analyze immersive images and videos, often achieving impressive results. For example, in [63], the authors deployed an end-to-end 3D convolutional neural network to predict the quality of VR videos without reference. In [64] and [65], the power of adversarial learning was utilized to successfully predict the quality of images.

2.2 Details of the Subjective Study

2.2.1 Image Capture

We used an Insta360 Pro camera [66] to capture the VR image in our 360 image database, due to its portability, raw format availability, high resolution (7680×3840), and good image quality. Instead of only capturing colorful, highly saturated images, we collected a wide variety of natural scenes,

including daytime/night scenes, sunny/cloudy backgrounds, indoor/outdoor scenes, and so on. We acquired 15 high-quality immersive 3D 360° reference images containing diverse content. Most of the scenes were captured in Austin, Texas. For each scene, 4 to 5 raw images (.dng format) were captured to ensure that one with the least amount of motion blur and stitching error could be selected. For each scene, an over-under equirectangular 3D image was generated. We selected the images to span a wide range of spatial information and colorfulness, as shown in Figure 2.1. Spatial Information (SI) is a measure that indicates the amount of spatial detail of a picture, and it is usually higher for more spatially complex scenes [67]. Color information is computed using Colorfulness (CF) as proposed in [68] which represents intensity and variety of colors in an image. Higher values indicate more colorful images. Figure 2.1(c) depicts a scatter plot of SI vs. CF, showing that our database includes a variety of images considering both metrics. Examples of images in our database are shown in Figure 2.2.



(a) (b) (c)

Figure 2.1: Plots of Spatial Information (SI) and Colorfulness (CF) of the VR images in the LIVE VR IQA Database

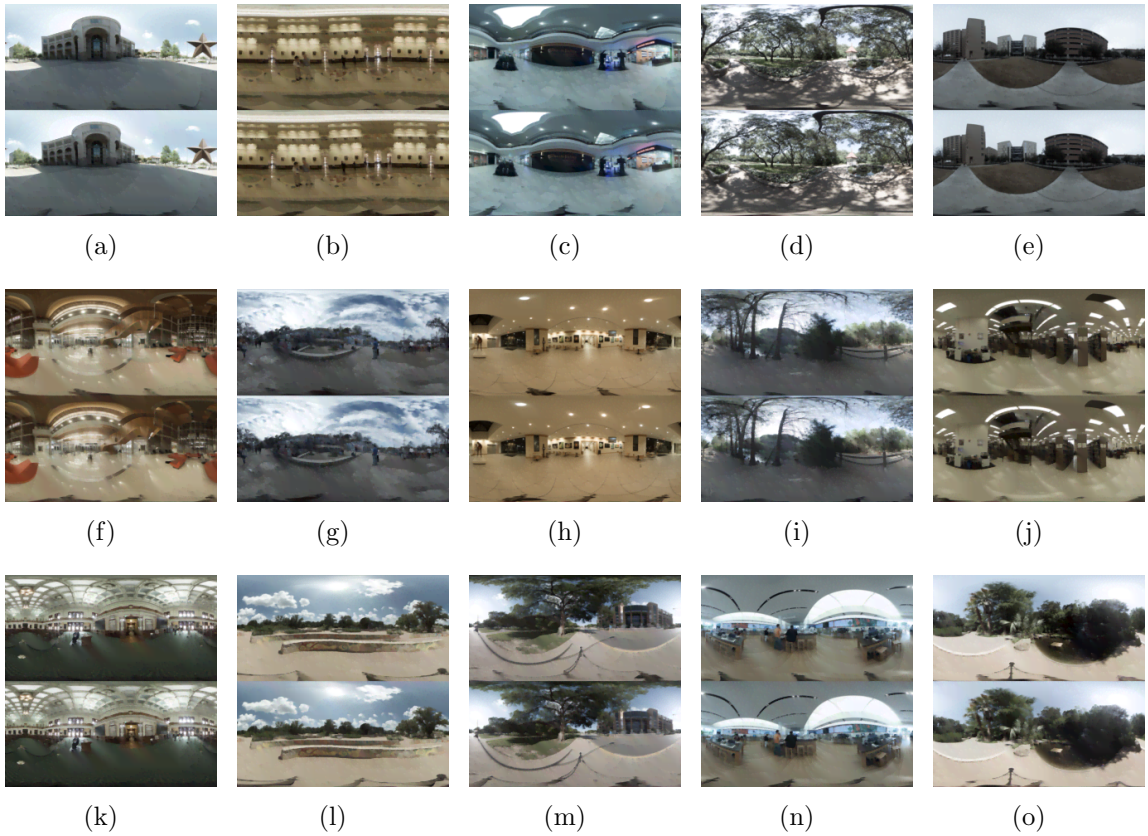


Figure 2.2: Exemplar VR images in the LIVE VR IQA Database

Figure 2.3: Insta360 Pro Camera

2.2.2 Test Images

Each of the selected 15 reference VR content was subjected to 6 types of distortion, including Gaussian noise, Gaussian blur, stitching distortion, down-

sampling distortion, VP9 compression, and H.265 compression. The driving goal of our study was to create a diverse and representative immersive stereoscopic 3D image quality database for developing, testing, and benchmarking VR-related IQA methods. We included the traditional distortions as well as VR-specific stitching distortions. We also included recent compression distortions, including VP9 and H.265, to study and model the way they compress and perceptually distort VR images.

The distortion levels were determined to ensure noticeable perceptual separation between severity levels while also avoiding obvious differences between neighboring levels. All of the distortions other than stitching distortions were applied directly to the equirectangular 3D image. The 360 images were generated using Insta360 Stitcher. Since the resolution of the original images was 7680 $\times$ 3840, we scaled the reference images to resolution 4096 $\times$ 2048 to match the resolution of the VR headset used in the study (as well as most commercial models) before applying the distortions. In the following sections, we explain the way each of the different distortions were applied to the 15 reference 3D VR images.

2.2.2.1 Gaussian Noise

Gaussian additive noise was applied to the unit normalized RGB channels with standard deviations in the range [0.002, 0.03].

2.2.2.2 Gaussian Blur

We separated the left and right images and applied a circular-symmetric 2-D Gaussain kernel to the RGB channels using standard deviations in the range of [0.7, 3.1] pixels. Each RGB channel in both the left and right image was blurred with the same kernel.

2.2.2.3 Downsampling

The left and right images were separated before adding downsampling distortion. Each original immersive image was downsampled to one of five reduced spatial resolutions using bicubic interpolation. We used the HTC Vive for our subjective experiments. This HMD presents a resolution of 1080×1200 and Field of View (FOV) of 110 degrees to each eye. The preferred resolution between 3K and 4K can be found by calculating the portion of solid angle that the FOV spans. We set the maximum total resolution to be 4096×2048, as also suggested in [69, 70], and the minimum resolution to be 820×820, thereby covering a wide range of qualities.

2.2.2.4 Stitching Distortion

We first separated the left and right images and captured 14 perspective views from each image using MATLAB, covering the entire spherical image to simulate a 14-head panoramic camera placed at the center of each scene [61]. The viewing directions we used are listed in Table 2.1, where ϕ represents the zenith angle, and θ represents the azimuth angle. The FOV was set to 110 degrees. An example of the 14 views is shown in Figure 2.4.

After obtaining a set of images captured by the virtual lenses, we imported the views into the popular stitching tool Nuke, and adjusted the orientation of each stitched image to have the same orientation as its reference image, to avoid introducing any further discomfort. Specifically, since the first viewing direction points to the zenith, we adjusted the ZXY rotation parameters in Nuke such that the first perspective view (generated by the first viewing direction) was on the right position. This was done by searching the rotation matrix space to find the parameters that would rotate the first view back to the zenith.

After adjusting the orientation of the stitched image, we tuned the stitching parameters, mainly the convergence distance, error threshold and whether 'refine' or 'reject' was applied, to generate different levels of the distortion. An example of different levels of stitching distortion is shown in Figure 2.5. The same procedure was applied on the left and right images, and we ensured that the stitching distortion created was at the same location in the two images to avoid further discomfort arising from binocular rivalry.

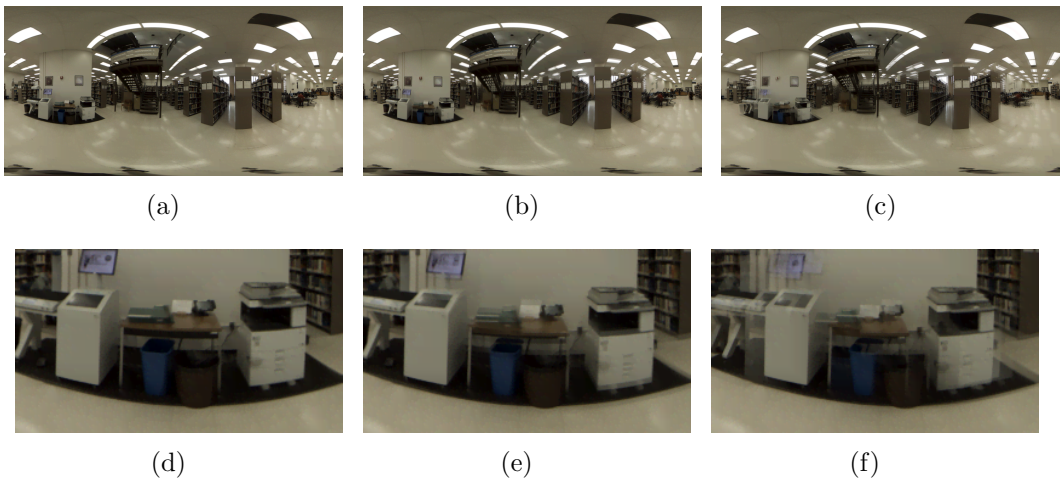
TABLE 2.1: Viewing Directions, where ϕ represents the zenith angle, and θ represents the azimuth angle

θ	0	$\pi/4$	$3\pi/4$	$5\pi/4$	$7\pi/4$	0	$\pi/2$	π	$3\pi/4$	$\pi/4$	$3\pi/4$	$5\pi/4$	$7\pi/4$	0
ϕ	0	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	$3\pi/4$	$3\pi/4$	$3\pi/4$	$3\pi/4$	π

2.2.2.5 VP9 Compression

VP9 compression was applied using the popular public domain software FFmpeg, using the libvpx-vp9 encoder. We varied the constant quality factor

Figure 2.4: Example of 14 perspective views that were stitched together



(a) (b) (c)

(d) (e) (f)

Figure 2.5: Different levels of stitching distortion. (a)-(c): Images of level 1, 3 and 5 (higher levels indicate more distortion). (d)-(e): Zoomed-in views of (a)-(c).

over the range [50, 63], where lower values indicate better quality.

2.2.2.6 H.265 Compression

H.265 (HEVC) compression distortion was applied using the FFmpeg libx265 encoder with different QP values ranging from 38 to 50, where higher values imply increased compression and worse quality.

2.2.3 Subjective Testing Design

We employed the Single Stimulus Continuous Quality evaluation methods described in the ITU-R BT 500.13 recommendation [71]. The human subjects entered their quality adjustments on a continuous rating scale from 0 to 100, where 0 indicates worst quality.

Each viewing session was limited to a duration of 30 minutes and the subjects were free to take rests at any time. The subjects were asked whether they were prone to discomfort when participating in either a VR or 3D environment beforehand, to eliminate subjects who were not suitable for this subjective study. The visual acuity of each subject was determined using the Snellen test, and each subject was asked to wear their corrective lenses to achieve normal vision when participating in the study. Each subject also participated in a RanDot Stereo test of their stereo vision and depth perception. If any test showed impairment, the subject was recommended not to take this test, but if the subject decided to perform the test, the results were discarded. The range of Interpupillary Distances (IPD) of the HTC Vive is 60.3mm-73.7mm. For those subjects whose IPD was outside of this range, a period of experimentation with the HMD was allowed. If the subject felt

uncomfortable, then it was recommended that he/she not perform the test. The data was also discarded when the subject did not follow instructions.

Each subject participated in three sessions separated by at least 24 hours apart. For each session, 9 contents and 60 distorted images were randomly selected. The "hidden" reference image was included in each session. To reduce the effects of memory comparisons, images of the same content were separated by at least five images of different content. The average viewing time for each session was 27 minutes, with the average viewing and rating time for each image being around 23 seconds.

2.2.4 Subjective Test Display

The subjective test was displayed on a HTC Vive VR headset with a built-in Tobii Pro eye tracking system [72], as depicted in Figure 2.6. The Tobii Pro Eye tracking is fully integrated into the HTC Vive HMD. It trackes the gaze direction using the Pupil Center Corneal Reflection technique. More specifically, it uses dark pupil eye tracking, where an illuminator is placed away from the optical axis causing the pupil to appear darker than the iris. Tobii Pro eye tracking has an accuracy of 0.5°, a latency of approximately 10ms, and a sampling frequency of 120 Hz. There are several data outputs for each eye: device and system timestamp, gaze origin, gaze direction, pupil position and absolute pupil size. Image playback was supported by a dedicated high performance server (Intel i7-6700, 32GB memory, 1TB hard drive, NVIDIA TITAN X). The interface was built using Unity Game Engine. Detailed procedures of the subjective test are described in the following sections.

Figure 2.6: HTC Vive integrated with the Tobii Pro Eye Tracking system.

2.2.4.1 Eye tracking

Eye tracking commenced at the beginning of each session. Subjects fixated on five red dots that flashed sequentially in the HMD at different positions [72], as shown in Figure 2.7. These points are mapped in normalized coordinates so that (0.0, 0.0) corresponds to the upper left corner and (1.0, 1.0) corresponds to the lower right corner of the current viewport. Each subject was asked to stare at each dot in succession, then after the last dot disappeared, the system used the recorded dot fixations to calibrate the eyetracker. The process was repeated if the calibration was not successful. If the calibration was still not successful after five trials, the subject would be asked to participate at another time. This situation happened twice during our experiments.

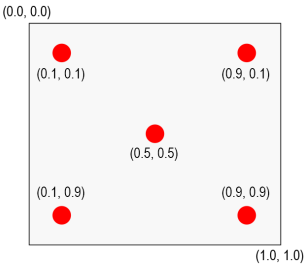


Figure 2.7: Calibration pattern.

2.2.4.2 Viewing and Scoring

The quality scale popped up automatically after 20 seconds of viewing to limit each subject's time viewing the images. To avoid having the subject view the image after the time limit, a grey canvas displayed as background of the rating bar, as shown in Figure 2.8. The quality scale was in the center of the subject's field of view, wherever they moved their head. Five Likert labels "Bad, Poor, Fair, Good, Excellent" indicated the range of ratings the subject could apply. To rate the images, the subjects used the hand controllers supplied with the VR headset to choose the desired score on the quality scale. After the subject was satisfied with the score chosen, they clicked on 'Submit and Next' to see the next image. Once the subject submitted the score, the name and score of the image were written to file. The submission timestamp was also recorded to determine the correspondences between the gaze data and the image. The subsequent image was randomly chosen from all the images in the session, subject to the previously mentioned constraints on the display order. Detailed gaze data was output by the Tobii Pro at the end of each session.

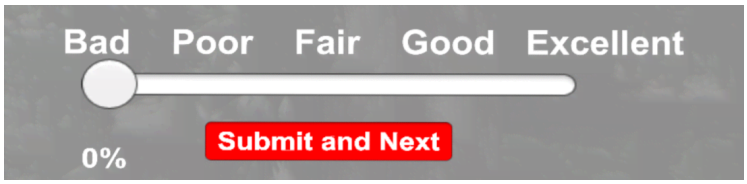


Figure 2.8: Rating bar used in the subjective study

2.2.5 Subjects and Training

All subjects were students at The University of Texas at Austin. The subject pool was inexperienced with image quality assessment and image distortions. A total of 40 students were involved in the study, and each image was rated by around 15 students.

Each subject was orally briefed about the goals of the study and presented with the detailed procedure in written form. A consent form was also signed by the subject. Each subject was asked to view the image as much as possible and score the images according to image quality only, without regard to the appeal of the content. Before the actual session, each subject viewed a training session of 10 images not included in the database. These images were distorted in the same way as the images in the database and spanned the same ranges of quality, to give the subject an idea of the quality and distortions that would be seen in the actual sessions. The subjects rated these images accordingly using the same technique as in the actual session to familiarize themselves with the controllers and the VR headset.

2.3 Data Analysis

Subjective Difference Mean Opinion Score (DMOS) were computed according to [73].The difference scores for reference images were 0 and were discarded for all sessions. Then per session Z-scores were computed from the difference scores and combined into a score matrix z_{ij} and a "viewed" matrix s_{ij}, where 0 indicates the image was not seen by the subject and 1 indicates

the image was seen by the subject.

Subject rejection was performed using the ITU-R BT 500.11 [71] to discard unreliable subjects. To proceed with subject rejection, we first determined whether the scores assigned by a subject were normally distributed, using the β_2 test by calculating the kurtosis coefficient of the function:

$$\beta_{2,j} = \frac{m_4}{(m_2)^2} \tag{2.1}$$

and

$$m_x = \frac{\sum_{i=1}^{M_{view}} (z_j - \bar{z_j})^x}{M_{view}}, \tag{2.2}$$

where M_{view} is the number of subjects that have seen image j. We calculated the mean score and standard deviation for each image:

$$\bar{z_j} = \frac{1}{M_{view}} \sum_{i=1}^{M_{view}} z_{ij} \tag{2.3}$$

$$\sigma_j = \sqrt{\sum_{i=1}^{M_{view}} \frac{(z_j - \bar{z_j})^2}{M_{view} - 1}} \tag{2.4}$$

If β_2 fell between 2 and 4, the scores were assumed to be normally distributed. Then:

$$\text{if } z_j \geq \bar{z_j} + 2\sigma_j, \text{ then } P_i = P_i + 1 \tag{2.5}$$

$$\text{if } z_j \leq \bar{z_j} - 2\sigma_j, \text{ then } Q_i = Q_i + 1 \tag{2.6}$$

If the scores were deemed to not be normally distributed, then:

$$\text{if } z_j \geq \bar{z_j} + \sqrt{20}\sigma_j, \text{ then } P_i = P_i + 1 \tag{2.7}$$

$$\text{if } z_j \leq \bar{z_j} - \sqrt{20}\sigma_j, \text{ then } Q_i = Q_i + 1 \tag{2.8}$$

To reject a subject, we determined whether the following two conditions hold:

$$\frac{P_i + Q_i}{N} > 0.5, \tag{2.9}$$

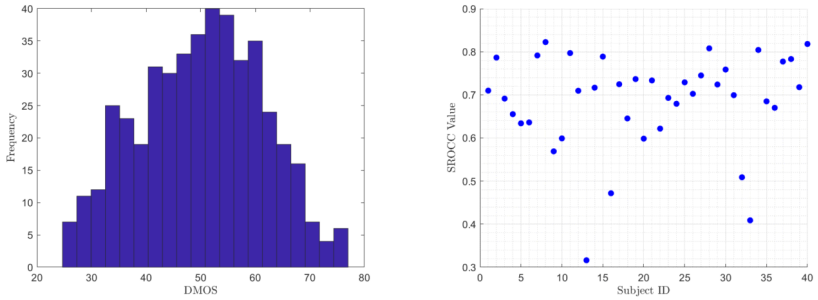
where N is the number of images in the study, and

$$\left|\frac{P_i - Q_i}{P_i + Q_i}\right| < 0.3. \tag{2.10}$$

If Equation 2.9 and Equation 2.10 were both found to hold, then a subject was rejected.

In our study, 2 out of 42 subjects were rejected. For the remaining subjects, we mapped their Z-score to [0, 100] using equation mentioned in [73]. Finally, the DMOS of each image was obtained by computing the mean of the rescaled Z-scores from 40 remaining subjects. A histogram of the recorded DMOS and a plot of the correlations between each subject's ratings and DMOS are shown in Figure 2.9. The DMOS were found to lie in the range [24.67, 76.99].

To explore the internal consistency of the subject data, we randomly divided the subjects into two equal size groups, and computed the Spearman's Rank Correlation Coefficient (SROCC) correlation between their scores. This was done 1000 times. After 1000 splits, the range of correlations was found to be between 0.80 and 0.90 with a median value of 0.87. Hence, there was



(a) (b)

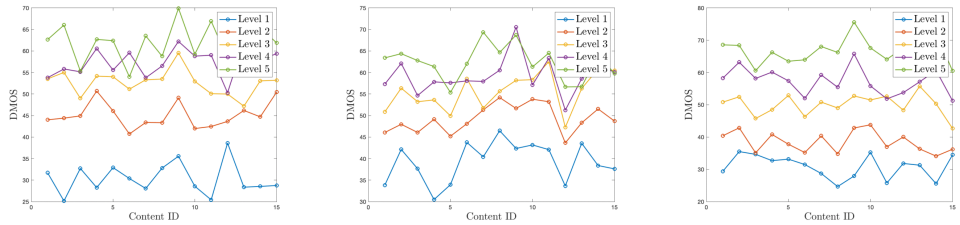
Figure 2.9: (a) Histogram of DMOS. (b) SROCC between subject ratings and DMOS.

a high degree of inter-subject agreement despite the more complex immersive viewing environment. We also calculated correlations by distortion category as shown in Table 2.2. Clearly, stitching distortion resulted in the lowest inter-subject correlation, which is not unexpected, since stitching distortions are highly localized distortions and their ratings are dependent on the amount of visual attention they received from each subject.

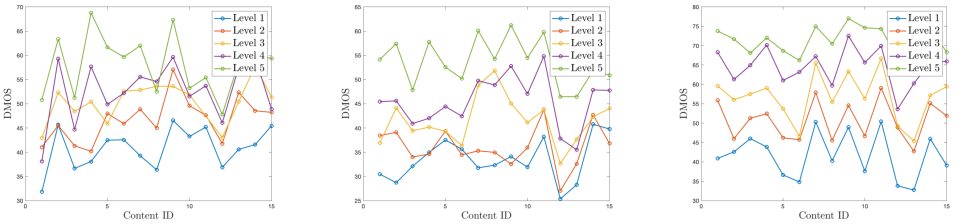
TABLE 2.2: Min, Max and Median SROCC between randomized subject groups for each distortion category

	GAUSSIAN BLUR	GAUSSIAN NOISE	DOWNSAMPLING	STITCHING	VP9	H.265
MIN	0.7778	0.6492	**0.8640**	**0.5669**	0.6056	0.8173
MAX	0.9316	0.8815	**0.9564**	**0.8535**	0.8793	0.9432
MEDIAN	0.8625	0.7897	**0.9146**	**0.7184**	0.7746	0.8951

Figure 2.10 plots the DMOS across all contents, where each color coded curve corresponds to a different distortion level. As shown in the figure, for downsampling and H.265 compression distortions, the DMOS associated with most of the contents decreased with distortion level and the DMOS for different distortion levels are clearly separated. Interestingly, for Gaussian noise and



(a) Gaussian Blur (b) Gaussian Noise (c) Downsampling

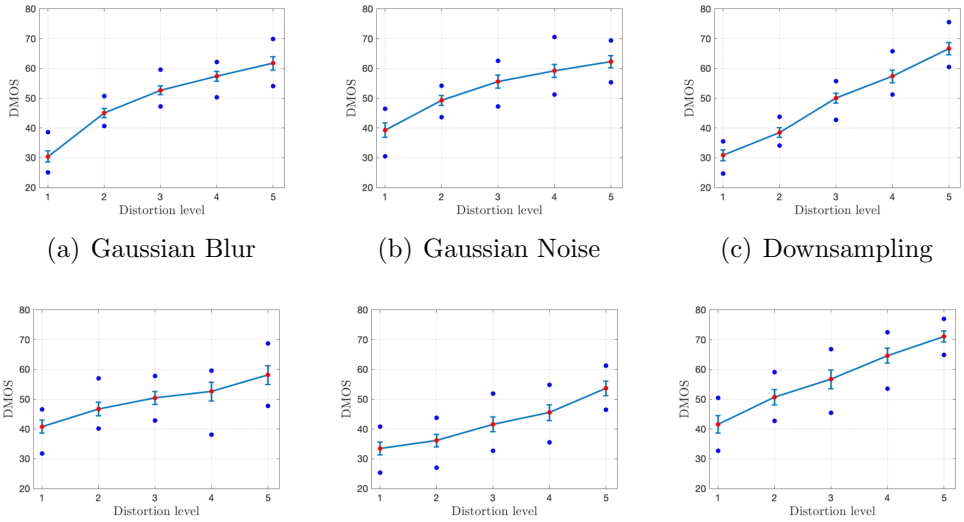


(d) Stitching Distortion (e) VP9 (f) H.265

Figure 2.10: DMOS of all contents for each level of applied distortion.

VP9 distortions, the DMOS given to some of the contents were not always monotonic with distortion level. For stitching distortions, the DMOS across distortion levels were mostly consistent but slighly entangled.

Figure 2.11 plots the DMOS ranges against distortion level for each distortion type. There were overlaps of the confidence intervals for Gaussian noise, stitching and VP9 distortions. Overlaps occurred at higher distortion levels for Gaussian noise, at lower distortion levels for VP9 and over all regions for stitching distortions. This indicates that more severe Gaussian noise distortions as were light VP9 distortions were rated similarly, while stitching distortions were less consistently rated overall.



(a) Gaussian Blur (b) Gaussian Noise (c) Downsampling

(d) Stitching Distortion (e) VP9 (f) H.265

Figure 2.11: Confidence intervals of DMOS over all contents for each applied level of distortion. The blue points indicate the maximum and the minimum DMOS for each distortion type and level. The red points indicate the mean DMOS and the blue bars are the 95% confidence intervals.

2.4 Objective IQA Model Comparison

When evaluating the performance of IQA methods, we computed the IQA scores separately on the left and right images, and used the average of these as the overall IQA score. Since BRISQUE requires training in advance, we split the database randomly, using 80% of the data for training, and 20% for testing. No contents were shared between training and testing. On each distortion type, BRISQUE was trained and tested using only features extracted on images having the corresponding distortion type, to allow measurement of the best case median performance, since performance of BRISQUE degrades in general when it has more distortions to measure. This process was done 1000 times and the median value was taken as the final IQA score. The IQA scores of the other methods were processed in the same way to avoid any bias. We tested and compared the following IQA models on our database.

1. *Peak Signal-to-Noise Ratio (PSNR)* is the negative logarithm of the pixel-wise mean squared error (MSE) function plus an additive offset between the reference and distorted images.

2. *Weighted-to-Spherically-Uniform PSNR (WS-PSNR)* [59] is a modification of PSNR that measures distortions in representation space and weights distortions according to the corresponding projection area in observation space.

3. *Structural Similarity Index (SSIM)* is a widely used full reference image

quality assessment model [45] which captures local luminance, contrast, and structural information.

4. *Multiscale SSIM (MS-SSIM)* [46] is a variation of SSIM that captures quality information across multiple spatial scales.

5. *Visual Saliency-Induced Index (VSI)* [50] is a full reference visual saliency-based IQA method that also integrates gradient magnitude and chrominance features.

6. *Gradient Magnitude Similarity Deviation (GMSD)* [49] is a simple gradient-based IQA method. It also uses spatial deviation pooling to aggregate the quality predictions.

7. *FSIM* [48] is a full reference IQA method that measures image quality based on local measurements of phase congruency and gradient magnitude.

8. *Mean Deviation Similarity Index (MDSI)* [51] is a full reference image quality evaluator that fuses gradient similarity, chromaticit, and deviation pooling features.

9. *Spherical Structural Similarity Index (S-SSIM)* [60] is a weighted-to-spherically-uniform VR-IQA method which scales pixels with equal mapped spherical areas by equal factors when measuring distortion using SSIM.

10. *BRISQUE* [52] is a NR IQA model that uses natural scene statistics features defined in the spatial domain.

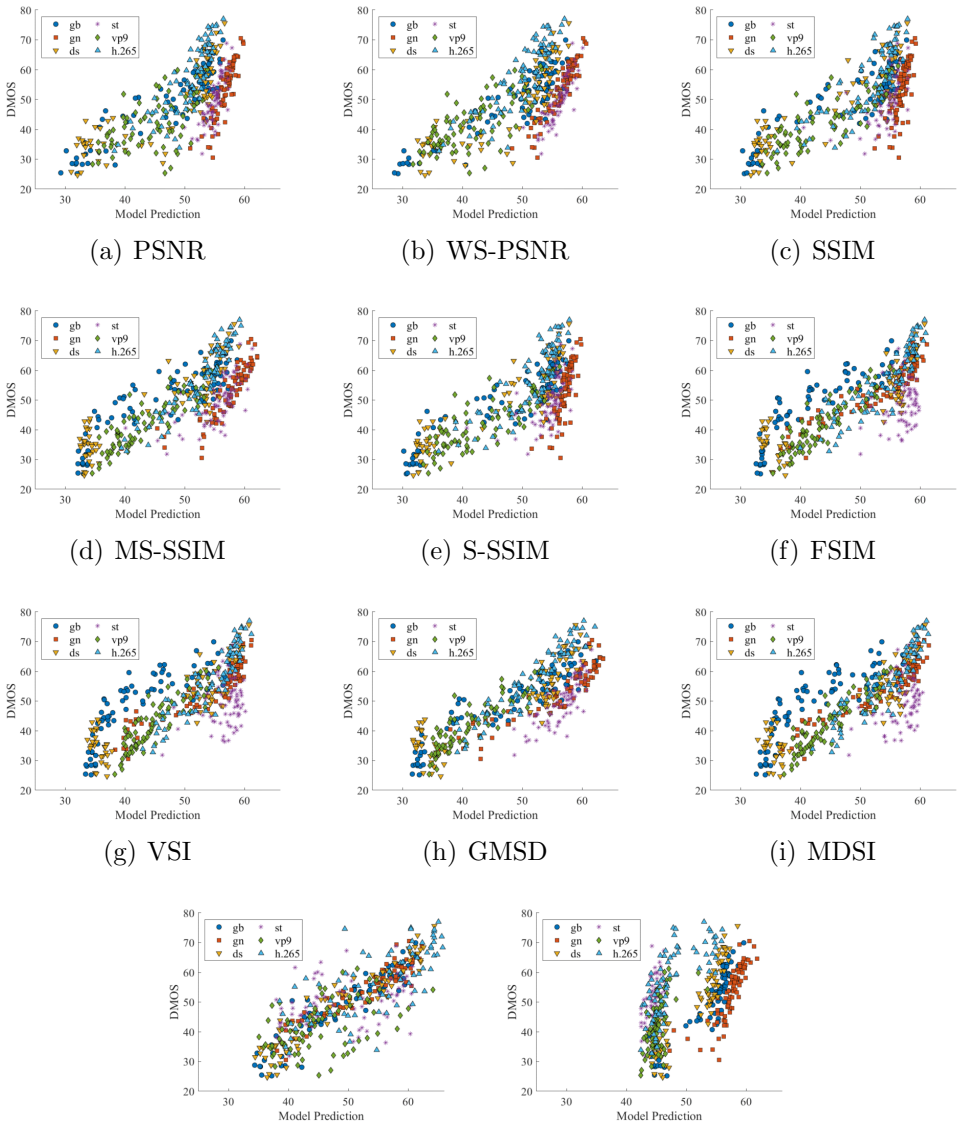
11. *NIQE* [53] is a completely blind (unsupervised) image quality assessment model, in which the quality of a distorted image is computed in terms of its distance from a learned NSS model.

2.4.1 Performance of Objective Methods

We tested the performance of the just-listed objective IQA models using three metrics: the Spearman's Rank Order Correlation Coefficient (SROCC), the Pearson Linear Correlation Coefficient (PLCC), and the Root Mean Square Error (RMSE). The SROCC assesses how well the relationship between an objective model prediction and human subjective scores can be described using a monotonic function. The PLCC measures the accuracy of prediction of different objective models after performing a nonlinear logistic regression. We used a five-parameter logistic function:

$$f(x) = \beta_1(\frac{1}{2} - \frac{1}{1 + e^{\beta_2(x-\beta_3)}}) + \beta_4 x + \beta_5 \tag{2.11}$$

where x are the predicted scores, $f(x)$ is the mapped score, and $\beta_i (i = 1, 2, 3, 4, 5)$ are parameters to be fitted that minimize the mean squared error between the mapped scores and the subjective scores. The Root Mean Squared Error is the standard deviation of the prediction errors. The performances of the compared IQA models is listed in Tables 2.3, 2.4 and 2.5. In addition, scatter plots of all of the considered objective IQA models against DMOS are shown in Figure 2.12.



(a) PSNR (b) WS-PSNR (c) SSIM

(d) MS-SSIM (e) S-SSIM (f) FSIM

(g) VSI (h) GMSD (i) MDSI

(j) BRISQUE (k) NIQE

Figure 2.12: Scatter plots of all pairs of objective and subjective IQA scores using different IQA algorithms. 'gb' refers to Gaussian blur, 'gn' refers to Gaussian noise, 'ds' refers to downsampling, and 'st' refers to stitching.

TABLE 2.3: SROCC of IQA Methods

	OVERALL	GAUSSIAN BLUR	GAUSSIAN NOISE	DOWNSAMPLING	STITCHING	VP9	H.265
PSNR	0.5755	**0.7893**	0.8929	0.8179	0.7321	**0.5036**	0.7714
WS-PSNR	0.6350	0.7911	**0.8875**	0.8286	0.7857	0.6304	0.8536
SSIM	0.6289	0.7821	0.9107	0.8321	0.5143	0.7643	0.7857
MS-SSIM	0.7187	0.8571	0.9107	0.8036	0.7250	0.8179	0.9250
S-SSIM	0.6420	0.8036	0.9143	0.8607	0.5857	0.7536	0.8214
FSIM	0.7007	0.9179	0.9143	**0.7893**	**0.8179**	**0.8821**	0.9357
VSI	0.6805	0.9143	0.9143	0.7929	0.7982	0.8464	0.9357
GMSD	**0.7963**	0.9000	0.9036	0.8179	0.7893	0.8429	0.9321
MDSI	0.6970	0.9179	0.9143	0.7964	0.8161	0.8714	**0.9429**
BRISQUE	0.7353	**0.9357**	0.9036	**0.9500**	0.4714	0.5750	0.7643
NIQE	**0.4635**	0.9321	0.8893	0.8714	**0.1357**	0.5411	**0.6946**

TABLE 2.4: PLCC of IQA Methods

	OVERALL	GAUSSIAN BLUR	GAUSSIAN NOISE	DOWNSAMPLING	STITCHING	VP9	H.265
PSNR	0.6645	0.8726	0.9200	0.8548	0.6740	0.5020	0.7370
WS-PSNR	0.6956	0.8862	0.9110	**0.8412**	**0.8061**	0.6616	0.8270
SSIM	0.7209	0.8777	**0.9377**	0.8758	0.5307	0.7741	0.7730
MS-SSIM	0.7692	0.8816	0.8937	0.8910	0.6887	0.8664	0.9011
S-SSIM	0.7297	0.8889	0.9395	0.8856	0.5874	0.7868	0.8087
FSIM	0.7644	0.8730	0.9365	0.8776	0.1131	**0.8937**	0.8834
VSI	0.7468	**0.8301**	0.9113	0.8951	**0.0320**	0.8627	0.8939
GMSD	**0.8230**	0.9254	0.9130	0.8912	0.7891	0.8703	**0.9247**
MDSI	0.7556	0.8434	0.9325	0.8906	0.0989	0.8856	0.9036
BRISQUE	0.7438	**0.9385**	0.9284	**0.9448**	0.1845	0.5998	0.7567
NIQE	**0.5348**	0.8976	**0.8764**	0.8945	0.1109	**0.4752**	**0.6963**

TABLE 2.5: RMSE of IQA Methods

	OVERALL	GAUSSIAN BLUR	GAUSSIAN NOISE	DOWNSAMPLING	STITCHING	VP9	H.265
PSNR	8.6283	6.3670	8.4388	7.9799	8.5940	7.5472	10.9798
WS-PSNR	8.3170	5.8372	7.5303	7.9487	9.1163	6.3785	10.9601
SSIM	7.9189	6.1000	**9.0154**	7.2637	**7.8567**	5.6369	9.9774
MS-SSIM	7.3847	6.8903	7.4371	7.1059	8.1978	**4.5863**	8.6862
S-SSIM	7.8286	5.7787	8.9576	7.1847	7.8860	5.5525	9.8875
FSIM	7.2500	8.1431	**3.5452**	6.4385	10.5177	4.9344	7.7393
VSI	7.6161	9.2142	3.9715	6.3541	**10.4909**	5.8952	7.8099
GMSD	**6.5393**	**4.9666**	4.7332	6.4346	8.6852	4.6751	8.2060
MDSI	7.4593	**9.3216**	3.5806	6.2573	10.3451	5.5139	**7.6793**
BRISQUE	7.6832	5.3178	4.2615	**5.6837**	9.9613	**8.4327**	8.3145
NIQE	**9.6537**	8.7418	8.4369	**9.5101**	9.0505	8.3211	**12.8910**

2.4.2 Statistical Evaluation

To evaluate whether two IQA methods are significantly different, we performed an F-test on the residuals between the IQA scores after non-linear mapping and the DMOS [35]. The assumption is that the two sets of residuals are Gaussian with zero means. Thus, to test whether they come from the same distribution depends on whether they have the same variance. The null hypothesis is that the residuals from one IQA come from the same distribution and are statistically indistinguishable from the residuals from another IQA. Each entry in the table consists of 6 symbols. A value of '1' in the table represents that the row algorithm is statistically superior to the column algorithm, while a value of '0' means the opposite. A value of '-' indicates that the row and column algorithms are statistically indistinguishable (or equivalent). The position of the symbols corresponds to the following datasets: Gaussian blur, Gaussian noise, downsampling, stitching, VP9, H.265, and all data. The results are shown in Table 2.6.

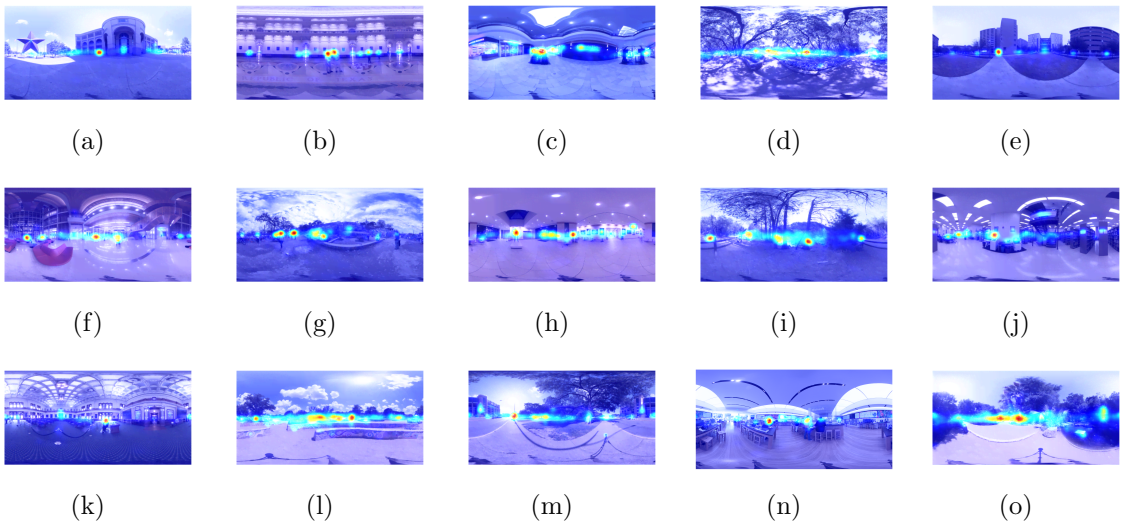
2.4.3 Analysis of Eye Tracking Data

We calculated gaze maps using the eye tracking data recorded by the Tobii Pro. To do so, we added all gaze points for the same content, treated each as an impulse, and smoothed them by applying a Gaussian function with standard deviation of 3.34° [74]. The computed gaze maps are plotted in Figure 2.13. We also calculated the distribution of viewing direction for all images, as shown in Figure 2.14. To visualize the distributions of viewing direction with

TABLE 2.6: Statistical Significance Matrix based on IQA-DMOS residuals. All statistical tests are performed at 95% confidence.

	PSNR	WS-PSNR	SSIM	MS-SSIM	S-SSIM	FSIM	VSI	GMSD	MDSI	BRISQUE	NIQE
PSNR	-------	0000000	-000000	0000000	0000000	0000000	0000000	0000000	0000000	000-000	1111111
WS-PSNR	1111111	-------	1010100	1000000	10-0100	0000000	0000000	0000000	0000000	0001000	1111111
SSIM	-111111	0101011	-------	0000000	0-000-0	0000000	00000-0	0000000	0000000	0101010	1111111
MS-SSIM	1111111	0111111	1111111	-------	1111111	010101-	0101011	0-0-0-0	0101011	0101010	1111111
S-SSIM	1111111	01-1011	1-111-1	0000000	-------	0000000	00000-0	0000000	0000000	0101010	1111111
FSIM	1111111	1111111	1111111	101010-	1111111	-------	1-1-111	0000000	10101-1	0101010	1111111
VSI	1111111	1111111	11111-1	1010100	11111-1	0-0-000	-------	0000000	-000000	0101010	1111111
GMSD	1111111	1111111	1111111	1-1-1-1	1111111	1111111	1111111	-------	1111111	0101111	1111111
MDSI	1111111	1111111	1111111	1010100	1111111	01010-0	-111111	0000000	-------	0101010	1111111
BRISQUE	111-111	1110111	1010101	1010101	1010101	1010101	1010101	1010000	1010101	-------	1111111
NIQE	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	-------

regards to the considered distortions, exemplar plots for four of the contents are shown in Figure 2.15 and Figure 2.16. Example gaze maps on different distortions of the same content are also shown in Figure 2.17.

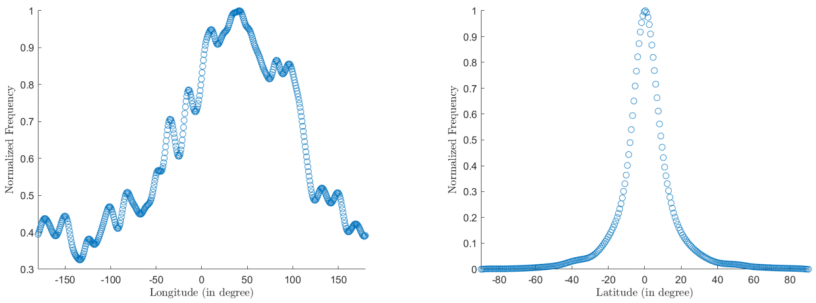


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

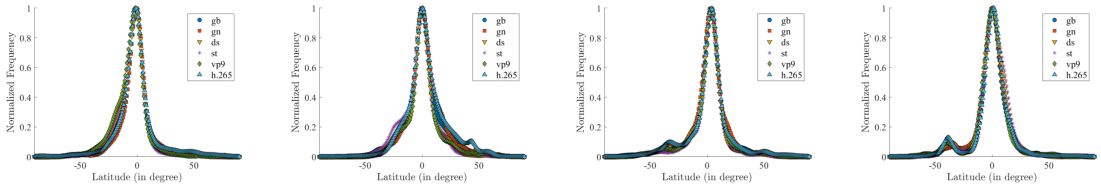
(k) (l) (m) (n) (o)

Figure 2.13: Example gaze maps



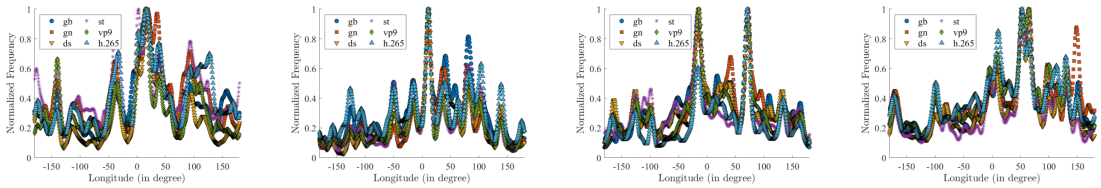
(a) Longitude (x axis) (b) Latitude (y axis)

Figure 2.14: Frequency of viewing directions.



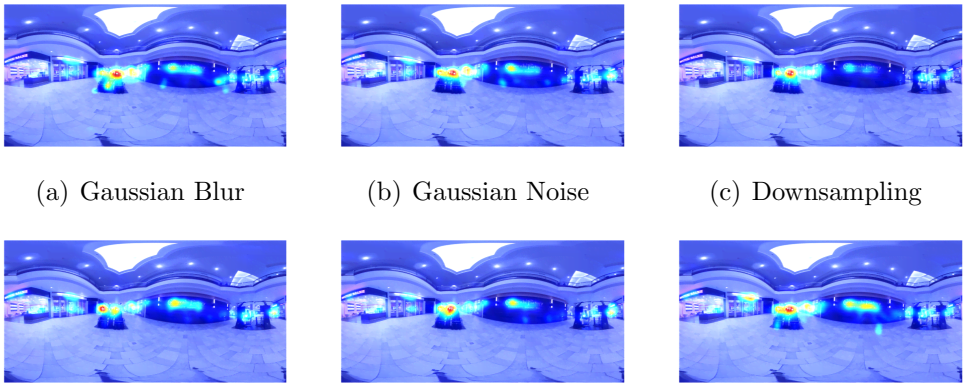
(a) (b) (c) (d)

Figure 2.15: Example frequency plots of latitude viewing directions for four exemplar contents.



(a) (b) (c) (d)

Figure 2.16: Example frequency plots of longitude viewing directions for four contents.



(a) Gaussian Blur (b) Gaussian Noise (c) Downsampling

(d) Stitching Distortion (e) VP9 (f) H.265

Figure 2.17: Example gaze maps on different distorted versions of a same content.

2.4.4 Discussion of Results

From Table 2.3, 2.4 and 2.5, we can conclude that among all methods tested, GMSD generally performed the best while NIQE performed the worst. While WS-PSNR seems to perform better with respect to PLCC on stitching distortions, from Table 2.6, we may see that GMSD provided better quality predictions overall as compared to all other models. WS-PSNR rewards locality, hence its good performance on the stitching distortions. Since stitching distortion is highly local and it greatly affects the overall score, the deviation pooling used in GMSD is more efficient in capturing it than methods using average pooling. In addition, stitching distortion adds weak edges, which can be detected using the gradient map of images. Though MDSI also uses deviation pooling, it utilizes a fused gradient similarity map which is less efficient in detecting weak edges. As a result, it did not perform as well. From the scatter plots, it is interesting to notice that for several algorithms, the correlations for stitching distortions were very poor. This might be because of the locality property of stitching distortion that makes it more difficult. It was also interesting that both WS-PSNR and S-SSIM performed better than their counterparts, which means that applying a reprojection weight to modify traditional IQA methods can help their performance on VR images. Overall, GMSD was statistically superior to all of the other compared methods, while NIQE was statistically inferior to almost all of the others. Training on the subject data of these 3D VR images was an important step of the NR models to capture the unique perceptual peculiarities of the distorted VR image viewing

experience. This is reinforced by the wide disparity in performance between the trained BRISQUE model and the training-free NIQE model, since they use the identical set of features!

From Figure 2.13 and 2.14 , we can also conclude that there exists an equator bias when viewing VR images. Subjects were more likely to view the center of the image (center bias), but this also depended on the content and whether there were objects of interest near the center. A good example is Figure 2.13(e), where the subjects' gaze was more attracted to the person in the image than to the building, although it is located at the center of the image. From Figure 2.15, we can see that on the various considered distortions, the distributions of the latitude viewing directions all followed the equator bias. But from Figure 2.16, it may be observed that this was not usually the case for the longitude viewing directions. For all of the considered distortions, the distributions tended to follow a similar trend, but on specific local distortions, the directions of interests might shift, as shown in Figure 2.16(a). By comparing the gaze maps of Figure 2.16(a) with Figure 2.17, we can see that the areas of interests shifted when stitching distortion was present. The appearance of stitching artifacts is much more localized as compared to other distortions.

Chapter 3

Learning to Compress Videos without Computing Motion

Motion estimation and compensation has occupied a significant amount of resources in both hybrid codec and deep learning-based methods. In order to reduce the overall computational complexity as well as increasing compression rate, we have formulated a new breed of deep video compression algorithms that are motion computation free, statistically motivated, and have perceptual relevance. In this work, we innovate the use of displaced frame differences to capture efficient representations of structures induced by motion, thus avoiding the computational overhead of motion estimation and motion compensation. In addition, we used a combined LTSM-UNet that efficiently captures both spatial and temporal information and uses to recreate video frames from the abstracted video code. The entire video compression system is collectively jointly optimized using a single loss function.

Our results show that video compression can be efficiently accomplished

[1]Meixu Chen, Todd Goodall, Anjul Patney, and Alan C Bovik. Learning to compress videos without computing motion. *Signal Processing: Image Communication*, page 116633, 2022.
Contributions: Meixu Chen: Writing, Software, Investigation, Formal Analysis; Todd Goodall, Anjul Patney: Conceptualization; Alan C. Bovik: Supervision, Conceptualization, Methodology, Review and Editing.

without explicitly computing motion predictions. We trained the new MOVI-Codec architecture end-to-end on the Kinetics-600 dataset and the Vimeo-90K dataset, using a single perceptual loss function (MS-SSIM), and tested it on the UVG dataset, the VTL dataset, and the HEVC Standard Test Sequences (Class B, Class C, Class D, and Class E). Our experimented results show that our new model outperforms the widely used video codec H.264 in LDP *veryfast* setting, and exceeds the performance of the latest standard video codec H.265 using the same setting. In addition, our network outperforms the latest H.266 (VVC) codec at higher bitrates, as assessed by the perceptually relevant MS-SSIM algorithm, on high resolution videos. The rest of this chapter is organized as follows. Section 3.1 briefly introduces current progress on learning-based methods for image/video compression and motion estimation. Section 3.2 describes details of the architecture and training protocol of the new MOVI-Codec model. Section 3.3 discusses the experiments we conducted and their outcomes, along with a data analysis along several dimensions.

3.1 Background

3.1.1 Deep Image Compression

A variety of standardized image compression engines have been proposed over the years to meet the needs of increasingly picture-centric technologies. JPEG algorithm [75], and later challengers JPEG 2000 [76], BPG [77], and VP9 [78]. These methods have proven to be quite practical, and in the case of JPEG, ubiquitous. Yet they are all handcrafted, highly modularized without the benefit of collective optimization of all their elements. Each of

these standards maps pixels to a less correlated representation, regardless of the attributes of the input image. These transformed values are then non-uniformly quantized, typically with reference to a human visual sensitivity model.

A variety of authors have recognized the potential of deep learning to advance progress on the image compression problem (a still timely goal given the senectitude of the prevailing JPEG standard), and many learning-based architectures have been devised [5–19]. Given that Convolutional Neural Networks (CNN) [79] were the first deep learning models to obtain standout performance on image analysis problems, it was natural that it be the first deep architecture to be applied to learning-based image compression. Ballé *et al.* [9] proposed a CNN-based image compression framework that was optimized end-to-end, which was shown to outperform JPEG2000 with respect to both MS-SSIM and PSNR image quality measures. Their framework was later extended by incorporating a hyperprior to capture spatial dependencies in the latent representation for entropy estimation [7]. In [15], Minnen *et al.* further enhanced the entropy model, by combining autoregressive and hierarchical priors to exploit the probabilistic structure in the latents. The resulting model was reported to outperform BPG with respect to both PSNR and MS-SSIM. Another architecture favored for learning-based image compression are Recurrent Neural Networks (RNN), because of their ability to exploit representative memories. Long Short-Term Memory (LSTM) models were proposed [80] to address the vanishing gradient problem of RNNs. Toderici *et al.* [5, 6] was

the first to deploy a deep RNN-based architecture for image compression by utilizing a scale-additive framework. This architecture allows for variable bit rates and only needs to be trained once. The authors also presented results using different types of RNNs, including LSTM, associative LSTM and a hybrid of a Gated Recurrent Unit (GRU) [81] and a ResNet, reporting that the performance of the model was better than JPEG. Generative Adversarial Networks (GAN) have been applied in several learning-based image compression models. Early on, Rippel *et al.* [12] proposed a GAN-based image compression framework that they claim outperformed all existing codecs with respect to MS-SSIM, while being lightweight and deployable. In [13], a GAN framework is presented to build an extreme image compression system which the authors report as achieving state-of-the-art performance, especially at very low bit rates, based on a user study.

3.1.2 Deep Video Compression

It is natural to also consider learning-based methods for video compression [20–23, 82–85]. Wu *et al.* [21] proposed a video compression architecture based on the idea that video compression is repeated image compression. They define two types of frames: key frames and other frames. Key frames are compressed using an RNN-based image compression network [6], while the other frames are interpolated in a hierarchical manner. Another hierarchical video compression architecture, called Hierarchical Learned Video Compression (HLVC), was proposed by Yang *et al.* [23]. In this method, there are three quality layers: an image compression layer, a Bi-Directional Deep Compression

(BDDC) layer, and a Single Motion Deep Compression (SMDC) layer. In an attempt to match the pipeline structure of hybrid codecs, Lu *et al.* proposed an end-to-end video compression model (DVC) [22] that replaces each traditional hybrid component, with deep learning models, then jointly optimized all the components against a single loss function. This work was further extended to two models, a lightweight version called DVC_Lite, and an advanced version called DVC_Pro, by adjusting various components of the architecture. Later, Habibian *et al.* [82] proposed a deep generative model for video compression using an autoregressive prior to conduct entropy coding. Generally, all learning-based video compression models implement traditional block-based motion estimation or optical flow, both of which have a high computational overhead.

The most related work to ours is [85], whereby an interpolation loop is used as an alternative to motion estimation/compensation. However, the frame interpolation network still requires training, which adds to the complexity of the overall method.

3.1.3 Motion Estimation and Motion Compensation

Motion estimation (ME) and motion compensation (MC) are crucial components in modern hybrid video codecs. These are used to exploit the temporal redundancy of video frames via inter-frame prediction. In traditional hybrid video codecs like H.264 and H.265, video frames are first partitioned into blocks, then motion vectors (MV) associated with each block are estimated with respect to predictions of neighboring reference frames via expensive block

search methods, which is the most intensive aspect of video compression. A few deep learning methods have been proposed to solve the ME problem. For example, Choi *et al.* [86] trained a CNN to measure the similarity of pairs of image patches and used this to estimate MVs. However, this method still requires a search process to find the best match. In [87], the authors developed a CNN that was trained to conduct both uni- and bi-directional ME, using separate networks so that motion information need not be transferred from the encoder to the decoder. The CNN does require two frames from the decoded picture buffer and their temporal indices as inputs, which it uses to produce filter coefficients that synthesize patches of a new frame, which is then used to predict the current frame. A drawback of this approach is that it requires the CNN to be resident at both the encoder and the decoder, which reduces decoding efficiency.

Another popular alternative to block matching algorithms are optical flow routines, which seek to obtain a dense vector field mapping the movements of pixel. A variety of deep learning based optical flow estimation methods have been proposed to reduce the computational overhead of dense optical flow vectors [88]. FlowNet [89] showed that it was possible to train a network from two input images to predict optical flow while matching or exceeding the accuracies of traditional methods. Later improvements introduced a stacked architecture that included warping of the second image via intermediate optical flow estimates, and a sub-network specialized to predict small motions [90]. Other approaches have tried to combine networks with traditional methods. Ranjan

et al. [88] proposed such a network called SpyNet, which adopted a traditional coarse-to-fine computational hierarchy using a spatial pyramid. Later, another network competitive with FlowNet2 was proposed, called LiteFlowNet [91], but with a significantly decreased model size. Our approach avoids even these methods of deep flow computation, by instead feeding the network a set of directional inter-frame residuals containing adequate information for the network to seek the most efficient perceptual representation.

3.2 Proposed Method

3.2.1 Framework

Figure 4.1 exemplifies the flow of our deep video compression network. A current frame is input to the network, along with multiple displaced frame differences from adjoining, previously coded and then decoded frames (lower part of figure). This is similar to the classic hybrid coding loop, which also includes the decoder as part of the encoder loop, to reduce reconstruction errors. The key components in our network is: Displacement Calculation Unit (DCU), Displacement Compression Network (DCN), and Frame Reconstruction Network (FRN). The details of each key component in our network will be discussed in the following sections.

The flow of our network is: Given an input video with frames $x_1, x_2, ..., x_T$, for every frame x_t, displaced frame differences between the current frame x_t and previous reconstructed frame $\hat{x}_{t-1}$ are calculated via the DCU, after which the displaced frame differences d_t are input into the DCN. The DCN com-

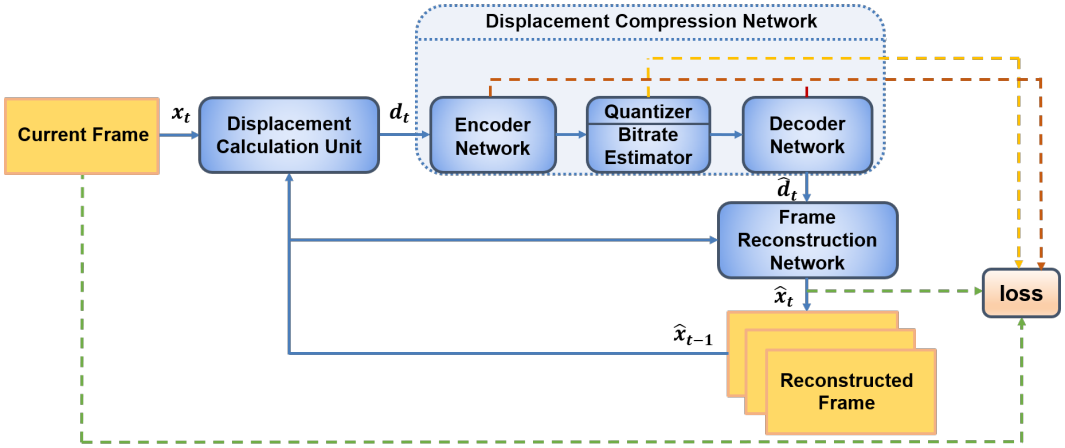


Figure 3.1: The overall network architecture of MOVI-Codec, which consists of three components: a Displacement Calculation Unit, a Displacement Compression Network and a Frame Reconstruction Network.

presses the incoming displaced frame differences which are used to capture statsitical redundancies. An illustration of displaced frame differences, i.e. differences between spatially displaced frames, is shown in Figure 4.2. Given a compressed output $\hat{d}_t$ from the DCN, FRN uses the reconstructed displaced frame differences $\hat{d}_t$ and the reconstructed previous frame $\hat{x}_{t-1}$ to reconstruct a current frame $\hat{x_t}$. Every frame is processed following this except for the first frame. The first frame x_1 is processed differently as it does not have previous reconstructed frame. As a result, an all-zero image is chosen as its previous reconstructed frame and it is otherwise processed the same as other frames. Pseudo code of the flow is shown in Algorithm 1. By using this architecture, we are able to reconstruct the videos without the use of motion.

Algorithm 1 Flow of MOVI-Codec for an Input Video

$\boldsymbol{x_1}$ **to** $\boldsymbol{x_T}$**:** video frames.
$\boldsymbol{\hat{x}_0}$**:** previous reconstructed frame for x_1.
$\boldsymbol{d_t}$**,** $\boldsymbol{\hat{d}_t}$**:** displaced frame differences and corresponding reconstructed ones, respectively.
$\boldsymbol{d_1}$**,** $\boldsymbol{\hat{d}_1}$**:** displaced frame differences between x_1 and $\hat{x}_0$, and corresponding reconstructed ones, respectively.

procedure MOVI-CODEC
    for t in 1 to T **do**
        if t is 1 **then**
            $\hat{x}_0 =$ all zero frame
            $d_1 \leftarrow \text{DCU}(x_1, \hat{x}_0)$
            $\hat{d}_1 \leftarrow \text{DCN}(d_1)$
            $\hat{x}_1 \leftarrow \text{FRN}(\hat{d}_1, \hat{x}_0)$
        else
            $d_t \leftarrow \text{DCU}(x_t, \hat{x}_{t-1})$
            $\hat{d}_t \leftarrow \text{DCN}(d_t)$
            $\hat{x}_t \leftarrow \text{FRN}(\hat{d}_t, \hat{x}_{t-1})$
        end if
    end for
end procedure

3.2.2 Displacement Calculation Unit (DCU)

In both traditional video codecs and recent deep learning-based ones, motion estimation and compensation has occupied a significant portion of the system resources. Motion estimation requires an expensive search process that we avoid, by instead training the network to efficiently represent the residuals between each current frame and a set of spatially-displaced neighboring frames. Computing a set of frame differences, even over many displacement directions is much cheaper than effective search processes. Moreover, while the statistics of motion are generally not regular, the intrinsic statistics of frame differences exhibit strong regularities [92], including those of differences between spatially displaced frames [93]. The strong internal structure of these frame differences makes them easier to efficiently represent in a deep architecture.

The DCU removes the need for any kind of motion vector search. Instead, it allows the DCU network to learn to optimally represent time-varying images as sets of spatially displaced frame differences. Given a video with T frames $x_1, x_2, ..., x_T$ of width w and height h, two directional (spatially displaced) temporal differences are computed between each pair of adjacent frames, as shown in Figure 4.2. In the DCU, the inputs are a current frame x_t and the reconstructed previous frame $\hat{x}_{t-1}$. Then, at each spatial coordinate (i, j), a set of spatially displaced differences is calculated as:

$$d_H(i,j)_t = x_t(i,j) - \hat{x}_{t-1}(i, j-s), \tag{3.1}$$

$$d_V(i,j)_t = x_t(i,j) - \hat{x}_{t-1}(i-s, j), \tag{3.2}$$

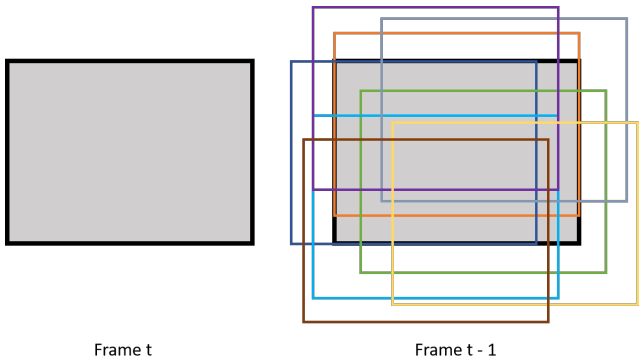


Figure 3.2: Concept of displaced frame differences, showing a frame t and previous frame $t-1$, and multiple spatially displaced versions of frame $t-1$.

where $s = 0, \pm 3, \pm 5, \pm 7$ in our experiment. The set of 13 displaced frame differences (residuals) is then fed into the Displacement Compression Network, which delivers as output the reconstructed set of displaced residuals $\hat{d}_t$. As shown in [93], the statistics of displaced frame differences are highly regular, and more so in the direction of local motion. This makes them good video representations to learn to exploit space-time redundancies, while avoiding the computational burden of motion estimation and compensation. Although the range of motion between frames can be larger than our largest choice of displacement, larger motions can be captured by various combinations of our set of displacements.

3.2.3 Displacement Compression Network (DCN)

3.2.3.1 Framework

After a set of 13 displaced frames are generated from the Displacement Calculation Unit, they are fed into the Displacement Compression Network, where each displacement occupies three channels (RGB), hence the overall in-

put to the DCN comprises 39 channels. The compression network comprises four parts, displacement encoder, displacement decoder, hyper encoder, and hyper decoder. Displacement encoder takes the displaced frame differences calculated from DCU and generates the latent representation y_t using several convolutional layers and convolutional LSTM layers similar to other deep learning-based compression architectures [6, 21]. LSTM [94] as a special RNN structure has proven stable and powerful for modeling long-range dependencies in sequence modeling. The major innovation of LSTM is its memory cell which keeps accumulating the state information. As a result, it helps hold the spatio-temporal information provided by displaced frame differences generated by DCU. The hyper autoencoder uses y_t as input to generate side information, which is then used to better compress quantized latent representation $\hat{y}_t$. Finally, the reconstructed $\hat{d}_t$ is generated using $\hat{y}_t$. The detailed processing flow of the hyper autoencoder is explained in later sections.

3.2.3.2 Quantizer

Traditional quantization inevitably produces zero gradients during back-propagation (BP) which halts network training. Our network deploys BP via stochastic gradient descent, which requires differentiability of all network elements. Hence, we implemented a modified quantizer as in [7], as follows, where $\hat{y}$ is the binarization of the latent representation of displaced frame differences, which lie between -1 and 1, and ϵ represents quantization noise:

$$\hat{y} = y + \epsilon \in -1, 1 \tag{3.3}$$

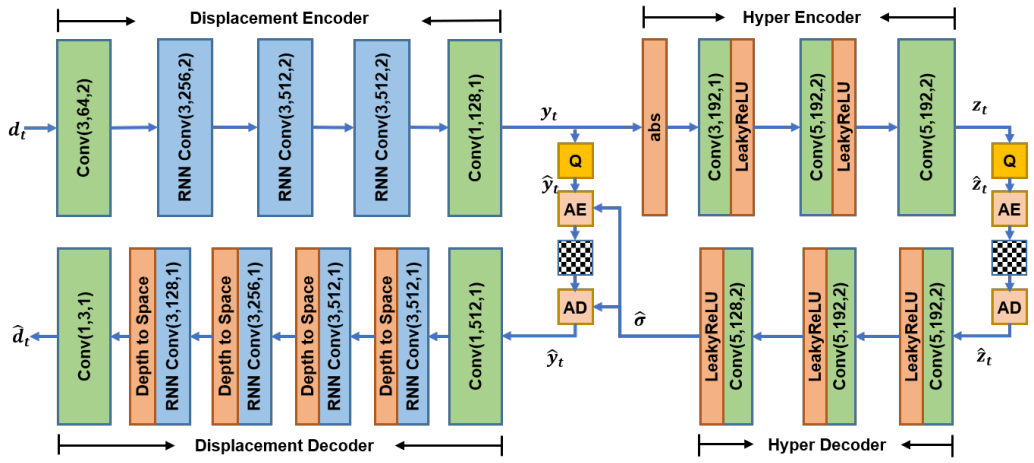


Figure 3.3: Flow diagram of the Displacement Compression Network. The left side shows the displacement autoencoder architecture, and the right side corresponds to the hyperprior autoencoder architecture. Q represents quantization, and AE, AD represent arithmetic encoder and arithmetic decoder, respectively. Conv(3,64,2) represents the convolution operation with kernel size of 3x3, 64 output channels and a stride of 2.

$$\epsilon \sim \begin{cases} 1-y & \text{with probability } \frac{1+y}{2} \\ -y-1 & \text{with probability } \frac{1-y}{2} \end{cases}. \tag{3.4}$$

Following quantization, the size of $\hat{y}$ is $\frac{H}{16} \times \frac{W}{16} \times C$, where H and W are the height and width of the frame, and C is the number of channels of the last convolution layer in the displacement encoder. In our architecture, $C = 128$, as shown in Figure 3.

3.2.3.3 Entropy Coding

To estimate the entropy of the compressed codes $H(\hat{y})$, where $\hat{y}$ is the quantized latent representation of y, we adopted the hyper-prior scheme proposed by Ballé *et al.* [7], where they use an additional set of random variables $\hat{z}$ to capture the spatial dependencies and model the latent representations $\hat{y}$

as Gaussian distribution as follows:

$$p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) \sim \mathcal{N}(\mu, \sigma), \tag{3.5}$$

where $p_{\hat{z}}(\hat{z})$ is modeled using the factorized entropy model [9].

The hyperprior autoencoder architecture is indicated by Hyper Encoder and Hyper Decoder in Figure 3.3, which is responsible for estimating the parameters of the Gaussian model used for entropy coding. After the displacement encoder encoded the input set of displaced frame differences d_t, the resulting latent representation y_t with spatially varying standard deviations is fed into the hyper encoder, which summarizes the distribution of standard deviations in the latent representation z_t. After quantization and arithmetic coding, the quantized $\hat{z}_t$ is transmitted as side information. The hyper decoder uses the quantized $\hat{z}_t$ as input to obtain Gaussian model parameter $\hat{\sigma}$ (μ=0 in our implementation). During modeling training, the Gaussian model parameters can be used to calculate $p_{\hat{y}_t}$ and then estimate $H(\hat{y}_t)$ to guide model optimization. While during model validation and/or testing, the Gaussian model can be used to calculate the cumulative distribution function (CDF) of $\hat{y}_t$ and then guide the arithmetic encoding and decoding of $\hat{y}_t$, which could further losslessly compress $\hat{y}_t$ to bitstream.

3.2.4 Frame Reconstruction Network (FRN)

Figure 3.4 shows the structure of the Frame Reconstruction Network (FRN). The FRN uses the reconstructed displaced frame differences $\hat{d}_t$ and the reconstructed previous frame $\hat{x}_{t-1}$ as the model input to reconstruct the

current frame. The architecture of FRN incorporates Convolutional LSTM (C-LSTM) blocks into a UNet architecture. The UNet architecture, which is an encoder-decoder style network with skip connections, makes it possible to extract and represent meaningful descriptors over multiple image scales. However, without modification, the UNet architecture cannot account for temporal relationships between frames of video data, which are deeply relevant to the efficiency of video compression. The C-LSTM is a convolutional version of the original LSTM, which replaces the matrix multiplication operation of the traditional LSTM with convolutions. It is quite useful for analyzing temporal image sequences, where the C-LSTM layers act as a temporal buffer and capture the long-short dependency of previously processed displaced frame differences. By introducing C-LSTM blocks into the UNet architecture, the FRN is able to process evolving frame properties over multiple scales, by relating compact representations of them in the C-LSTM memory units, leading to better reconstructed frame quality and higher compression rates.

3.2.5 Training Strategy

We modeled the loss function considering the rate-distortion trade-off as follows:

$$\begin{aligned} L &= D + \lambda R \\ &= [D_1(x_t, \hat{x}_t) + \beta D_2(d_t, \hat{d}_t)] + \lambda[H(\hat{y}_t) + H(\hat{z}_t)], \end{aligned} \tag{3.6}$$

where D and R represent the distortion and rate, respectively. λ controls the trade-off between the number of bits and distortion. D_1 denotes the distortion between the input frame x_t and reconstructed frame $\hat{x}_t$ measured by MS-SSIM

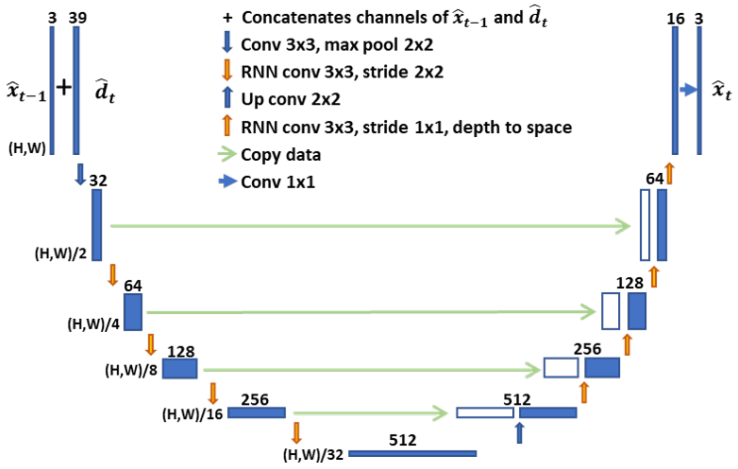


Figure 3.4: LSTM-UNet architecture used in Frame Reconstruction Network.

or MSE, and D_2 denotes the distortion between displaced frame differences d_t and the reconstructed displaced frame differences $\hat{d}_t$ measured by MSE. β controls the trade-off between the perceptual distortion D_1 and the pixel-to-pixel distortion D_2. $H(\cdot)$ represents the bitrates for encoding the latent representations $\hat{y}$ and $\hat{z}$ estimated by the hyperprior autoencoder.

To leverage multi-frame information using our RNN-based codec structure, we update the network parameters every set of N frames during model training, using the loss function in Equation 4.7 but modified as a sum of losses over the kth set of the N frames indexed $x_{t_k+1}, ..., x_{t_k+N}$:

$$L_k = \frac{1}{N}\sum_{n=1}^{N}[D_1(x_{t_k+n}, \hat{x}_{t_k+n}) + \beta D_2(d_{t_k+n}, \hat{d}_{t_k+n})] + \lambda[H(\hat{y}_{t_k+n}) + H(\hat{z}_{t_k+n})]. \quad (3.7)$$

3.3 Experiments

3.3.1 Settings

The MOVI-Codec networks were trained end-to-end on the Kinetics-600 dataset [95, 96] and the Vimeo-90K dataset [97]. The Kinetics-600 videos are downloaded from YouTube, each video having duration of about 10s and various resolutions and frame rates. We used part of the testing set from Kinetics-600, which consists of around 10,000 videos, to conduct our experiments. From each video, a random 128 $\times$ 128 patch with 49 frames was randomly selected for training, and the values of each input video were normalized to [-1,1]. We randomly downsampled the original frame and extracted a 128 $\times$ 128 patch to reduce any previously introduced compression artifacts. The Vimeo-90K dataset consists of 4,278 videos of fixed resolution 448 $\times$ 256. Since the Vimeo-90K dataset has 7 frames per video, we randomly selected a patch of the same size as mentioned before with 7 frames for training. In the Vimeo-90K dataset, the consecutive frames are selected so that the average motion magnitude is between 1-8 pixels, whereas there is no limitation to the motion magnitude between frames in the Kinetics-600 dataset. The mini-batch size is set as 8 for training, and the step length N in our recurrent network is set as 7. By training on both the Vimeo-90K and the Kinetics-600 dataset, we are able to generalize our model to a wide range of natural motions. We tested the MOVI-Codec on the VTL dataset [98], the JCT-VC [99] (Class B, C, D and E) datasets, and the UVG datasets [100]. These datasets cover a variety of resolutions as shown in Table 3.1. For fair comparison with [22, 101]

and [23], we tested our framework on the JCT-VC datasets using the first 100 frames, and tested on VTL and UVG using all frames.

TABLE 3.1: Resolutions of different datasets used for evaluation

Dataset	VTL	UVG	JCT-VC Class B	JCT-VC Class C	JCT-VC Class D	JCT-VC Class E
Resolution	352 × 288	1920 × 1080	1920 × 1080	832 × 480	416 × 240	1280 × 720

To evaluate the quality of the reconstructed videos, we used two quality models: the perception-based MS-SSIM [46] and the non-perceptual PSNR. Multiscale SSIM (MS-SSIM) is a widely used image quality assessment model which captures local luminance, contrast, and structural information. For each quality metric, we trained 5 models with different values of the weighting parameter λ to cover different bitrate ranges. For the MS-SSIM model, λ was set to 0.01, 0.05, 0.1, 0.5 and 1.0, respectively. For PSNR based models, λ was set to 0.0005, 0.0025, 0.005, 0.025 and 0.05. We fixed $\beta = 1$, since we didn't observe any significant differences in model performance as it was varied over the range 0.1 to 10.0.

We compared our method with both traditional and recent deep learning models. H.264 [102], H.265 [103] and the most recent H.266 [104] were included as representatives of traditional hybrid compression codecs. We follow [22] [23], and used the x264 and x265 "LDP very fast" mode. For H.266, we followed [105] to implement the "faster" mode. So that we could compare against another motion-free method, we also included the H.265 zero motion setting, using x265 with *merange* set to zero, which allows exploiting temporal redundancy using an IB prediction structure but without performing motion

estimation. In this regard, this setting is most similar to our architecture [106]. Among recent deep learning models, DVC [22] and Wu *et al.* [21] are optimized for PSNR, Habibian *et al.* [82] and Cheng *et al.* are optimized for MS-SSIM, and HLVC [23] has both MS-SSIM optimized and PSNR optimized results.

3.3.2 Results

In this section, we compare our video compression engine against the standards H.264, HEVC, and H.266/VVC, and with other deep learning-based video compression architectures (Wu [21], DVC [22, 101], and Cheng [85]) on the UVG dataset, the VTL dataset, and the HEVC Standard Test Sequences (Class B, Class C, Class D, and Class E). When compressing videos using the H.264 and HEVC codecs, we followed the settings in [22] and used FFmpeg with the *very fast* mode[1]. When implementing H.266, we followed [105] using the *faster* mode. We also provide visual examples of our approach against other approaches in Figure 3.5. More exemplar reconstructed videos are included on our project page with link given in the Abstract.

Figures 3.6, 3.7, 3.8, and 3.9 show the experimental results on the VTL dataset, the UVG dataset, and the HEVC Standard Test Sequences (Class B, Class C, Class D, and Class E). These results show that our network outperformed both H.264 and the HEVC standard against MS-SSIM. On

[1]H.264: *ffmpeg -pix_fmt yuv420p -s WxH -r FR -i Video.yuv -vframes N -c:v libx264 -preset veryfast -tune zerolatency -crf Q -g GOP -bf 2 -b_strategy 0 -sc_threshold 0 output.mkv*
H.265: *ffmpeg -pix_fmt yuv420p -s WxH -r FR -i Video.yuv -vframes N -c:v libx265 -preset veryfast -tune zerolatency -x265-params "crf=Q:keyint=GOP" output.mkv*
FR, N, Q, GOP represents the frame rate, the number of encoded frames, quality, GOP size, respectively. *N* is set to 100 for HEVC datasets.

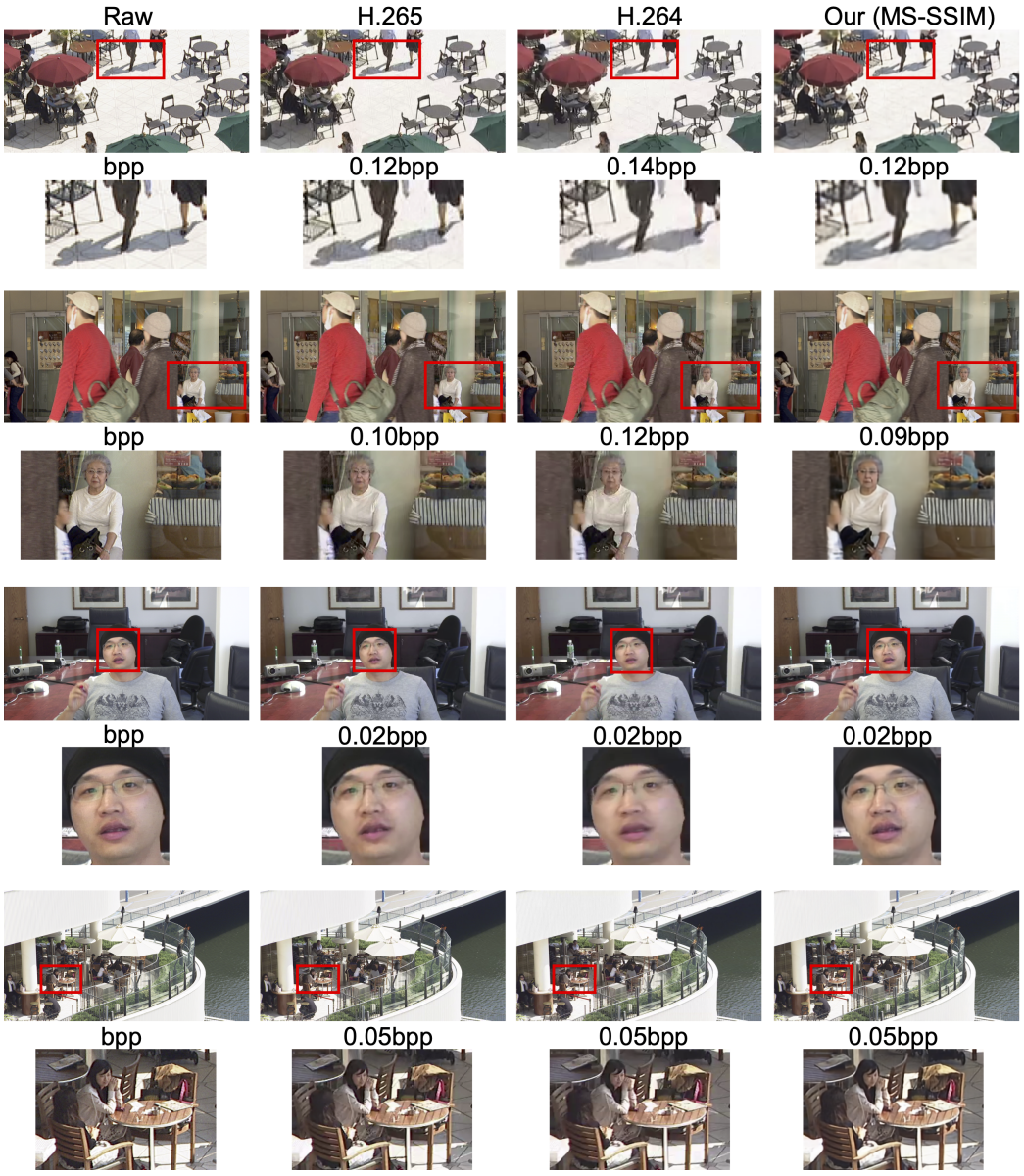


Figure 3.5: Visual examples of our method as compared with H.264 and HEVC.

datasets with higher resolution videos (UVG dataset, HEVC Class B dataset, and HEVC Class E dataset), our network was able to outperform the latest

H.266 codec at higher bitrates as assessed using the perceptually relevant MS-SSIM algorithm. We also compared our model against several deep learning-based compression models, including a frame interpolation-based model by Wu *et al.* [21], DVC [101], HLVC [23] and the video compression framework proposed by Cheng *et al.*, which uses an added spatial energy compaction penalty in the loss function [85]. Among these, DVC and HLVC were trained on both PSNR and MS-SSIM, to obtain better results against each metric. In our comparison, we include the best performance for these two methods for each metric. It is worth noting that our model only uses one previous frame as input, whereas in Wu's framework, both neighboring frames are utilized when reconstructing the middle frame. Additionally, our framework replaces the classical motion estimation and compensation module by instead training the network to optimally interpolate displaced frame differences. For completeness, we also evaluated all models against the PSNR, where MOVI-Codec did not always perform as well. However, this is a problem with the PSNR, which is not perceptually relevant, and which produces significantly inferior quality predictions than perception-based quality predictors like MS-SSIM [4]. Indeed, the high quality of the reconstructions that we make available on the model page (see link in Abstract) further attests to this. As has been observed by others [9, 12, 13, 83], perceptual measures are better arbiters of deep compressed video quality than absolute fidelity models like the PSNR. It is worth noting that when comparing our model against the H.265 zero motion setting, while both methods do not utilize motion estimation, MOVI-Codec was able to

perform better with respect to MS-SSIM than the H.265 zero motion setting, while delivering similar performance against PSNR.

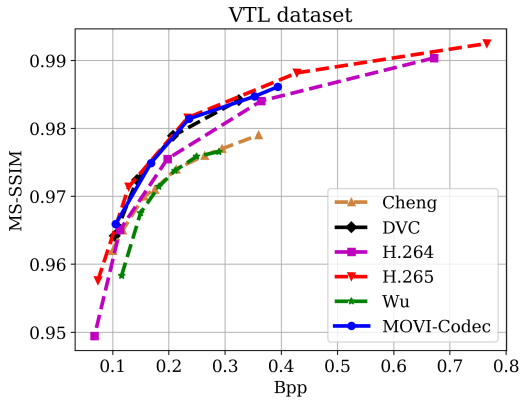


Figure 3.6: MS-SSIM on the VTL dataset (352 × 288) for different compression codecs. Our method is competitive with the state of the art over varying bit rates on these low-resolution videos.

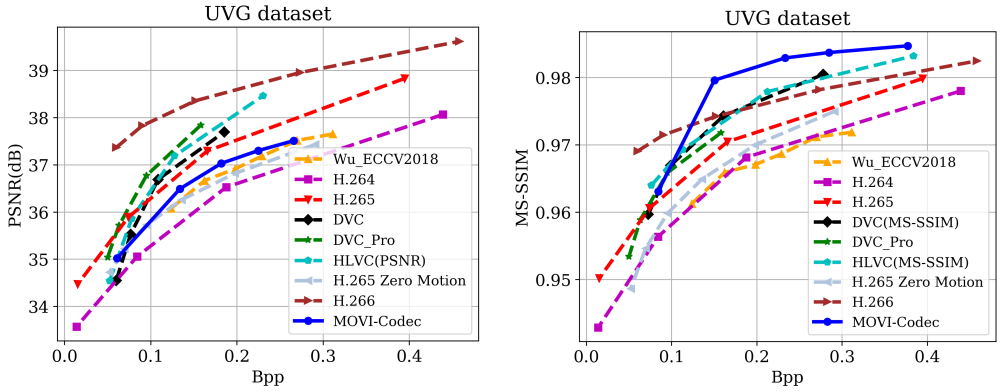


Figure 3.7: PSNR and MS-SSIM on the UVG dataset (1920 × 1080) for different compression codecs. Our method outperformed all compression methods against the perceptually relevant MS-SSIM, while remaining highly competitive against the non-perceptual PSNR.

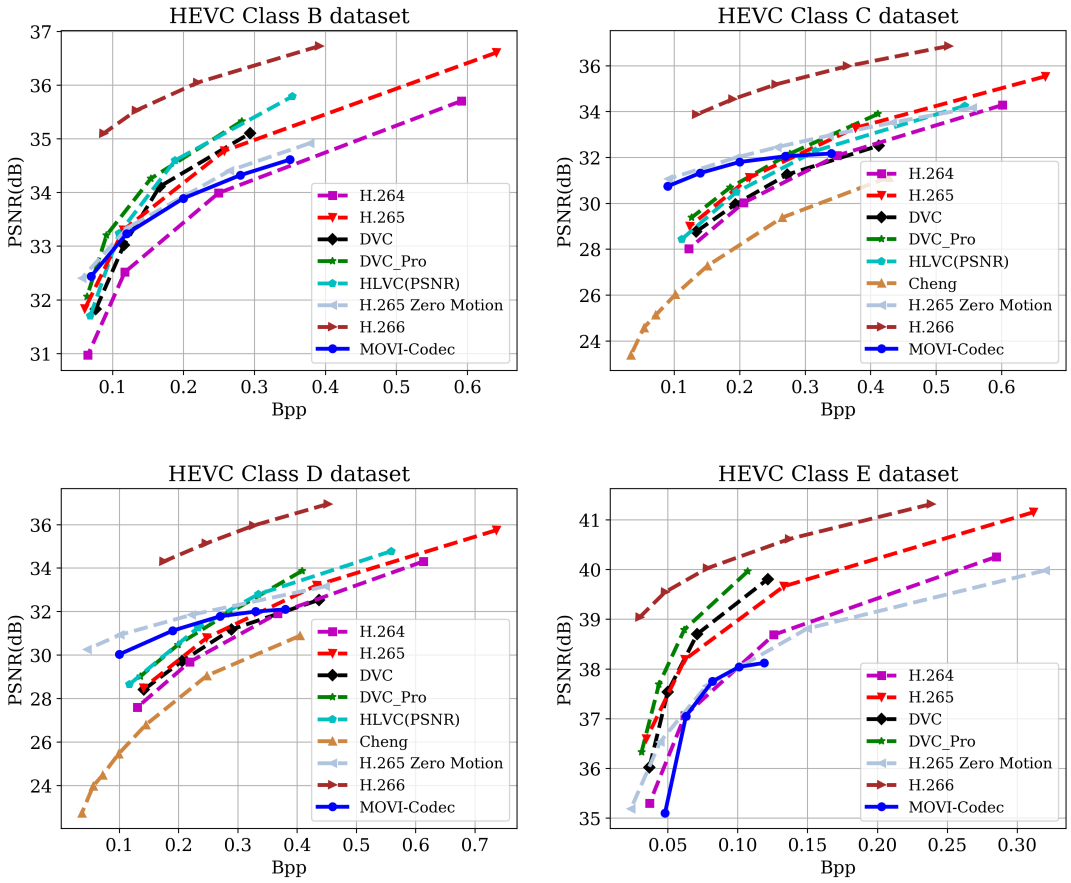


Figure 3.8: PSNR of HEVC test sequences for different compression codecs. The resolution of Class B is 1920 × 1080, of Class C is 832 × 480, of Class D is 416 × 240 and of Class E is 1280 × 720. Overall, our method is competitive with H.265, and is particularity good at lower bit rates on lower resolution datasets.

3.3.3 Ablation Studies

We conducted ablation studies to assess the choices we made in our approach, specifically with respect to the choice of displaced frame differences, and the effectiveness of the proposed LSTM-UNet. The results are shown in Figure 3.10 and Figure 3.11.

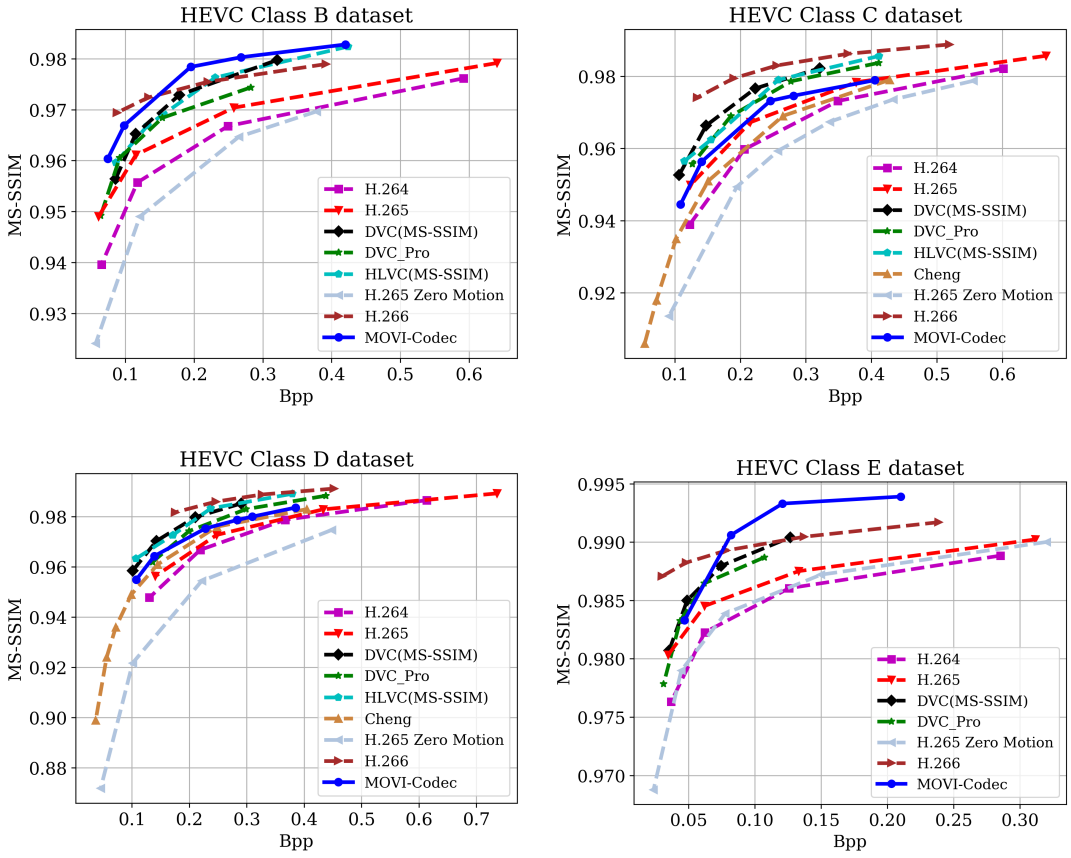


Figure 3.9: MS-SSIM of HEVC test sequences for different compression codecs, where the resolution of Class B is 1920 × 1080, of Class C is 832 × 480, of Class D is 416 × 240 and of Class E is 1280 × 720. Our method outperformed H.265 and is competitive with other state of the art deep learning models.

3.3.3.1 Displaced Frame Difference Combination

Figure 3.10 shows the experimental results on different combinations of displaced frame differences, where $s = 0$ refers to frame differences with no displacements, which gives the worst performance of all combinations evaluated. This shows the value of "displaced" frame differences as a way of

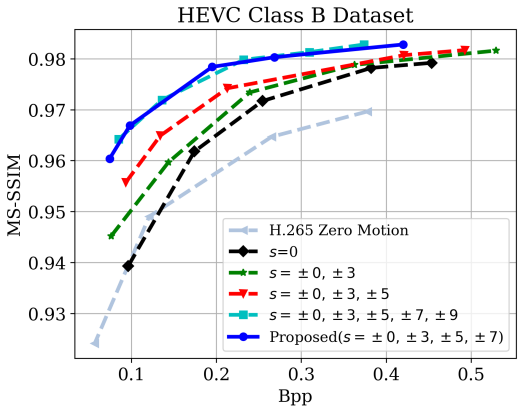


Figure 3.10: Ablation study of displaced frame difference combinations.

training the network on more diverse motion induced displacements. Including displacements as large as $s = 7$ greatly increases the overall performance, by allowing interpolation of larger motions in videos. We also tried adding $s = 9$ to our choice of displacement combinations, but this new combination did not improve the overall performance, meaning that our combination of displacements was adequate to capture motions of various sizes. It is worth noting that as compared with the H.265 zero motion configuration, which also does not utilize motion estimation, our network was able to perform better as assessed by the perceptually relevant MS-SSIM, including when $s = 0$.

3.3.3.2 Effectiveness of the LSTM-UNet

Figure 3.11 shows the experimental results on the HEVC Class B dataset when using UNet and LSTM-UNet to reconstruct frames, respectively. As shown in the example, LSTM-UNet extends the advantage of UNet for extracting and representing spatial descriptors to include spatio-temporal descriptors using C-LSTM blocks, yielding better reconstruction performance.

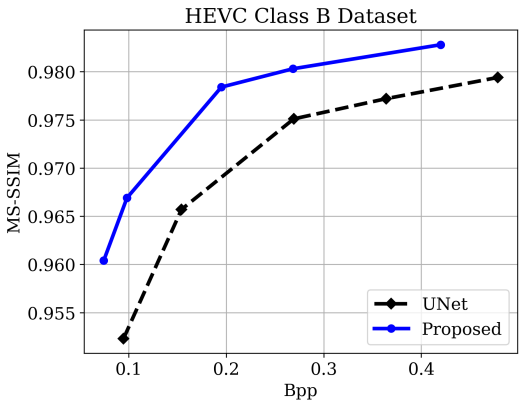


Figure 3.11: Ablation study of the effectiveness of the proposed LSTM-UNet. In addition, LSTM-UNet converges faster than the UNet counterparts, shortening the training time of the network.

3.3.4 Motion Vector Analysis

To verify that MOVI-Codec can capture large motions with the chosen set of displacement combination, we calculated the optical flow of adjacent frames in the testing datasets using a pre-trained network called SPynet [88]. To emphasize large motions, we calculated all motion vectors against adjacent frames, and only picked the minimum and maximum motion vectors in the x and y directions. As a result, we ended up with four values of motion vectors for each adjacent frames. Figure 3.12 shows the distribution of the picked motion vectors on all videos in the HEVC Class B dataset, which is the dataset having the highest resolution videos among our testing datasets. From the figure, we can conclude that our model produced a similar distribution as the original frame pairs, hence our model was able to capture large motions using a set of small displacements.

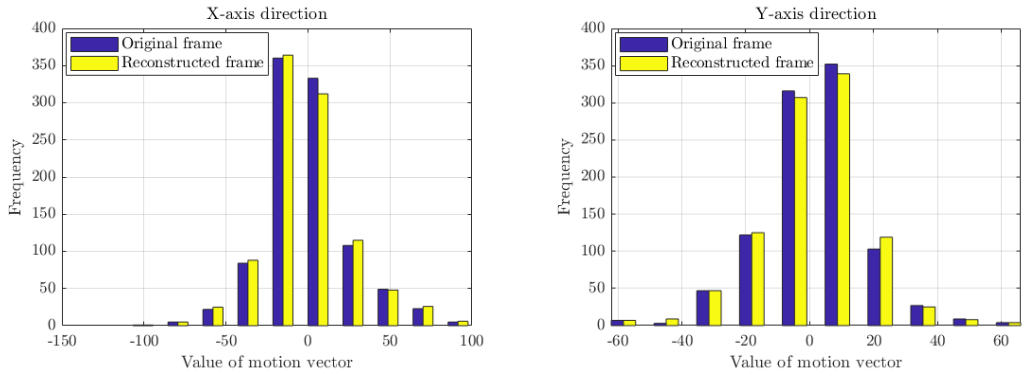
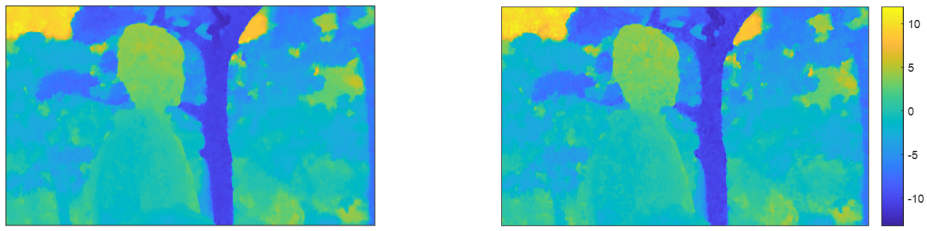


Figure 3.12: Distributions of the maximum and minimum motion vector components along the horizontal (left) and vertical (right) axes of the HEVC Class B dataset.

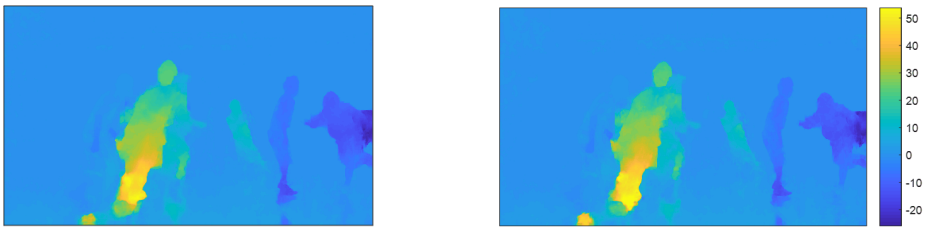
Figure 3.13 and 3.14 illustrate the accuracy of our motion reconstruction. The test video in Figure 3.13 shows the x axis optical flow between two adjacent frames from the Kimono video, which is a video with a moving background and slow motion, whereas Figure 3.14 shows the optical flow images of two adjacent frames in Basketball Drive video, which has a static background and large motions. In both videos, our model was able to reconstruct motion accurately.



(a) Original frame

(b) Reconstructed frame

Figure 3.13: Optical flow along the horizontal direction between two adjacent frames in the Kimono video.



(a) Original frame (b) Reconstructed frame

Figure 3.14: Optical flow along the horizontal directionbetween two adjacent frames in the Basketball Drive video.

3.3.5 Model Analysis

To compare the computational complexity of the different codecs, we tested two deep learning models: the one proposed by Wu *et al.* [21], the light version of DVC called DVC Lite [101], and the commercial software x265 for H.265 compression, using a server with an Intel Core i9-9940X CPU and GTX 1080Ti on video sequences of resolution 1920×1080. The experimental results are provided in Figure 3.15.

The overall encoding speed of our framework is mostly invariant of bitrate, whereas since Wu's framework adopts a progressive coding scheme, its encoding speed varies with the target bitrate. In our framework, although we adopted an RNN-based compression method on displaced frame differences, we utilized the RNN unit to store temporal dependencies and did not use a progressive coding scheme for compression. DVC Lite is a lightweight version of DVC with a more efficient motion estimation module and a lightweight motion compression network, which can be twice as fast as the original DVC model in

terms of encoding speed [101]. Our framework is faster than the lightweight model, further justifying the use of learned interpolation of displaced frame differences. Since the arithmetic coding at lower bitrates is faster than at larger ones, there is a slight slope to our encoding speed curve. But overall, the complexity of our model is invariant to bitrate, which means that our model maintains a stable encoding speed regardless of video content or bitrate for a given resolution.

As shown in Figure 3.15, compared with the traditional hybrid codec, our model is faster than the latest codec HEVC with *slower* setting. However, using the *very fast* setting on x264 and x265, the encoding speed can run at 110 fps and 30 fps, respectively. Of course, by applying model acceleration techniques such as model distillation, model quantization, or by decreasing the model size, it should be possible to similarly accelerate the encoding speed of our framework.

3.3.6 Discussion

From Figures 3.6, 3.7, 3.8, 3.9, and 3.15, we can conclude that our model delivers better compression performance than LDP *veryfast* setting of traditional hybrid codecs like H.264 and HEVC in terms of MS-SSIM, at a low computational complexity. This justifies our use of displaced frame differences as motion information for video compression. Although our model was able to acheive competitive performances as lower settings of traditional codec having low computational complexity, and without the complicated motion estimation and compensation modules other deep learning-based models use, our model

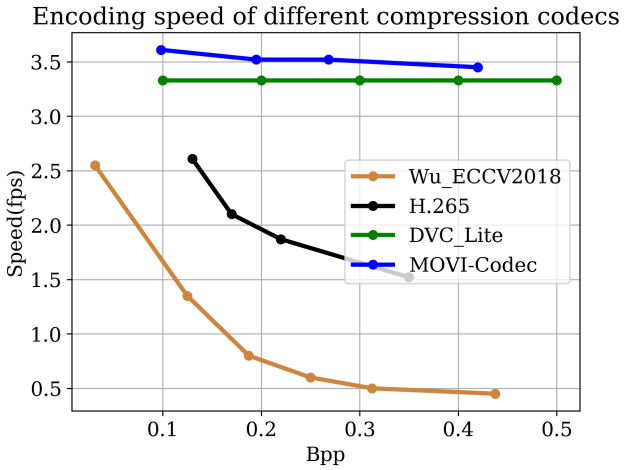


Figure 3.15: Encoding speed of different compression codecs. H.265 refers to the encoding speed of the x265 codec *slower* setting.

did not outperform all of the state-of-the-art models. Nonetheless, the performance achieved by our model provides a new way of motion computation that may prove quite useful for video compression. In our model, we designed the set of spatial displacements used by our network to cover a reasonable range of natural motions. A promising future direction is to automatically assign displacement combinations as a function of resolution. The encoding speed of our model is state-of-the-art among deep learning models, but has not yet been optimized to match compute-optimized traditional codecs like HEVC or VVC, e.g. by model acceleration methods.

Chapter 4

Foveation-based Deep Video Compression without Motion Search

Virtual Reality (VR) and its applications have attracted significant and increasing attention. However, the requirements of much larger file sizes, different storage formats, and immersive viewing conditions pose significant challenges to the goals of acquiring, transmitting, compressing, and displaying high-quality VR content. At the same time, the great potential of deep learning to advance progress on the video compression problem has driven a significant research effort. Because of the high bandwidth requirements of VR, there has also been significant interest in the use of space-variant, foveated compression protocols. We have integrated these techniques to create an end-to-end deep learning video compression framework. A feature of our new compression model is that it dispenses with the need for expensive search-based motion prediction computations. This is accomplished by exploiting statistical regularities inherent in video motion expressed by displaced frame

[1]Meixu Chen, Richard Webb, and Alan C. Bovik, Foveation-based Deep Video Compression without Motion Search, *arXiv preprint arXiv 2203.16490*, 2022.
Contributions: Meixu Chen: Writing, Software, Investigation, Formal Analysis; Richard Webb: Conceptualization; Alan C. Bovik: Supervision, Conceptualization, Methodology, Review and Editing.

differences. Foveation protocols are desirable since, unlike traditional flat-panel displays, only a small portion of a video viewed in VR may be visible as a user gazes in any given direction. Moreover, even within a current field of view (FOV), the resolution of retinal neurons rapidly decreases with distance (eccentricity) from the projected point of gaze. In our learning based approach, we implement foveation by introducing a Foveation Generator Unit (FGU) that generates foveation masks which direct the allocation of bits, significantly increasing compression efficiency while making it possible to retain an impression of little to no additional visual loss given an appropriate viewing geometry. Our experiment results reveal that our new compression model, which we call the Foveated MOtionless VIdeo Codec (Foveated MOVI-Codec), is able to efficiently compress videos without computing motion, while outperforming foveated version of both H.264 and H.265 on the widely used UVG dataset and on the HEVC Standard Class B Test Sequences. The rest of this chapter is organized as follows. Section 4.1 introduces the research related to this project. Section 4.2 details the architecture and training protocol used to create the Foveated MOVI-Codec model. Section 4.3 explains the experiments on algorithm performance and comparisons that we conducted.

4.1 Background

4.1.1 Foveated Video Compression

Since the turn of the millennium, there has been a slowly growing interest in the use of foveation for such diverse image and video processing tasks as quality assessment [107], segmentation [108], and watermarking [109].

Methods of foveating visual content can be categorized into three ways: geometric transformations, space-varying filters, and space-variant multiresolution decompositions [110]. In the first of these, a foveated retinal sampling geometry is used to either apply a foveating coordinate transformation on an original uniform resolution image [111], or to average and map local pixel groups into superpixels [112, 113]. Filter-based methods process images with space-varying low-pass filter with cut-off frequencies determined by foveated resolution-reduction protocols [114, 115]. Multiresolution methods foveation involves decomposing images into bandpass scales, and only retaining scales specified by a foveal fall-off function defined relative to a measured or presumed fixation point [26, 116].

Recently, given significant advances in high resolution and immersive displays technologies, along with concurrent increases in VR content, interest of foveation as an efficient processing tool has quickened. Recent related models include [117], where a neurobiological model of visual attention is used to predict high saliency regions and to generate saliency maps. A guidance map is also generated, using foveation to guide bit allocations when tuning quantization parameters in video compression system. Li *et al.* [118] trained a content-weighted CNN to conduct image compression, whereby the bitrates allocated to different parts of an image are adapted to the local content. Their system significantly outperforms JPEG and JEPG2000 in terms of SSIM when operating in a low bitrate regime. Mentzer *et al.* [14] proposed a similar but simpler model, by incorporating a second channel at the output of the encoder

that is expanded into a mask which is used to modify the latent representations. DeepFovea [29] is a foveated reconstruction model, that employs a generative adversarial neural network. A peripheral video is reconstructed from a small fraction of pixels, by finding a closest matching video to the sparse input stream of pixels that lies on the learned manifold of natural videos. This method is fast enough to drive gaze-contingent head-mounted displays in real time.

4.1.2 Foveated Video Quality Assessment

When designing foveated compression systems, it is desirable to be able to access their perceptual efficiencies using quality measurement tools that account for the foveation. However, almost all available image quality measurement tools, such as SSIM [45], operate on spatially uniform resolution contents. However, there are a few foveated video quality assessment models, which can be conveniently divided into several types. One type of foveated VQA model uses purely static, spatial foveation, whereby measurement or prediction of the user's point of gaze guides the space variant measurement of quality as a function of eccentricity. For example, the Foveated Wavelet Image Quality Index (FWQI) utilizes wavelets to extract position-dependent spatial quality information [27, 119]. Several factors are taken into consideration, including the spatial contrast sensitivity function, which is used to determine local visual cutoff frequencies, which guides modeling of human visual sensitivity across the available wavelet subbands, when combined with assumption on viewing distance and the display resolution. Lee *et al.* [107] proposed a

foveal signal-to-noise ratio (FSNR) to evaluate the quality of picture or video streams. In this method, a foveated image is obtained by a foveated coordinate transformation on the original image(s) to be quality-accessed.

A second type of foveated VQA model is based on retinal velocity. In addition to static foveation mechanisms, these kinds of models also take advantage of the fact that the contrast sensitivity of HVS to an object in a moving scene is influenced by the velocity of its map on the retina. Movement in a video may cause two effects: loss of acuity of the moving objects, modifications of perceived quality. Further, two factors can contribute to losses of acuity: increases of retinal image velocity, and increases of eccentricity relative to the foveal center. Based on these observations, Riomac-Drlje *et al.* [120] proposed a foveated mean squared error (FMSE) that models the effects of spatial acuity reduction due to motion. Another model called the foveation-based content Adaptive Structural SIMilarity index (FA-SSIM), which is based on the popular IQA model SSIM [121] combines SSIM with a foveation-based sensitivity function.

You *et al.* [122] proposed a full reference attention-driven foveated video quality metric (AFViQ) that accounts for the localization of fixations in images and videos. All of the algorithms mentioned above assume that the point of fixation is the center of the image, which is not always true, and can lead to an invalid foveation model. As a result, algorithms based on automatic fixation detection have also been proposed. AFViQ attempted to solve this problem by integrating foveation into a wavelet-based distortion visibility model.

4.2 Proposed Method

4.2.1 Framework

Figure 4.1 illustrates the overall architecture of our network, which extends our previous MOVI-Codec [123]. The compression network is comprised of four components: a Displacement Calculation Unit (DCU), a Displacement Compression Network (DCN), a Foveation Generator Unit (FGU), and a Frame Reconstruction Network (FRN). The DCU computes displaced frame differences between the current frame and the previous reconstructed frame; the FGU generates foveation masks that later direct the allocation of bits in DCN; the DCN compresses displaced frame differences generated from DCU; and the FRN reconstructs input frames from the previous reconstructed frame and the reconstructed displaced frame differences.

The flow of our network is: Given an input video with frames $x_1, x_2, ..., x_T$, for every frame x_t, calculated displaced frame differences between the current frame x_t and previous reconstructed frame $\hat{x}_{t-1}$ via the DCU, after which the displaced frame differences d_t are input into the DCN. In the FGU, a perception-based foveation map P is generated from [26, 119] and used to generate a set of foveation masks $M(P)$. After the set of displaced frame differences d_t are encoded into latent representations y_t, the masks generated from the FGU direct the allocation of bits via element-wise multiplication of y_t and $M(P)$, producing a masked latent representation c_t, which is then quantized (via rounding) and decoded to $\hat{d}_t$. Finally, the FRN reconstructs the input

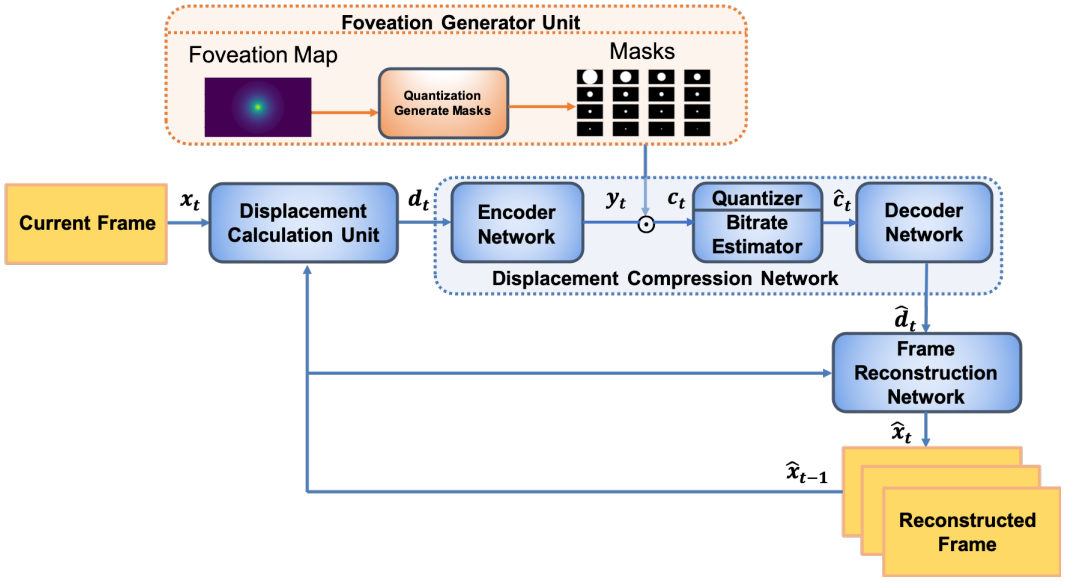


Figure 4.1: Overall network architecture of the Foveated MOVI-Codec, which consists of four components: a Displacement Calculation Unit, a Displacement Compression Network, a Foveation Generation Unit, and a Frame Reconstruction Network.

frame $\hat{x}_t$ from the reconstructed displaced frame differences $\hat{d}_t$ and the previous reconstructed frames $\hat{x}_{t-1}$. The DCN and FRN are defined identically as in [123], so we do not further elaborate them here. We explain the DCU and FGU in the following.

4.2.2 Displacement Calculation Unit (DCU)

The DCU removes the need for any kind of motion vector search. Instead, the DCU learns to optimally represent time-varying images as sets of spatially displaced frame differences. Given a video with T frames $x_1, x_2, ..., x_T$ of width W and height H, two directional (spatially displaced) temporal differences are computed between each pair of adjacent frames, as shown in Figure

4.2. Assume that the inputs to the DCU a current frame x_t and a reconstructed previous frame $\hat{x}_{t-1}$. Then, at each spatial coordinate (i, j), a set of spatially displaced differences is calculated as:

$$d_H(i,j)_t = x_t(i,j) - \hat{x}_{t-1}(i, j-s), \tag{4.1}$$

$$d_V(i,j)_t = x_t(i,j) - \hat{x}_{t-1}(i-s, j). \tag{4.2}$$

In our current implementation, $s = 0, \pm 3, \pm 5, \pm 7$. This set of 13 displaced frame differences (residuals) is then fed into the DCU, which delivers as output the reconstructed set of displaced residuals $\hat{d}_t$. As shown in [93], the statistics of displaced frame differences are regular, and more so in the direction of local motion. This makes them good video representations to learn to exploit space-time redundancies, while avoiding the computational burdens of search-based motion estimation and compensation. Although the range of motions between frames can be larger than our largest choice of displacement, larger motions can be captured by combinations of our set of displacements.

4.2.3 Foveation Generator Unit (FGU)

In the DCN, the encoded video data that is output from the quantizer is still spatially invariant, and arithmetic coding is used to further compress the code. However, the goal of the FGU is to exploit the non-uniform distribution of ganglion cells and photoreceptors across the visual field. Our basic tool to accomplish this is an established model of the contrast sensitivity function (CSF) expressed in terms of eccentricity. We use this to enable increased compression of the image in a manner such that the reconstructed frames are

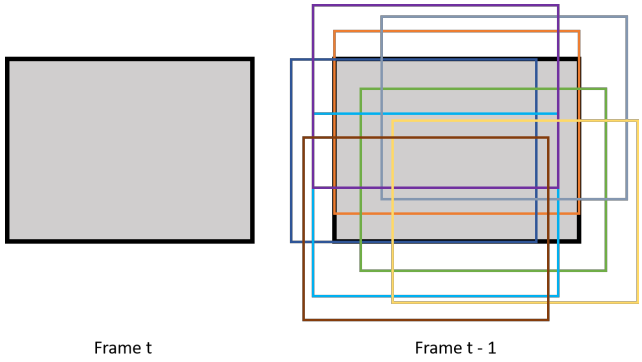


Figure 4.2: Concept of displaced frame differences, showing a frame t and a previous frame $t-1$, and multiple spatially displaced versions of frame $t-1$ that can also be differenced with frame t.

indistinguishable from the original around the point of fixation, as well as with increasing eccentricity.

A good model of the contrast threshold is given by [26]:

$$CT(f,e) = CT_0 exp(\alpha f \frac{e+e_2}{e_2}), \tag{4.3}$$

where f is spatial frequency, e is retinal eccentricity, CT_0 is a specialized minimum contrast threshold, α is a spatial frequency decay constant and e_2 is the half-resolution eccentricity. We follow best fitting parameter values given in [26] are $\alpha = 0.106$, $e2 = 2.3$, and $CT0 = 1/64$ in our experiment. The CSF is then:

$$CS(f,e) = \frac{1}{CT(f,e)}. \tag{4.4}$$

The authors of [119] defined a foveation-based error sensitivity in terms of viewing distance D, frequency f, and location (x,y):

$$S_f(D,f,x,y) = \begin{cases} \frac{CS(f,e(D,x,y))}{CS(f,0)} = exp(-\alpha f \frac{e(D,x,y)}{e_2}) & \text{for } f \leq f_m(x) \\ 0 & \text{for } f > f_m(x) \end{cases}, \tag{4.5}$$

where f_m is the cutoff frequency.

In our model, we fix the frequency in Equation 4.5 to be the maximum frequency that can be presented on the display without aliasing. The FGU uses these models to generate a foveation map that is used to guide bit allocation and rate control. During training, foveation maps are generated using Equation 4.5, assuming a fixed screen resolution, center gaze, and viewing distance following [29] as shown in Figure 4.3. In Equation 4.5, the contrast sensitivity decays forwards zero beyond the cutoff frequency. Our approach to foveation is quantum; rather than changing the displayed resolutions in a smooth and graded manner, which makes the problem more complex, it is instead quantized. Quantization is applied to yield n levels of the foveation map, and $n = 16$ in the current implementation. We also make sure that the contrast sensitivity for the last level is larger than zero to be able to reconstruct all periperal information. Figure 4.3 shows a quantized foveation map. Since the latent representations for the displaced frame differences d_t are 128 channels, the same mask is assigned for every 8 channels of latent representations. The quantized map in x axis is shown in Figure 4.4. After a set of n masks $M(P)$ are generated, we element-wise multiply $M(P)$ and the encoder output y_t to obtain quantized spatially variant (foveated) codes c_t which are then subjected to entropy coding and bitrate estimation, using the same procedure as [123].

While quantized foveation maps are used to train our model, during application (testing) we instead use isotropic 2D gaussian shaped foveation maps, where the gaussians are defined to follow the modified fall-off of visual

acuity. This allows for smoother perceived changes of foveation, with the significant added benefit of making it possible to effect variable rate control by varying the widths (σ) of the gaussians. We define this parameter as foveation mask space constant (FMSC).

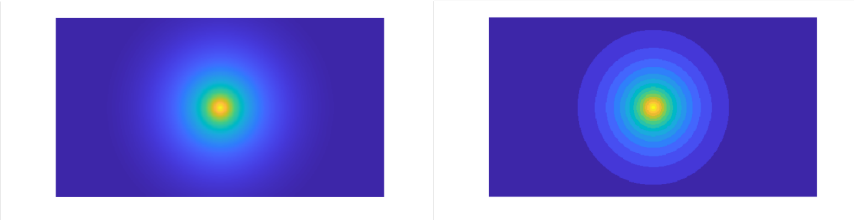


Figure 4.3: Foveation map (left) and quantized foveation map (right), where brighter regions corresponds to larger value.

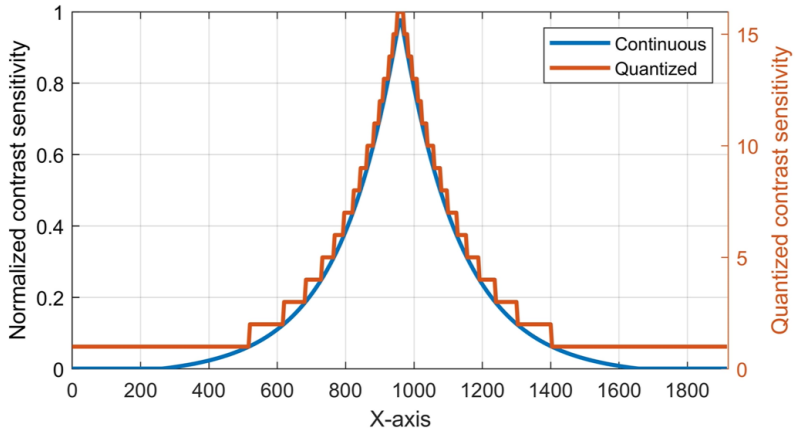


Figure 4.4: Quantized contrast sensitivity function.

4.2.4 Bit Rate Allocation

Given an input frame x, let $y = E(x) \in R^{c \times h \times w}$ be the output of the encoder network, which includes c feature maps of sizes of $h \times w$. Also let

$p = P(x)$ denote a $h \times w$ non-negative foveation map to be applied. The expand y using masks $\mathbf{m} \in R^{c \times h \times w}$ as follows:

$$m(i,j,k) = \begin{cases} 1 & \text{if } p(i,j) \geq \lfloor \frac{k}{c/L} \rfloor \cdot \frac{1}{L} \\ 0 & \text{others} \end{cases}, \tag{4.6}$$

where c is the number of channels in the latent representations y, and L is the number of desired compression levels across the foveation regions. In this way, more bits are allocated to the foveal region, preserving visual details with less sacrifice of bit rate. The sum of the foveation maps $\sum_{i,j} p_{i,j}$ naturally serves as a continuous estimate of compression rate, and can be directly adopted as a compression rate controller. Because of the flexibility of this foveation map approach, it is not necessary to apply entropy rate estimation when training the encoder and decoder, using a simple binarizer for quantization of latent representations y.

4.2.5 Training Strategy

We are able to model the loss function considering only the distortion as follows:

$$D = [D_1(x_t, \hat{x}_t) + \beta D_2(d_t, \hat{d}_t)], \tag{4.7}$$

where D represents the distortion, and D_1 is the distortion between the input frame x_t and reconstructed frame $\hat{x}_t$, measured by foveation-weighted SSIM as detailed below at the end of this subsection. D_2 is the distortion between the displaced frame differences d_t and the reconstructed displaced frame differences $\hat{d}_t$, as measured by the MSE. The weight β controls the trade-off between the perceptual distortion D_1 and the pixel-to-pixel distortion D_2.

To leverage multi-frame information in our RNN-based codec structure, we update the network parameters every set of N frames during model training, using the loss function in Equation 4.7, but modified to be a sum of losses over the kth set of N frames indexed $x_{t_k+1}, ..., x_{t_k+N}$:

$$D_k = \frac{1}{N}\sum_{n=1}^{N}[D_1(x_{t_k+n}, \hat{x}_{t_k+n}) + \beta D_2(d_{t_k+n}, \hat{d}_{t_k+n})]. \tag{4.8}$$

During training, we selected a random $W \times W$ patch from each training video, and also randomly sampled a patch of the same size from the foveation map, to generate foveation masks from the patch. Foveation-weighted SSIM scores were calculated by applying a low-pass filter (Haar's filter) on the SSIM scores of each frame patch, then multiplying them by the foveation map patchs. The overall workflow is shown in Figure 4.5.

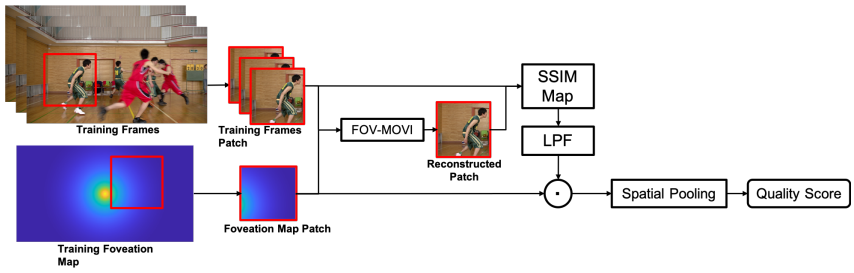


Figure 4.5: Training strategy.

4.3 Experiments

4.3.1 Settings

The Foveated MOVI-Codec networks that we experimented with were trained end-to-end on the Kinetics-600 dataset [95, 96] and on the Vimeo-90K dataset [97]. We used part of the testing set from Kinetics-600, which

consists of around 10,000 videos, to conduct our experiments. From each video, a random 192 $\times$ 192 patch containing 49 frames was randomly selected for training, and normalized the values of each input video to [-1,1]. Since Kinetics-600 dataset consist of YouTube videos of different resolutions, we randomly downsampled each original frames and extracted a 192 $\times$ 192 patches from the foveation maps to reduce any previously introduced compression artifacts . We randomly sampled 192 $\times$ 192 patches from the foveation maps to generate foveation masks for bitrate allocation. The Vimeo-90K dataset consists of 4,278 videos of fixed resolution 448 $\times$ 256. Since the videos in this dataset each have 7 frames, we randomly selected patches from each of the same size as mentioned earlier (overall 7 frames) for training.

We fixed the mini-batch size to 8 for training, while the step length N of the recurrent network was set as 7. We used Adamax optimizer for training and set the initial learning rate to 0.0001. The whole system is implemented based on PyTorch and using one Titan RTX GPU. By training on both the Vimeo-90K and the Kinetics-600 datasets, we are able to generalize our model to a wider range of natural motions. We tested the Foveated MOVI-Codec on the JCT-VC Class B datasets [103] and the UVG datasets [100]. Both of these testing datasets have HD resolution contents (1920 $\times$ 1080).

In order to assess the reconstruction quality of the foveation compressed videos, we utilized the perceptually relevant FWQI foveated video quality measurement tool follwing the same settings in [29], with screen width being 0.02 meters and display distance being 0.012 meters. We also used the foveated

SSIM model which deploys a fixed foveation map generated from the error sensitivity function from [119]. During testing, videos having different bitrates were generated using gaussian shape foveation maps with different foveation mask space constants, e.g. FMSCs of $\frac{H}{10}, \frac{H}{8}, \frac{H}{6}, \frac{H}{4}, \frac{H}{3}$, and $\frac{H}{2}$, where H is the height of the input frame. Examplar 1D slices through the gaussians are shown in Figure 4.6, while the corresponding quantized maps are shown in Figure 4.7. We fixed $\beta = 1$.

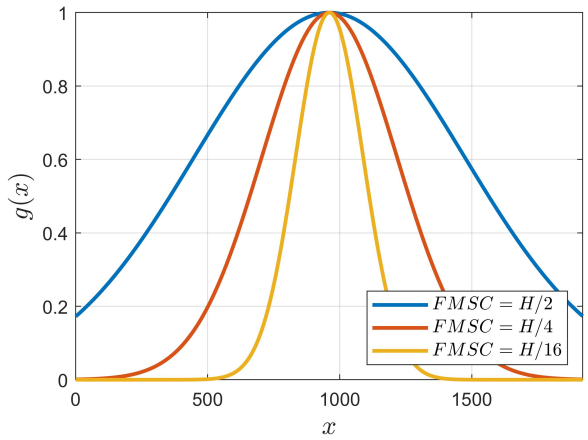


Figure 4.6: Normalized sliced profiles of gaussian foveation masks.

4.3.2 Results

4.3.2.1 Rate-Distortion Curve

We compared our video compression engine against the standardized hybrid codecs H.264 and H.265, and also against our previous non-foveated model, the MOVI-Codec, on the UVG dataset and the HEVC Standard Test Sequences Class B. In addition, we also implemented a foveated version of

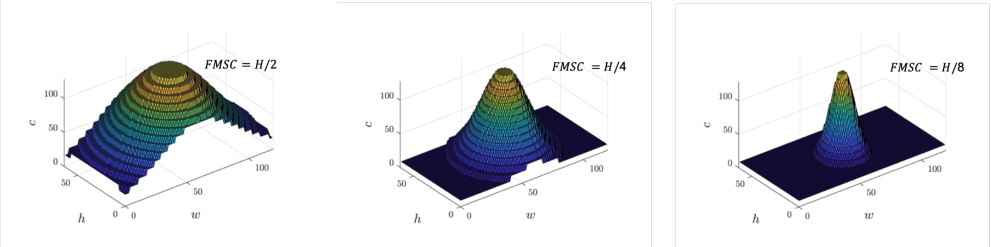


Figure 4.7: Examplar quantized gaussian foveation masks with different foveation mask space constants σ.

the hybrid codecs using the foveation method mentioned [26]. Both testing datasets have resolutions 1920×1080.

Figure 4.8 shows the results obtained on the UVG and HEVC Class B datasets. Unsurprisingly, the foveated version of the hybrid codecs outperforms their foveated counterparts in terms of FWQI. These results also show that our foveated model outperformed the non-foveated MOVI-Codec on both datasets. Moreover, the Foveated MOVI-Codec outperformed both H.264 and H.265, as well as their foveated counterparts, on both datasets. It is worth noting that the measured qualities of the reconstructed videos produced by Foveated MOVI-Codec produced did not vary much with respect to bitrate, suggested that our model is able to maintain a high quality fovea, while decreasing the bitrate derived from the periphery without sacrificing perceptual video quality. Visualizations of example frames compressed using different levels of bitrates and qualities are shown in Figure 4.9. More exemplar reconstructed videos are included on our project page with link given in the Abstract. In these reconstructed frames, we selected three regions for detailed comparison: one in the foveal region and the other two others in peripheral.

Our model is able to reconstruct videos having higher quality foveas and peripheral regions than the compared models, both visual and in terms of FWQI.

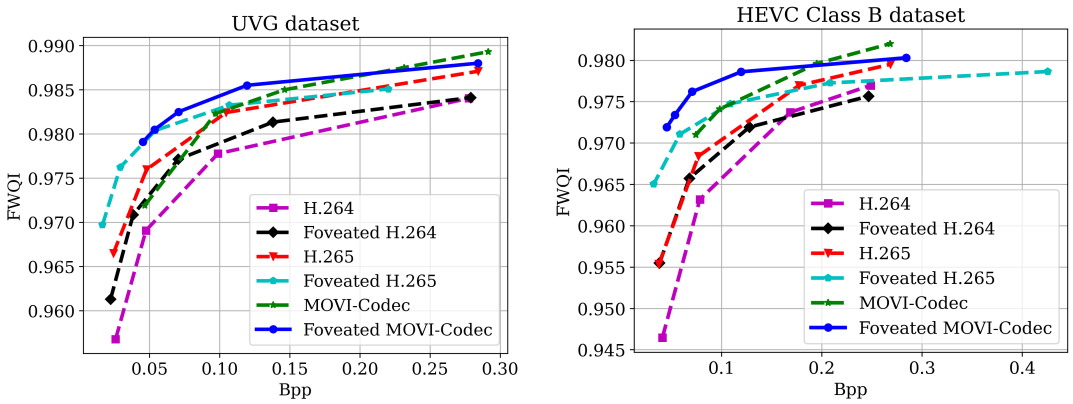
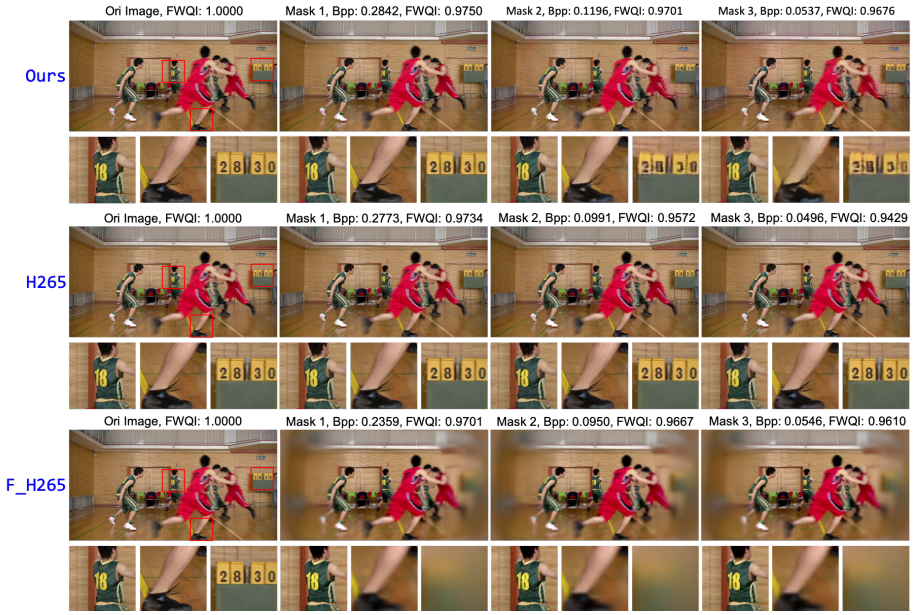


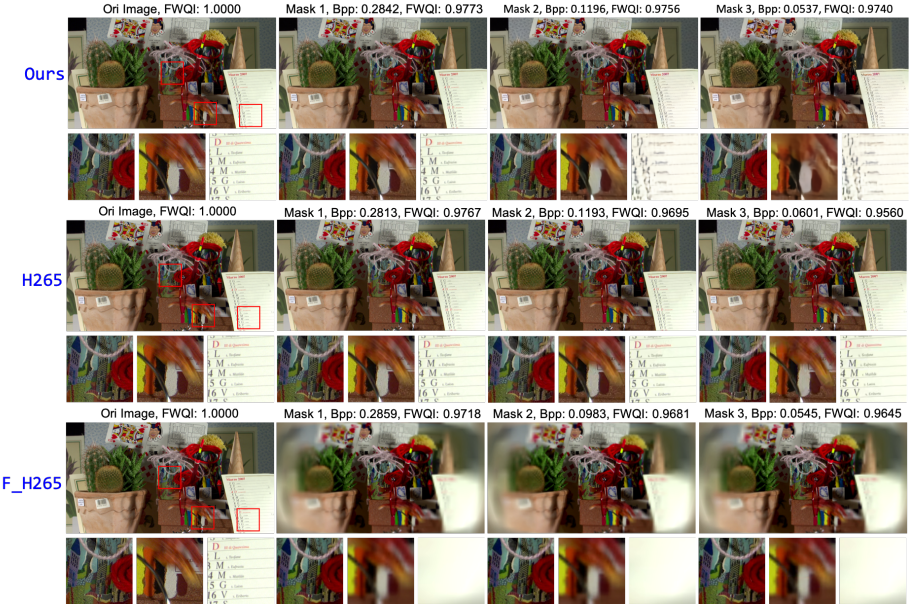
Figure 4.8: FWQI of the compared models on the UVG dataset and HEVC B test sequences. All video resolutions are 1920 × 1080.

4.3.2.2 Latent Representations

As mentioned in Section 4.2.3, the Foveated MOVI-Codec uses foveation maps to mediate bit allocations as a function of eccentricity relative to visual fixation. To visualize this process, we compared the latent representations (the encoded outputs) y_t in the Foveated MOVI-Codec against the encoded outputs y'_t of the original MOVI-Codec as shown in Figure 4.10. In the figure, the first row corresponds to reconstructed frames under different models, where the first column shows reconstructed frames from the MOVI-Codec, the second column contains reconstruction from the Foveated MOVI-Codec trained with a uniform (non-foveated) importance map with the masks of first N channels being one and zero elsewhere, and N is a random number during training.



(a) Basketball Drive



(b) Cactus

Figure 4.9: Visualizations of examplar foveated frames reconstructed by FOVMOVI-Codec, H.265, and Foveated H.265 (denoted F_265) on the videos (a) Basketball drive and (b) Cactus.

The remaining two columns show reconstructions from the Foveated MOVI-Codec with foveation mask space constants FMSCs equal to $H/2$ and $H/4$, respectively. The second row shows the corresponding accumulated feature maps, where brighter colors correspond to larger numbers of more features. This shows that more features are used to represent the foveal region, as the foveation maps become narrower (smaller FMSC). The last row of Figure 4.10 shows the latent representations at each level (8 channels per level) of the reconstructed frames. Figure 4.11 also shows the sum of latent representations of y_t and y'_t (foveated and non-foveated, respectively). As shown in Figure 4.11, the sum is roughly flat for the non-foveated compressor, whereas the sum decreases as with the channel number for the foveated compressor. This suggests that the foveated network was able to learn more relevant features in the first few channels. From Figures 4.10-4.11, we may conclude that the model learned efficient features across channels and bit allocation, even without the masked multiplication.

4.3.2.3 Bit Allocation

Figure 4.12 shows the reconstructed frames from the Foveated MOVI-Codec, differenced frames between original frames and reconstructed frames, and bits and SSIM profiles, when using different foveation space constants. From the differenced frames, we can conclude that our model is able to reconstruct a foveal region similar to the original frame regardless of the mask used. The third row of Figure 4.12 shows the both a bit allocation plot and the SSIM map profile for different models. From the SSIM map profile, it may

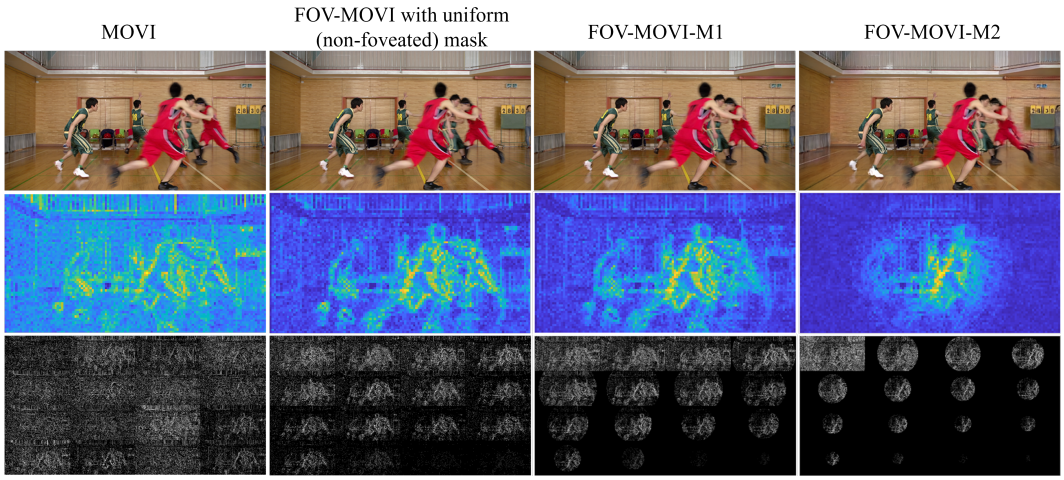
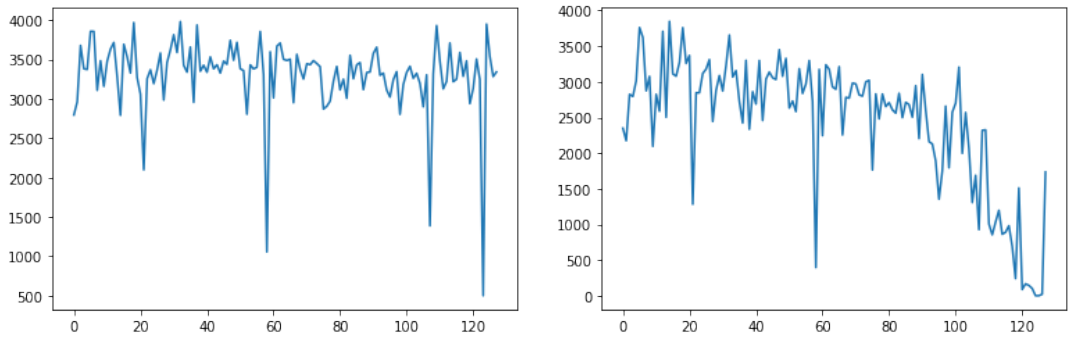


Figure 4.10: Latent representations generated from four models. The first row correponds to reconstructed frames from each model, the second row shows the cumulative latent representations, and the last row shows the latent representations at each compression level. FOV-MOVI-M1 is Foveated MOVI-Codec with foveation mask space constant $FMSC = H/2$ and FOV-MOV-M2 is Foveated MOVI-Codec with $FMSC = H/4$, where H is the height of the frame.

be observed that the lower number of bits allocated to the peripheral does not result in lower quality, since the quality of the reconstructed frames remain similar overall.

4.3.3 Discussion

Our experiments have shown that deploying foveation masks leads to much more efficient video compression for suitable environments, such as VR. Our new model outperformed H.264, H.265 and their foveated counterparts against FWQI across all testing sequences. Our model is best targeted at high resolution, gaze contingent foveated compression applications in VR and AR. The hierarchical masks make it possible to transmit scalably, viz., the first lev-



(a) MOVI-Codec (b) Foveated MOVI-Codec

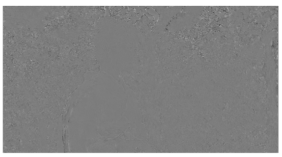
Figure 4.11: Sum of latent representations for each channel, where the sum is decreasing in foveated version.

els of content when bandwidth is limited, followed by the other levels. Since foveation masks are used in our model, the first transmitted levels correspond to foveal regions which draw the attention, and are the most important, supplying additional efficiency related to traditional hybrid codecs. Further, the new method is faster than MOVI-Codec since it does not require arithmetic coding. In the current model, the foveation masks are fixed with respect to frame height. One future direction is to train sets of masks adaptive to contents. Another direction is to extend the framework to generate a foveation map based on frequency as well and use it to allocate the contents learnt in the latent representation.

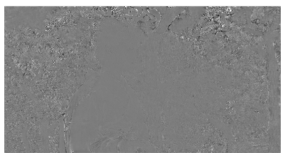
(a) Reconstructed frame with $FMSC = H/2$

(b) Reconstructed frame with $FMSC = H/4$

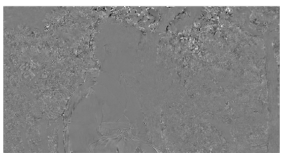
(c) Reconstructed frame with $FMSC = H/6$



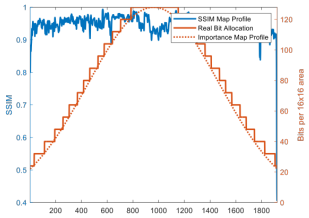
(d) Differenced frames with $FMSC = H/2$



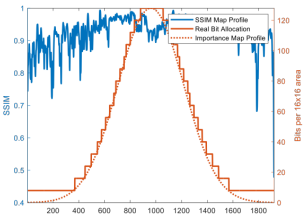
(e) Differenced frames with $FMSC = H/4$



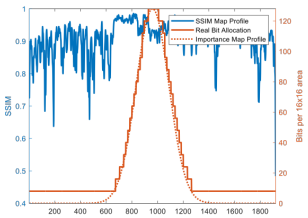
(f) Differenced frames with $FMSC = H/6$



(g) Bits and SSIM Profile with $FMSC = H/2$



(h) Bits and SSIM Profile with $FMSC = H/4$



(i) Bits and SSIM Profile with $sFMSC = H/6$

Figure 4.12: Reconstructed frames, differenced frames and bit-SSIM profiles under different foveation space constants (FMSCs).

Chapter 5

Conclusion and Future Work

With the increasing interest and applications of VR, it is essential to develop subjective and objective tools to understand and assessment immersive VR content quality. In addition, since VR contents are much larger in size and require higher bandwidth to transmit, it is also important to design codecs for better compressing VR content. In this dissertation, I presented my work towards exploring two perspectual aspects of VR: Quality and Compression. With regard to quality aspect of VR, we've built a 3D VR picture database

[1]Meixu Chen, Yize Jin, Todd Goodall, Xiangxu Yu, and Alan C. Bovik. Study of 3D virtual reality picture quality. *IEEE Journal of Selected Topics in Signal Processing,* 14(1):89–102, 2019.
Contributions: Meixu Chen: Writing, Software, Investigation, Formal Analysis; Yize Jin: Software, Investigation, Formal Analysis; Todd Goodall: Conceptualization; Xiangxu Yu: Investigation; Alan C. Bovik: Supervision, Conceptualization, Methodology, Review and Editing.

[2]Meixu Chen, Todd Goodall, Anjul Patney, and Alan C Bovik. Learning to compress videos without computing motion. *Signal Processing: Image Communication*, page 116633, 2022.
Contributions: Meixu Chen: Writing, Software, Investigation, Formal Analysis; Todd Goodall, Anjul Patney: Conceptualization; Alan C. Bovik: Supervision, Conceptualization, Methodology, Review and Editing.

[3]Meixu Chen, Richard Webb, and Alan C. Bovik, Foveation-based Deep Video Compression without Motion Search, *arXiv preprint arXiv 2203.16490*, 2022.
Contributions: Meixu Chen: Writing, Software, Investigation, Formal Analysis; Richard Webb: Conceptualization; Alan C. Bovik: Supervision, Conceptualization, Methodology, Review and Editing.

with eye tracking and made it publicly available. Towards the compression aspect of VR, we've developed a motionless deep learning based video compression codec called MOVI-Codec. In addition, we developed a foveated deep learning video compression without motion search, which we call Foveated MOVI-Codec.

3D VR image quality assessment. The free-viewing of high resolution, immersive VR implies significant data volume, which leads to challenges when storing, transmitting and rendering the images which can affect the viewing quality. Therefore, it is important to be able to analyze and predict the perceptual quality of immersive VR image. Towards meeting this challenge, we have created a comprehensive 3D immersive image database with 15 different contents and 6 distortion categories rated by 40 subjects. This database is the first to evaluate the gaze-tracked quality of stereoscopic 3D VR images in an immersive environment. We also evaluated the performance evaluation of eleven popular image quality assessment algorithms. The new LIVE 3D VR IQA Database is being made publicly and freely available for others to develop improved 2D and 3D VR IQA algorithms. Future work will focus on the use of visual saliency models using the eye tracking data provided with this database, as well as developing algorithms that target VR-specific distortions.

Deep learning-based video compression without motion. In both traditional video codecs and recent deep learning-based ones, motion estimation and compensation has occupied a significant portion of the system resources. Motion estimation requires an expensive search process that we avoid, by in-

stead training the network to efficiently represent the residuals between each current frame and a set of spatially-displaced neighboring frames. Computing a set of frame differences, even over many displacement directions is much cheaper than effective search processes. Moreover, while the statistics of motion are generally not regular, the intrinsic statistics of frame differences exhibit strong regularities. Inspired by this regularies exhibit in frame differences, we proposed an end-to-end deep learning video compression framework that renovated motion prediction. To be specific, we proposed the use of displaced frame differences as indicators of motion information, and fed them into a deep space-time compression network, which learns optimal between-frame interpolated representations to achieve efficiency. Additionally, we proposed a new version of UNet, called LSTM-UNet, that utilizes both spatial and temporal information to conduct frame reconstruction. Our experimental results show that our approach outperforms the LDP *veryfast* setting of the standard codecs H.264 and H.265 in terms of MS-SSIM. In addition, our network was able to outperform the latest H.266 codec at higher bitrates as assessed by the perceptual MS-SSIM algorithm, on high resolution videos. The reduced complexity of the framework and the avoidance of motion search could make it easier to implement on resrouce-limited devices. In MOVI-Codec, the selection of displacements between previous and current frame are hand-picked. It is possible to utilize another network to select a variety of combinations of displacements based on content and resolution of the frame, such that a more reasonable range of motion is capture.

Foveated deep learning-based video compression without motion search One advantage of VR is that the two eyes have fixed positions, aside from eye movements, relative to the viewing screen. Because of this, the eye movements, and associated points of gaze on the displays can be measured. This makes it possible to exploit the fact that the density of retinal photosensors is highly non-uniform. Since the density of photoreceptors falls away quite rapidly with increased eccentricity relative the fovea, much more efficient representations of what is perceived can be obtained by judiciously removing redundant information from peripheral regions. Based on this, we have proposed an end-to-end deep learning video compression framework that assigns bits according to a foveation protocol, assuming known visual fixations. We also achieve efficiency by training a deep space-time compression network to use displaced frame differences to compute efficient motion information by learning optimal between-frame interpolated representations. Our experimental results show that our approach, which we call FOV-MOVI-Codec, outperforms both H.264 and H.265 and foveated versions of them. The low complexity of our model, which avoids motion search and take advantage of the visual acuity falloff of the human visual system, could make it amenable for implementations on gaze-contigent devices. Future directions could include extending the current frame from generating foveation map based on a spatially varying contrast sensitivity function, to generating foveation map based on frequency as well, and use it to allocation the contents learnt in the latent represetations.

Bibliography

[1] Ennèl van Eeden and Wilson Chow. Perspectives from the global entertainment & media outlook 2018-2022, 2018. [Online] Available: https://www.pwc.com/gx/en/entertainment-media/outlook/perspectives-from-the-global-entertainment-and-media-outlook-2018-2022.pdf.

[2] Cisco Visual Networking Index. Cisco visual networking index: Forecast and methodology, 2016–2021. *Complete Visual Networking Index (VNI) Forecast*, 12(1):749–759, 2017.

[3] Cisco. Cisco annual internet report (2018–2023) white paper, 2020. [Online] Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[4] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

[5] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar.

Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085*, 2015.

[6] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5306–5314, 2017.

[7] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[8] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc V Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems*, pages 1141–1151, 2017.

[9] Johannes Ballé, Valero Laparra, and Eero Simoncelli. End-to-end optimized image compression. 2019.

[10] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4385–4393, 2018.

[11] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *International Conference on Learning Representations*, 2017.

[12] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *International Conference on Machine Learning*, pages 2922–2930, 2017.

[13] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. pages 221–231.

[14] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. pages 4394–4402, 2018.

[15] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10771–10780, 2018.

[16] Yash Patel, Srikar Appalaraju, and R Manmatha. Deep perceptual compression. *arXiv preprint arXiv:1907.08310*, 2019.

[17] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018.

[18] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pages 675–685, 2019.

[19] F Mentzer, E Agustsson, M Tschannen, R Timofte, and L Van Gool. Practical full resolution learned lossless image compression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10621–10630, 2019.

[20] Tong Chen, Haojie Liu, Qiu Shen, Tao Yue, Xun Cao, and Zhan Ma. Deepcoder: A deep neural network based video compression. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017.

[21] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 416–431, 2018.

[22] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11006–11015, 2019.

[23] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with hierarchical quality and recurrent enhancement. *arXiv preprint arXiv:2003.01966*, 2020.

[24] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. Foveated 3D graphics. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[25] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016.

[26] Wilson S Geisler and Jeffrey S Perry. Real-time foveated multiresolution system for low-bandwidth video communication. 3299:294–305, 1998.

[27] Zhou Wang and Alan C Bovik. Embedded foveation image coding. *IEEE Transactions on image processing*, 10(10):1397–1410, 2001.

[28] Zhou Wang, Ligang Lu, and Alan C Bovik. Foveation scalable video coding with automatic fixation selection. *IEEE Transactions on Image Processing*, 12(2):243–254, 2003.

[29] Anton S Kaplanyan, Anton Sochenov, Thomas Leimkühler, Mikhail Okunev, Todd Goodall, and Gizem Rufo. Deepfovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[30] D Purves, GJ Augustine, D Fitzpatrick, LC Katz, AS LaMantia, JO McNamara, and SM Williams. Functional specialization of the rod and

cone systems. *Neuroscience*, 2, 2001.

[31] Ben M Harvey and Serge O Dumoulin. The relationship between cortical magnification factor and population receptive field size in human visual cortex: constancies in cortical architecture. *Journal of Neuroscience*, 31(38):13604–13612, 2011.

[32] Heinz Wässle, Ulrike Grünert, Jürgen Röhrenbeck, and Brian B Boycott. Retinal ganglion cell density and cortical magnification factor in the primate. *Vision research*, 30(11):1897–1911, 1990.

[33] Brian Cheung, Eric Weiss, and Bruno Olshausen. Emergence of foveal image sampling from learning to attend in visual scenes. *arXiv preprint arXiv:1611.09430*, 2016.

[34] Cornelius Weber and Jochen Triesch. Implementations and implications of foveated vision. *Recent Patents on Computer Science*, 2(1):75–85, 2009.

[35] Hamid R Sheikh, Muhammad F Sabir, and Alan C Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. Image Process.*, 15(11):3440–3451, 2006.

[36] Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and C.-C. Jay Kuo. Image database TID2013: Pe-

culiarities, results and perspectives. *Signal Process., Image Commun.*, 30:57–77, 2015.

[37] Eric C Larson and DM Chandler. Categorical image quality (CSIQ) database, 2010.

[38] Deepti Ghadiyaram and Alan C Bovik. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Trans. Image Process.*, 25(1):372–387, 2016.

[39] Huiyu Duan, Guangtao Zhai, Xiaokang Yang, Duo Li, and Wenhan Zhu. IVQAD 2017: An immersive video quality assessment database. In *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–5. IEEE, 2017.

[40] Evgeniy Upenik, Martin Řeřábek, and Touradj Ebrahimi. Testbed for subjective evaluation of omnidirectional visual content. In *2016 Picture Coding Symposium (PCS)*, pages 1–5. IEEE, 2016.

[41] Wei Sun, Ke Gu, Guangtao Zhai, Siwei Ma, Weisi Lin, and Patrick Le Calle. CVIQD: Subjective quality evaluation of compressed virtual reality images. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3450–3454. IEEE, 2017.

[42] Huiyu Duan, Guangtao Zhai, Xiongkuo Min, Yucheng Zhu, Yi Fang, and Xiaokang Yang. Perceptual quality assessment of omnidirectional im-

ages. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.

[43] Chen Li, Mai Xu, Xinzhe Du, and Zulin Wang. Bridge the gap between vqa and human behavior on omnidirectional video: A large-scale dataset and a deep learning model. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 932–940, New York, NY, USA, 2018. ACM.

[44] Mai Xu, Chen Li, Yufan Liu, Xin Deng, and Jiaxin Lu. A subjective visual quality assessment method of panoramic videos. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 517–522. IEEE, 2017.

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[46] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Proc. 37th Asilomar Conf. Signals Syst. Comput.*, volume 2, pages 1398–1402, Nov. 2003.

[47] Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Trans. Image Process.*, 15(2):430–444, 2006.

[48] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. FSIM: A feature similarity index for image quality assessment. *IEEE Trans.*

Image Process., 20(8):2378–2386, 2011.

[49] Wufeng Xue, Lei Zhang, Xuanqin Mou, and Alan C Bovik. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE Trans. Image Process.*, 23(2):684–695, 2014.

[50] Lin Zhang, Ying Shen, and Hongyu Li. VSI: A visual saliency-induced index for perceptual image quality assessment. *IEEE Trans. Image Process.*, 23(10):4270–4281, 2014.

[51] Hossein Ziaei Nafchi, Atena Shahkolaei, Rachid Hedjam, and Mohamed Cheriet. Mean deviation similarity index: Efficient and reliable full-reference image quality evaluator. *IEEE Access*, 4:5579–5590, 2016.

[52] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.*, 21(12):4695–4708, 2012.

[53] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.

[54] Michele A Saad and Alan C Bovik. Blind quality assessment of videos using a model of natural scene statistics and motion coherency. In *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 332–336. IEEE, 2012.

[55] Peng Ye, Jayant Kumar, Le Kang, and David Doermann. Unsupervised feature learning framework for no-reference image quality assessment. In *Proc. IEEE Conf. Comp. Vis. Pattern Recog.*, pages 1098–1105. IEEE, 2012.

[56] Matt Yu, Haricharan Lakshman, and Bernd Girod. A framework to evaluate omnidirectional video coding schemes. In *2015 IEEE International Symposium on Mixed and Augmented Reality*, pages 31–36. IEEE, 2015.

[57] Vladyslav Zakharchenko, Kwang Pyo Choi, and Jeong Hoon Park. Quality metric for spherical panoramic video. In *Optics and Photonics for Information Processing X*, volume 9970, page 99700C. International Society for Optics and Photonics, 2016.

[58] Mai Xu, Chen Li, Zhenzhong Chen, Zulin Wang, and Zhenyu Guan. Assessing visual quality of omnidirectional videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[59] Yule Sun, Ang Lu, and Lu Yu. Weighted-to-spherically-uniform quality evaluation for omnidirectional video. *IEEE signal processing letters*, 24(9):1408–1412, 2017.

[60] Sijia Chen, Yingxue Zhang, Yiming Li, Zhenzhong Chen, and Zhou Wang. Spherical structural similarity index for objective omnidirectional video quality assessment. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018.

[61] Luyu Yang, Zhigang Tan, Zhe Huang, and Gene Cheung. A content-aware metric for stitched panoramic image quality assessment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2487–2494, 2017.

[62] Suiyi Ling, Gene Cheung, and Patrick Le Callet. No-reference quality assessment for stitched panoramic images using convolutional sparse coding and compound feature selection. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018.

[63] Jiachen Yang, Tianlin Liu, Bin Jiang, Houbing Song, and Wen Lu. 3D panoramic virtual reality video quality assessment based on 3D convolutional neural networks. *IEEE Access*, 6:38669–38682, 2018.

[64] Heaun-Taek Lim, Hak Gu Kim, and Yang Man Ra. VR IQA net: Deep virtual reality image quality assessment using adversarial learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6737–6741. IEEE, 2018.

[65] Hak Gu Kim, Heoun-taek Lim, and Yong Man Ro. Deep virtual reality image quality assessment with human perception guider for omnidirectional image. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[66] Insta 360 Pro, 2019. [Online] Available: `https://www.insta360.com/product/insta360-pro/`.

[67] P ITU-T RECOMMENDATION. Subjective video quality assessment methods for multimedia applications. *International telecommunication union*, 1999.

[68] David Hasler and Sabine E Suesstrunk. Measuring colorfulness in natural images. In *Human vision and electronic imaging VIII*, volume 5007, pages 87–96. International Society for Optics and Photonics, 2003.

[69] Peiyao Guo, Qiu Shen, Zhan Ma, David J Brady, and Yao Wang. Perceptual quality assessment of immersive images considering peripheral vision impact. *arXiv preprint arXiv:1802.09065*, 2018.

[70] Rongbing Zhou, Mingkai Huang, Shuyi Tan, Lijun Zhang, Du Chen, Jie Wu, Tao Yue, Xun Cao, and Zhan Ma. Modeling the impact of spatial resolutions on perceptual quality of immersive image/video. In *2016 International Conference on 3D Imaging (IC3D)*, pages 1–6. IEEE, 2016.

[71] Int. Telecommun. Union. Methodology for the subjective assessment of the quality of television pictures ITU-R recommendation BT.500-13. *Tech. Rep.*, 2012.

[72] Tobii pro SDK, 2019. [Online] Available: `http://developer.tobiipro.com/index.html`.

[73] Kalpana Seshadrinathan, Rajiv Soundararajan, Alan Conrad Bovik, and Lawrence K Cormack. Study of subjective and objective quality assess-

ment of video. *IEEE transactions on Image Processing*, 19(6):1427–1441, 2010.

[74] Yashas Rai, Jesús Gutiérrez, and Patrick Le Callet. A dataset of head and eye movements for 360 degree images. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 205–210. ACM, 2017.

[75] Gregory K Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.

[76] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal processing magazine*, 18(5):36–58, 2001.

[77] Fabrice Bellard. BPG image format, 2015. [Online] Available: `https://bellard.org/bpg`.

[78] Debargha Mukherjee, Jim Bankoski, Adrian Grange, Jingning Han, John Koleszar, Paul Wilkins, Yaowu Xu, and Ronald Bultje. The latest open-source video codec vp9-an overview and preliminary results. In *Picture Coding Symposium (PCS)*, pages 390–393, 2013.

[79] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):1995, 1995.

[80] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[81] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014.

[82] Amirhossein Habibian, Ties van Rozendaal, Jakub M Tomczak, and Taco S Cohen. Video compression with rate-distortion autoencoders. *IEEE International Conference on Computer Vision*, pages 7033–7042, 2019.

[83] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G Anderson, and Lubomir Bourdev. Learned video compression. *IEEE International Conference on Computer Vision*, pages 3454–3463, 2019.

[84] Zhibo Chen, Tianyu He, Xin Jin, and Feng Wu. Learning for video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):566–576, 2019.

[85] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learning image and video compression through spatial-temporal energy compaction. pages 10071–10080, 2019.

[86] Giyong Choi, PyeongGang Heo, Se Ri Oh, and HyunWook Park. A new motion estimation method for motion-compensated frame interpolation using a convolutional neural network. pages 800–804, 2017.

[87] Hyomin Choi and Ivan V Bajić. Deep frame prediction for video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[88] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017.

[89] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. pages 2758–2766, 2015.

[90] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. pages 2462–2470, 2017.

[91] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. pages 8981–8989, 2018.

[92] Rajiv Soundararajan and Alan C Bovik. Video quality assessment by reduced reference spatio-temporal entropic differencing. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4):684–694, 2012.

[93] Dae Yeol Lee, Hyunsuk Ko, Jongho Kim, and Alan C Bovik. On the space-time statistics of motion pictures. *JOSA A*, 38(7):908–923, 2021.

[94] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[95] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[96] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[97] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

[98] Video Trace Library. VTL test sequences, 2020.

[99] Frank Bossen et al. Common test conditions and software reference configurations. *JCTVC-L1100*, 12:7, 2013.

[100] Ultra Video Group. UVG test sequences, 2020.

[101] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. An end-to-end learning framework for video compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[102] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[103] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

[104] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

[105] Fraunhofer Heinrich Hertz Institute. Fraunhofer Versatile Video Encoder (VVenC), 2021.

[106] Catarina Brites and Fernando Pereira. Distributed video coding: Assessing the hevc upgrade. *Signal Processing: Image Communication*, 32:81–105, 2015.

[107] Sanghoon Lee, Marios S Pattichis, and Alan C Bovik. Foveated video quality assessment. *IEEE Transactions on Multimedia*, 4(1):129–132, 2002.

[108] Giuseppe Boccignone, Angelo Chianese, Vincenzo Moscato, and Antonio Picariello. Foveated shot detection for video segmentation. *IEEE*

Transactions on Circuits and Systems for Video Technology, 15(3):365–377, 2005.

[109] Alper Koz and A Aydin Alatan. Foveated image watermarking. 3:657–660, 2002.

[110] Zhou Wang and Alan C Bovik. Foveated image and video coding. In *Digital Video Image Quality and Perceptual Coding*, pages 431–458. CRC Press, 2005.

[111] Zhou Wang. *Rate-Scalable Foveated Image and Video Communications*. The University of Texas at Austin, 2001.

[112] Richard S Wallace, Ping-Wen Ong, Benjamin B Bederson, and Eric L Schwartz. Space variant image processing. *International Journal of Computer Vision*, 13(1):71–90, 1994.

[113] Norimichi Tsumura, Chizuko Endo, Hideaki Haneishi, and Yoichi Miyake. Image compression and decompression based on gazing area. 2657:361–367, 1996.

[114] Shizhong Liu and Alan C Bovik. Foveation embedded dct domain video transcoding. *Journal of Visual Communication and Image Representation*, 16(6):643–667, 2005.

[115] Hamid R Sheikh, Shizhong Liu, Zhou Wang, and Alan C Bovik. Foveated multipoint videoconferencing at low bit rates. 2:II–2069, 2002.

[116] Peter J Burt. Smart sensing within a pyramid vision machine. *Proceedings of the IEEE*, 76(8):1006–1015, 1988.

[117] Zhicheng Li, Shiyin Qin, and Laurent Itti. Visual attention guided bit allocation in video compression. *Image and Vision Computing*, 29(1):1–14, 2011.

[118] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. pages 3214–3223, 2018.

[119] Zhou Wang, Alan Conrad Bovik, Ligang Lu, and Jack L Kouloheris. Foveated wavelet image quality index. 4472:42–53, 2001.

[120] Snježana Rimac-Drlje, Mario Vranješ, and Drago Žagar. Foveated mean squared error - a novel video quality metric. *Multimedia tools and applications*, 49(3):425–445, 2010.

[121] Snježana Rimac-Drlje, Goran Martinović, and Branka Zovko-Cihlar. Foveation-based content adaptive structural similarity index. pages 1–4, 2011.

[122] Junyong You, Touradj Ebrahimi, and Andrew Perkis. Attention driven foveated video quality assessment. *IEEE Transactions on Image Processing*, 23(1):200–213, 2014.

[123] Meixu Chen, Todd Goodall, Anjul Patney, and Alan C Bovik. Learning to compress videos without computing motion. *Signal Processing: Image Communication*, page 116633, 2022.

[124] Zhou Wang and Qiang Li. Information content weighting for perceptual image quality assessment. *IEEE Trans. Image Process.*, 20(5):1185–1198, 2011.

[125] Rajiv Soundararajan and Alan C Bovik. RRED indices: Reduced reference entropic differencing for image quality assessment. *IEEE Trans. Image Process.*, 21(2):517–526, 2012.

[126] Laboratory for Image & Video Engineering. Image & video quality assessment at LIVE, 2019. [Online] Available: `http://live.ece.utexas.edu/research/Quality/index.htm`.

[127] Joseph J Atick and A Norman Redlich. Towards a theory of early visual processing. *Neural Computation*, 2(3):308–320, 1990.

[128] Fred Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183, 1954.

[129] Dawei W Dong and Joseph J Atick. Temporal decorrelation: a theory of lagged and nonlagged responses in the lateral geniculate nucleus. *Network: Computation in Neural Systems*, 6(2):159–178, 1995.

[130] Martina Poletti and Michele Rucci. A compact field guide to the study of microsaccades: Challenges and functions. *Vision research*, 118:83–97, 2016.

[131] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[132] Sungoh Kim, Chansik Park, Hyungju Chun, and Jaemoon Kim. A novel fast and low-complexity motion estimation for uhd hevc. pages 105–108, 2013.

[133] EJ Chichilnisky and Rachel S Kalmar. Functional asymmetries in on and off ganglion cells of primate retina. *Journal of Neuroscience*, 22(7):2737–2747, 2002.

[134] Ralf Engbert. Microsaccades: A microcosm for research on oculomotor control, attention, and visual perception. *Progress in Brain Research*, 154:177–192, 2006.

[135] Michele Rucci and Jonathan D Victor. The unsteady eye: an information-processing stage, not a bug. *Trends in Neurosciences*, 38(4):195–206, 2015.

[136] Shan Zhu and Kai-Kuang Ma. A new diamond search algorithm for fast block-matching motion estimation. *IEEE Transactions on Image Processing*, 9(2):287–290, 2000.

[137] Gisle Bjontegaard. Calculation of average PSNR differences between RD-curves. *VCEG-M33*, 2001.

[138] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[139] Meixu Chen, Yize Jin, Todd Goodall, Xiangxu Yu, and Alan Conrad Bovik. Study of 3D virtual reality picture quality. *IEEE Journal of Selected Topics in Signal Processing*, 14(1):89–102, 2019.

[140] Christine A Curcio, Kenneth R Sloan, Robert E Kalina, and Anita E Hendrickson. Human photoreceptor topography, 1990.

Vita

Meixu Chen received the B.Eng. degree in information engineering from Xi'an Jiaotong University, Xi'an, China, and the M.S. degree in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, in 2016 and 2019, respectively. Since 2017, she has been a Research Assistant with the Laboratory for Image and Video Engineering, The University of Texas at Austin. Her research interests focus on image and video processing, machine learning, and perception.

Contact: chenmx@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.