FAST DEPTH CODING IN 3D-HEVC USING DEEP LEARNING

A DISSERTATION
SUBMITTED TO THE
DEPARTMENT OF ELECTRONIC AND INFORMATION ENGINEERING
OF THE HONG KONG POLYTECHNIC UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Zhen-xiang WANG
November 2017

CERTIFICATE OF ORIGINALITY

I hereby declare that this dissertation is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written nor material which has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________________(Signed)

______________________________(Date)

Abstract

The 3D Extension of the High Efficiency Video Coding standard (3D-HEVC), which has been finalized by the Joint Collaborative Team on Video Coding (JCT-VC) in February 2015, is the new industry standard for 3D applications. The 3D-HEVC provides plenty of advanced coding tools specifically for addressing the coding of auto-stereoscopic videos which have the format of multiple texture views along with the depth maps which are responsible for synthesising intermediate views with sufficient quality for auto-stereoscopic display. The provided tools take advantage of the statistical redundancies amongst texture views and depth maps in the video sequences, as well as the unique characteristics of depth maps to significantly shrink the bit-rate while preserving the objective visual quality of the 3D videos. However, those tools with high capability in terms of compression come with the high complexity of computation which has made the encoding time of the 3D video sequences much longer than ever by traversing a lot more candidates, calculating time-consuming RD Cost for each of them, especially in the wedgelet searching process for depth maps. While this full-search style method can promise to find the best candidate in depth intra mode decision, the time cost is expensive.

In this dissertation we address the time cost by presenting a new intra mode decision method for depth maps, leveraging the deep convolutional neural networks to predict the wedgelet angles for the depth blocks. The predictions from the learned models are capable of reducing the number of wedgelet candidates by half as well as the angular modes in depth map coding. The size of the neural network has been carefully designed to balance the trade-off between the time cost of model prediction and the model prediction accuracy. Confusion matrix is used to monitor the training process. Top-K criteria is employed for the prediction. We have integrated the learned models into the reference software of 3D-HEVC for the experiments. The compiled executable binaries are able to harness the power of the simultaneous computation of CPU, as well as the parrallel computation of GPU to accelerate the predictions. The simulation results show that the proposed algorithm provides 64.6% time reduction in average while the BD performance has a tiny decrease comparing with the state-of-the-art 3D-HEVC standard.

Acknowledgments

First and foremost, I would like to give sincere thanks to my supervisor, Dr.Yui-Lam Chan, for his extremely generous support, most insightful advices and innumerable yet constructive feedback. I learned from him to first identify a problem, by reading a vast amount of articles to know what people have achieved and what bottlenecks they have encountered. I learned how to read papers, how to organize them to become the inner comprehension. He guided me to use the machine learning approach to solve the problem that has been found in the first stage. Without his guidance I will not have the idea to learn the deep learning technology and apply it to optimize the video coding. His encyclopedic knowledge and charming personalities made him my mentor in both research and life. I wish to thank Dr.Sik-Ho Tsang, for our in-depth discussions from which I can always find useful clues to proceed to next step. His great expertise in video coding significantly benefits me during my intensive period of learning. Also I would like to thank my friends Alex and Jacky, for our extensive discussions about artificial intelligence and their applications. Finally thank you my parents, for the great love and constant encouragement which give me confidence to face and handle all the challenges at every moment.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Video is the medium to record, copy, playback, broadcast and display the motion images in an electronic style [1]. Watching videos is becoming an important way for our entertainment as well as education. The high definition (HD) and ultra high definition (UHD) video are increasingly demanding nowadays. People prefer videos with higher definitions than those with lower resolutions because the former one provides much better viewing experience. However, challenges emerged for delivering videos with high definition. HD videos typically contain much more information in every picture frame than the standard definition videos. More data needs to be squeezed into the same capacity for transmission. For example, the uncompressed video with the dimension 720 x 480 at 30 frames per second requires 0.03 gigabytes per second, while the uncompressed video with the dimension 2880 x 2048 at 120 frames per second requires 2.12 gigabytes per second. Since bit rate is proportional to system bandwidth for transmission [2], and expanding the bandwidth in a large scale is too expensive, the significantly increased bit rate for transmitting the video data is becoming one of the major obstacles for HD video services.

To cope with the growing need for higher compression of moving pictures [3], Joint Collaborative Team on Video Coding (JCT-VC) [4] has developed the High Efficiency Video Coding standard which is the newest international video coding standard for substantially ameliorate the compression performance against the previous standards. Comparing with the H.264 Advanced Video Compression Standard [5], the H.265 High Efficiency Video Coding Standard provides fifty percent bit rate reduction while maintaining the objective video quality at the same level.

While Two-dimensional video is the most common video type, Three-dimensional (3D) video has been brought to market via lots of ways, including Blu-Ray disc, cable and satellite transmission, terrestrial broadcast, and streaming or downloading from the Internet [6]. 3D video provides the perception of depth information which augments the vividness of the video contents. Currently most 3D videos in the market are using stereo display technology. Two similar views, one for left eye, the other for right eye, are presented at the same time with the multiplexing techniques

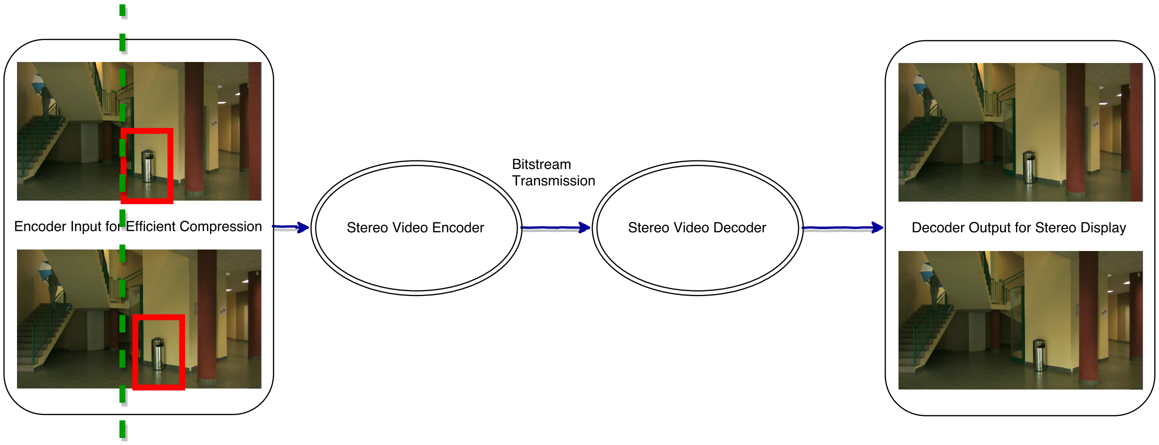

Encoder Input for Efficient Compression
Stereo Video Encoder
Bitstream Transmission
Stereo Video Decoder
Decoder Output for Stereo Display

Figure 1.1: System Structure for transmitting videos targeting stereo display.

enabling the adjustments of video geometry information [7] to provide the 3D effect. Figure 1.1 illustrates the typical system structure for transmitting videos targeting stereo display. It can be observed that there exists a displacement between the two views. The green vertical left margins of the red rectangles in the two views at encoder side are different. Such a displacement is the visual disparity for 3D perception. Stereoscopic videos [8] have achieved great profitability for movie theatres in recent years. For example, IMAX 3D has became the most popular one that offering the immersing multimedia experiences around the world. Special 3D glasses are needed for watching the IMAX 3D movies. The current 3D film industry is very successful in terms of attracting customers, however, it is not the end of the story. Myopic people do not like to wear one more pair of glasses when watching 3D movies. Some people will experience discomfort after wearing the 3D glasses for a period of two hours. To get rid of the undesired 3D glasses, autostereoscopic multi-view technology [8] is coming to our rescue. The two major different characteristics between stereo display and autostereoscopic display are listed in Table 1.1 [9]. The impact of different view numbers for autostereoscopic display is shown in Table 1.2 [9]. Comparative ease can be brought to the 3D video audience since they do not need to wear 3D glasses for watching autostereoscopic videos. At each different view position, scenes with minor differences are available from multiple stereo pairs which are provided by autostereoscopic display [9]. As a result, when audience make a move for various view positions, scenes not viewable from the previous locations are revealed during the movement.

Table 1.1: Characteristics Comparison of Stereoscopic Display and Autostereoscopic Display

Characteristic	Stereo Display	Autostereoscopic Display
Glass-Free	No	Yes
Multiple Stereo Pairs	No	Yes

Table 1.2: Impact of Available View Amount for Autostereoscopic Display

Characteristic	Small Number of Views	Large Number of Views
Seamless View Transition	No	Yes
High Quality of Scene Depth	No	Yes

The autostereoscopic multi-view display demands more than two views. With a sufficient amount of views present in autostereoscopic display, the disparities between every two adjacent views can be small enough to offer seamless transitions from scene to scene, such that when multiple views meet eyes sequentially, the scenes as a whole can be gorgeous. The visual quality of the autostereoscopic display is highly proportional to the number of available views. Due to limited available bandwidth, transmitting arbitrary number of views is not practical. Researchers have proposed a new format which only requires limited number of view and their associated depth maps for the capability of generating arbitrary amount of views theoretically. The typical system structure for using this new format to compress and supply 3D video resources is shown in Figure 1.2. An enormous amount of views in the medium positions which are able to guarantee the high quality of the 3D video can be synthesized from the decoded texture frames in combination with decoded depth maps.

To employ multi-view plus depth format for 3D video, efficient compressing methods are desired, which has led to the 3D Video Coding Extension of the High Efficiency Video Coding Standard (3D-HEVC) by the Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) [10]. The 3D Extension of the HEVC standard gives extra coding efficiency for encoding a few texture views along with the corresponding depth maps by using new tools which exploit the redundancies amongst texture and depth views, and pay attention to the unique characteristics of

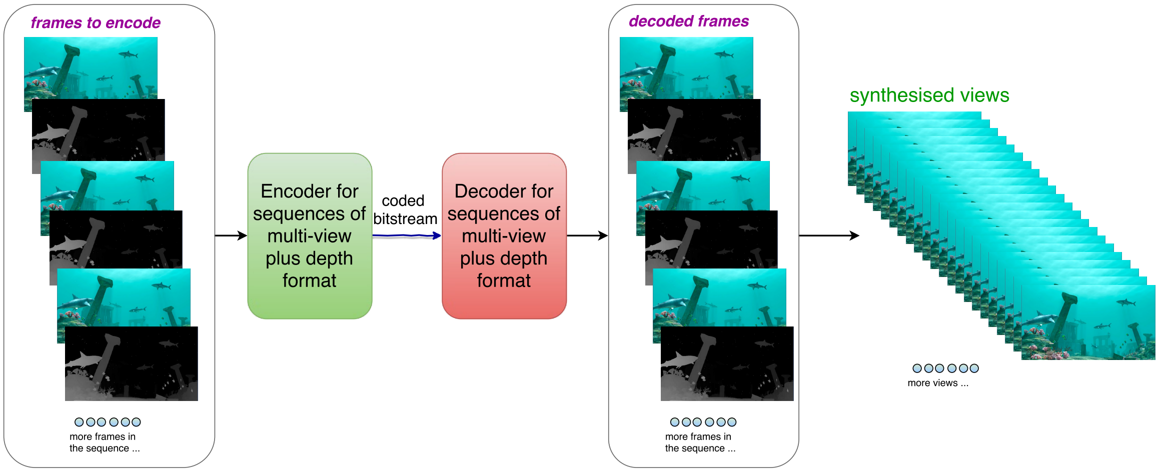

frames to encode
more frames in the sequence ...
Encoder for sequences of multi-view plus depth format
coded bitstream
Decoder for sequences of multi-view plus depth format
decoded frames
more frames in the sequence ...
synthesised views
more views ...

Figure 1.2: System Structure for transmitting videos of Multi-view Plus Depth format.

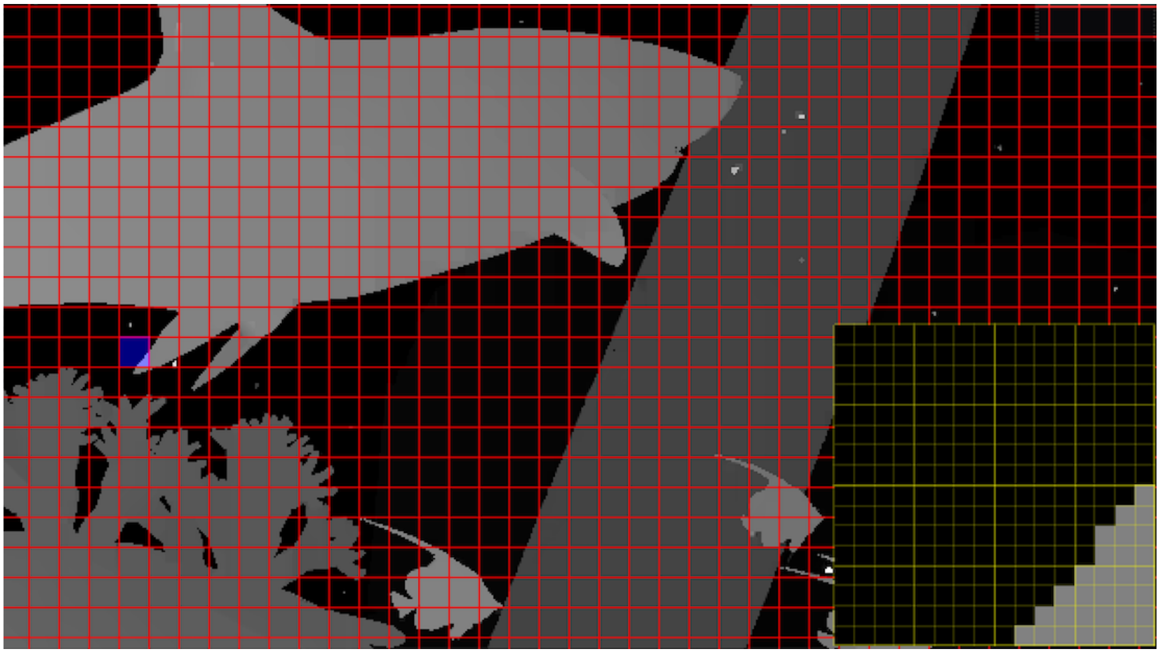


Figure 1.3: Example of wedgelet partition in a block of size 16 by 16 in depth map from Shark video sequence.

the depth maps, such as large homogeneous regions separated by sharp boundaries [11].

Depth information measures of the distance between the object in the far position and the object in the near position from a static viewpoint, which is expressed in the format of depth map. Instead of presenting depth maps directly to the viewer, views in the medium positions are generated by Depth-Image-Based Rendering (DIBR) technique. The qualities of the depth maps are vital to the DIBR process. Corona artifacts (a.k.a. ringing artifacts) can be discovered in synthesized views if the edge sharpness in depth maps can not be well preserved. Therefore, retaining the edge sharpness in depth map is the key to avoid the artifacts in the synthesized views. In 3D-HEVC, new intra-picture prediction tools and residual coding methods have been applied to preserve the special properties of depth maps. Depth Modelling Mode (DMM) which is one of the new intra-picture prediction tools, is designed to provide much more granularity for encoding the depth maps than the normal angular intra prediction modes. DMM is more capable of approximating the depth maps to be encoded due to the fact that it provides a vast amount of non-rectangle partitions. Figure 1.3 presents an example of the wedgelet partition from the depth map in Shark video sequence. The small block highlighted by blue color amongst the blocks separated by the red grid is magnified at the right-bottom position in Figure 1.3. A straight line is used for the partition in wedgelet mode. Figure 1.4 shows a sample of the contour partition from the same depth map as Figure 1.3. The partition pattern comprises contour lines instead of one single straight line. Wedgelet partition and contour partition for depth

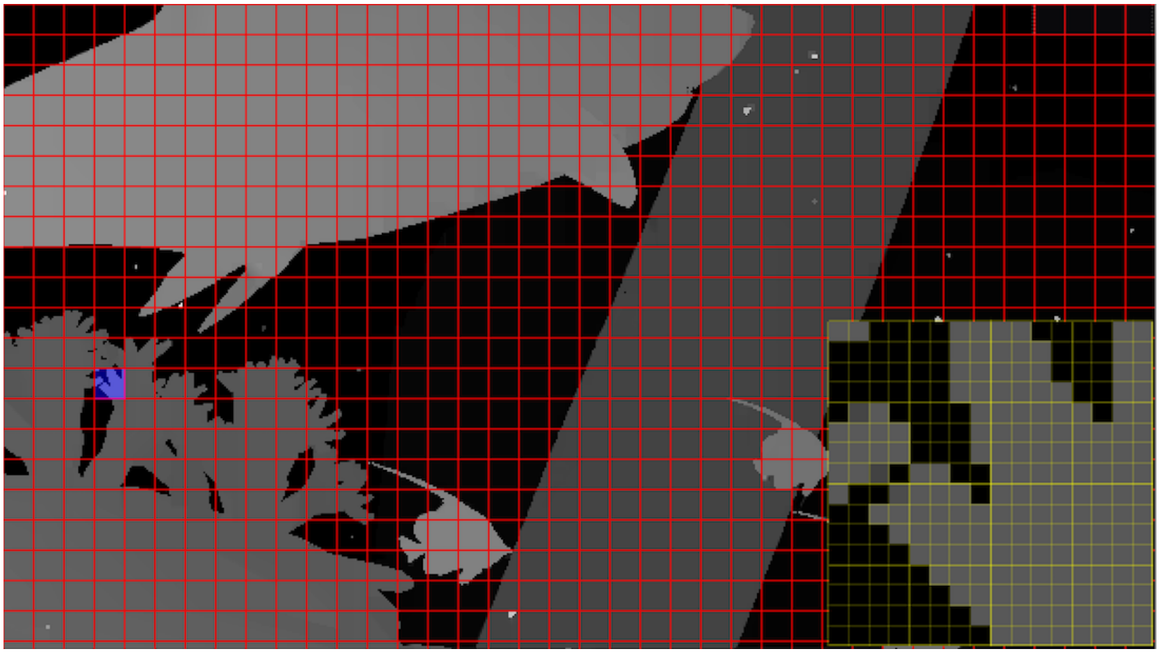


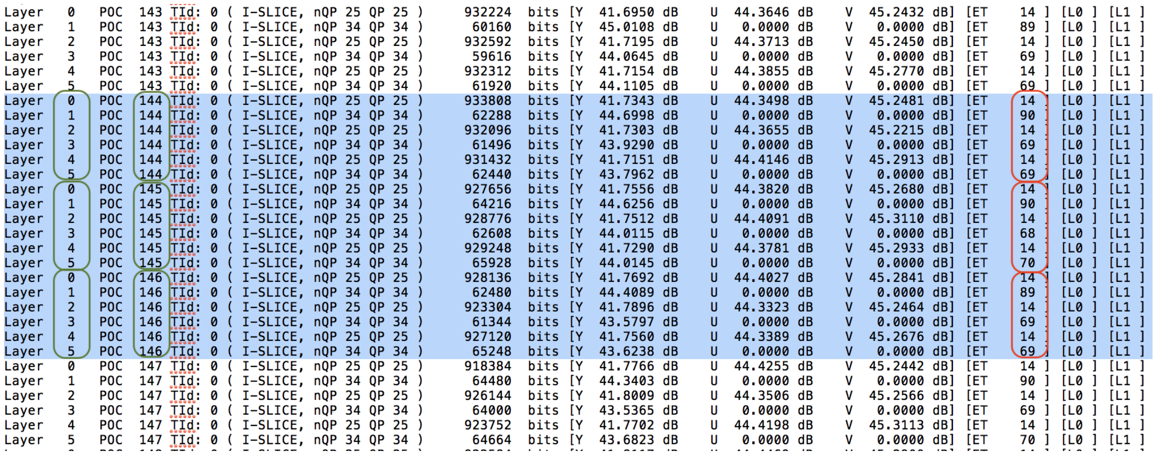
Figure 1.4: Example of contour partition in a block of size 16 by 16 in depth map from Shark video sequence.

maps are enabled by DMM1 and DMM4 separately.

1.1 Motivation

The idea of this work originates from the discovery of the computational complexity of the wedgelet searching process in depth modelling modes. The immense complexity for searching the best wedgelet candidate lead to the strongly marked increase of encoding time. The time consumed for compressing a single depth map in 3D-HEVC encoder is roughly a sixfold increase relevant to the encoding time of a single texture frame wherein the all intra configuration in HTM-16.2 is used. Thus we designed a deep neural network architecture which is trained subsequently for predicting the most probable wedgelet candidates. The learned model achieves 92.2% to 97.3% top-16 accuracy for various block sizes. The inference engine is integrated into the reference software (HTM-16.2) of 3D-HEVC. The learned models reduce roughly half of the wedgelet searching candidates. It provides 64.6% time reduction in average while the BD performance has a negligible decrease comparing with the unmodified 3D-HEVC encoder.

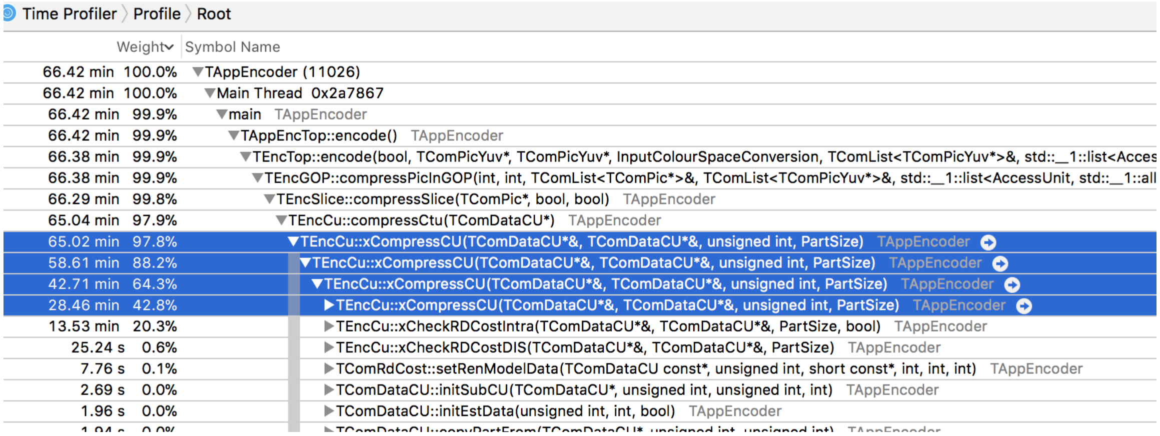
Motivation for Wedgelet Candidates Reduction: The encoding time consumed in HTM-16.2 encoder for each view (including both texture and depth) by default can be observed from the command line outputs. Figure 1.5 shows a piece of command line outputs from the encoding process



```
Layer  0   POC  143 TId: 0 ( I-SLICE, nQP 25 QP 25 )     932224  bits [Y  41.6950 dB   U  44.3646 dB    V  45.2432 dB] [ET   14 ] [L0 ] [L1 ]
Layer  1   POC  143 TId: 0 ( I-SLICE, nQP 34 QP 34 )      60160  bits [Y  45.0108 dB   U   0.0000 dB    V   0.0000 dB] [ET   89 ] [L0 ] [L1 ]
Layer  2   POC  143 TId: 0 ( I-SLICE, nQP 25 QP 25 )     932592  bits [Y  41.7195 dB   U  44.3713 dB    V  45.2450 dB] [ET   14 ] [L0 ] [L1 ]
Layer  3   POC  143 TId: 0 ( I-SLICE, nQP 34 QP 34 )      59616  bits [Y  44.0645 dB   U   0.0000 dB    V   0.0000 dB] [ET   69 ] [L0 ] [L1 ]
Layer  4   POC  143 TId: 0 ( I-SLICE, nQP 25 QP 25 )     932312  bits [Y  41.7154 dB   U  44.3855 dB    V  45.2770 dB] [ET   14 ] [L0 ] [L1 ]
Layer  5   POC  143 TId: 0 ( I-SLICE, nQP 34 QP 34 )      61920  bits [Y  44.1105 dB   U   0.0000 dB    V   0.0000 dB] [ET   69 ] [L0 ] [L1 ]
Layer  0   POC  144 TId: 0 ( I-SLICE, nQP 25 QP 25 )     933808  bits [Y  41.7343 dB   U  44.3498 dB    V  45.2481 dB] [ET   14 ] [L0 ] [L1 ]
Layer  1   POC  144 TId: 0 ( I-SLICE, nQP 34 QP 34 )      62288  bits [Y  44.6998 dB   U   0.0000 dB    V   0.0000 dB] [ET   90 ] [L0 ] [L1 ]
Layer  2   POC  144 TId: 0 ( I-SLICE, nQP 25 QP 25 )     932096  bits [Y  41.7303 dB   U  44.3655 dB    V  45.2215 dB] [ET   14 ] [L0 ] [L1 ]
Layer  3   POC  144 TId: 0 ( I-SLICE, nQP 34 QP 34 )      61496  bits [Y  43.9290 dB   U   0.0000 dB    V   0.0000 dB] [ET   69 ] [L0 ] [L1 ]
Layer  4   POC  144 TId: 0 ( I-SLICE, nQP 25 QP 25 )     931432  bits [Y  41.7151 dB   U  44.4146 dB    V  45.2913 dB] [ET   14 ] [L0 ] [L1 ]
Layer  5   POC  144 TId: 0 ( I-SLICE, nQP 34 QP 34 )      62440  bits [Y  43.7962 dB   U   0.0000 dB    V   0.0000 dB] [ET   69 ] [L0 ] [L1 ]
Layer  0   POC  145 TId: 0 ( I-SLICE, nQP 25 QP 25 )     927656  bits [Y  41.7556 dB   U  44.3820 dB    V  45.2680 dB] [ET   14 ] [L0 ] [L1 ]
Layer  1   POC  145 TId: 0 ( I-SLICE, nQP 34 QP 34 )      64216  bits [Y  44.6256 dB   U   0.0000 dB    V   0.0000 dB] [ET   90 ] [L0 ] [L1 ]
Layer  2   POC  145 TId: 0 ( I-SLICE, nQP 25 QP 25 )     928776  bits [Y  41.7512 dB   U  44.4091 dB    V  45.3110 dB] [ET   14 ] [L0 ] [L1 ]
Layer  3   POC  145 TId: 0 ( I-SLICE, nQP 34 QP 34 )      62608  bits [Y  44.0115 dB   U   0.0000 dB    V   0.0000 dB] [ET   68 ] [L0 ] [L1 ]
Layer  4   POC  145 TId: 0 ( I-SLICE, nQP 25 QP 25 )     929248  bits [Y  41.7290 dB   U  44.3781 dB    V  45.2933 dB] [ET   14 ] [L0 ] [L1 ]
Layer  5   POC  145 TId: 0 ( I-SLICE, nQP 34 QP 34 )      65928  bits [Y  44.0145 dB   U   0.0000 dB    V   0.0000 dB] [ET   70 ] [L0 ] [L1 ]
Layer  0   POC  146 TId: 0 ( I-SLICE, nQP 25 QP 25 )     928136  bits [Y  41.7692 dB   U  44.4027 dB    V  45.2841 dB] [ET   14 ] [L0 ] [L1 ]
Layer  1   POC  146 TId: 0 ( I-SLICE, nQP 34 QP 34 )      62480  bits [Y  44.4089 dB   U   0.0000 dB    V   0.0000 dB] [ET   89 ] [L0 ] [L1 ]
Layer  2   POC  146 TId: 0 ( I-SLICE, nQP 25 QP 25 )     923304  bits [Y  41.7896 dB   U  44.3323 dB    V  45.2464 dB] [ET   14 ] [L0 ] [L1 ]
Layer  3   POC  146 TId: 0 ( I-SLICE, nQP 34 QP 34 )      61344  bits [Y  43.5797 dB   U   0.0000 dB    V   0.0000 dB] [ET   69 ] [L0 ] [L1 ]
Layer  4   POC  146 TId: 0 ( I-SLICE, nQP 25 QP 25 )     927120  bits [Y  41.7560 dB   U  44.3389 dB    V  45.2676 dB] [ET   14 ] [L0 ] [L1 ]
Layer  5   POC  146 TId: 0 ( I-SLICE, nQP 34 QP 34 )      65248  bits [Y  43.6238 dB   U   0.0000 dB    V   0.0000 dB] [ET   69 ] [L0 ] [L1 ]
Layer  0   POC  147 TId: 0 ( I-SLICE, nQP 25 QP 25 )     918384  bits [Y  41.7766 dB   U  44.4255 dB    V  45.2442 dB] [ET   14 ] [L0 ] [L1 ]
Layer  1   POC  147 TId: 0 ( I-SLICE, nQP 34 QP 34 )      64480  bits [Y  44.3403 dB   U   0.0000 dB    V   0.0000 dB] [ET   90 ] [L0 ] [L1 ]
Layer  2   POC  147 TId: 0 ( I-SLICE, nQP 25 QP 25 )     926144  bits [Y  41.8009 dB   U  44.3506 dB    V  45.2566 dB] [ET   14 ] [L0 ] [L1 ]
Layer  3   POC  147 TId: 0 ( I-SLICE, nQP 34 QP 34 )      64000  bits [Y  43.5365 dB   U   0.0000 dB    V   0.0000 dB] [ET   69 ] [L0 ] [L1 ]
Layer  4   POC  147 TId: 0 ( I-SLICE, nQP 25 QP 25 )     923752  bits [Y  41.7702 dB   U  44.4198 dB    V  45.3113 dB] [ET   14 ] [L0 ] [L1 ]
Layer  5   POC  147 TId: 0 ( I-SLICE, nQP 34 QP 34 )      64664  bits [Y  43.6823 dB   U   0.0000 dB    V   0.0000 dB] [ET   70 ] [L0 ] [L1 ]
```

Figure 1.5: An example showing a piece of the command line outputs during the encoding process for Shark sequence.

of Shark sequence. The numbers in red blocks stands for the encoding time of certain views, while the corresponding layer Id and Picture Order Count (POD) are in the green blocks. A repetitive pattern of the encoding time for each view can be observed every six numbers vertically. A simple calculation using six numbers within the top-most red block, (90+69+69)/(90+69+69+14∗3) ≈ 0.84, shows that approximately 84% of the total encoding time is busy with encoding the depth maps. Similarly, it is reported in [12] that the coding for depth map consumes near 86% of total 3D-HEVC encoding time. A trial of time profiling for 3D-HEVC encoder is performed using Instruments which is available on macOS. After encoding the Newspaper sequence for more than one hour, Figure 1.6 clearly shows 97.8% time is used to compress the CUs recursively. The first recursive xCompressCU function (denoted as XC1 thereafter) is for CUs of size 64x64, the second recursive xCompressCU



Time Profiler 〉 Profile 〉 Root

Weight⌄		Symbol Name
66.42 min	100.0%	▼TAppEncoder (11026)
66.42 min	100.0%	▼Main Thread 0x2a7867
66.42 min	99.9%	▼main TAppEncoder
66.42 min	99.9%	▼TAppEncTop::encode() TAppEncoder
66.38 min	99.9%	▼TEncTop::encode(bool, TComPicYuv*, TComPicYuv*, InputColourSpaceConversion, TComList<TComPicYuv*>&, std::__1::list<Acces
66.38 min	99.9%	▼TEncGOP::compressPicInGOP(int, int, TComList<TComPic*>&, TComList<TComPicYuv*>&, std::__1::list<AccessUnit, std::__1::all
66.29 min	99.8%	▼TEncSlice::compressSlice(TComPic*, bool, bool) TAppEncoder
65.04 min	97.9%	▼TEncCu::compressCtu(TComDataCU*) TAppEncoder
65.02 min	97.8%	▼TEncCu::xCompressCU(TComDataCU*&, TComDataCU*&, unsigned int, PartSize) TAppEncoder
58.61 min	88.2%	▼TEncCu::xCompressCU(TComDataCU*&, TComDataCU*&, unsigned int, PartSize) TAppEncoder
42.71 min	64.3%	▼TEncCu::xCompressCU(TComDataCU*&, TComDataCU*&, unsigned int, PartSize) TAppEncoder
28.46 min	42.8%	▶TEncCu::xCompressCU(TComDataCU*&, TComDataCU*&, unsigned int, PartSize) TAppEncoder
13.53 min	20.3%	▶TEncCu::xCheckRDCostIntra(TComDataCU*&, TComDataCU*&, PartSize, bool) TAppEncoder
25.24 s	0.6%	▶TEncCu::xCheckRDCostDIS(TComDataCU*&, TComDataCU*&, PartSize) TAppEncoder
7.76 s	0.1%	▶TComRdCost::setRenModelData(TComDataCU const*, unsigned int, short const*, int, int, int) TAppEncoder
2.69 s	0.0%	▶TComDataCU::initSubCU(TComDataCU*, unsigned int, unsigned int, int) TAppEncoder
1.96 s	0.0%	▶TComDataCU::initEstData(unsigned int, int, bool) TAppEncoder
1.94 s	0.0%	▶TComDataCU::copyPartFrom(TComDataCU*, unsigned int, unsigned int) TAppEncoder

Figure 1.6: A screen capture of the time profiling information for Newspaper sequence.

Table 1.3: The summary of the time percentages occupied by DMM1 searching in the process for compressing CUs

size of CU	Recursive xCompressCU Function	Time percentage of DMM1 searching process
32 by 32	XC2	30.0%
16 by 16	XC3	25.6%
8 by 8	XC4	18.8%

Table 1.4: The summary of the time percentages occupied by VSO in DMM1 searching

size of CU	process	Time percentage of VSO in DMM1 searching
32 by 32	VSO in DMM1 searching from XC2	80.1%
16 by 16	VSO in DMM1 searching from XC3	83.7%
8 by 8	VSO in DMM1 searching from XC4	78.8%

(denoted as XC2 thereafter) is targeting CUs of size 32x32, the third one (denoted as XC3 thereafter) is dedicated to CUs of size 16x16, and the last one (denoted as XC4 thereafter) is bound to CUs of size 8x8. It is observed that the most time consuming part during the process of compressing the depth CUs is DMM1 searching. The DMM1 searching time percentages are summarised in Table 1.3 wherein the summary for XC1 is omitted since DMM1 is not applicable to CUs of size 64 by 64 in HTM-16.2. [t] The major reason leading to the time consuming property of DMM1 searching is the View Synthesis Optimization (VSO) Method for improving quality of synthesized views [13], wherein the Synthesized View Distortion CChange (SVDC) is computed. The time percentages of the VSO processes in DMM1 searching are summarised in Table 1.4. In HTM-16.2, many wedgelet candidates are evaluated using the VSO which has a high computational complexity. Evaluating less wedgelet candidates will help to relieve the burden of heavy computation required by VSO, thereby certain time reduction can be achieved.

Motivation for Using Deep Learning: Deep learning such as Multi-Layer Perceptron (MLP) is a subfield of representation learning, which is in turn a major subset of machine learning [14]. Machine learning such as the support vector learning [15] is applied to many methods in the domain of Artificial Intelligence (AI). Deep learning based on back propagation training has been found hard to proceed in the late 1980s [16], however, starting from 2012, it kicks off the glorious comeback. The deep Convolutional Neural Network (CNN) has won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) from 2012 to 2015 with the CNN architecture of the winner going deeper and deeper year by year. The great achievements attract attentions from people all over the world and make deep learning the hottest topic in our daily lives. Inspired by the fact that supervised deep learning can learn multiple layers of abstract representations in the visual recognition tasks, it should be applicable to recognize the angular modes of the intra-picture prediction in the 3D-HEVC. The final DMM1 candidate selected in depth map coding is essentially determined by the angle pattern of the depth blocks. If we can make use of deep learning to predict the most probable angles of the target pixel block, a vast amount of angular modes and DMM1 wedgelet candidates can be naturally skipped by which the time saving can be achieved without decreasing the coding performance.

Motivated by the discussions above, we adopt deep learning approach with deep convolutional neural network to accelerate the depth map coding in 3D-HEVC.

1.2 Contribution and Dissertation Outline

We accelerate the depth map coding in 3D-HEVC leveraging the power of deep learning. The contributions of the dissertation are:

- A deep convolutional neural network with 32 layers comprising ResNet units [17] has been designed and trained for recognizing the angular directions of the blocks from intra-picture prediction in 3D-HEVC encoder. The learned models have high top-k precisions which work well on the tasks of recognizing intra angular patterns in 3D-HEVC.
- A way of integrating the learned model into the HTM-16.2 encoder has been suggested. By making use of Bazel [18] to compile the encoder binary, the data level parallelism (instead of concurrency) functionality in CPU as well as the parallel architecture in GPU are fully utilized for efficient computations of matrix operations.
- An algorithm, illustrated in Figure 1.7 on on page 9, for fast depth map coding based on the predictions from learned deep models has been proposed and implemented. The simulation results show that the proposed algorithm is capable of reducing 64.6% time in wedgelet searching during 3D-HEVC encoding process while the BD performance only has a trivial decrease.

The first two contributions lay the foundation for the third one, which is the main objective of this work: to accelerate the depth map encoding process in 3D-HEVC.

Chapter 2 supplies the background of video coding history, video coding standards, and deep learning using artificial neural network. Prior arts in video coding and deep learning are surveyed in this chapter.

Chapter 3 describes the methodology which has been implemented to collect the data to be used for deep learning. The pro-processing steps for the data are provided in details along with the reasons behind the scene. We also visualize lots of collected data to help with the understanding of their properties.

Chapter 4 presents the designed deep convolutional neural network which has been adopted in the deep learning process. Discussions on choosing proper hyper-parameters for the devised neural network are given. The stopping criteria are presented with the training results.

Chapter 5 provides evaluation results for the learned models. Block resizing with different approaches are compared using the accuracy of prediction.

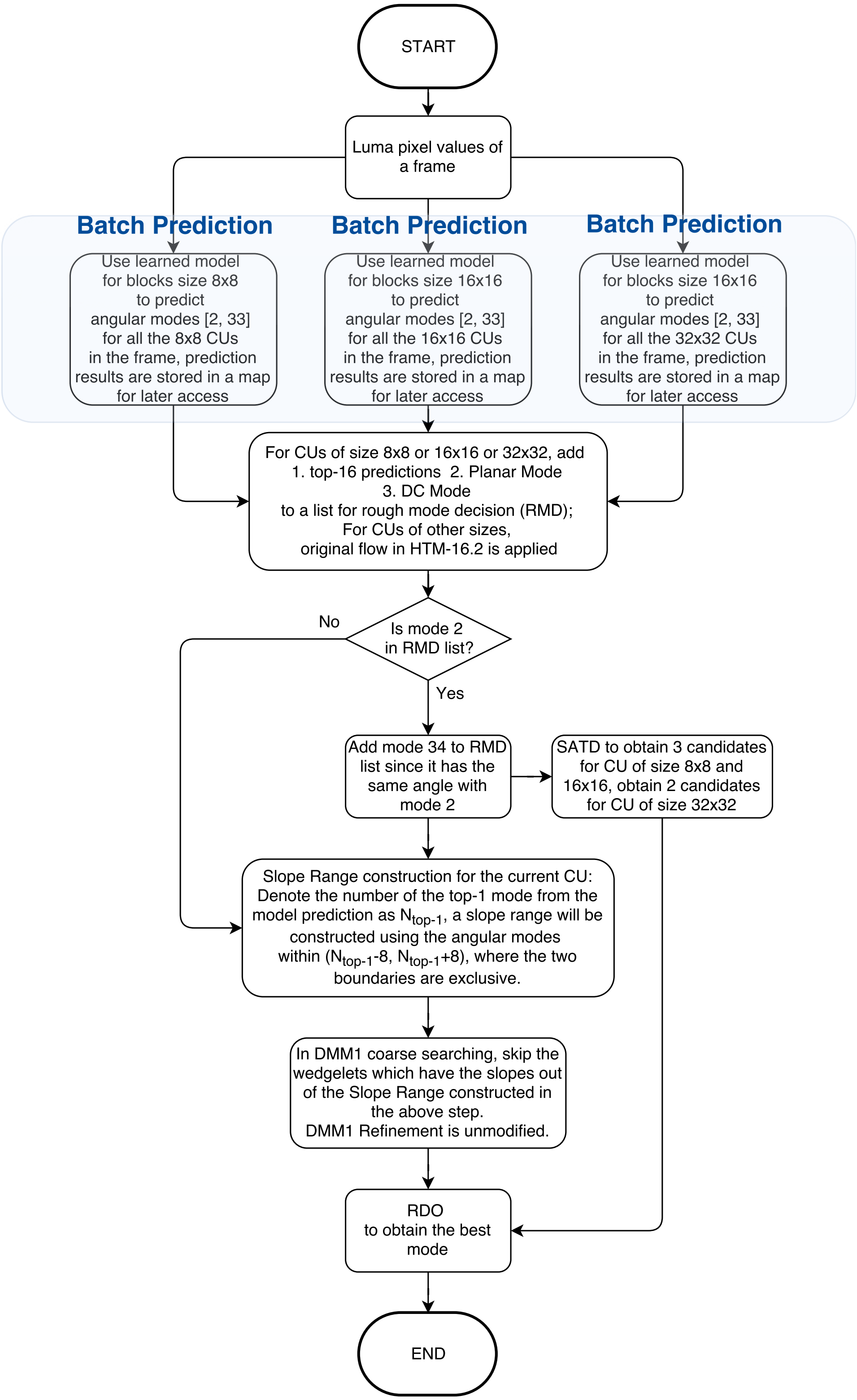

START
Luma pixel values of a frame
Batch Prediction
Batch Prediction
Batch Prediction
Use learned model for blocks size 8x8 to predict angular modes [2, 33] for all the 8x8 CUs in the frame, prediction results are stored in a map for later access
Use learned model for blocks size 16x16 to predict angular modes [2, 33] for all the 16x16 CUs in the frame, prediction results are stored in a map for later access
Use learned model for blocks size 16x16 to predict angular modes [2, 33] for all the 32x32 CUs in the frame, prediction results are stored in a map for later access
For CUs of size 8x8 or 16x16 or 32x32, add 1. top-16 predictions 2. Planar Mode 3. DC Mode to a list for rough mode decision (RMD); For CUs of other sizes, original flow in HTM-16.2 is applied
No
Is mode 2 in RMD list?
Yes
Add mode 34 to RMD list since it has the same angle with mode 2
SATD to obtain 3 candidates for CU of size 8x8 and 16x16, obtain 2 candidates for CU of size 32x32
Slope Range construction for the current CU: Denote the number of the top-1 mode from the model prediction as N_{top-1}, a slope range will be constructed using the angular modes within (N_{top-1}-8, N_{top-1}+8), where the two boundaries are exclusive.
In DMM1 coarse searching, skip the wedgelets which have the slopes out of the Slope Range constructed in the above step. DMM1 Refinement is unmodified.
RDO to obtain the best mode
END

Figure 1.7: Flowchart for proposed fast depth coding algorithm.

Chapter 6 shows the methods been used for integrating the learned model into the 3D-HEVC encoder. Advantages and drawbacks of the integration are discussed. Simulation results comparing with the original HTM-16.2 are given in this chapter.

Chapter 7 concludes the thesis and discusses the future work for fast depth coding using deep learning.

Chapter 2

Background

With the rising popularity of the high definition videos, the new standard termed High Efficiency Video Coding (HEVC) for compressing videos in a more efficient way comparing with previous standards, such as H.264/AVC, has emerged under the efforts from the Joint Collaborative Team on Video Coding (JCT-VC). In the meanwhile, five extensions of the HEVC standard, comprising Format Range Extension (RExt), Scalability Extension (SHVC), Multi-view Extension (MV-HEVC), 3D Extension (3D-HEVC), Screen Content Coding Extension (SCC), have been finalized from 2014 to 2016 to fulfill extra requirements in various video coding scenarios.

2.1 Video Coding

Chapter 3

Prepare the Data for Deep Learning

With the rising popularity of the high definition videos, the new standard termed High Efficiency Video Coding (HEVC) for compressing videos in a more efficient way comparing with previous standards, such as H.264/AVC, has emerged under the efforts from the Joint Collaborative Team on Video Coding (JCT-VC). In the meanwhile, five extensions of the HEVC standard, comprising Format Range Extension (RExt), Scalability Extension (SHVC), Multi-view Extension (MV-HEVC), 3D Extension (3D-HEVC), Screen Content Coding Extension (SCC), have been finalized from 2014 to 2016 to support fulfill extra requirements in various scenarios.

3.1 Viccccdeo Coding

Chapter 4

Train the Deep Model for Prediction

With the rising popularity of the high definition videos, the new standard termed High Efficiency Video Coding (HEVC) for compressing videos in a more efficient way comparing with previous standards, such as H.264/AVC, has emerged under the efforts from the Joint Collaborative Team on Video Coding (JCT-VC). In the meanwhile, five extensions of the HEVC standard, comprising Format Range Extension (RExt), Scalability Extension (SHVC), Multi-view Extension (MV-HEVC), 3D Extension (3D-HEVC), Screen Content Coding Extension (SCC), have been finalized from 2014 to 2016 to support fulfill extra requirements in various scenarios.

4.1 Video Cnbvvnbvnboding

Chapter 5

Evaluate the Learned Deep Model

With the rising popularity of the high definition videos, the new standard termed High Efficiency Video Coding (HEVC) for compressing videos in a more efficient way comparing with previous standards, such as H.264/AVC, has emerged under the efforts from the Joint Collaborative Team on Video Coding (JCT-VC). In the meanwhile, five extensions of the HEVC standard, comprising Format Range Extension (RExt), Scalability Extension (SHVC), Multi-view Extension (MV-HEVC), 3D Extension (3D-HEVC), Screen Content Coding Extension (SCC), have been finalized from 2014 to 2016 to support fulfill extra requirements in various scenarios.

5.1 Video Codhgvingg

Chapter 6

Employ the Learned Deep Model

With the rising popularity of the high definition videos, the new standard termed High Efficiency Video Coding (HEVC) for compressing videos in a more efficient way comparing with previous standards, such as H.264/AVC, has emerged under the efforts from the Joint Collaborative Team on Video Coding (JCT-VC). In the meanwhile, five extensions of the HEVC standard, comprising Format Range Extension (RExt), Scalability Extension (SHVC), Multi-view Extension (MV-HEVC), 3D Extension (3D-HEVC), Screen Content Coding Extension (SCC), have been finalized from 2014 to 2016 to support fulfill extra requirements in various scenarios.

6.1 Video Cobnbvnbvnbvnding

Chapter 7

Conclusion

With the rising popularity of the high definition videos, the new standard termed High Efficiency Video Coding (HEVC) for compressing videos in a more efficient way comparing with previous standards, such as H.264/AVC, has emerged under the efforts from the Joint Collaborative Team on Video Coding (JCT-VC). In the meanwhile, five extensions of the HEVC standard, comprising Format Range Extension (RExt), Scalability Extension (SHVC), Multi-view Extension (MV-HEVC), 3D Extension (3D-HEVC), Screen Content Coding Extension (SCC), have been finalized from 2014 to 2016 to support fulfill extra requirements in various scenarios.

7.1 Video Codvbnvnbvcnbvnbing

Bibliography

[1] *Video*, Web Page, 2017. [Online]. Available: http://hidefnj.com/video.html.

[2] C. E. Shannon, *The mathematical theory of communication*. Urbana: Urbana : University of Illinois Press, 1949.

[3] "Itu-t recommendation database," 2017. [Online]. Available: http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=12905&lang=en.

[4] "Jct-vc - joint collaborative team on video coding," 2017. [Online]. Available: http://www.itu.int/en/ITU-T/studygroups/2013-2016/16/Pages/video/jctvc.aspx.

[5] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed.., ser. H.264 Advanced Video Compression Standard 2e. Hoboken: Hoboken : Wiley, 2010.

[6] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, 2011, ISSN: 0018-9219. DOI: 10.1109/JPROC.2010.2098830.

[7] J. Konrad and M. Halle, "3-d displays and signal processing," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 97–111, 2007, ISSN: 1053-5888. DOI: 10.1109/msp.2007.905706.

[8] I. Sexton and P. Surman, "Stereoscopic and autostereoscopic display systems," *Signal Processing Magazine, IEEE*, vol. 16, no. 3, pp. 85–99, 1999, ISSN: 1053-5888. DOI: 10.1109/79.768575.

[9] Mu, amp, X, K. Ller, P. Merkle, and T. Wiegand, "3-d video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, 2011, ISSN: 0018-9219. DOI: 10.1109/JPROC.2010.2091090.

[10] "Jct-3v - joint collaborative team on 3d video coding extension development," 2017. [Online]. Available: http://www.itu.int/en/ITU-T/studygroups/2013-2016/16/Pages/video/jct3v.aspx.

[11] G. Tech, K. Ying Chen, J.-R. Muller, A. Ohm, A. Vetro, and A. Ye-Kui Wang, "Overview of the multiview and 3d extensions of high efficiency video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 26, no. 1, pp. 35–49, 2016, ISSN: 1051-8215. DOI: 10.1109/TCSVT.2015.2477935.

[12] H.-B. Zhang, C.-H. Fu, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, “Probability-based depth intra mode skipping strategy and novel vso metric for dmm decision in 3d-hevc,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2016, ISSN: 1051-8215. DOI: `10.1109/TCSVT.2016.2612693`.

[13] H. Dou, Y.-L. Chan, K.-B. Jia, and W.-C. Siu, “Segment-based view synthesis optimization scheme in 3d-hevc,” *Journal of Visual Communication and Image Representation*, vol. 42, pp. 104–111, 2017, ISSN: 1047-3203. DOI: `10.1016/j.jvcir.2016.11.012`.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, ser. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2016, xxii, 775 pages, ISBN: 9780262035613 0262035618.

[15] B. Schlkopf, C. J. C. Burges, and A. J. Smola, *Advances in kernel methods : support vector learning*. Cambridge, Mass. ; London: Cambridge, Mass. ; London : MIT Press, 1999.

[16] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks : the official journal of the International Neural Network Society*, vol. 61, p. 85, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.

[18] *Bazel*, Web Page, 2017. [Online]. Available: `https://www.bazel.build/`.