DEEP LEARNING-BASED ARTIFACTS REMOVAL IN VIDEO COMPRESSION

A Dissertation
in
Electrical and Electronics Engineering
and
Computer Science

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

DOCTOR OF PHILOSOPHY

by
WEI JIA

M.S., Beijing University of Posts and Telecommunications, Beijing, China, 2012
B.S., Beijing University of Posts and Telecommunications, Beijing, China, 2009

Kansas City, Missouri
2021

DEEP LEARNING-BASED ARTIFACTS REMOVAL IN VIDEO COMPRESSION

Wei Jia, Candidate for the Doctor of Philosophy Degree

University of Missouri–Kansas City, 2021

ABSTRACT

The block-based coding structure in the hybrid video coding framework inevitably introduces compression artifacts such as blocking, ringing, etc. To compensate for those artifacts, extensive filtering techniques were proposed in the loop of video codecs, which are capable of boosting the subjective and objective qualities of reconstructed videos. Recently, neural network-based filters were presented with the power of deep learning from a large magnitude of data. Though the coding efficiency has been improved from traditional methods in High-Efficiency Video Coding (HEVC), the rich features and information generated by the compression pipeline has not been fully utilized in the design of neural networks. Therefore, we propose a learning-based method to further improve the coding efficiency to its full extent.

In addition, the point cloud is an essential format for three-dimensional (3-D) objects capture and communication for Augmented Reality (AR) and Virtual Reality (VR) applications. In the current state of the art video-based point cloud compression (V-PCC), a dynamic point cloud is projected onto geometry and attribute videos patch by patch, each represented by its texture, depth, and occupancy map for reconstruction. To deal with occlusion, each patch is projected onto near and far depth fields in the geometry video. Once there are artifacts on the compressed two-dimensional (2-D) geometry video, they would be propagated to the 3-D point cloud frames. In addition, in the lossy compression, there always exists a tradeoff between the rate of bitstream and distortion (RD). Although some methods were proposed to attenuate these artifacts and improve the coding efficiency, the non-linear representation ability of Convolutional Neural Network (CNN) has not been fully considered. Therefore, we propose a learning-based approach to remove the geometry artifacts and improve the compressing efficiency.

Besides, we propose using a CNN to improve the accuracy of the occupancy map video in V-PCC. To the best of our knowledge, these are the first learning-based solutions of the geometry artifacts removal in HEVC and occupancy map enhancement in V-PCC. The extensive experimental results show that the proposed approaches achieve significant gains in HEVC and V-PCC compared to the state-of-the-art schemes.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled " Deep Learning-based Artifacts Removal in Video Compression ," presented by Wei Jia, candidate for the Doctor of Philosophy degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

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

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

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

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

Baek-Young Choi, Ph.D.
Department of Computer Science & Electrical Engineering

CONTENTS

ABSTRACT iii

ILLUSTRATIONS viii

TABLES xviii

ACKNOWLEDGEMENTS xxi

Chapter

1 Residual-Guided In-Loop Filter Using Convolution Neural Network 1

1.1 Background 2

1.2 In-loop filters in video compression 5

1.3 Residual-assisted in-loop RRNet 12

1.4 Experimental results 21

1.5 Summary 32

2 DEEP LEARNING GEOMETRY COMPRESSION ARTIFACTS REMOVAL FOR VIDEO-BASED POINT CLOUD COMPRESSION 34

2.1 Background 35

2.2 Artifacts removal in V-PCC 39

2.3 Geometry in V-PCC 45

2.4 Two-step geometry artifacts removal 49

2.5 Experimental results 59

2.6 Summary 65

3 Convolutional Neural Network-Based Occupancy Map Accuracy Improvement for Video-based Point Cloud Compression 75

3.1 Background 76

3.2 Occupancy map improvement 79

3.3 Occupancy map video in V-PCC 83

3.4 Occupancy map refinement OGCNN with two inputs 86

3.5 Experimental results 94

3.6 Summary 102

REFERENCE LIST 112

VITA 129

ILLUSTRATIONS

Figure Page

1 Typical example of the Kimono residual under $QP37$ with intra mode. The color has been adjusted for clear viewing. The inverse transformed residual signal provides the comprehensive partition information of the transforming units. It is obvious to see the 32×32, 16×16, 8×8, and 4×4 partition blocks of TU in the residual. For instance, the shapes of the woman's body and tree trunks are more easily discernable. Meanwhile, the residual contains a large amount of dense, detailed textures. For example, we can see many needle leaves on the trees. This information can help to augment the considerable variation in some areas of the reconstruction. 7

2 RRNet with sub-networks of both the Residual Network and the Reconstruction Network. Residuals are fed into the Residual Network to provide the TU partition information and the detailed textures information. The Residual Network relies on residual blocks to learn features effectively with residual learning. We feed the reconstruction into the Reconstruction Network. The Reconstruction Network executes the downsampling and upsampling strategy to patch up the local and global information. This enhances reconstruction quality and aids with the residual learning approach. 8

3 Residual feature maps of Cactus and BQSquare derived from the Residual Network of RRNet under $QP37$. The residual features of Cactus with abundant context including pokers, calender and metal circle demonstrates its prominent contribution for enhancing the quality of the video frame. The residual features of BQSquare which are a flat example show a great amount of details involving chairs and tables. 10

4 The location of RRNet embedded in HEVC. We insert the RRNet into HEVC as an in-loop method. The RRNet would input residual from extracting module and reconstruction into the Residual Network and the Reconstruction Network, respectively. The RRNet is executed instead of DF and SAO filters. 11

5 Comparison of RD curves in HEVC with DF and SAO, VRCNN and proposed RRNet on luminance. The compared RD curves of BasketballDrill(a), FourPeople(b), Johnny(c) and Traffic(d) are shown. It is obvious that our proposed RRNet outperforms HEVC with DF and SAO and VRCNN for all theses sequences under all tested QPs including $22, 27, 32$ and 37. 22

6 The comparison of QP 32 model with respective QP models. The ΔPSNR on $\Delta QP = 0$ means the QP32 model compared to itself on PSNR is zero. Except QP34 setting, the PSNR of individual QP model is better than the one for the QP32 settings on the other QP parameters. The ΔPSNR increases significantly with the absolute value of ΔQP on the each side of $\Delta QP = 0$. 29

7 Visual comparisons between the ground truths, HEVC anchor, VRCNN, and proposed RRNet approach on the luminance of $QP37$ in Johnny and BasketballDrill sequences, respectively. The groups of figures (a), (b), (c), and (d) are the original video, the video generated using HEVC, the video generated using VRCNN, the video generated using RRNet, respectively. (Zoom in for better visual effects.) 30

8 The projection process [1] from 3-D to 2-D in V-PCC. The segmented 3-D patches are projected to six planes of its bounding box. 44

9 The correlations between 3-D point clouds sequence and 2-D geometry video. A 2-D patch with picture order count (POC) $2N$ of near layer and one with $2N+1$ of far layer are both derived from the same 3-D point cloud patch with POC N. 45

10 Near (a) and far (b) layer frames in 2-D geometry video. (C) is the difference δ between the near layer frame with POC $2N$ and its corresponding POC $2N+1$ far layer frame. Since generally, the difference δ is a minimal value, it is difficult to directly see the difference between the near and far layer frames unless making subtraction with them as (c). 67

11 The process of 3-D point-cloud reconstruction [1] in V-PCC. The atlas (a) is first reconstructed. Only padded patch rough sketches are built. After the decoded occupancy map is merged into the reconstruction (b), the points occupancy status becomes clearly shown because the occupancy map has removed the padded area. When the geometry information is added into the reconstruction (c), all 3-D point clouds are almost created except the color attributes. Finally, the attributes are drawn on the reconstruction (d). This visualizing reconstruction process demonstrates how the 2-D geometry enormously impacts the subjective quality of 3-D point clouds reconstruction. 68

12 Impact of the 2-D geometry artifacts to the 3-D point cloud artifacts. The sub-figures of the first row (a), (b), and (c) are the 2-D geometry, attributes, and 3-D point cloud of ground truths, respectively. With the same arrangement, the sub-figures of the second row are from anchor reconstructions. Because it is difficult to directly recognize the difference of the geometry value between the anchor and ground truth with our eyes, we visualize their difference in (d). White and black pixels represent the different ones, while gray pixels are the same ones. We can clearly see the geometry value distortion of the gun in the enlarged area of (d). From the enlarged areas, we can clearly see that the V-PCC propagates the 2-D geometry distortion to attributes reconstruction. Then, the V-PCC brings the attributes artifacts into the 3-D point-cloud reconstruction. 69

13 The proposed two-step method is embedded in the V-PCC encoding scheme. The target of the first step is to denoise the coarse artifacts of the reconstructed near layer r_0 and far layer r_1 first. The augmented near layer reconstruction $r_0^{'}$ as PMC iteratively participates into the far layer g_{p1} prediction to generate the far layer reconstruction r_1. Afterward, r_1 is fed into the far layer step one CNN to produce a better reconstruction $r_1^{'}$. The objective of the second step is to utilize the interactive information $I_s(r_0^{'}, r_1^{'})$ between near layer $r_0^{'}$ and far layer $r_1^{'}$ in full extent for further removing artifacts. We input the outputs of step one including $r_0^{'}$ and $r_1^{'}$ into XInteractNet to achieve enhanced corresponding near layer reconstruction $r_0^{''}$ and far layer reconstruction $r_1^{''}$. They are finally used for reconstructing the 3-D point cloud. 70

14 XInteractNet architecture. In the X down block, to share the interactive information $I_d(x_0(i), x_1(i))$ at level i before the max-pooling operation, the previous convolutional near layer's feature $x_0(i-1)$ and far layer's feature $x_1(i-1)$ are merged into $x(i-1)$ first. The No. $i+1$ convolutional near and far layers take the pooled $x(i)$ feature as input to tackle the next convolution computation. Similarly, the X up block shares the interactive information $I_u(y_0(j), y_1(j))$ at level j before the transposed-convolution operation. The previous convolutional near layer's feature $y_0(j-1)$ and far layer's feature $y_1(j-1)$ are shared and merged into $y(j-1)$. The obtained $y_0(j)$ and $y_1(j)$ features are concatenated again and fed into the next NO. $j+1$ convolutional near and far layers for computation. 71

15 Number of points N_R in reconstructed 3-D point clouds within first 32 frames of four CTC sequences. The bitrate of Y axis gradually increases from level $r1$ to level $r4$. The number of points N_R of our proposed two-step method clearly restore more points than V-PCC geometry pading and smoothing methods. These statistics fully demonstrate that the proposed two-step method effectively restores the 3-D point clouds. 72

16 Comparison of Geometry point-to-point and point-to-plane Rate-Distortion (RD) curves on Loot sequence in all the intra cases. As shown, the point-to-point and point-to-plane PSNRs of first three rate points in the proposed two-step method is higher than the ones of all other V-PCC geometry padding and smoothing methods. This proves that the proposed algorithm performs obviously better than SOTA methods on improving coding efficiency. 73

17 Visual comparisons of ground truth, point-cloud reconstructions of geometry smoothing, padding and proposed two-step methods. The figures are derived from Loot, RedandBlack, and Longdress. From the three sequences, we can clearly see from the red and green rectangles that there are significant artifacts and noises in the reconstructions of V-PCC Geometry smoothing and padding methods, while the reconstructions of the proposed two-step method show a smoother effect. The visual results obviously demonstrate that compared to the SOTA, the proposed two-step method brings better subjective qualities. 74

18 Occupancy map comparison of the full resolution and quarter resolution with the same occupancy distribution. A grid represents a pixel in the occupancy map video. The bold border square is denoted as a 4×4 block. The red pixels indicate the occupied pixels in the full-resolution occupancy map video, while the blue pixels indicate the occupied pixels in the reconstructed full-resolution occupancy map video. 104

19 Comparison of the occupancy maps, geometry, and attributes of the quarter-resolution and full-resolution videos. The reconstructions of the occupancy maps, geometry, and attributes in the first and second rows are derived from the configurations of the quarter-resolution and full-resolution videos, respectively. We can see from the enlarged areas that, compared to the full resolution, the edges of the body in the quarter-resolution occupancy map, geometry and attribute videos show a more serious zigzag artifact. 105

20 The proposed OGCNN framework includes two subnetworks: the Occupancy Network and the Geometry Network. The Occupancy Network uses the quarter-resolution occupancy map video as input. It derives occupancy segmentation feature maps from the occupancy map. The Geometry Network uses the reconstructed geometry video as input and derives geometry segmentation feature maps from the geometry. The occupancy map and geometry segmentation features are then concatenated together and used as the input of the remaining convolutional layers. 106

21 Occupancy map boundary blocks. A boundary block is an $N \times N$ square that consists of both occupied and unoccupied pixels. The white grids represent the occupied pixels, while the black grids represent the unoccupied pixels. The red, blue, yellow squares indicate the 16×16, 32×32, and 64×64 boundary blocks, respectively. 107

22 Comparison of the numbers of points N_R in the reconstructed 3D point clouds of the V-PCC [2], OR [3], PBF [4], Occupancy Network and OGCNN. The Y axis is the bitrate, which gradually increases from low bitrate $r1$ to high bitrate $r5$. We can see that for all dynamic point clouds, the number of points N_R in our proposed Half OGCNN and Full OGCNN are less than those in the V-PCC anchor, SOTAs, and the Occupancy Network. 108

23 Geometry R-D curve comparison of the V-PCC [2], OR [3], PBF [4], Occupancy Network and OGCNN for the all intra case. We can see that the D2 PSNRs of the proposed OGCNN at all five rate points are higher than those of the V-PCC anchor, SOTAs and Occupancy Network. 109

24 2D occupancy map comparison of the ground truth, V-PCC anchor [2] and proposed Full OGCNN. For Loot, (a), (b), (c) and (d) are the occupancy map reconstructions of the Full OGCNN and the V-PCC anchor, the difference between the two, and the ground truth, respectively. (e) and (f) are the enlarged areas of the gold and blue blocks in (c). For Longdress, the same order is followed. In (c) and (i), the green pixels denote the unoccupied pixels of the V-PCC anchor correctly removed by the Full OGCNN. The red pixels denote the occupied pixels of the V-PCC anchor wrongly removed by the Full OGCNN. 110

25 3D visual comparison of the original point clouds, the point clouds reconstructed by the V-PCC anchor and the proposed Half OGCNN. 111

TABLES

Tables Page

1 The Residual Network Parameters of conv layers 12

2 The Reconstruction Network Parameters of Conv And Transposed Conv Layers 12

3 Training parameters 18

4 BD-rate of the SOTAs and proposed RRNet against HEVC under All Intra case 20

5 The computational complexity of VRCNN and proposed RRNet against HEVC under All Intra case 21

6 BD-rate of VRCNN and proposed RRNet against HEVC under Random Access case 23

7 The dual-input Residual and Reconstruction approach and the dual-input Partition and Reconstruction [5] approach versus Reconstruction only approach on the BD-rate 26

8 The computational complexity of the dual-input Partition and Reconstruction method [5] and the dual-input Residual and Reconstruction approach against HEVC 26

9 Convolutional Parameters of EDSR Residual Blocks [6] 26

10 BD-rate of RRNet against the dual-input Residual and Reconstruction with EDSR Residual Blocks [6] 28

11 2-D and 3-D geometry PSNR comparison between lower and higher bitrates of V-PCC Anchor [7] within soldier first 32 frames 47

12 The convolutional and transposed convolutional layers parameters of the first X Up Block in XInteractNet 54

13 Proposed two-step method and geometry padding method [8] against the geometry smoothing method [9] respectively on BD-rate and time complexity within the first 32 frames of sequences under all intra 55

14 Training parameters 58

15 Proposed two-step method and one-step method of near and far CNNs against the one-step method of mixed geometry respectively on BD-rate and time complexity within the first 32 frames of sequences under the all intra case 58

16 Proposed two-step method against the geometry smoothing [9] on BD-rate and Time complexity under the random access case 60

17 V-PCC anchor [2] performance comparison of the quarter-resolution, half-resolution, and full-resolution occupancy maps within the first 32 frames . 84

18 Occupancy Network Parameters of the Conv and Transposed Conv Layers 89

19 Training parameters 92

20 Performance comparison of the full-resolution OGCNN, half-resolution OGCNN and quarter-resolution V-PCC [2] under the all intra case 93

21 Occupancy accuracy comparison of the V-PCC anchor and OGCNNs on boundary blocks 93

22 Performance comparison of the Half OGCNN and V-PCC anchor [2] under Random Access 98

23 Performance comparison of the Half OGCNN, OR [3], and PBF [4] under the all intra case 99

24 Performance comparison of the Half OGCNN and Occupancy Network under the all intra case 100

ACKNOWLEDGEMENTS

I would first like to thank my supervisor, Professor Zhu Li, whose expertise was invaluable in formulating the research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I am grateful to Dr. Li Li for his professional advice, continuous support, and patience during my Ph.D. study. His immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life.

I would like to acknowledge my colleagues and classmates Zhaobin Zhang, Yangfan Sun, Yue Li, and Han Zhang for their wonderful collaboration. I thank your patient support and for all of the opportunities I was given to further my research.

I would like to thank my wife Xin An foremost. She always encourages my any decision which matters our family direction and treasures our love most which exceeds anything else. I would like to thank my daughter Vivian who is a gift from God. We cherish the happiness of our family union. I would like to thank my parents, sister, and mother-in-law for their real help, wise counsel and sympathetic ear. You are always there for us.

Finally, I would like to express gratitude to my friends for their encouragement and support all through my studies.

CHAPTER 1

RESIDUAL-GUIDED IN-LOOP FILTER USING CONVOLUTION NEURAL NETWORK

The block-based coding structure in the hybrid video coding framework inevitably introduces compression artifacts such as blocking, ringing, etc. To compensate for those artifacts, extensive filtering techniques were proposed in the loop of video codecs, which are capable of boosting the subjective and objective qualities of reconstructed videos. Recently, neural network based filters were presented with the power of deep learning from a large magnitude of data. Though the coding efficiency has been improved from traditional methods in High-Efficiency Video Coding (HEVC), the rich features and information generated by the compression pipeline has not been fully utilized in the design of neural networks. Therefore, in this paper, we propose the Residual-Reconstruction-based Convolutional Neural Network (RRNet) to further improve the coding efficiency to its full extent, where the compression features induced from bitstream in form of prediction residual is fed into the network as an additional input to the reconstructed frame. In essence, the residual signal can provide valuable information about block partitions and can aid reconstruction of edge and texture regions in a picture. Thus, more adaptive parameters can be trained to handle different texture characteristics. The experimental results show that our proposed RRNet approach presents significant BD-rate savings compared to HEVC and the state-of-the-art CNN-based schemes, indicating that residual signal plays a significant role in enhancing video frame reconstruction.

1.1 Background

Advanced Video Coding (H.264/AVC) [10], High-Efficiency Video Coding (H.265/HEVC) [11] are existing popular video coding standards. Versatile Video Coding (VVC) [12] is the emerging next-generation standard under the development of the Moving Pictures Expert Group (MPEG). These video coding standards adopt the so-called hybrid coding frameworks, where the major procedures include prediction, transform, quantization, and entropy coding. In the hybrid coding framework, a video frame is partitioned into non-overlapping coding blocks. These blocks form the basis coding units (CU), prediction units (PU), transform units (TU), etc. A block-based coding scheme is hardware-friendly and easy to implement. It also lends itself to useful coding functionalities such as parallelization.

However, block-wise operation inevitably introduces video quality degradation near the block boundaries, known as block artifacts. Beyond that, coarse quantization is another major factor in causing video quality degradation, especially at the regions with sharp edges known as the ringing artifacts. This ripple phenomenon induces poor visual quality and leads to a bad user-experience [13]. Given this, extensive in-loop filters have been proposed to compensate for artifacts and distortions in video coding. The in-loop filters can be classified into two categories based on whether the deep learning techniques are used.

The first category is traditional signal processing based methods, including Deblocking Filter (DF) [14, 15], Sample Adaptive Offset (SAO) [16–18], Adaptive Loop Filter (ALF) [19], non-local in-loop filter [20] and many others. DF can reduce blocking artifacts at PU and TU boundaries. SAO compensates the pixel-wise residuals by explicitly signaling offsets for pixel groups with similar characteristics. ALF is essentially a Wiener filter where the current pixel is filtered as a linear combination of neighboring pixels. The three filters mentioned above are based on neighbor-pixel statistics. In contrast, the non-local in-loop filter takes advantage of non-local similarities in natural images.

Traditional methods improve the video quality with relatively low complexity. Therefore, they have been successfully applied in video coding standards. Recently, however, the deep learning based in-loop filters have been proposed to achieve further improvements [21–23]. One type of CNN utilizes the principle of the Kalman filter to construct a deep learning filter. Another type of CNN consists of the highway or content-aware block units to achieve flexibility. People have realized that these deep learning based schemes have at least two benefits from traditional methods. One is that non-linear filtering operations are involved in the system. It is critical to capture and compensate for the distortions caused by codecs because these coding distortions are essentially non-linear by themselves. Another benefit is that deep learning can learn features from a large amount of data automatically, which would be more efficient than handcraft features. Though the coding efficiency has been improved from traditional methods in HEVC, the coding information has not been fully utilized in the design of neural networks. In [24], the authors proposed to utilize partition information in the design of neural networks, indicating introducing more coding information can benefit the overall performance.

Motivated by these, we propose a novel in-loop algorithm by introducing the residual signal to the network and devising two sub-networks for residual and reconstruction signals, respectively. They are the Residual Network and the Reconstruction Network. The major contributions of this work are three-fold:

- First, we supply the residual signal as the supplementary information and feed it into the neural network in pair with the reconstructed frame. To the best of our knowledge, this is the first work that utilizes the residual signal to devise an in-loop filter for video coding.
- Second, the network structure is carefully designed for the dual-input CNN to utilize the underlying features in different input channels fully. The residual blocks are used for Residual-Network. A hierarchical autoencoder network with skip connections is used for Reconstruction-Network.
- Third, extensive experiments have been conducted to compare with existing algorithms to demonstrate its effectiveness of the proposed scheme. Throughout analyses are provided to give more insights into the problem based on the experimental results.

Note that a residual introduced deblocking method has been proposed in our previous work [25]. This paper provides more motivation, analysis, experimental results, and comparison of related works on the residual-based loop filter. Additionally, in order to validate the efficiency of our RRNet design, we recurs more three inputs-based methods for comparison. The experimental results show that the customized Residual Network and Reconstruction Network is significantly beneficial for bitrate savings.

We organize the remainder of this paper as follows. In Section 1.2, we describe related works. Section 1.3 introduces the proposed RRNet approach. In Section 1.4, we report and analyze the experimental results. Finally, Section 1.5 summarizes this paper and discusses future works.

1.2 In-loop filters in video compression

In this section, we briefly review the prior works related to loop filters of video coding, including the traditional signal processing based methods and deep learning based methods.

1.2.1 Traditional signal processing based methods

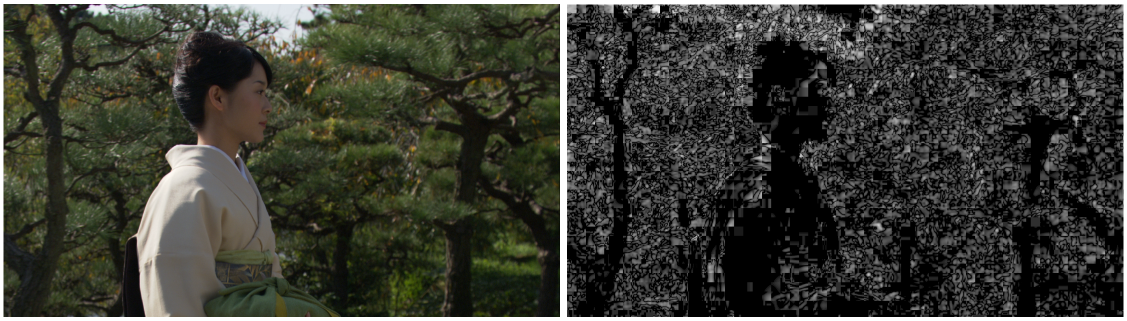
Relying on the signal processing theory, the following in-loop filter methods have been proposed.

1) Deblocking Filter (DF). List *et al.* [26] devised the first version of an adaptive deblocking filter, which was adopted by H.264/AVC standard. It depressed distortions at block boundaries by applying an appropriate filter. Zhang *et al.* [27] proposed a three-step framework considering task-level segmentation and data-level parallelization to efficiently parallelize the deblocking filter. Tsu-Ming *et al.* [14] then proposed a high-throughput deblocking filter. In HEVC, Norkin *et al.* [15] designed a DF with lower complexity and better parallel-processing capability. Li *et al.* [28] provided deblocking with a shape-adaptive low-rank before preserving edges well and an extra before restoring the lost high-frequency components.

2) Sample Adaptive Offset (SAO) [29]. Chien and Karczewicz proposed an adaptive loop filtering technique [30] based on the Laplacian energy and classifications of the reconstructed pixel value. This approach obtains obvious performance improvements but with high complexity. Ken *et al.* [31] designed an extrema correcting filter (EXC) and a boundary correcting filter (BDC). Huang *et al.* [32] developed a picture-based boundary offset (PBO), picture-based border offset (PEO) and picture-based adaptive constraint (PAC). Fu *et al.* [16, 17] devised an algorithm that can adaptively select the optimal pixel-classification method. However, computational complexity is still very high. To address this, Fu and Chen *et al.* [18] proposed a sample adaptive offset (SAO) method, which was finally adopted by HEVC. It provides a better trade-off between performance and complexity.

3) Adaptive Loop Filter (ALF). Tsai *et al.* [19] proposed the ALF method to decrease the mean square error between original frames and decoded frames by Wiener-based adaptive filter. The filter coefficients are trained for different pixel regions at the encoder. The coefficients are then explicitly signaled to the decoder. Besides, ALF activates the filter at different regions by signaling control flags.

4) Non-local Mean Models. The non-local mean methods improve the efficiency of in-loop filters as well. To suppress the quantization noise optimally and improve the quality of the reconstructed frame, Han *et al.* [33] proposed a quadtree-based non-local Kuan's (QNLK) filter. Ma *et al.* [20] proposed the group-based sparse representation with image local and non-local self-similarities. This model lays a



(a) Ground Truth (b) Residual

Figure 1: Typical example of the Kimono residual under $QP37$ with intra mode. The color has been adjusted for clear viewing. The inverse transformed residual signal provides the comprehensive partition information of the transforming units. It is obvious to see the 32×32, 16×16, 8×8, and 4×4 partition blocks of TU in the residual. For instance, the shapes of the woman's body and tree trunks are more easily discernable. Meanwhile, the residual contains a large amount of dense, detailed textures. For example, we can see many needle leaves on the trees. This information can help to augment the considerable variation in some areas of the reconstruction.

solid groundwork for the in-loop filter design. Zhang *et al.* [34] utilized image nonlocal prior knowledge to develop a loop filter by imposing the low-rank constraint on similar image patches for compression noise reduction.

1.2.2 Deep learning based methods

Recently, the deep learning based in-loop filters have been proposed. For images, Dong *et al.* [35] designed a compact and efficient model, known as Artifacts Reduction Convolutional Neural Networks (AR-CNN). This model was effective for reducing various types of coding artifacts. Kang *et al.* [36] propose to learn sparse image representations for modeling the relationship between low-resolution and high-resolution image patches in terms of the learned dictionaries for image patches with and without blocking

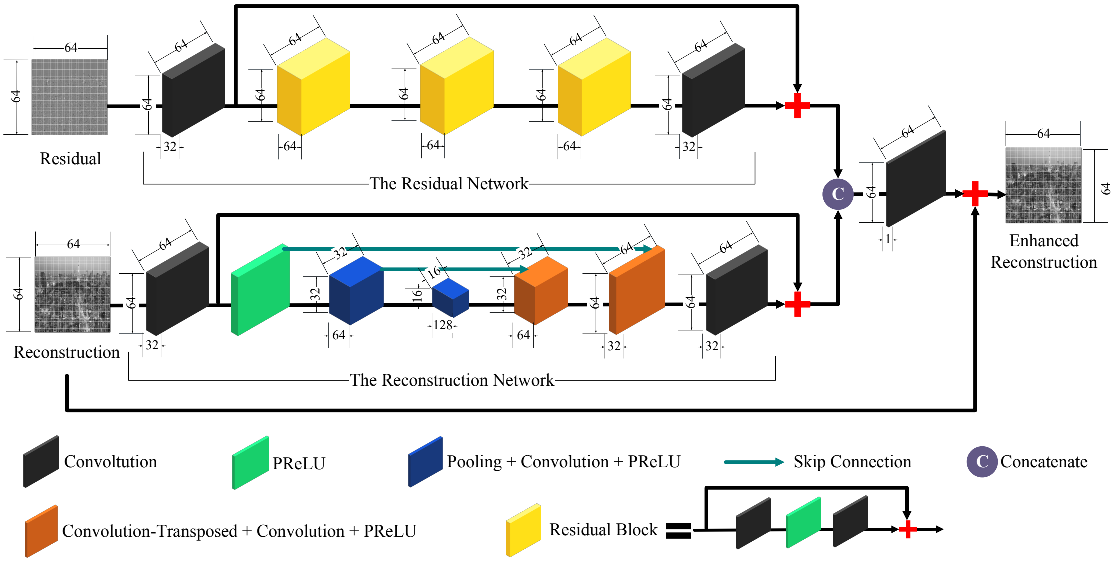
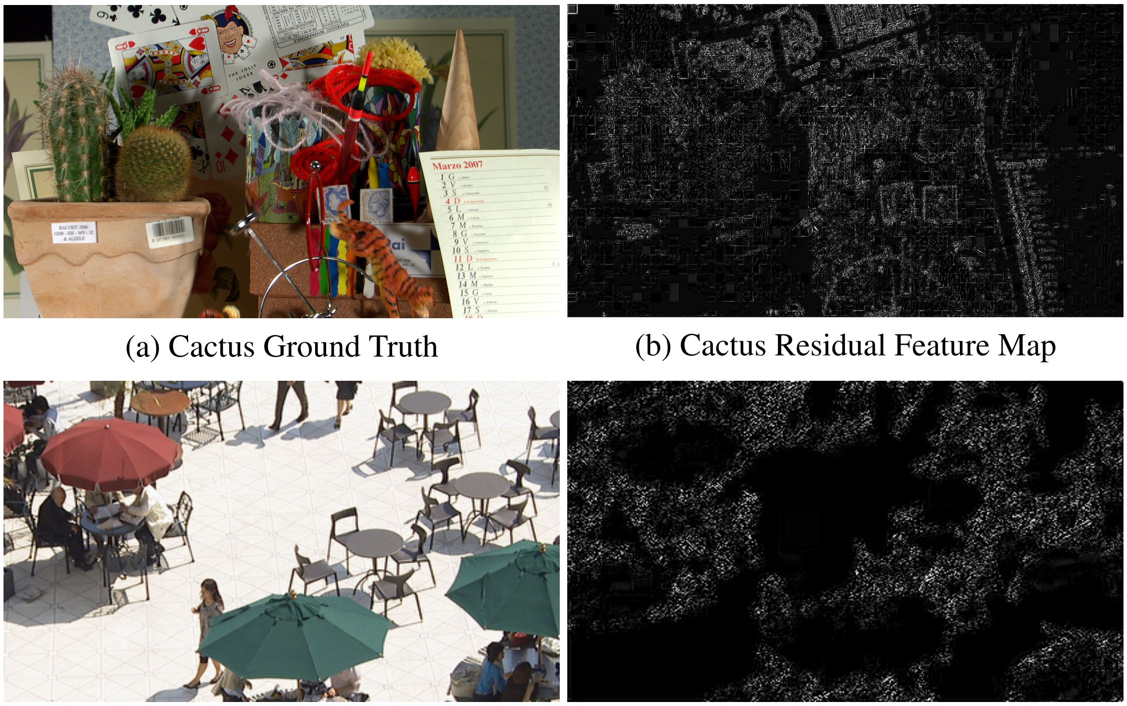


Figure 2: RRNet with sub-networks of both the Residual Network and the Reconstruction Network. Residuals are fed into the Residual Network to provide the TU partition information and the detailed textures information. The Residual Network relies on residual blocks to learn features effectively with residual learning. We feed the reconstruction into the Reconstruction Network. The Reconstruction Network executes the downsampling and upsampling strategy to patch up the local and global information. This enhances reconstruction quality and aids with the residual learning approach.

artifacts, respectively. Wang *et al.* [37] devised a Deep Dual-Domain (D^3) based fast restoration framework to recover high-quality images from JPEG compressed images. The D^3 model increased the large learning capacity of deep networks.

For videos, Xue *et al.* [38] proposed the task-oriented flow (TOFlow), where a motion representation was learned for video enhancement. Tao *et al.* [39] proposed a sub-pixel motion compensation (SPMC) model, which has shown its efficiency in video super-resolution applications. In the framework of video coding, Dai *et al.* [40] designed a Variable-filter-size Residual-learning CNN (VRCNN) that achieved 4.6% bit-rate gain. Yang *et al.* [41, 42] developed the Quality Enhancement Convolutional Neural Network (QE-CNN) method in HEVC. With the residual learning [43], Wang *et al.* [44] designed the dense residual convolutional neural network (DRN), which exploits the multi-level features to recover a high-quality frame from a degraded one. Other CNN-based video compression works, including [45–47] pushed the horizon of in-loop filtering techniques as well. Most recently, Zhang *et al.* [21] devised the residual highway convolutional neural network (RHCNN) in HEVC. Lu *et al.* [22] modeled loop filtering for video compression as a Kalman filtering process. Jia *et al.* [23] proposed a content-aware CNN based in-loop filtering for HEVC. However, most of these frameworks are designed for one specific restoration task. To address this issue, Jin *et al.* [48] proposed a flexible deep CNN framework that exploits the frequency characteristics of different types of artifacts.

The aforementioned deep learning methods only took the reconstructed low-quality video frame as input. However, the coding information was not efficiently utilized. To better use coding information, Lin and He *et al.* [5,24] proposed a partition-masked CNN,



(a) Cactus Ground Truth

(b) Cactus Residual Feature Map

(c) BQSquare Ground Truth

(d) BQSquare Residual Feature Map

Figure 3: Residual feature maps of Cactus and BQSquare derived from the Residual Network of RRNet under $QP37$. The residual features of Cactus with abundant context including pokers, calender and metal circle demonstrates its prominent contribution for enhancing the quality of the video frame. The residual features of BQSquare which are a flat example show a great amount of details involving chairs and tables.

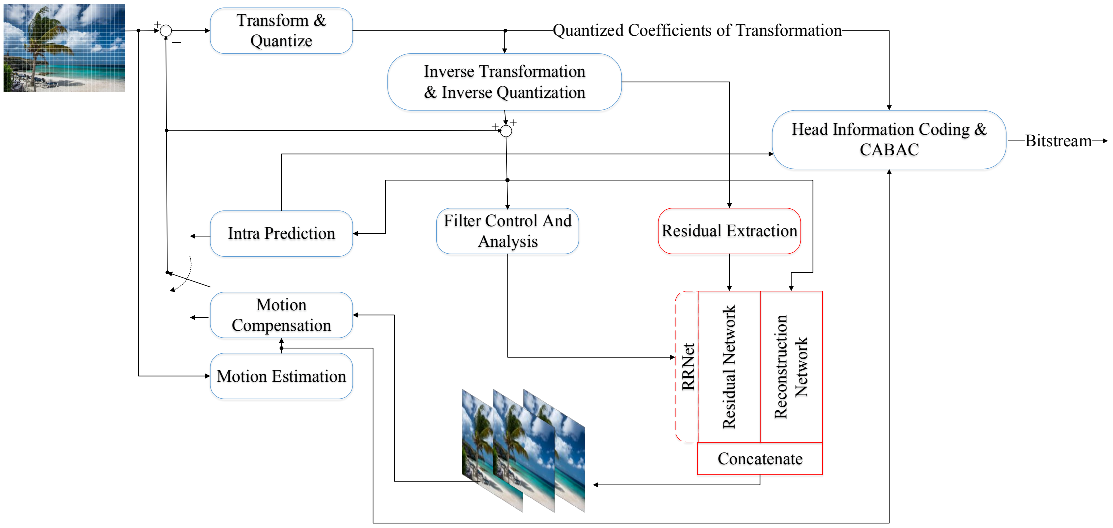


Figure 4: The location of RRNet embedded in HEVC. We insert the RRNet into HEVC as an in-loop method. The RRNet would input residual from extracting module and reconstruction into the Residual Network and the Reconstruction Network, respectively. The RRNet is executed instead of DF and SAO filters.

Table 1: The Residual Network Parameters of conv layers

Layers	Kernel Size	Feature maps Number	Stride	Padding
Conv 1	3×3	32	1	1
Residual Block 1 (2 convs)	3×3	64	1	1
Residual Block 2 (2 convs)	3×3	64	1	1
Residual Block 3 (2 convs)	3×3	64	1	1
Conv 8	3×3	32	1	1

Table 2: The Reconstruction Network Parameters of Conv And Transposed Conv Layers

Type of Layer	Conv1	Conv2	Conv3	Transposed Conv1	Conv4	Transposed Conv2	Conv5	Conv6
Kernel Size	3×3	3×3	3×3	2×2	3×3	2×2	3×3	3×3
Feature Map Number	32	64	128	64	64	32	32	32
Stride	1	1	1	2	1	2	1	1
Padding	1	1	1	0	1	0	1	1

where the block partition information was utilized for improving the quality of the reconstructed frames. It has shown additional improvements in terms of coding efficiency over the reconstruction-only methods.

1.3 Residual-assisted in-loop RRNet

This section will discuss the proposed RRNet scheme in detail, including a more in-depth discussion on the architecture of the RRNet, loss function, dataset, and training process.

1.3.1 Architecture of the proposed RRNet framework

Fig. 2 shows the overall architecture of the proposed RRNet framework. The proposed RRNet framework includes two sub-networks: the reconstruction network and the residual network. The reconstruction network uses the reconstruction as input and derives reconstruction feature maps from the input. The residual network uses the residual as input and derives residue feature maps from the input. The feature maps derived from the two sub-networks are concatenated together and used as the input of the last convolutional layer. In addition, we use the residual learning method that learns the difference between the input and the label to accelerate the training process.

As explained in the last paragraph, both the reconstruction and residual are utilized as the inputs of the proposed network. Applying the reconstruction as input is the same as most existing works since our target is to enhance the reconstruction. However, why the residual is used as the other input for our proposed RRNet network?

First, we believe that the residual can provide accurate transform unit (TU) partitions and great textures beneficial for the enhancement. Fig. 1 gives a typical example of the residual from the sequence Kimono. We can see clear TU boundaries from the residual figure. As we know, the basic unit of encoding the residual is a TU. Each TU transforms and quantizes independently. Therefore, it is more probable to have severe artifacts in the block boundary than the block center. The TU boundary information is a good indicator that implicates where the distortion is more severe and guides the network to learn more distinct features. In addition, we can see from the residual frame in Fig. 1 that, within each TU, the texture information is still visible. They can illustrate the body shapes of the girl and tree trunks clearly. This texture information also contributes to the reconstruction enhancement.

Second, the residual signal suffers from frame prediction accuracy, most notably in the areas where the residual contains non-zero values. This essentially means that the encoder does not accurately predict the regions where the residual values are large. Accordingly, the residual is beneficial for the CNN learning process, especially in areas where the residual contains non-zero values. From the extracted residual feature maps as shown in Fig. 3, we can see that the residual signal is useful for improving the capability of the CNN to learn sharp edges and complex shape information that would otherwise be missed by the encoder.

In addition to introducing the residual as the dual input, we can also see from Fig. 2 that we use different sub-networks for the reconstruction and residual. As we know, the characteristics of the reconstruction and residual are different. The residual is more sensitive, while the reconstruction consisting of residual and prediction contains more global information. We should design specific sub-networks to optimize the features derived from various inputs and improve the reconstructed frame quality. A detailed introduction of the two sub-networks will be described in detail in the next two subsections.

To give a better illustration of how we embed the above-introduced framework in HEVC, we give a modified HEVC encoder in Fig. 4. We replace the deblocking and SAO filters using the proposed RRNet framework. The output frame from our framework will be used as a reference for the to-be-encoded frames in the future. Note that in the proposed RRNet framework, we need to extract the residual from the bitstream in addition to the reconstruction.

1.3.2 Design of the Residual Network

We develop a Residual Network consisting of several residual blocks [43] to adapt to the residual features. The residual block could effectively keep the residual features and the gradient information on the shallow layers. Therefore, the proposed Residual Network can derive the distinct features from the residual frame. Considering the complexity, we use only 8 convolutional layers to derive the residual features. Because the network consisting of residual blocks [43] could effectively keep the residual features and the gradient information on the shallow layers [49], we adopt the residual block as the basic unit of our Residual Network.

The network based on the residual blocks brings apparent advantages. In the residual blocks based network, the collection of multiple routes substitutes the simple sole route. Based on the multiple routes property, because of the independence of the routes in the residual block-based network, this uncorrelated property enhances the canonical effect of the Residual Network. Because the contributions for the gradient information are mainly from the shallow layers, adding the weights of the short routes could effectively prevent from vanishing gradient.

In Fig. 2, the upper pathway shows the detailed architecture of our proposed Residual Network. Table 1 shows the convolutional layers configurations. The Residual Network includes three residual blocks consisting of six convolutional layers and two convolutional layers at the beginning and end. We set the Kernel Size for each convolutional layer as 3×3, the Feature Map Number as 32, Stride as 1, and Padding as 1. As the Parametric Rectified Linear Unit (PReLU) [50] has been demonstrated to be more effective than the ReLU, we employ it as the activation function in the Residual Network. We compute the feature maps of the Residual Network as follows:

$$\begin{cases} F_i^{res}(x) = A(W_i * F_{i-1}^{res}(x) + B_i), i \in \{2,4,6,8\} \\ F_j^{res}(x) = A(W_j * F_{j-1}^{res}(x) + B_j) + \ F_{j-2}^{res}(x), \ j \in \{3,5,7\} \end{cases} \tag{1.1}$$

where x denotes the input of residual, A is the activation function, W_i and B_i are the weights and bias matrices respectively.

1.3.3 Design of the Reconstruction Network

Simultaneously, we consider the reconstruction signal as the other input. Therefore, we design a Reconstruction Network containing several downsampling and upsampling pairs to learn the reconstruction features. The Reconstruction Network adopts the classic autoencoder architecture [51, 52] with the skip connection concatenating the encoder and decoder parts [53]. In this way, the reconstruction network can recover the global information and details as much as possible.

The Reconstruction Network has the following advantages. On the encoder side, downsampling the reconstruction helps extract more useful reconstruction features of low space dimensions. Based on the downsampling operation, upsampling the small reconstruction features helps derive the more extensive reconstruction features on the decoder side. The skip connection concatenating the reconstruction features from the encoder side could help the decoder to recover the global and detailed information of the reconstruction.

In Fig 2, the lower pathway shows the detailed structure of our proposed Reconstruction Network. We adopt the pooling and transposed convolutional layer to perform downsampling and upsampling, respectively. In the encoder phase, downsampling reduces the redundancy effectively in the reconstruction and keeps useful information. However, it may cut the global context as well. Hence, we execute the upsampling in the decoder phase to propagate the global information of the reconstruction to the next convolutional layer. Next, in the skip connection phase, we concatenate the concentrated reconstruction features from the encoder to the upsampling reconstruction features from the decoder. This is to provide the network with both the brief features and global context in the reconstruction. The Reconstruction Network is a difference learning network as well. Table 18 shows the detailed configurations. For the convolutional layers, we set the Kernel Size to 3×3, Stride to 1, Padding to 1, Feature Map Number to 32, 64 or 128. For the transposed convolutional layers [54], we set the Kernel Size to 2×2, Stride to 2, Padding to 1, Feature Map Number to 64 or 32. The reconstruction network can be formulated as follows,

$$F_i^{rec}(z) = P(W_i * F_{i-1}^{rec}(z) + B_i), i \in \{1, 2\} \tag{1.2}$$

where z is the reconstruction signal input, and P represents the sequential functions for activation and max-pooling. We choose PReLU as the activation function in the Reconstruction Network.

Table 3: Training parameters

Parameters	QP 37
Base Learning Rate	$1e^{-4}$
γ Adjusting Coefficient	0.1
Adjusting Epochs Interval	100
Weight Decay	$1e^{-4}$
Momentum	0.9
Total Epochs	120

$$\begin{aligned} F_5^{rec}(z) &= C(P(W_5 * F_4^{rec}(z) + B_5), F_2^{rec}(z)) \\ F_7^{rec}(z) &= C(P(W_7 * F_6^{rec}(z) + B_7), F_1^{rec}(z)) \end{aligned} \tag{1.3}$$

where C denotes the concatenating function for jointing features.

After concatenating the features of the Residual Network and the Reconstruction Network, we calculate them with a convolutional layer of 1 channel. Then we obtain the final output $F_{out}(x, z)$ which is the same size as input.

1.3.4 Loss function, dataset and training

Loss function. We employ Mean Squared Error (MSE) [55] as the loss function for our proposed RRNet as follows,

$$L(\Theta) = \frac{1}{N} \sum_{i=1}^{N} ||\Upsilon(Y_i|\Theta) - X_i||_2^2 \tag{1.4}$$

where Θ encapsulates the whole parameter set of the network containing weights and bias and $\Upsilon(Y_i|\Theta)$ denotes the network module. X_i is a pixel of the original frame, where i indexes each pixel. Y_i is the corresponding pixel of the reconstruction, that is compressed by HEVC when we turn off its deblocking and SAO. N is the number of pixels.

Dataset. We employ the DIV2K [56, 57] dataset comprising 800 training images and 100 validating images of $2k$ resolution as the original frames. Because modern video codecs operate on YUV color domain, we convert the original 900 PNG images to YUV videos with FFMPEG [58] of GPU acceleration. A modified HEVC reference software is then used to encode original frames to generate the reconstruction and residual with $QP22$, $QP27$, $QP32$, and $QP37$, respectively. We finally extract 64×64 blocks from the Luma component of the reconstructed, residual, and original frames and use them as the inputs and labels for training our proposed RRNet. In total, there are $522,939$ groups of inputs and labels for training and $66,650$ groups for validation.

Training. Once we obtain the residual and reconstruction patches of divided components, we feed them into the Residual Network and the Reconstruction Network, respectively, by batch-size of 16. Table 19 exhibits the parameters of training procedure for $QP37$ samples. We experiment with a larger learning ($1e^{-3}$) rate and a smaller learning rate ($1e^{-5}$), but the former one leads to the gradient explosion while the later one learns too slowly. Therefore, $1e^{-4}$ is the appropriate base learning rate of $QP37$ model. We adopt the Adaptive Moment Estimation (Adam) [59] algorithm with the momentum of 0.9 and the weight decay of $1e^{-4}$. These parameter values are selected according to experience values. When the model is trained less than 120 epochs, the loss has not been convergent. Accordingly, the $QP37$ model is trained with 120 epochs. After 100 epochs, we decrease the learning rate by 10 times. After the $QP37$ model is derived, we fine tune it with 20 epochs to obtain the other models: $QP22$, $QP27$, $QP32$. Finally, we obtain the models for all the QPs for testing.

Table 4: BD-rate of the SOTAs and proposed RRNet against HEVC under All Intra case

Class	Sequence	VRCNN [40] vs. HEVC	EDSR Residual Blocks [6] vs. HEVC	Partition-aware CNN [5] vs. HEVC	RRNet vs. HEVC
A	Traffic	−8.1%	−8.5%	−8.7%	**-10.2**%
	PeopleOnStreet	−7.7%	−7.8%	−8.2%	**-9.4**%
B	Kimono	−5.9%	−6.6%	−6.9%	**-8.6**%
	ParkScene	−6.2%	−6.6%	−6.9%	**-8.1**%
	Cactus	−2.7%	−4.9%	−5.4%	**-5.8**%
	BasketballDrive	−5.2%	−4.6%	−4.7%	**-7.7**%
	BQTerrace	−2.9%	−2.9%	−2.9%	**-4.2**%
C	BasketballDrill	−10.6%	−10.9%	−11.3%	**-13.8**%
	BQMall	−7.3%	−7.0%	−7.4%	**-9.3**%
	PartyScene	−4.6%	−4.5%	−4.8%	**-5.6**%
	RaceHorses	−5.8%	−5.0%	−5.3%	**-7.1**%
D	BasketballPass	−7.6%	−7.3%	−7.8%	**-9.5**%
	BQSquare	−5.3%	−5.4%	−5.8%	**-6.3**%
	BlowingBubbles	−5.5%	−5.5%	−5.7%	**-6.7**%
	RaceHorses	−8.9%	−8.8%	−9.1%	**-10.2**%
E	FourPeople	−10.0%	−10.4%	−10.9%	**-12.8**%
	Johnny	−9.1%	−8.1%	−8.7%	**-12.5**%
	KristenAndSara	−9.4%	−9.0%	−9.6%	**-11.8**%
	Class A	−7.9%	−8.2%	−8.5%	**-9.8**%
	Class B	−4.6%	−5.1%	−5.4%	**-6.9**%
	Class C	−7.1%	−6.9%	−7.2%	**-8.9**%
	Class D	−6.8%	−6.7%	−7.1%	**-8.2**%
	Class E	−9.5%	−9.2%	−9.7%	**-12.4**%
Avg.	All	−6.8%	−6.9%	−7.2%	**-8.9**%

Table 5: The computational complexity of VRCNN and proposed RRNet against HEVC under All Intra case

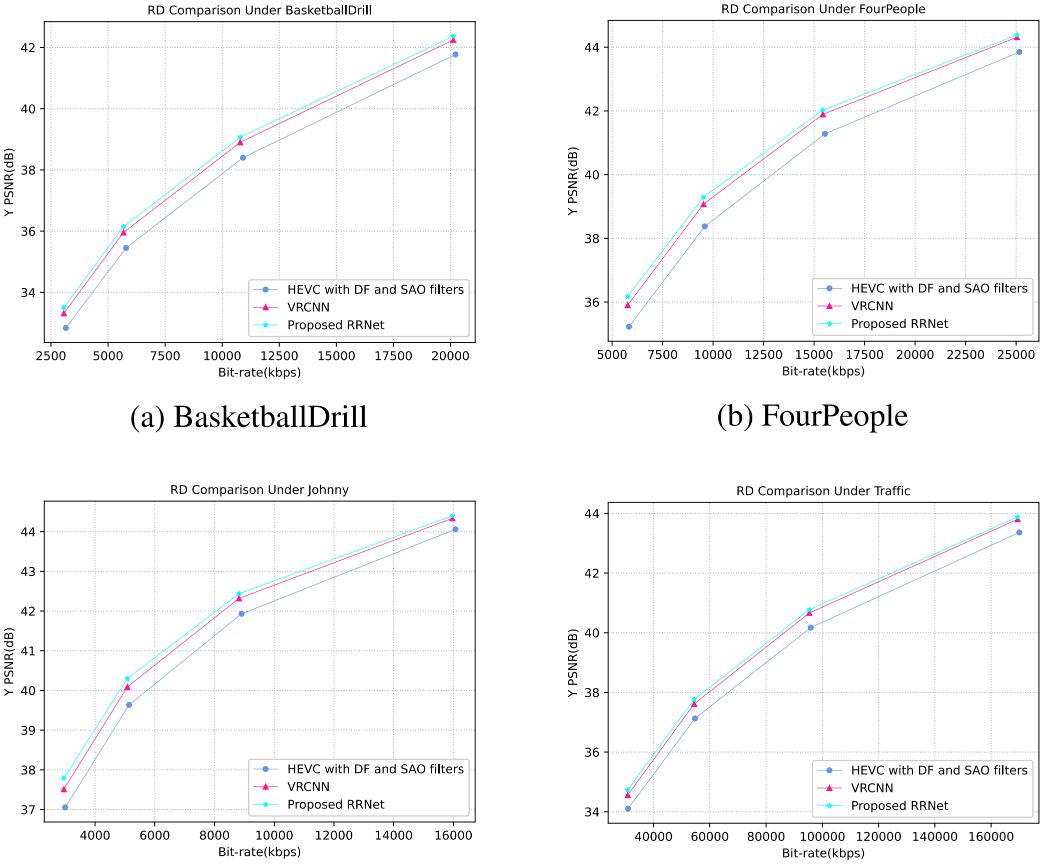
Approches	Frame-work	Encoding Time	Decoding Time
VRCNN	Pytorch(C++)	108.72%	420.41%
RRNet	Pytorch(C++)	117.48%	1238.78%

1.4 Experimental results

To test the performance of the proposed algorithm, we embedded the proposed RRNet scheme into HEVC reference software as shown in Fig. 4. In this section, we first compare the proposed RRNet with VRCNN [40], EDSR Residual Blocks [6], Partition-aware CNN [5], and HEVC on BD-rate [60], respectively. Subsequently, we validate the multiple inputs function by comparing the dual-input residual and reconstruction with the solo input reconstruction. Meanwhile, we compare the dual-input Residual and Reconstruction approach with the dual-input Partition and Reconstruction approach [5]. Afterward, we evaluate the efficiency of different networks on the same inputs by comparing RRNet and EDSR Residual Blocks with the dual-input of residual and reconstruction. For the test, we test all the sequences defined in HM-16.19 CTC [61] under the intra-coding and inter-coding configurations.

1.4.1 Performances of the proposed RRNet algorithm

Table 4 shows the comparison results of VRCNN [40], EDSR Residual Blocks [6], Partition-aware CNN [5], and the proposed RRNet against HEVC under the all intra case.



(a) BasketballDrill

(b) FourPeople

(c) Johnny

(d) Traffic

Figure 5: Comparison of RD curves in HEVC with DF and SAO, VRCNN and proposed RRNet on luminance. The compared RD curves of BasketballDrill(a), FourPeople(b), Johnny(c) and Traffic(d) are shown. It is obvious that our proposed RRNet outperforms HEVC with DF and SAO and VRCNN for all theses sequences under all tested QPs including $22, 27, 32$ and 37.

Table 6: BD-rate of VRCNN and proposed RRNet against HEVC under Random Access case

Class	Sequence	VRCNN vs. HEVC	RRNet vs. HEVC
A	Traffic	−5.0%	**-6.0**%
	PeopleOnStreet	−1.4%	**-1.6**%
B	Kimono	−1.9%	**-2.6**%
	ParkScene	−2.7%	**-3.4**%
	Cactus	−3.2%	**-3.9**%
	BasketballDrive	−1.4%	**-1.9**%
	BQTerrace	−5.2%	**-5.8**%
C	BasketballDrill	−3.1%	**-4.3**%
	BQMall	−2.0%	**-2.5**%
	PartyScene	−0.5%	**-1.0**%
	RaceHorses	−1.3%	**-1.4**%
D	BasketballPass	−0.7%	**-0.9**%
	BQSquare	−1.4%	**-2.1**%
	BlowingBubbles	−1.8%	**-2.4**%
	RaceHorses	−1.5%	**-1.6**%
E	FourPeople	−8.2%	**-9.5**%
	Johnny	−7.6%	**-10.2**%
	KristenAndSara	−6.9%	**-7.6**%
	Class A	−3.2%	**-3.8**%
	Class B	−2.9%	**-3.5**%
	Class C	−1.7%	**-2.3**%
	Class D	−1.4%	**-1.7**%
	Class E	−7.6%	**-9.1**%
Avg.	All	−3.1%	**-3.8**%

Note that to ensure fairness, the EDSR Residual Blocks and Partition-aware CNN all employ eight convolutional layers, including three residual blocks as shown in Table 9, which have the same convolution layer depth as the one of the Residual Network in the proposed RRNet. We train $QP37$ models of VRCNN, EDSR Residual Blocks, and Partition-aware CNN with 120 epochs on the whole DIV2K dataset and then achieve the models of $QP32$, $QP27$ and $QP22$ by fine tuning the trained $QP37$ model with 20 epochs. These are identical to the process used to train RRNet as stated in Section 3.4.5.

We can see that the proposed RRNet algorithm outperforms VRCNN, EDSR Residual Blocks, and Partition-aware CNN by an average of 2.1%, 2.0%, and 1.7%, respectively. Additionally, the RRNet method surpasses VRCNN, EDSR Residual Blocks, and Partition-aware CNN in every sequence in BD-rate. Specifically, the proposed RRNet scheme outperforms VRCNN, EDSR Residual Blocks, and Partition-aware CNN by 2.9%, 3.2%, and 2.7% on Class E, respectively. Similarly, compared to the HEVC anchor, RRNet realizes a substantial gain on BD-rate with an average of -8.9%. The most remarkable individual difference occurs on BasketballDrill sequence with a gain of -13.8% on BD-rate. This sequence contains particularly complex textures with very dramatic variations. These performances demonstrate that RRNet effectively enhances the reconstruction by introducing the residual signal and developing customized networks for residual and reconstruction inputs.

Fig. 23 shows the luminance Rate-Distortion (RD) curves of the proposed RRNet approach, VRCNN, and HEVC anchor. As illustrated, the PSNR of the proposed RRNet method is higher than the one of VRCNN and HEVC with in-loop filters under every QP in BasketballDrill, FourPeople, Johnny, and Traffic sequences. This clearly shows that the proposed RRNet model is superior to the VRCNN and HEVC baseline approaches to enhance the quality of compressed video frames.

The time complexity [62] is exhibited in Table 5. In all cases, we apply the same test environment. Specifically, the GPU configuration is GTX 1080ti. Due to the huge computation of CNN on the encoder side, VRCNN takes 8.72% longer than HEVC. Meanwhile, because of the dual-input networks, RRNet takes 17.48% longer than HEVC. On the decoder side, the results reflect a similar situation for complexity. HEVC computes fastest while RRNet complexity overhead is 1238.78%. We can adopt the methods of model compression and acceleration [63–65] to reduce the redundancy of the proposed RRNet model. The solutions of model compression and acceleration includes parameter pruning, quantization, low-rank factorization, compact convolutional filters, and knowledge distillation. We can use the parameter pruning and quantization based approaches to remove the redundancy of the RRNet parameters. In addition, the low-rank factorization based methods are utilized to calculate the useful parameters of RRNet. The compact convolutional filters are structurally designed to shrink the parameter space of RRNet and save computation and storage resources. The approaches based on knowledge distillation is used to train a more compact RRNet or learn a distilled RRNet model.

Table 6 shows the experimental results in random access case. We can see that the proposed algorithm can bring an average of -0.7% and -3.8% BD-rate gain compared to VRCNN and HEVC, respectively. Again, we can also see that RRNet outperforms the other two methods in every class. Moreover, the peak difference between RRNet and

Table 7: The dual-input Residual and Reconstruction approach and the dual-input Partition and Reconstruction [5] approach versus Reconstruction only approach on the BD-rate

	Partition and Reconstruction [5] vs. Reconstruction	Residual and Reconstruction vs. Reconstruction
Class A	-0.4%	**-1.0**$\%$
Class B	-0.2%	**-0.9**$\%$
Class C	-0.4%	**-1.1**$\%$
Class D	-0.4%	**-0.8**$\%$
Class E	-0.6%	**-1.6**$\%$
Avg. All	-0.4%	**-1.0**$\%$

Table 8: The computational complexity of the dual-input Partition and Reconstruction method [5] and the dual-input Residual and Reconstruction approach against HEVC

Approches	Frame-work	Encoding Time	Decoding Time
Partition Reconstruction [5]	Pytorch(C++)	122.24%	1581.63%
Residual Reconstruction	Pytorch(C++)	123.81%	1669.39%

VRCNN reaches 1.5% on Class E. This demonstrates that the benefits brought by RRNet can be propagated to inter frames. Thus the RRNet can bring significant performance improvements in random access case.

Table 9: Convolutional Parameters of EDSR Residual Blocks [6]

Kernel Size	3×3
Feature Map Number	32
Stride	1
Padding	1

1.4.2 Results analysis of multiple inputs approaches

Here we compare the method with residual and reconstruction inputs to the method with only reconstruction input. Additionally, we compare the dual-input Residual and Reconstruction approach with another multiple inputs approach that utilizes the mean mask of the PU partition [5] and Reconstruction. Note to guarantee a fair comparison, all reconstruction sub-networks utilize the same network with eight convolutional layers, including three EDSR residual blocks shown in Table 9.

Table 7 exhibits the comparison of the dual-input Residual and Reconstruction scheme against Reconstruction only method and the comparison of the dual input PU Partition and Reconstruction method against Reconstruction only method. On the one hand, the dual-input Residual and Reconstruction saves an average of -1.0% BD-rate compared with Reconstruction only method. On the other hand, the dual-input Residual and Reconstruction method saves an average of -0.6% BD-rate over the dual input Partition and Reconstruction method. Specifically, the dual-input Residual and Reconstruction approach leads -1.6% BD-rate on Class E against the only Reconstruction method. The peak difference of BD-rate between the dual-input Partition and Reconstruction method and the only Reconstruction method on Class E is -0.6%. In every class, the dual-input of the Residual and Reconstruction approach is better than the only Reconstruction method and the dual-input of the Partition and Reconstruction method on BD-rate.

These performances clearly show that based on the same network architecture for video reconstruction, the residual signal provides useful information for augmenting the quality. This is reasonable because the inverse transformed residual provides the TU

Table 10: BD-rate of RRNet against the dual-input Residual and Reconstruction with EDSR Residual Blocks [6]

Class	RRNet vs. Residual and Reconstruction with EDSR Residual Blocks
Class A	-0.8%
Class B	-1.4%
Class C	-1.0%
Class D	-0.5%
Class E	-2.2%
Avg. All	-1.2%

partition information and the detailed textures used to enhance the reconstruction. Hence, introducing the residual signal augments the quality of the compressed video frame prominently. In conclusion, compared to the only Reconstruction method and another multiple input methods based on the mean mask of the partition, the dual-input Residual and Reconstruction approach clearly augments the reconstruction. On the aspect of the time complexity, as shown in Table 8, the dual-input Residual and Reconstruction approach and the dual-input Partition and Reconstruction method are approximately on the same level.

1.4.3 Results analysis of network architecture

We compare the proposed RRNet approach with the dual-input of residual and reconstruction method with EDSR Residual Blocks to evaluate the performance of the proposed Residual Network and Reconstruction Network. Note that both the RRNet and the second method have the same inputs. The second method utilizes the EDSR Residual Blocks on both residual and reconstruction. Table 10 shows the compared results between RRNet and the dual-input of residual and reconstruction approach with EDSR

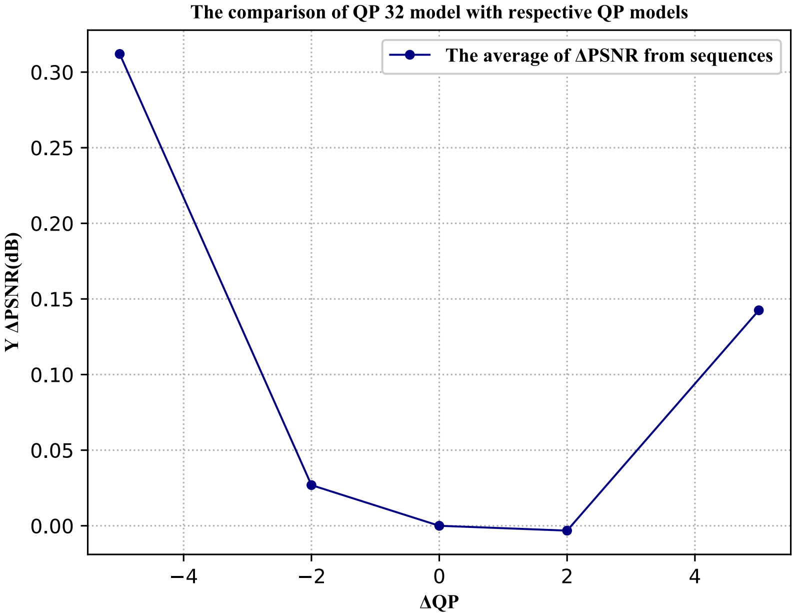


Figure 6: The comparison of QP 32 model with respective QP models. The ΔPSNR on $\Delta QP = 0$ means the QP32 model compared to itself on PSNR is zero. Except QP34 setting, the PSNR of individual QP model is better than the one for the QP32 settings on the other QP parameters. The ΔPSNR increases significantly with the absolute value of ΔQP on the each side of $\Delta QP = 0$.

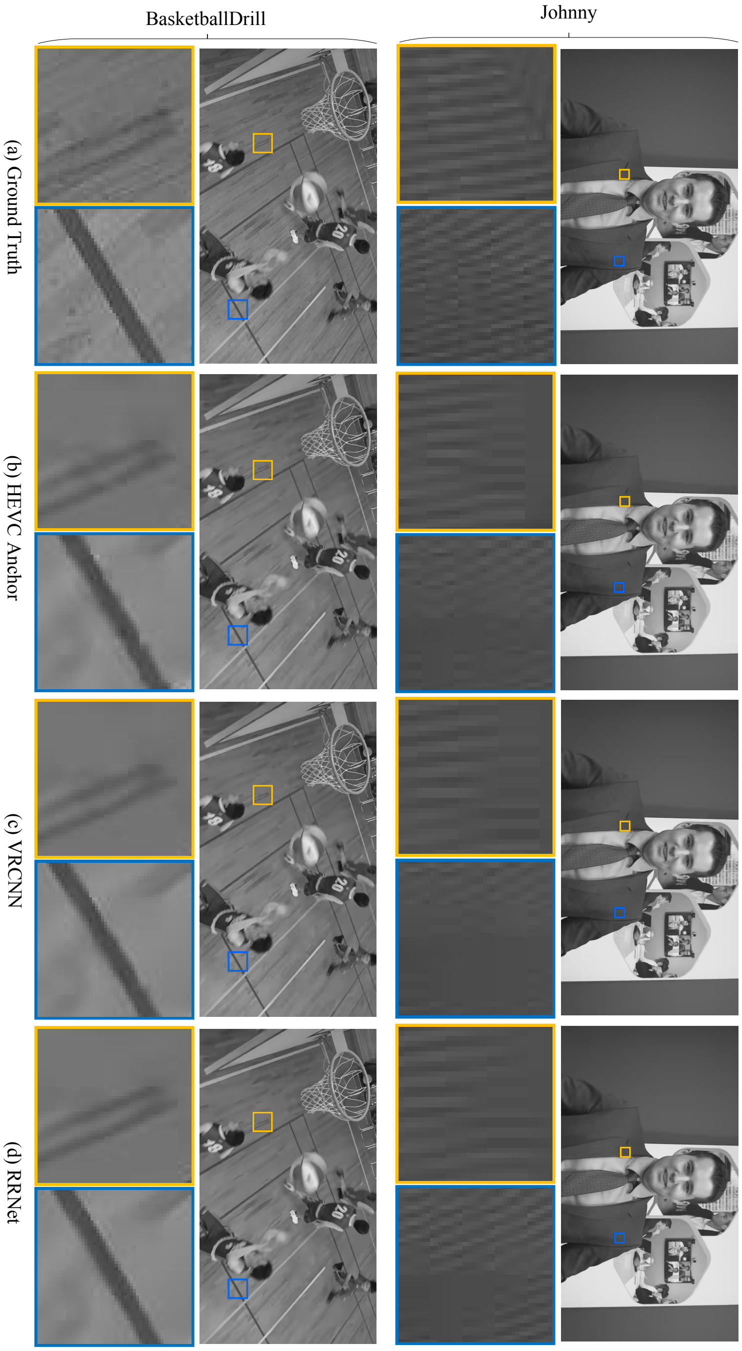


Figure 7: Visual comparisons between the ground truths, HEVC anchor, VRCNN, and proposed RRNet approach on the luminance of $QP37$ in Johnny and BasketballDrill sequences, respectively. The groups of figures (a), (b), (c), and (d) are the original video, the video generated using HEVC, the video generated using VRCNN, the video generated using RRNet, respectively. (Zoom in for better visual effects.)

Residual Blocks. RRNet gains an average of -1.2% BD-rate against the latter method. Specifically, the proposed RRNet outperforms the dual-input of residual and reconstruction method with EDSR Residual Blocks in every class sequence for BD-rate. The largest difference of BD-rate is -2.2% on the Class E sequence. These demonstrate that both the Residual Network and the Reconstruction Network fit their respective signals very well. The results also clearly demonstrate that processing the residual and reconstruction with unique architectures is beneficial. Additionally, the validation of comparison provides evidence that the RRNet network shows an obvious improvement in the quality of coded frames.

1.4.4 The performance from a specific QP model on different QPs

To validate the performance from an assigned QP model on other QP settings, as illustrated in Fig. 6, we compared the PSNR of QP32 when reconstructed by other QP models. The ΔPSNR on $\Delta QP = 0$ means that the QP32 model compares itself on PSNR, and it should be zero. Except for QP34, the PSNR of other QP models evaluated on itself is better than when it is evaluated on the QP32 model. The ΔPSNR increases dramatically with the absolute value of ΔQP on both positive and negative sides. Accordingly, specific QP tuned models outperform the other QP models when tuned for that specific setting. In summary, based on Fig. 6, a model can be reused to replace another model in the range of -2 to 2 ΔQP.

1.4.5 Subjective Results

Fig. 7 exhibits the visual comparisons between the ground truths, HEVC anchor, VRCNN, and proposed RRNet approach on the luminance of $QP37$ in Johnny and BasketballDrill sequences, respectively. The groups of figures (a), (b), (c), and (d) are the original video, the video generated using HEVC, the video generated using VRCNN, the video generated using RRNet, respectively. In the Johnny, from the zoomed gold blocks, we can see that there are evident distortions and textures miss in the HEVC and VRCNN frames, while the RRNet frame shows smoother and more abundant textures. We can see from the zoomed blue rectangles that the HEVC and VRCNN frames blur more severely than the RRNet frame. From the BasketballDrill, we can see from the zoomed gold and blue blocks that the distortions in HEVC and VRCNN frames are more serious than the one of the RRNet frame. The experimental results demonstrate that the proposed RRNet can bring better subjective qualities than the previous in-loop filtering methods.

1.5 Summary

In this paper, we propose a new video deblocking solution that utilizing both reconstructed pixels as well as rich information and features available from the compression pipeline. The coding residual signal unique from compression pipeline is utilized as an additional input for improving the CNN based in-loop filter for HEVC. In essence, it is introduced to enhance the quality of reconstructed compressed video frames. In this process, we first import the residual as an independent input to reinforce the textures and details. Then, we custom designed RRNet approach that involves two separate CNNs:

the Residual Network and the Reconstruction Network. Each customized layer aims to reveal specific features that are characteristic of each type of frame. In the Residual Network, we apply residual blocks to minimize the difference between the input frame and the output frame. In the Reconstruction Network, we utilize both downsampling and upsampling ladders to adapt to learn the features for the reconstruction frames. The experimental results demonstrate that the proposed algorithms significantly reduce artifacts from both objective and subjective perspectives. From the objective point of view, the BD-rate is significantly improved. From the subjective point of view, the reconstruction quality of the compressed video frames is superior. These results demonstrate that the proposed schemes improved the current state of the art significantly in BD rate reduction. In the future, we will try to create more advanced in-loop methods for video coding, while develop complexity reduction for the inference time model.

CHAPTER 2

DEEP LEARNING GEOMETRY COMPRESSION ARTIFACTS REMOVAL FOR VIDEO-BASED POINT CLOUD COMPRESSION

Point cloud is an essential format for three-dimensional (3-D) object modelling and interaction in Augmented Reality (AR) and Virtual Reality (VR) applications. In the current state of the art video-based point cloud compression (V-PCC), a dynamic point cloud is projected onto geometry and attribute videos patch by patch, each represented by its texture, depth, and occupancy map for reconstruction. To deal with occlusion, each patch is projected onto near and far depth fields in the geometry video. Once there are artifacts on the compressed two-dimensional (2-D) geometry video, they would be propagated to the 3-D point-cloud frames. In addition, in the lossy compression, there always exists a tradeoff between the rate of bitstream and distortion (RD). Although some geometry-related methods were proposed to attenuate these artifacts and improve the coding efficiency, the interactive correlation between projected near and far depth fields has been ignored. Moreover, the non-linear representation ability of Convolutional Neural Network (CNN) has not been fully considered. Therefore, we propose a learning-based approach to remove the geometry artifacts and improve the compressing efficiency. We have the following contributions. We devise a two-step method working on the near and far depth fields decomposed from geometry. The first stage is learning-based Pseudo-Motion Compensation. The second stage exploits the potential of the strong correlations between near and far depth fields. Our proposed algorithm is embedded in the V-PCC reference software. To the best of our knowledge, this is the first learning-based solution of the geometry artifacts removal in V-PCC. The extensive experimental results show that the proposed approach achieves significant gains on geometry artifacts removal and quality improvement of 3-D point-cloud reconstruction compared to the state-of-the-art schemes.

2.1 Background

Due to the massive demands for stereoscopic experience, three-dimensional (3-D) sensing and scanning instruments including Light Detection and Ranging (LIDAR) scanners [66, 67] and RGB-D cameras [67, 68] are developing unprecedentedly. Those 3-D devices daily generate an enormous amount of data. To visualize these 3-D data vividly, some 3-D representing methods, such as point clouds, light fields, and polygon meshes, progress rapidly. Those stereo expressing approaches are capable of representing the 3-D volumetric data in a realistic and immersive way. Point cloud is especially popular among these methods since we can acquire them more easily, render them more realistically, and manipulate them more feasibly.

Point cloud is an important format for various 3-D based volumetric technologies such as virtual reality (VR), augmented reality (AR), and mixed reality (MR) [69] all advancing rapidly. The number of 3-D applications [70, 71] is therefore increasing significantly based on these immersive and realistic technologies. For instance, an applying case is navigation [72, 73]. The mobile navigation [73] aims to create a 3-D map with localization data, global positioning system (GPS) images, and depth data. Other applications are VR/AR immsersive videos, games [74] and telecommunications [75]. Since it is feasible to render and visualize point clouds, we can use a collection of 3-D point clouds to represent the low delay 3-D stream of the high quality (4K or 8K) immersive telecommunications. In addition, the historic relic [76] is another interesting application. This type of heritage application is capable of providing an immersive stereo experience by visualizing the real historical relic to billions of points. However, point clouds contain a lot of digital data, so that storing and streaming such massive data is very difficult. The conflict between demands and capacity of the storage pushes an emergence of useful point cloud compression (PCC) solutions.

Driven by this technique requirement, Moving Pictures Experts Group (MPEG) starts the standardization [77] of PCC. Owing to the different densities of point clouds, there are two types of schemes incorporating video-based point cloud compression (V-PCC) [77] and geometry-based point cloud compression (G-PCC) [77]. V-PCC mainly works on dense point clouds while G-PCC works on sparse point clouds such as large-scale point cloud maps that are produced by simultaneous localization and mapping (SLAM) algorithms [78]. We mainly discuss the V-PCC development in this work. V-PCC divides a point cloud into many 3-D patches at first. V-PCC then projects the generated 3-D patches onto 2-D planes and packs them into a 2-D geometry video and a texture video. Subsequently, to encode 2-D videos efficiently, V-PCC maintains spatial continuity by padding the void area of geometry and texture videos before the 2-D compression. V-PCC eventually compresses the padded geometry and texture videos with 2-D video codecs such as Advanced Video Coding (AVC) [10], High Efficiency Video Coding (HEVC) [11], and Versatile Video Coding (VVC) [12] in the lossy mode.

Due to this lossy compression of V-PCC, distortions exist in the 2-D geometry reconstruction and 3-D point-cloud reconstruction. Essentially, the 2-D geometry video is the depth information of the 3-D point cloud. Once the 2-D geometry reconstruction distorts, the 3-D point-cloud reconstruction will also have artifacts. For instance, when the 2-D geometry reconstruction loses some pixels, the corresponding points of the 3-D point-cloud reconstruction will miss as well. Similarly, if the 2-D geometry reconstruction inserts a few noisy redundant pixels, the 3-D point-cloud reconstruction will introduce the corresponding redundant points. In addition, if some values of the 2-D geometry reconstruction verify, compared to the origin, the corresponding points of the 3-D reconstruction will locate in mistaken positions. All these cases degrade the quality of 3-D point-cloud reconstruction and result in artifacts.

To attenuate artifacts, researchers proposed some 2-D geometry-related methods [4, 9, 79, 80]. Among them, [81] as a geometry padding approach proposed padding the empty space between patches with neighboring patch information. This method is especially beneficial for coding efficiency in all the intra cases. To further decrease the distance between the point-cloud reconstruction and its origin, [8] searched and picked up a depth value from a depth candidate list in the 2-D geometry video for padding the 3-D geometry. This encoder-only method is the first method using geometry reconstruction to process the inserted redundant positions. Although these methods have achieved successes, the non-linear representation ability of the learning-based [82] approach has not been fully considered. There is still considerable space to develop a better learning-based geometry distortion removal algorithm.

Therefore, to take care of the tradeoff between the distortion and bitrate, in this work, we propose for the first time a learning-based approach removing the geometry artifacts for a better quality of the 3-D point-cloud reconstruction. We make the following contributions.

- To address the geometry artifacts problem, we propose a two-step method working on the near and far depth fields decomposed from geometry. The first stage is learning-based Pseudo-Motion Compensation. The second stage exploits the potential of the strong correlations between near and far depth fields. To the best of our knowledge, this is the first learning-based solution of the geometry artifacts removal in V-PCC.

- Our proposed algorithm is embedded in the V-PCC reference software for simulation. We have conducted extensive experiments to compare with state-of-the-art (SOTA) methods to demonstrate the effectiveness of the proposed method. We thoroughly analyze the experimental results to give more insights into the problem.

We organize the remainder of this paper as follows. We review the related works on point cloud compression in Section 2.2, followed by our motivation and observations on geometry in Section 2.3. We introduce the proposed geometry artifacts removal with two stages in Section 2.4. In section 2.5, we report and analyze the experimental results comprehensively. We summarize this paper in Section 2.6 briefly.

2.2 Artifacts removal in V-PCC

This section briefly reviews the previous point cloud compression (PCC) works and the geometry improvement methods in V-PCC.

2.2.1 Point cloud compression

The PCC methods can be roughly summarized into two groups, which are the group of 3-D-based and 2-D-based approaches and the group of deep learning-based approaches.

1) 3-D-based and 2-D-based methods. Because there is no strong time correlation between the neighboring frames, the 3-D-based methods can not precisely estimate a motion between points in neighboring frames. Kammerl *et al.* [83] devised a lossy compression approach for dynamic point cloud streaming. The co-located octree node of the reference point-cloud frame predicted the current point-cloud frame. However, we can only apply this approach to a few moved point-cloud frames. Thanou *et al.* [84] used a set of graphs to represent the time-varying geometry of these point-cloud frames. Based on this, they cast 3-D motion estimation as a feature-matching issue between consecutive point-cloud frames. However, this method did not precisely estimate the motion vectors of some objects in point-cloud frames.

Queiroz *et al.* [85] proposed a simple codec. This coder segmented the voxelized point cloud at each frame into blocks of voxels. Their proposed method executed the 3-D translational motion estimation block by block to find the corresponding block in the reference point-cloud frame. In addition, Mekuria *et al.* [86] further imported iterative closest point (ICP) instead of translational motion model to better formulate the motions in neighboring point-cloud frames. These methods could relieve the suffering from 3-D motion estimation and motion compensation to some extent.

To solve the bottleneck that streaming and caching the point clouds requires large bandwidth and storage space, Sun *et al.* [87] proposed a clustering method starting with a range image-based 3-D segmentation. In addition, it introduced a prediction with the depth modeling modes for depth map coding. Nevertheless, without flexible block partition and more efficient motion estimation schemes, the coding efficiency of the dynamic point cloud compression (DPCC) still cannot be compatible with the 2-D-based approach.

Because codecs such as AVC, HEVC, and VVC have proven that the 2-D video compression algorithms are efficient, researchers proposed 2-D-based methods to transform the 3-D dynamic point cloud to 2-D videos for compression. Budagavi *et al.* [88] developed a method to code a projected 2-D video acquired from ordering points in a 3-D point cloud with HEVC. However, this work could not further utilize the inter-prediction information since the obtained video did not have many spatial and temporal correlations. To attenuate this deficiency, He *et al.* [89] proposed a cubic projection method to convert a 3-D dynamic point cloud to a 2-D video. Although this work improved video coding performance, this approach resulted in missing points due to occlusion.

To minimize the number of occluded points, Lasserre *et al.* [90] proposed an approach that combined the octree and projection. Mammou *et al.* [91] devised a method that projected a 3-D dynamic point cloud to 2-D videos by a patch-based scheme. Their motivation was to consider projecting more points while reducing the bit cost on 2-D video coding as much as possible. Packing a group of patches that consists of 2-D pixels converted from 3-D points was the main philosophy of the patching method. This work packed these 2-D patches onto a video then compressed by codecs such as HEVC. Li *et al.* [92] proposed a general model utilizing the 3-D motion and 3-D to 2-D correspondence to calculate the 2-D motion vector (MV). Compared to other proposals, the patch-based method [93] performed better in coding efficiency. MPEG Immersive media working group (MPEG-I) adopted [93] as a V-PCC standard. This approach has shown its efficiency with excellent performance. However, we have not improved the geometry video to its full potential extent, which intrinsically guides the 3-D point-cloud reconstruction process.

2) Deep learning-based methods. Point cloud processing is the fundamental component for any deep-learning-based point-cloud applications. Rather than compress point cloud data directly, Tu *et al.* [94] proposed converting the packet data, which is raw point cloud data, losslessly into range images previously. To avoid unnecessarily voluminous rendering data, Qi *et al.* [95] proposed a type of neural network that directly consumes point clouds, which well respects the permutation invariance of points in the input. However, many works process the 3-D videos frame-by-frame either through 2-D convents or 3-D perception algorithms. Choy *et al.* [96] proposed a new sparse tensor-based 3-D point-cloud processing method called Minkowski. This network for spatio-temporal perception can directly process such 3-D videos using high-dimensional convolutions. Gojcic *et al.* [97] proposed an end-to-end algorithm for joint learning of both parts of initial pairwise and the globally consistent refinement.

The deep learning-based PCC methods utilized the non-linear ability of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to improve the efficiency of PCC. To require less volume while giving the same decompression accuracy, Tu *et al.* [98] used an RNN and residual blocks to compress one frame from 3-D LiDAR. However, these two methods did not use joint optimization. Quach *et al.* [99] performed the joint optimization of both rate and distortion with a tradeoff parameter on a data-driven method. They created this method for the geometry static point cloud compression (SPCC) based on learned convolutional transforms and uniform quantization. Nevertheless, there is still space to improve the architecture of the network. Huang *et al.* [100] presented a new 3-D geometry PCC method based on an auto-encoder network. They used the extracted features of the raw model in the encoder to compress the original data to be bitstream. They further compressed the bitstream with sparse coding. Nevertheless, this method worked on the well-segmented objects.

To reduce the bitrate, Huang *et al.* [101] proposed a deep compression method to decrease the memory footprint of LiDAR point clouds with the sparsity and structural redundancy between points. Nevertheless, the spatio-temporal relationships had not been fully considered. To reduce the bitrate of both geometry and intensity values, Biswas *et al.* [102] exploited spatio-temporal relationships across multiple LiDAR sweeps and proposed a conditional entropy model. This models the probabilities of the octree symbols by utilizing both coarse level geometry, previous sweeps' geometric, and intensity information.

2.2.2 Geometry related methods in V-PCC

MPEG calls some geometry-related works to improve the point cloud quality and compression efficiency during the V-PCC standardization. Guede *et al.* [79] proposed encoding the projected information as an absolute depth value instead of an error between the far layer and near layer. However, this solution can not improve the quality of the point-cloud reconstruction. To obtain a better tradeoff between these artifacts and bitrate, Olivier *et al.* [103] proposed improving the projection of connected components into patches. Nevertheless, it did not resolve the empty space between patches, which directly impacted the coding efficiency. In order to take care of the empty space, Rhyu *et al.* [81] proposed dilating the gap between patches by expanding the geometry from boundaries of patches. Although this way minimized block artifacts in the decoded 2-D video, it still compressed the near and far geometry frames separately. To get better encoding efficiency, Dawar *et al.* [80] proposed using one frame instead of two frames to encode 2-D near and far layers. However, this method interpolated pixels from spatial neighbors leading to distortions to some extent and did not save decoding time.

To keep the reconstruction quality but decrease the decoding time complexity, Nakagami *et al.* [9] proposed an upgraded geometry smoothing. The geometry smoothing of V-PCC aims at alleviating potential discontinuities that may arise at the patch boundaries due to compression artifacts. This proposed approach moved boundary points to the centroid of their nearest neighbors. It skipped the smoothing for points inside a patch and pre-calculated the neighbor points centroid instead of the nearest neighbor (NN) search.

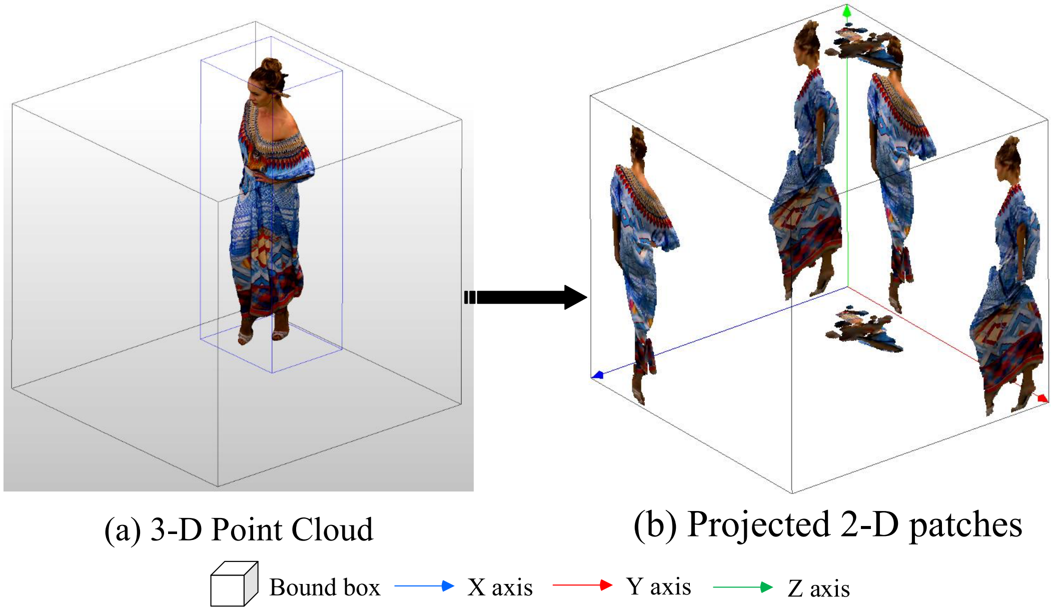


Figure 8: The projection process [1] from 3-D to 2-D in V-PCC. The segmented 3-D patches are projected to six planes of its bounding box.

Nevertheless, this approach did not exploit the geometry information of the 3-D reconstruction.

Graziosi *et al.* [8] therefore proposed padding the geometry with the reconstructed depth value for those positions introduced by occupancy map rescaling. To decrease the distance between the geometry reconstruction and origin, this encoder-only method searched and selected a depth value from a range of possible geometry depth values for an inserted point. This method only used some limited neighborhoods and geometry characteristics. The color consistency and surface consistency of the lines in the reconstruction could be beneficial. In addition, this method could still be improved further by occupancy

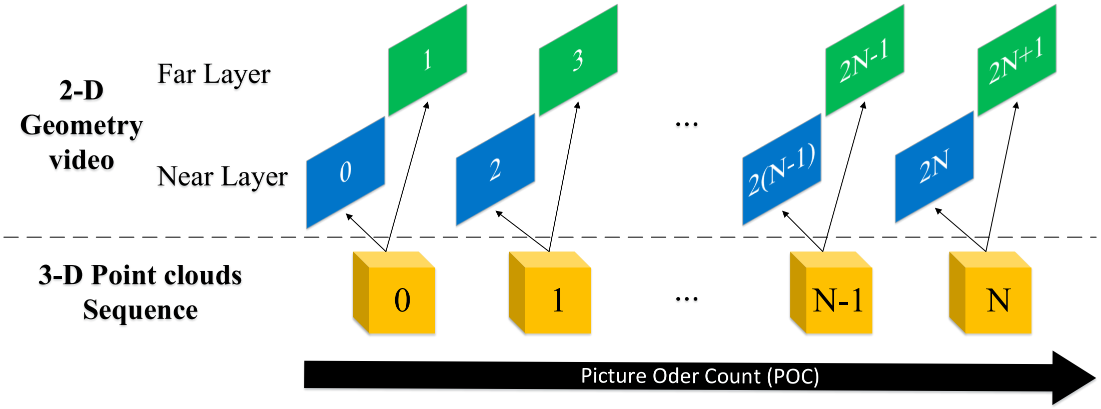


Figure 9: The correlations between 3-D point clouds sequence and 2-D geometry video. A 2-D patch with picture order count (POC) $2N$ of near layer and one with $2N + 1$ of far layer are both derived from the same 3-D point cloud patch with POC N.

map reconstruction methods.

Although V-PCC adopted these two works into the reference software due to their good performances, they did not fully exploit the strong correlations and interactions between the geometry near and far layers. In addition, the non-linear representation ability of CNN has not been considered carefully in the V-PCC geometry distortion removal.

2.3 Geometry in V-PCC

To explain the importance and necessity of the 2-D geometry improvement in V-PCC, we need to figure out the property of the geometry step by step. We first elaborate on how V-PCC converts the 3-D point cloud to the 2-D near and far layers of geometry frames in the projecting process. Then we state the strong correlations between near and far layers of the geometry. We explain all these above in Section 2.3.1. To better understand that how geometry propagates its impact on point-cloud reconstruction, we previously introduce the important role that geometry plays in the point cloud reconstructing process in Section 2.3.2. Finally, we explain the impact of 2-D geometry on 3-D point-cloud reconstruction in both objective and subjective performance in Section 2.3.3.

2.3.1 The projection from 3-D to 2-D

A point-cloud frame consists of a collection of points within a 3-D volumetric space that is with its coordinates, geometry, and attributes information, as shown in Fig. 8 (a). To use the proven powerful 2-D codecs such as HEVC and VCC, V-PCC first projects the volumetric 3-D point-cloud frames to 2-D video frames. The whole projection process contains three stages: patch segmentation, patch generation, and patch packing. The V-PCC patch denotes a set of information that describes the point cloud in a 3-D bounding box. This information incorporates points, corresponding geometry, and attribute descriptions along with the atlas information.

The patch segmentation aims to decompose the 3-D point clouds into many patches. Then during the process of patch generation, as shown in Fig. 8, V-PCC projects the segmented 3-D patches to six planes of its bounding box. To appropriately resolve the problem that different 3-D points are projected onto the identical 2-D pixel, V-PCC projects each patch onto two depth fields. More specifically, we assume that $P(u, v)$ is the collection of points of a patch projected to the identical sample (u, v). The near layer stashes the point of $P(u, v)$ with the lowest depth d_0. The far layer projects the point of $P(u, v)$ with

Table 11: 2-D and 3-D geometry PSNR comparison between lower and higher bitrates of V-PCC Anchor [7] within soldier first 32 frames

PSNR (dB)	Metrics	$r1$ (lower)	$r2$ (higher)
2-D Geometry video	Near Layer	46.8062	**49.1117**
	Far Layer	46.4561	**48.7152**
	Avg. All	46.6312	**48.9134**
3-D point clouds	Point-to-Point Error	65.66	**67.45**
	Point-to-Plane Error	67.42	**69.48**

the highest depth within $[d_0, d_0 + \delta]$, where δ denotes the thickness of the projected surface. Intrinsically, as illustrated in Fig. 9, a 2-D near layer with picture order count (POC) $2N$ and a far layer with POC $2N + 1$ are both derived from the same 3-D point-cloud frame with POC N. Fig. 10 visualizes an example of the near layer frame with POC $2N$, corresponding POC $2N + 1$ far layer frame and their difference. Since generally, the δ is a minimal value, it is difficult to directly see the difference between the near and far layer frames unless we make subtraction with them as Fig. 10 (c). Afterward, to generate the 2-D geometry and attributes videos, V-PCC arranges the projected 2-D patches compactly onto a 2-D frame with size $W \times H$. We call this stage patch packing. Once we obtain 2-D geometry videos, codecs can compress them efficiently.

2.3.2 The reconstruction process

Before reconstructing point clouds, the V-PCC decoder first demultiplexes the compressed bitstream into geometry, attributes, occupancy map, and atlas streams. The atlas mainly contains auxiliary patch information. The occupancy map is a binary signal video implicating whether a 2-D pixel exists in the original 3-D point cloud as a point. Once the reconstructing process starts, as described in Fig. 11, V-PCC reconstructs the atlas first. We can see from Fig. 11 (a) that V-PCC only builds padded patch rough sketches.

After V-PCC merges the decoded occupancy map into the reconstruction, the occupancy status of the points becomes clear because the occupancy map removes the padded area.

However, there is still no clear person shape information. When V-PCC adds the geometry information into the reconstruction, all 3-D point clouds are almost created except the color attributes. Finally, V-PCC draws the attributes on the reconstruction.

2.3.3 The impact of the 2-D geometry to the 3-D point cloud

The visualizing reconstruction process above has demonstrated that how the 2-D geometry enormously impacts the subjective quality of 3-D point clouds reconstruction. Furthermore, Fig. 12 shows that how the V-PCC propagates the 2-D geometry artifacts significantly to the 3-D point-cloud reconstruction. The sub-figures of the first row (a), (b), and (c) are the 2-D geometry, attributes, and 3-D point cloud of ground truths, respectively. With the same arrangement, the sub-figures of the second row are from anchor reconstructions. Because it is difficult to directly recognize the difference of the geometry value between the anchor and ground truth with our eyes, we visualize their difference in (d). We can clearly see the geometry value distortion of the gun in the enlarged area of (d). As explained in Section 2.3.2, the V-PCC reconstructs the 2-D attributes video with a 2-D reconstructed geometry video, containing useful pixel location information. The V-PCC finally reconstructs the 3-D point clouds with recolored and smoothed attributes video in 3-D space. From the enlarged areas, we can clearly see that the V-PCC propagates the 2-D geometry distortion to attributes reconstruction. Then, the V-PCC brings the attributes artifacts into the 3-D point-cloud reconstruction.

On the aspect of objective influence, we can find the same importance of geometry video in Table 11. The higher bitrate $r2$ outperforms the lower one $r1$ on the PSNR of 2-D geometry video on either near or far layer. These gains are obviously propagated to 3-D point clouds on either point-to-point or point-to-plane [104] PSNR. These altogether demonstrate the consistency of objective geometry qualities on both 2-D and 3-D sides. Based on the analysis and observations above, if an efficient algorithm improving 2-D geometry can be carefully devised, we can expect similar ideal performance on the 3-D point cloud.

2.4 Two-step geometry artifacts removal

Based on the observations and analysis above in Section 2.3, the 2-D geometry impacts the 3-D point-cloud reconstruction significantly. Therefore, to remove artifacts of the near and far layers in geometry, we develop an algorithm with a two-step strategy. In the first step, we not only improve the near and far layers with individual CNNs, but also use the enhanced near reconstruction as the Pseudo-Motion Compensation (PMC) for augmenting the far layer. In the second step, we dive into the interactions between near and far layers and devise an X-like interacting network (XInteractNet) to fully use their strong similarities for further enhancement. Specifically, we elaborate an in-depth discussion on the two-step scheme in Section 2.4.1, design of XInteractNet in Section 2.4.2, loss function in Section 3.4.4, dataset in Section 2.4.4, and training process in Section 3.4.5.

Algorithm 1 The flow of two-step approach

```
Input: The near depth field reconstruction $r_0$, the far depth field $g_{p1}$.
Output: The artifacts removed near depth field $r_0^{''}$ and far depth field $r_1^{''}$.

if Step one initialization successes then
    Input $r_0$ into the step one $S_0(\cdot)$ CNN and output the augmented near reconstruction $r_0^{'}$;
    Input $g_{p1}$ and Pseudo-Motion Compensation $r_0^{'}$ into the predictor and output the far reconstruction $r_1$;
    Input $r_1$ into the step one $S_1(\cdot)$ CNN and output the augmented far reconstruction $r_1^{'}$;
end
if Step two initialization successes then
    Input $r_0^{'}$ and $r_1^{'}$ into the step two XInteractNet;
    Compute the down features $x_0^d$ and $x_1^d$ with X Down Block $\Phi^d(x_0(i), x_1(i))$;
    Compute the up features $y_0^u$ and $y_1^u$ with X Up Block $\Phi^u(y_0(i), y_1(i))$;
    Output the artifacts removed near depth field $r_0^{''}$ and far depth field $r_1^{''}$;
end
```

2.4.1 Artifacts removal of geometry with two steps

We carefully devise our algorithm on two aspects. On the one hand, as observed in Section 2.3.2 and Section 2.3.3, the geometry plays an important role in the 3-D point-cloud reconstruction. We, therefore, focus on removing artifacts of geometry to improve the quality of 3-D point clouds further. On the other hand, as elaborated in Section 2.3.1, in the projection process of geometry, V-PCC projects a 3-D point-cloud frame to two 2-D different layer frames, including a near one denoted as g_0 with depth d_0 and a far one denoted as g_1 with depth d_1. Since we essentially acquire these two 2-D frames from the same 3-D point-cloud frame, we consider utilizing their similarities and interactive information $I_s(g_0, g_1)$ as much as possible for denoising geometry. Hence, we propose a

CNN-based two-step method that uses the interaction $I_s(g_0, g_1)$ to enhance near layer g_0 and far layer g_1 as frequently as possible.

Fig. 13 describes the upgraded V-PCC encoding scheme embedded in the proposed two-step approach. Initially, the upgraded V-PCC uses the 3-D input of point clouds $P(i)$ to generate patch information $T(j)$. V-PCC then generates the occupancy map $O(k)$ with these patches information $T(j)$ during the patch packing process. Afterwards, V-PCC generates the 2-D near layer g_0 and far layer g_1 from the 3-D input $P(i)$, patch information $T(j)$ and occupancy map $O(k)$ for further padding. V-PCC then pads far layer g_0 and near layer g_1 to generate corresponding g_{p0} and g_{p1} that are beneficial for predicting, transforming and quantizing.

The target of the first step is to denoise the coarse artifacts of the reconstructed near layer r_0 and far layer r_1 first. This way can provide step two with reconstructions of better quality as dual inputs. The inter prediction in HEVC adopts motion compensation technologies. The principle of motion compensation in codec is to search out a reference frame containing reference blocks for predicting the current frame. Since we replace the reference frame with an enhanced near layer reconstruction in the upgraded codec for predicting the far layer reconstruction, we call this process as PMC. Specifically, at first, we feed r_0 into a CNN $S(\cdot)$ (near CNN) that is an autoencoder structure with four convolutional layers. Then the augmented near layer reconstruction $r_0^{'}$ as PMC iteratively participates into the padded far layer g_{p1} prediction to generate the far layer reconstruction r_1. We embed the algorithm of step one into the V-PCC and HECV encoders, configuring the near layer reconstruction $r_0^{'}$ as an I frame while the geometry far layer reconstruction

r_1 as a P frame. The overhauled codec sets the near layer $r_0^{'}$ (I frame) as a reference frame for predicting the far layer reconstruction r_1 (P frame). The updated encoder estimates and compensates the motions for the far layer reconstruction r_1 with the enhanced near layer reference frame $r_0^{'}$. Accordingly, due to this PMC, if the quality of the near layer reference frame $r_0^{'}$ is better, the quality of the far layer reconstruction r_1 will be better. Afterward, we feed the far reconstruction r_1 into the far layer step one CNN (far CNN), which has the same architecture as near CNN, to produce a better reconstruction $r_1^{'}$.

The objective of the second step is to utilize the interactive information $I_s(r_0^{'}, r_1^{'})$ between near layer $r_0^{'}$ and far layer $r_1^{'}$ in full extent for further removing artifacts. Accordingly, we devise a network with two inputs and two outputs, namely XInteractNet mining the interactive information $I_s(r_0^{'}, r_1^{'})$ as much as possible. We input the outputs of step one including $r_0^{'}$ and $r_1^{'}$ into XInteractNet to achieve enhanced corresponding near layer reconstruction $r_0^{''}$ and far layer reconstruction $r_1^{''}$. We finally use $r_0^{''}$ and $r_1^{''}$ for reconstructing the 3-D point cloud. We elaborate the proposed two-step artifacts removal algorithm of near and far depth fields in Algorithm 1.

2.4.2 XInteractNet

The architecture of the proposed XInteractNet in the second step is illustrated in Fig. 14. Let us define $x_0(i)$ and $x_1(i)$ are the near and far layer feature map of No. i level of the XInteractNet, respectively. In order to share the interactive information $I_s(x_0(i), x_1(i))$ between near and far layers as much as possible, we devise a network model pair called X interactive down and up blocks as the basic unit of XInteractNet.

According to the observations above in Section 2.3, for a given near and far layer pairs, there exists a high correlation between the near and far layer feature maps at every level in the network. Therefore, the main idea behind the design of the proposed x interactive down and up blocks is sharing the interactive information $I_s(x_0(i), x_1(i))$ after every convolutional computation level.

Specifically, in the X down block, to share the interactive information $I_d(x_0(i), x_1(i))$ at level i before the max-pooling operation, we merge the previous convolutional near layer's feature $x_0(i-1)$ and far layer's feature $x_1(i-1)$ into $x(i-1)$ first. Then we fed $x(i-1)$ into the max-pooling model $f(x(i-1); w)$ to decrease its size from $H \times W$ to $(H/2) \times (W/2)$ while aggregate more features from C channels to $2 \times C$ channels. The No. $i+1$ convolutional near and far layers take the pooled $x(i)$ feature as input to tackle the next convolution computation. We set the kernel size to 3×3, stride to 1, padding to 1 for all convolutional layers of X down block. Formally, the X interactive down block is able to be concluded as

$$\Psi^d\left(x_0(i), x_1(i)\right) = C(\Phi_0^d * x_0\left(i\right), \Phi_1^d * x_1\left(i\right)) \tag{2.1}$$

where $x_0(i)$ and $x_1(i)$ are the near and far features input that stands on the No. i layer. Φ_0^d and Φ_1^d are denoted as the the near and far layers model parameters including corresponding weights and bias matrices, respectively. C denotes the concatenation function.

Similarly, the X up block shares the interactive information $I_u(y_0(j), y_1(j))$ at level j before the transposed-convolution operation. The previous convolutional near layer's feature $y_0(j-1)$ and far layer's feature $y_1(j-1)$ share interactions for each other and we merge them into $y(j-1)$. Afterward, the No. j transposed-convolutional

Table 12: The convolutional and transposed convolutional layers parameters of the first X Up Block in XInteractNet

Layer	1	2	3
Level	Near Conv Far Conv	Near Transposed Conv Far Transposed Conv	Near Conv Far Conv
Kernel Size	3×3 3×3	2×2 2×2	3×3 3×3
Feature Map Number	64 64	64 64	32 32
Stride	1 1	2 2	1 1
Padding	1 1	0 0	1 1

layer takes it as input to recover its size from $(H/2) \times (W/2)$ back to $H \times W$ while decreasing the feature channels from $2C$ to C. Then we concatenate the achieved $y_0(j)$ and $y_1(j)$ features again and feed it into the next NO. $j+1$ convolutional near and far layers for computation. Table 12 shows the convolutional layer parameters of X up block. Mathematically, the X interactive up block is able to be represented by

$$\Psi^u\left(y_0(j), y_1(j)\right) = C(\Phi_0^u * y_0\left(j\right), \Phi_1^u * y_1\left(j\right)) \tag{2.2}$$

where $y_0(j)$ and $y_1(j)$ are the near and far features input that stands on the No. j layer. Φ_0^u and Φ_1^u denote the the near and far layers model parameters including corresponding weights and bias matrices, respectively. C denote the concatenation function.

2.4.3 Interactive loss function

To properly fit the second step of XInteractNet, we devise a loss function called Interactive loss function to measure the XInteractNet. During the design process of the interactive loss function, we have two primary considerations.

First, as a dual inputs and dual outputs supervised network, XInteractNet should

Table 13: Proposed two-step method and geometry padding method [8] against the geometry smoothing method [9] respectively on BD-rate and time complexity within the first 32 frames of sequences under all intra

Class	Sequence	V-PCC with geometry padding method (SOTA) [8]					Proposed two-step method				
		Geom.BD-Totalrate		Attr.BD-Totalrate			Geom.BD-Totalrate		Attr.BD-Totalrate		
		D1 ↓	D2 ↓	Luma ↓	Cb ↓	Cr ↓	D1 ↓	D2 ↓	Luma ↓	Cb ↓	Cr ↓
A	Loot	−4.4%	−9.9%	4.4%	4.8%	5.4%	**-20.8**%	**-18.2**%	−1.2%	−1.4%	−0.5%
	Redandblack	−0.8%	−7.5%	4.4%	5.9%	5.0%	**-10.1**%	**-11.3**%	−1.1%	−0.9%	−2.4%
	Soldier	−1.5%	−7.8%	4.4%	7.3%	6.4%	**-13.8**%	**-12.8**%	−1.9%	0.1%	−0.8%
B	Longdress	−1.3%	−8.2%	2.3%	3.5%	3.2%	**-12.8**%	**-13.7**%	−2.9%	−1.4%	−2.3%
	Class a	−2.2%	−8.4%	4.4%	6.0%	5.6%	**-14.9**%	**-14.1**%	−1.4%	−0.7%	−1.2%
	class b	−1.3%	−8.2%	2.3%	3.5%	3.2%	**-12.8**%	**-13.7**%	−2.9%	−1.4%	−2.3%
Avg.	All	−2.0%	−8.3%	3.9%	5.4%	5.0%	**-14.4**%	**-14.0**%	−1.8%	−0.9%	−1.5%
	Enc.Self	102%					103%				
	Enc.Child	101%					101%				
	Dec.Self	102%					121%				
	Dec.Child	101%					212%				

be capable of recovering the near and far depth fields to be close to their original ones on the pixel level, respectively. We, therefore, define the near Mean Square Error (MSE):

$$L_0(\Theta_0) = \frac{1}{N}\sum_{i=1}^{N} ||\Upsilon_0(r_0^{''}(i)|\Theta_0) - g_0(i)||_2^2 \tag{2.3}$$

where Θ_0 encapsulates the whole near depth field parameter set of the XInteractNet, that contains weights and bias. $\Upsilon_0(r_0^{''}(i)|\Theta_0)$ is denoted as corresponding near depth field modules in the XInteractNet that output $r_0^{''}$. As explained in Section 2.4.2, $g_0(i)$ is the original near depth field, where i indexes each of them. $r_0^{''}(i)$ is the near depth field output of the XInteractNet. N is the number of frames. Similarly, the far MSE is defined as

$$L_1(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N} ||\Upsilon_1(r_1^{''}(i)|\Theta_1) - g_1(i)||_2^2 \tag{2.4}$$

Hence, the sum of near MSE $L_0(\Theta_0)$ and far MSE $L_1(\Theta_1)$ can be defined as Dual MSE

(DMSE)

$$L_D(\Theta_0, \Theta_1) = L_0(\Theta_0) + L_1(\Theta_1) \tag{2.5}$$

Second, as explained in Section 2.4.2 above, the XInteractNet aims to utilize the interactive information $I_s(r_0, r_1)$ between the near and far depth fields as much as possible. Relying on the interactive information, the XInteractNet should be able to handle two types of depth field problems well. Specifically, one non-occlusive case is that the original near depth field is the same as the original far depth field. V-PCC essentially converts the same 3-D point to the original near and far pixel with the same value. The XInteractNet should recognize this non-occlusive case and do its best to assimilate them with their $I_s(g_0, g_1)$. In this way, the XInteractNet can be helpful to reconstruct only one point in 3-D space correctly. In the other occlusive case, the correlation between near and far depth fields is quite weak, and this means that the near depth field is quite different from the far depth field. The XInteractNet should then learn from $I_s(g_0, g_1)$ to discriminate them and help reconstruct two different points in 3-D space. Based on the considerations above, we need to design a special term in the loss function that can recover the interactive status of $I_s(r_0, r_1)$ to be as close as possible to the original interactive status $I_s(g_0, g_1)$. We, therefore, devise an interactive term

$$L_{IT}(r_0^{''}, r_1^{''}, g_0, g_1) = \lambda L_{s1}(|r_0^{''} - r_1^{''}|, |g_0 - g_1|) \tag{2.6}$$

where L_{s1} denotes smooth L one loss function. $r_0^{''}$ and $r_1^{''}$ are the near and far depth fields outputs of XInteractNet while g_0 and g_1 are their corresponding origins. λ is a tuning coefficient and is set to 0.001 by default. This interactive term pushes the XInteractNet to learn from interactive information so that it could squeeze the difference between near and far reconstructions and the difference of their labels.

We finally propose the interactive loss function:

$$\begin{aligned} L_I((\Theta_0, \Theta_1); (r_0^{''}, r_1^{''}, g_0, g_1)) = L_D(\Theta_0, \Theta_1) + \\ L_{IT}(r_0^{''}, r_1^{''}, g_0, g_1) \end{aligned} \tag{2.7}$$

The interactive loss function $L_I((\Theta_0, \Theta_1); (r_0^{''}, r_1^{''}, g_0, g_1))$ effectively constrain the XInteractNet to learn from the interactive information $I_s(g_0, g_1)$ for correctly recognizing the correlation between the near and far depth fields.

2.4.4 Dataset

We adopt the Common Test Conditions(CTC) [105] consisting of dynamic point cloud sequences recommended by MPEG for training, validating, and testing. $8i$ captured and collected these raw 3-D point cloud sequences. For our two-step algorithm, to train and validate our near and far $S(\cdot)$ CNNs of step one and XInteractNet of step two models, we use Queen sequence. We test the proposed models with the other four sequences containing Loot, RedandBlack, Soldier, and Longdress, as shown in Section 2.5. On step one, V-PCC generates 250 near depth field and 250 far depth field origins and reconstructions of Queen. Among these data, for both near and far depth fields, we use 192 frames for training while 58 frames for validating. Then, we extract the 64×64 Coding Tree Units (CTU) from the luminance component of the generated near and far depth field of origins and reconstructions. Finally, we generate a total of $76,800$ near and far depth field frames for training, and $23,200$ frames for validating our $S(\cdot)$ CNNs. In step two, we use

Table 14: Training parameters

Parameters	Value
Base Learning Rate	$1e^{-4}$
γ Adjusting Coefficient	0.1
Adjusting Epochs Interval	50
Weight Decay	$1e^{-4}$
Momentum	0.9
Total Epochs	60

Table 15: Proposed two-step method and one-step method of near and far CNNs against the one-step method of mixed geometry respectively on BD-rate and time complexity within the first 32 frames of sequences under the all intra case

Class	Sequence	One-step method of near and far CNNs					Proposed two-step method				
		Geom.BD-Totalrate		Attr.BD-Totalrate			Geom.BD-Totalrate		Attr.BD-Totalrate		
		D1 ↓	D2 ↓	Luma ↓	Cb ↓	Cr ↓	D1 ↓	D2 ↓	Luma ↓	Cb ↓	Cr ↓
A	Loot	−4.4%	−1.9%	0.0%	1.1%	0.6%	**-12.2**%	**-8.0**%	−0.3%	−0.1%	−1.0%
	Redandblack	−3.3%	−2.1%	−0.1%	−0.2%	−0.1%	**-6.1**%	**-4.9**%	−0.4%	−1.1%	−1.2%
	Soldier	−3.8%	−1.5%	−0.1%	0.4%	0.0%	**-8.1**%	**-4.8**%	−0.8%	−0.7%	−1.3%
b	Longdress	−4.2%	−1.8%	−0.2%	0.1%	0.2%	**-5.9**%	**-3.9**%	−1.2%	−1.0%	−1.1%
	Class A	−3.9%	−1.8%	−0.1%	0.4%	0.1%	**-8.8**%	**-5.9**%	−0.5%	−0.6%	−1.1%
	Class B	−4.2%	−1.8%	−0.2%	0.1%	0.2%	**-5.9**%	**-3.9**%	−1.2%	−1.0%	−1.1%
Avg.	All	−3.9%	−1.8%	−0.1%	0.3%	0.2%	**-8.1**%	**-5.4**%	−0.7%	−0.7%	−1.1%
	Enc.Self	100%					100%				
	Enc.Child	103%					103%				
	Dec.Self	108%					118%				
	Dec.Child	101%					113%				

the V-PCC embedded step one models, instead of anchor V-PCC reference software, to generate both training and validating data. The other data preparation process is the same as the step one generation process. The amount of training or validating CTUs is the same as the one of step one as well.

2.4.5 Training

To train the first step $S(\cdot)$ CNNs, we feed the near and far depth field reconstruction CTUs obtained from V-PCC anchor into the near and far $S(\cdot)$ CNNs, respectively. The corresponding near and far origins generated from the anchor supervise the training procedure. During training the XInteractNet of the second step, we feed the generated reconstruction CTUs of near and far depth fields from step one $S(\cdot)$ CNNs into XInteractNet by batch-size of 16, respectively. The corresponding near and far origins generated from $S(\cdot)$ CNNs supervise the XInteractNet training procedure. Table 19 shows the parameters of the XInteractNet training process. Once the interactive loss is convergent, the training state is considered as completion. According to our experiments and observations, the loss is convergent before 60 epochs so we set the total XInteractNet training epochs to 60. Additionally, we set the base learning rate to $1e^{-4}$. We degrade the learning rate by multiplying γ of 0.1 after each interval of 50 epochs. In fact, γ just means the learning rate adjusting coefficient. We apply the Adaptive Moment Estimation (Adam) [59] algorithm as the gradient optimizer. We set Adam momentum to 0.9 and the weight decay to $1e^{-4}$. We use these hyper-parameters for the first step $S(\cdot)$ training as well.

2.5 Experimental results

To evaluate the performance of the proposed approaches, we implement our proposed two-step and one-step algorithms into the V-PCC and HEVC reference software. As explained in Section 2.4, the two-step method represents the proposed geometry artifacts removal approach in two stages, including step one with near and far CNNs and step

Table 16: Proposed two-step method against the geometry smoothing [9] on BD-rate and Time complexity under the random access case

Sequence	Geom.BD-TotalRate		Attr.BD-TotalRate		
	D1 ↓	D2 ↓	Luma ↓	Cb ↓	Cr ↓
A.Loot	**-20.2**%	**-19.2**%	−6.0%	−3.8%	−5.9%
A.Red&black	**-10.3**%	**-11.5**%	−2.0%	−1.2%	−2.2%
A.Soldier	**-6.1**%	**-6.6**%	−5.0%	−1.6%	−1.9%
B.Longdress	**-12.1**%	**-14.1**%	−3.9%	−2.4%	−2.8%
Class A	**-12.2**%	**-12.4**%	−4.3%	−2.2%	−3.3%
Class B	**-12.1**%	**-14.1**%	−3.9%	−2.4%	−2.8%
Avg. All	**-12.2**%	**-12.8**%	−4.2%	−2.3%	−3.2%
Enc.Self	103%				
Enc.Child	100%				
Dec.Self	119%				
Dec.Child	211%				

two using our designed XInteractNet. The one-step method of near and far CNNs means it executes one stage utilizing the near CNN and far CNN for corresponding depth fields. The one-step method of mixed geometry is inputted with a mixture of near and far depth fields. In this section, we compare the V-PCC geometry smoothing method [9], state-of-the-art(SOTA), namely geometry padding method [8], the one-step methods above, and the proposed two-step method. On the aspect of test data, we experiment within the first 32 frames of four V-PCC CTC [105] sequences, as mentioned in Section 3.4.5.

2.5.1 Comparison with SOTA under all intra

As Table 13 shown, we compare the proposed two-step method and SOTA geometry padding [8] against the V-PCC geometry smoothing [9] under all intra case within the first 32 frames of four CTC sequences [105]. Note that, to be fair, we set both the proposed two-step method and the other two V-PCC methods to 8 bits for the geometry encoder. We experiment with all these methods under four level bitrate settings. On point-to-plane D2 Geom.BD-TotalRate, we can see that the proposed two-step approach outperforms V-PCC geometry smoothing by an average of -14.0% while the SOTA geometry padding method gains -8.3% on this metric. In addition, the proposed two-step method surpasses SOTA and geometry smoothing methods in every sequence on point-to-plane D2. Specifically, the proposed two-step approach performs better than SOTA by -5.7% and -5.5% related to Class A and B on point-to-plane D2, respectively. Especially, the peak difference even reaches -8.3% on Loot of Class A.

Compared to the SOTA geometry padding method, the proposed two-step method gains an average of -12.4% on point-to-point D1 Geom.BD-TotalRate. We can see that the proposed two-step method outperforms SOTA and geometry smoothing methods in all sequences on point-to-point D1. Specifically, the proposed two-step method surpasses SOTA on Class A and B correspondingly by -12.7% and -11.5% at the point-to-point error D1. In addition, the top difference climbs to -16.4% on Loot of Class A.

Two reasons lead to these comparison results. Both geometry smoothing [9] and geometry padding [8] do not exploit the strong correlations and interactions between the geometry near and far layers. In addition, the non-linear representation ability of CNN has not been considered in these two geometry distortion removal methods. All these above implicate that our proposed two-step mechanism and designed XInteractNet achieves a clear improvement on the problem of geometry artifacts removal. Besides, the design of XInteractNet effectively exploiting the strong correlations between near and far depth fields is beneficial for the enhancement of geometry.

As illustrated in Fig. 22, we count the number of points n_r of the reconstructed

3-D point cloud within the first 32 frames of four CTC sequences. From level $r1$ to level $r4$, the bitrate labeled on the Y axis gradually increases. Our proposed algorithm aims to remove the artifacts and restore the points that initially exist in ground truth as much as possible. We can obviously see that the number of points n_r of our proposed two-step method is more than the one of SOTA, geometry smoothing. These statistics fully demonstrate that the proposed two-step methods efficiently restore the original points in the 3-D point-cloud reconstruction to improve its quality.

Fig. 23 shows the comparison of Geometry Rate-Distortion (RD) curves in all the intra cases on Loot sequence. As shown, the point-to-plane D2 PSNRs of all 4 bitrate settings in the proposed two-step method is higher than the ones of the SOTA and geometry smoothing method. Similarly, the point-to-point D1 PSNRs of $r1$, $r2$, and $r3$ bitrate levels in the proposed two-step method are higher than those of the SOTA and geometry smoothing method as well. These results significantly prove that the proposed two-step method is superior to the SOTA and geometry smoothing on removing geometry artifacts and improving the PCC quality.

The time complexity [62] is shown in Table 13 as well. For all methods, we configure the same environments. Specifically, the CPU configuration is Intel core i5-8400 CPU @ 2.80GHz, and the GPU configuration is GTX 1080ti. The 'Enc.Self' represents the encoder side of V-PCC, and 'Enc.Child' means the encoder side of HEVC. Similarly, the 'Dec.Self' represents the decoder side of V-PCC, and 'Dec.Child' means the decoder side of HEVC. Under the all intra case, on the encoder side of V-PCC, the proposed two-step method takes 103% time of geometry smoothing method. On the decoder side of

V-PCC, it takes 121% time of geometry smoothing method. Their time complexities are similar.

2.5.2 Comparison with one-step methods

Table 15 shows the BD-rate and time complexity comparison between the proposed two-step method and the one-step method of near and far CNNs all against the one-step method inputted into mixed geometry within the first 32 frames of four CTC sequences. The proposed two-step method outperforms the one-step method of near and far CNNs by an average of -4.2% and -3.6% on point-to-point error D1 and point-to-plane D2, respectively. Meanwhile, the proposed two-step method performs better than the one-step method inputted into mixed geometry -8.1% and -5.4% on point-to-point error D1 and point-to-plane error D2, respectively. These performances significantly demonstrate that the proposed two-step method outperforms one-step methods on objective qualities and PCC coding efficiency. These results also prove that XInteractNet in the second step explores the similarities between the near layer and far layer effectively. In addition, thanks to the architecture design with X similarity down and up blocks, XInteractNet could mine the interactions between the near layer and far layer in a full extent.

Regarding time complexity, the proposed two-step method takes almost the same time as the geometry smoothing method on the encoder side of V-PCC and HEVC. Meanwhile, on the decoder side of V-PCC and HEVC, it takes 118% and 113% time of geometry smoothing method, respectively.

As exhibited in Fig. 22, we compare the number of points n_r in the reconstructed

3-D point cloud between the proposed two-step method and one-step methods within the first 32 frames of four CTC sequences as well. We can see that the number of points n_r of the proposed two-step method is more than either the one of the one-step method of near and far CNNs or the one of the one-step method inputted into mix geometry in all sequences. These statistics fully prove that the proposed two-step method significantly performs better than other one-step methods on restoring points for artifact removal. Additionally, Fig. 23 shows the RD curves comparison between the proposed two-step method and the other one-step methods. The proposed two-step method leads the higher PSNRs of point-to-point error D1 and points to plane error D2 than other one-step methods.

2.5.3 Performances of the proposed two-step algorithm under random access case

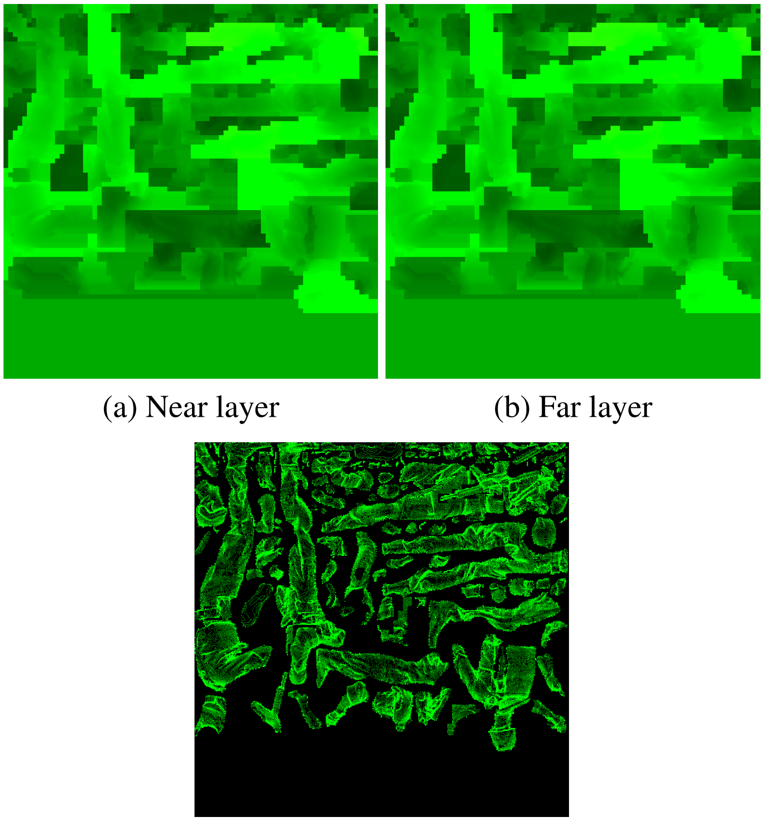
As shown in Table 16, in the random access case, we can see that compared to the V-PCC geometry smoothing method, the proposed two-step method gains an average of -12.2% and -12.8% on Geom.BD-TotalRate point-to-point error D1 and point-to-plane error D2, respectively. Again, we can also see that the proposed two-step method leads the geometry smoothing method on point-to-point error D1 and point-to-plane error D2 in every class. Additionally, the top difference between the two-step method and geometry smoothing method climbs to -20.2% and -19.2% at Loot on point-to-point error D1 and point-to-plane D2, respectively. This comparison fully proves that the benefits brought by the proposed two-step methods can be similarly propagated to random access case.

2.5.4 Subjective results

Fig. 25 shows the visual comparisons of ground truth, point-cloud reconstructions of geometry smoothing, geometry padding (SOTA), and the proposed two-step method. These figures are from sequences of Loot, RedandBlack, and Longdress. We generate the figures (a), (b), (c), and (d) from the point cloud ground truth, the point-cloud reconstructions of geometry smoothing, geometry padding (SOTA), and the proposed two-step methods, respectively. From these sequences, we can clearly see from the zoomed red and green blocks that there are apparent distortions in the reconstructions of geometry smoothing, geometry padding. However, the reconstructions of the proposed two-step method exhibit better visual qualities. For instance, the green rectangles of Loot and Longdress and red ones of Redandblack and Longdress show significantly smoother boundary edges processed by the proposed two-step method. Meanwhile, the green rectangle of Redandblack and the red one of Loot exhibits obviously more abundant graph information as well. The subjective results demonstrate that compared to the geometry smoothing and geometry padding (SOTA) methods, the proposed two-step approach could significantly bring better visual qualities.

2.6 Summary

The geometry video intrinsically represents the depth fields of 3-D point clouds. Once there are artifacts on the compressed 2-D geometry video, they would be propagated to the 3-D point-cloud frames. In the lossy compression, there always exists a tradeoff between the rate of bitstream and distortion. This paper proposes a learning-based approach to remove the geometry artifacts and improve the compressing efficiency. We devise a two-step method working on the near and far depth fields decomposed from geometry. The first stage is learning-based Pseudo-Motion Compensation. The second stage exploits the potential of the strong correlations between near and far depth fields. We embed the proposed algorithm into the V-PCC reference software. To the best of our knowledge, this is the first learning-based solution of the geometry artifacts removal in V-PCC. The extensive experimental results show that the proposed approach achieves significant gains on geometry artifacts removal and quality improvement of 3-D point-cloud reconstruction compared to state-of-the-art schemes. In the future, we can still exploit the temporal relationships of the 2-D geometry and 3-D point-cloud frames to assist the artifact removal further. The multi-frames-based method is another reasonable potential solution utilizing the learned features for motion compensation. In addition, we consider using sparse convolution for PCC, which is a promising direction as well.



(a) Near layer

(b) Far layer

(c) The difference of near and far layer

Figure 10: Near (a) and far (b) layer frames in 2-D geometry video. (C) is the difference δ between the near layer frame with POC $2N$ and its corresponding POC $2N+1$ far layer frame. Since generally, the difference δ is a minimal value, it is difficult to directly see the difference between the near and far layer frames unless making subtraction with them as (c).

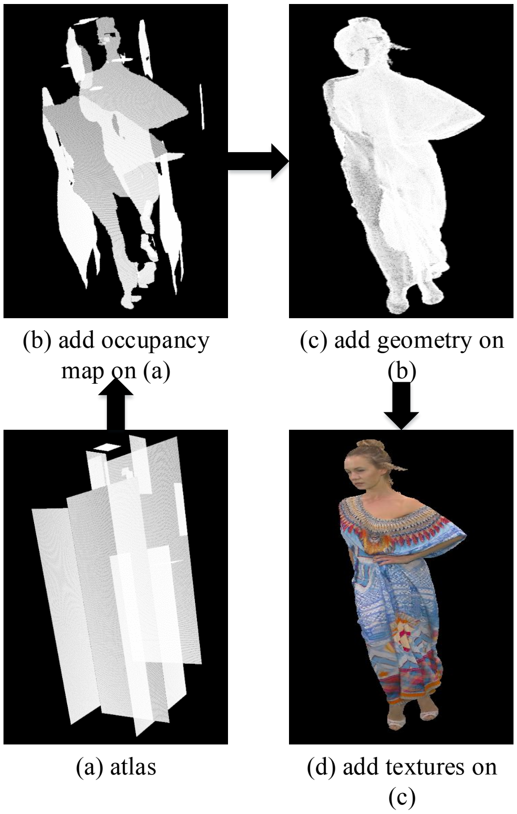


Figure 11: The process of 3-D point-cloud reconstruction [1] in V-PCC. The atlas (a) is first reconstructed. Only padded patch rough sketches are built. After the decoded occupancy map is merged into the reconstruction (b), the points occupancy status becomes clearly shown because the occupancy map has removed the padded area. When the geometry information is added into the reconstruction (c), all 3-D point clouds are almost created except the color attributes. Finally, the attributes are drawn on the reconstruction (d). This visualizing reconstruction process demonstrates how the 2-D geometry enormously impacts the subjective quality of 3-D point clouds reconstruction.

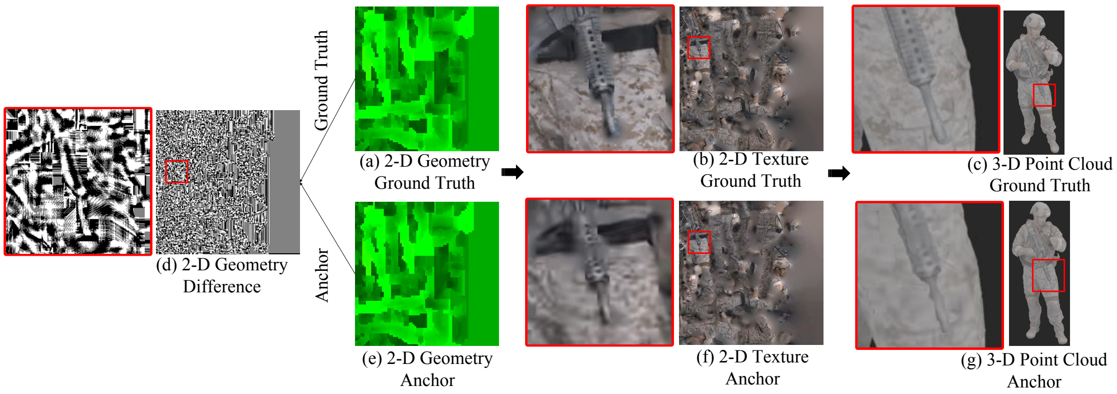


Figure 12: Impact of the 2-D geometry artifacts to the 3-D point cloud artifacts. The sub-figures of the first row (a), (b), and (c) are the 2-D geometry, attributes, and 3-D point cloud of ground truths, respectively. With the same arrangement, the sub-figures of the second row are from anchor reconstructions. Because it is difficult to directly recognize the difference of the geometry value between the anchor and ground truth with our eyes, we visualize their difference in (d). White and black pixels represent the different ones, while gray pixels are the same ones. We can clearly see the geometry value distortion of the gun in the enlarged area of (d). From the enlarged areas, we can clearly see that the V-PCC propagates the 2-D geometry distortion to attributes reconstruction. Then, the V-PCC brings the attributes artifacts into the 3-D point-cloud reconstruction.

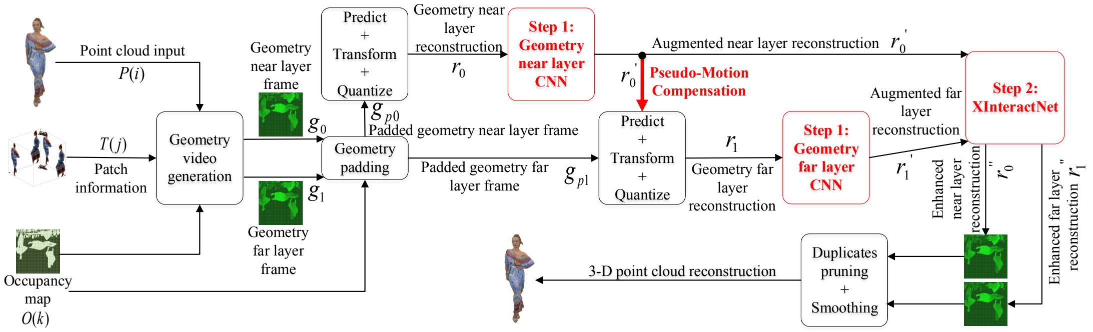

Point cloud input
P(i)
T(j)
Patch information
Occupancy map
O(k)
Geometry video generation
Geometry near layer frame
g_0
Geometry far layer frame
g_1
Geometry padding
g_{p0}
Padded geometry near layer frame
Padded geometry far layer frame
g_{p1}
Predict + Transform + Quantize
Geometry near layer reconstruction
r_0
Step 1: Geometry near layer CNN
Augmented near layer reconstruction r_0'
r_0'
Pseudo-Motion Compensation
Predict + Transform + Quantize
r_1
Geometry far layer reconstruction
Step 1: Geometry far layer CNN
r_1'
Augmented far layer reconstruction
Step 2: XInteractNet
Enhanced near layer reconstruction
r_0''
Enhanced far layer reconstruction r_1''
Duplicates pruning + Smoothing
3-D point cloud reconstruction

Figure 13: The proposed two-step method is embedded in the V-PCC encoding scheme. The target of the first step is to denoise the coarse artifacts of the reconstructed near layer r_0 and far layer r_1 first. The augmented near layer reconstruction $r_0^{'}$ as PMC iteratively participates into the far layer g_{p1} prediction to generate the far layer reconstruction r_1. Afterward, r_1 is fed into the far layer step one CNN to produce a better reconstruction $r_1^{'}$. The objective of the second step is to utilize the interactive information $I_s(r_0^{'}, r_1^{'})$ between near layer $r_0^{'}$ and far layer $r_1^{'}$ in full extent for further removing artifacts. We input the outputs of step one including $r_0^{'}$ and $r_1^{'}$ into XInteractNet to achieve enhanced corresponding near layer reconstruction $r_0^{''}$ and far layer reconstruction $r_1^{''}$. They are finally used for reconstructing the 3-D point cloud.

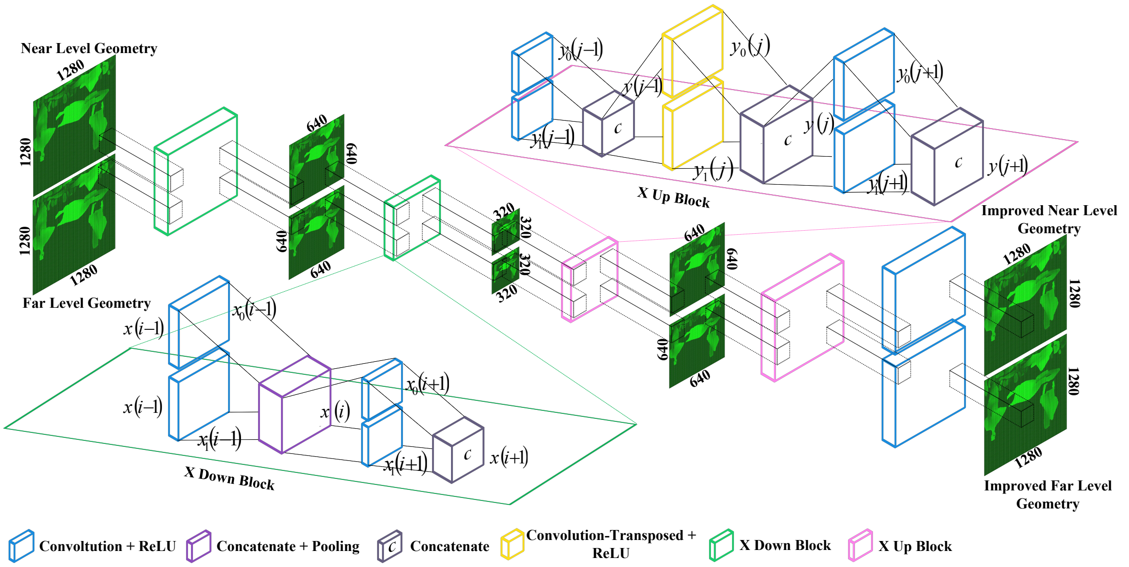
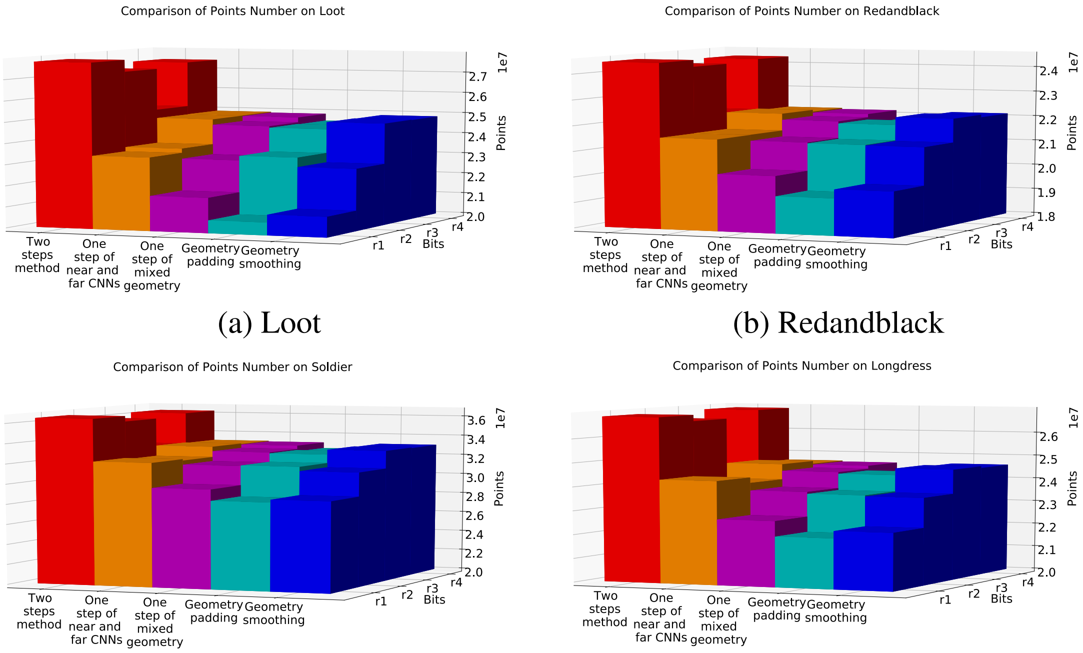


Figure 14: XInteractNet architecture. In the X down block, to share the interactive information $I_d(x_0(i), x_1(i))$ at level i before the max-pooling operation, the previous convolutional near layer's feature $x_0(i-1)$ and far layer's feature $x_1(i-1)$ are merged into $x(i-1)$ first. The No. $i+1$ convolutional near and far layers take the pooled $x(i)$ feature as input to tackle the next convolution computation. Similarly, the X up block shares the interactive information $I_u(y_0(j), y_1(j))$ at level j before the transposed-convolution operation. The previous convolutional near layer's feature $y_0(j-1)$ and far layer's feature $y_1(j-1)$ are shared and merged into $y(j-1)$. The obtained $y_0(j)$ and $y_1(j)$ features are concatenated again and fed into the next NO. $j+1$ convolutional near and far layers for computation.



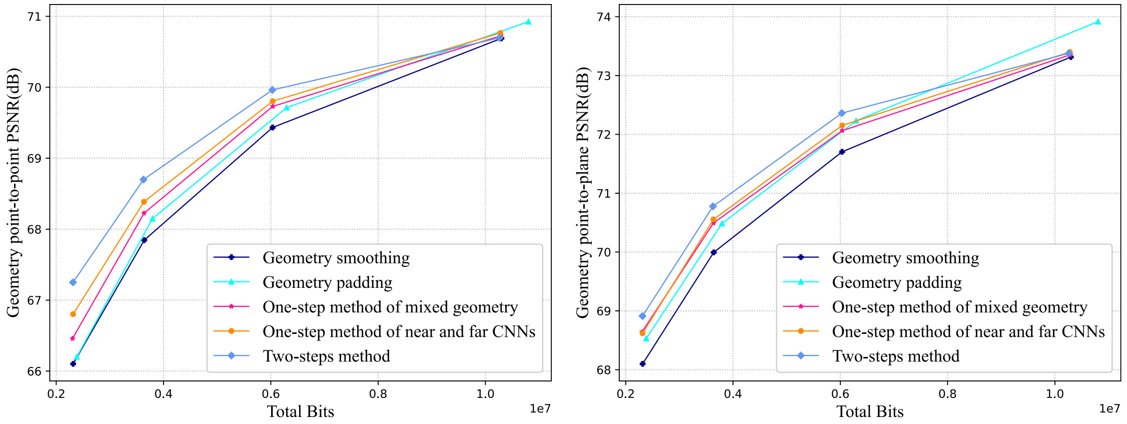
(a) Loot

(b) Redandblack

(c) Soldier

(d) Longdress

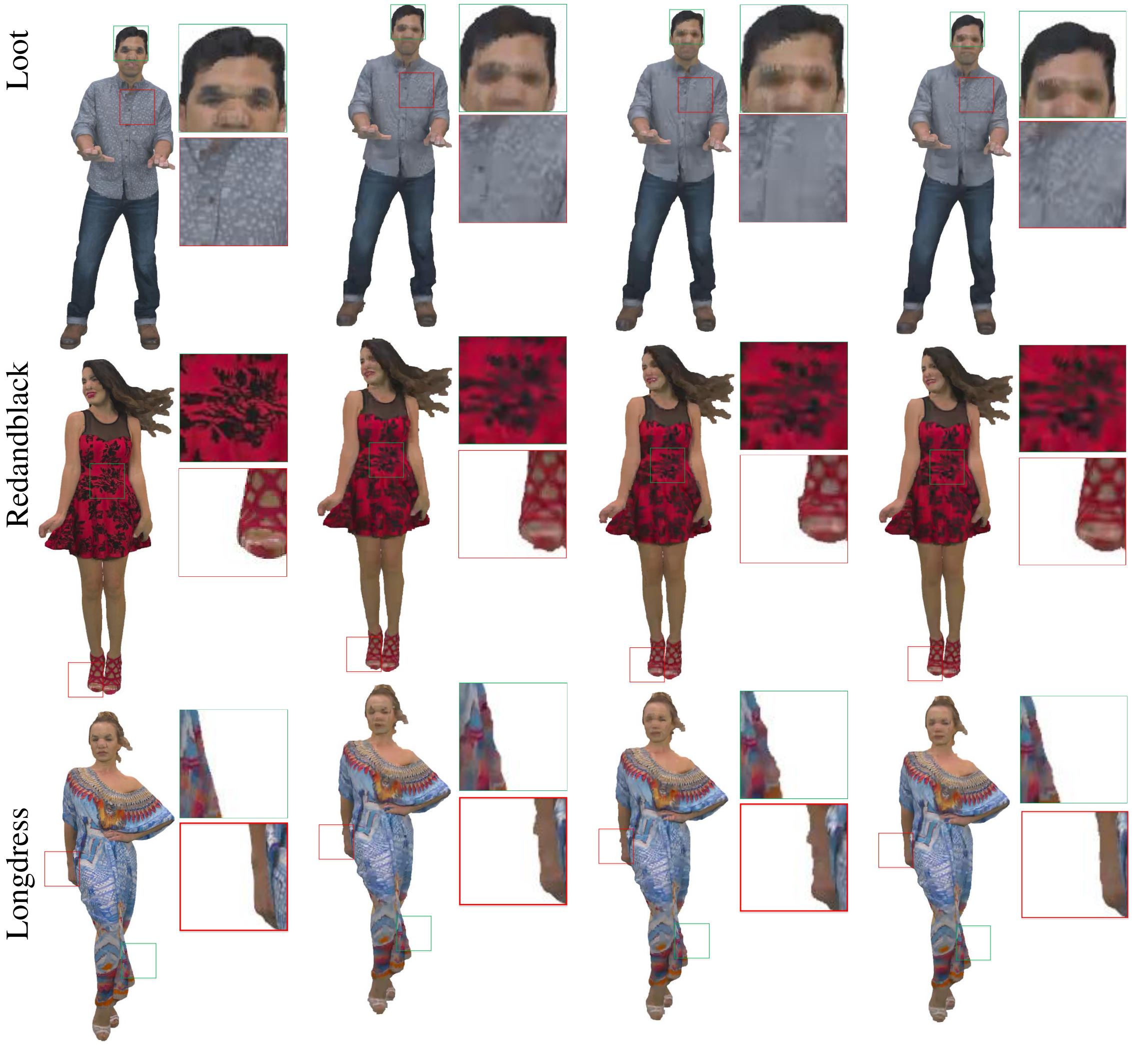
Figure 15: Number of points N_R in reconstructed 3-D point clouds within first 32 frames of four CTC sequences. The bitrate of Y axis gradually increases from level $r1$ to level $r4$. The number of points N_R of our proposed two-step method clearly restore more points than V-PCC geometry pading and smoothing methods. These statistics fully demonstrate that the proposed two-step method effectively restores the 3-D point clouds.



(a) Point-to-point RD Curves on Loot

(b) Point-to-plane RD Curves on Loot

Figure 16: Comparison of Geometry point-to-point and point-to-plane Rate-Distortion (RD) curves on Loot sequence in all the intra cases. As shown, the point-to-point and point-to-plane PSNRs of first three rate points in the proposed two-step method is higher than the ones of all other V-PCC geometry padding and smoothing methods. This proves that the proposed algorithm performs obviously better than SOTA methods on improving coding efficiency.



(a) Ground truth

(b) Geometry smoothing method

(c) Geometry padding method

(d) Proposed two-step method

Figure 17: Visual comparisons of ground truth, point-cloud reconstructions of geometry smoothing, padding and proposed two-step methods. The figures are derived from Loot, RedandBlack, and Longdress. From the three sequences, we can clearly see from the red and green rectangles that there are significant artifacts and noises in the reconstructions of V-PCC Geometry smoothing and padding methods, while the reconstructions of the proposed two-step method show a smoother effect. The visual results obviously demonstrate that compared to the SOTA, the proposed two-step method brings better subjective qualities.

CHAPTER 3

CONVOLUTIONAL NEURAL NETWORK-BASED OCCUPANCY MAP ACCURACY IMPROVEMENT FOR VIDEO-BASED POINT CLOUD COMPRESSION

In video-based point cloud compression (V-PCC), a dynamic point cloud is projected onto geometry and attribute videos patch by patch for compression. In addition to the geometry and attribute videos, an occupancy map video is compressed into a V-PCC bitstream to indicate whether a two-dimensional (2D) point in the projected geometry video corresponds to any point in three-dimensional (3D) space. The occupancy map video is usually downsampled before compression to obtain a tradeoff between the bitrate and the reconstructed point cloud quality. Due to the accuracy loss in the downsampling process, some noisy points are generated, which leads to severe objective and subjective quality degradation of the reconstructed point cloud. To improve the quality of the reconstructed point cloud, we propose using a convolutional neural network (CNN) to improve the accuracy of the occupancy map video. We mainly make the following contributions. First, we improve the accuracy of the occupancy map video by formulating the problem as a binary segmentation problem since the pixel values of the occupancy map video are either 0 or 1. Second, in addition to the downsampled occupancy map video, we introduce a reconstructed geometry video as the other input of the CNN to provide more useful information in order to indicate the occupancy map video. To the best of our knowledge, this is the first learning-based work to improve the performance of V-PCC.

Compared to state-of-the-art schemes, our proposed CNN-based approach achieves much more accurate occupancy map videos and significant bitrate savings.

3.1 Background

Three-dimensional (3D) industry- and consumer-level scanning equipment, such as RGBD cameras [67, 68] and light detection and ranging (LIDAR) [66, 67], are becoming more common and less expensive than ever before. These sensing devices are capable of scanning and producing a massive amount of 3D data. Due to their ability to represent 3D data in a more immersive and realistic pattern, 3D visual representation approaches such as polygon meshes, light fields, and point clouds are becoming increasingly popular. Among these 3D volumetric digital representation formats, point clouds achieve a good tradeoff among ease of acquisition, realistic rendering, and facilitating data manipulation and processing. Therefore, point clouds are being adopted more frequently.

Point clouds lay a solid foundation for unprecedented visual technologies, including immersive virtual reality (VR), augmented reality (AR), and mixed reality (MR) [69]. These advanced technologies are useful in many applications [70, 71], including historic site [76] and art museum exploration, immersive real-time remote telecommunications [75], interactive games [74], and mobile navigation [72] [73]. However, point clouds are typically represented by an extremely large amount of data. Consequently, it is impossible to cache, stream, and render these large amounts of raw point cloud data. This barrier has created the necessity for efficient point cloud compression (PCC).

Recently, the Moving Pictures Experts Group (MPEG) initiated a standardization activity [77] on PCC. The diversity of point clouds in terms of density has led to the development of two technologies: video-based point cloud compression (V-PCC) [77] and geometry-based point cloud compression (G-PCC) [77]. In this paper, we mainly focus on some improvements based on V-PCC. In V-PCC, a point cloud is initially segmented into 3D patches. Then, these 3D patches are projected onto two-dimensional (2D) planes and packed into geometry and attribute videos. Afterwards, the empty space in the geometry and attribute videos is padded to keep the spatial continuity to improve the video compression efficiency. Finally, the geometry and attribute videos are compressed with high-efficiency video coding (HEVC) [11].

Due to the padding process and the loss caused by compression, it is difficult to determine whether one pixel in the reconstructed geometry video corresponds to a valid 3D point. To address this problem, in addition to the geometry and attribute videos, an occupancy map video is compressed into the V-PCC bitstream. The pixels in the occupancy map video are used to indicate whether the pixels in the geometry and attribute videos correspond to any points in 3D space. Ideally, the occupancy map video should be coded with the same resolution as the geometry and attribute videos (full-resolution), but this incurs a high bitrate cost. To save bitrates, the V-PCC encoder downscales the full-resolution occupancy map video to a half-resolution or quarter-resolution video before compression. The V-PCC decoder then upscales this downscaled occupancy map video back to the full resolution for reconstructing the 3D point cloud. Some noisy pixels are thus introduced in the boundary areas of the upsampled full-resolution occupancy map video. These 2D noisy pixels are reconstructed into 3D noisy points, which leads to serious quality degradation of the reconstructed point cloud.

Through V-PCC standardization, a variety of occupancy map refinement methods [3, 4, 106–112] have been proposed to improve the occupancy map accuracy. Among them, the methods in [4] and [3] have been adopted in the V-PCC encoder, although they are disabled by default. In [4], a patch border filter (PBF) was proposed to manipulate occupancy map and geometry videos to reduce the distance between the contours of patches. However, this method may still introduce some error pixels on the contours. In [3], an occupancy refinement (OR) method is proposed to iteratively refine the occupancy flags of blocks with fewer pixels to avoid introducing noisy ones in the occupancy map. However, this method can still insert some noisy pixels. These deficiencies all lead to degradations in the quality of 3D point cloud reconstructions. Therefore, these two methods have not been adopted as part of the V-PCC common test condition (CTC) [105], and there is still considerable space to develop a better method to improve the accuracy of occupancy map videos.

In this paper, we propose an occupancy-geometry-based convolutional neural network (OGCNN) to improve the accuracy of occupancy map videos in order to improve the quality of reconstructed 3D point clouds. To the best of our knowledge, this work is the first CNN-based solution for improving the efficiency of V-PCC. This work mainly makes the following technical contributions.

- We formulate the problem of occupancy map accuracy improvement as a binary segmentation problem. The binary cross entropy loss is adopted as the loss function to train the CNN.

- A reconstructed geometry video is introduced as the other input of the proposed CNN in addition to an occupancy map video. The geometry contains useful information that can help improve the accuracy of the occupancy map video.
- The proposed algorithm is implemented in the V-PCC reference software. Extensive experiments have been conducted to compare the algorithm in this paper with state-of-the-art (SOTA) algorithms to demonstrate the effectiveness of the proposed scheme.

We organize the remainder of this paper as follows. We review the related works on point cloud compression in Section 3.2, followed by our motivation and observations on occupancy map video enhancement in Section 3.3. We introduce the proposed CNN-based occupancy map accuracy improvement method in Section 3.4. In Section 3.5, we comprehensively report and analyze the experimental results. A summary of this paper is presented in Section 3.6.

3.2 Occupancy map improvement

This section briefly reviews the prior works on dynamic point cloud compression and accuracy improvements based on occupancy map videos in V-PCC.

3.2.1 Dynamic point cloud compression

There are roughly two types of compression methods, 3D-based approaches and 2D-based approaches, for dynamic point cloud compression. As indicated by its name, a 3D-based approach directly performs 3D motion estimation and motion compensation in

3D space. Kammerl *et al.* [83] proposed a lossy compression method for dynamic point cloud streaming that uses the colocated octree node of the reference frame to predict that of the current frame. This method, however, can only be applied to frames with small motions. Thanou *et al.* [84] formulated 3D motion estimation as a feature-matching problem between successive graphs after representing the time-varying geometry of these point cloud frames with a set of graphs. Nonetheless, the motion vectors of some objects in point cloud frames are not accurately estimated. Queiroz *et al.* [85] developed a simple coder that breaks the voxelized point cloud at each frame into blocks of voxels. The 3D translational motion estimation was performed block by block to find the corresponding block of the reference frame. In addition, Mekuria *et al.* [86] further introduced the iterative closest point (ICP) instead of a translational motion model to better characterize the motions in neighboring frames. These schemes can attenuate the deficiencies of 3D motion estimation and motion compensation to some extent. Nevertheless, without flexible block partitioning and more efficient motion estimation algorithms, the compression performance of dynamic point clouds is still incomparable with that of 2D-based methods.

The 2D-based methods that are dedicated to converting a 3D dynamic point cloud to 2D videos for compression through 2D video coding standards have been proven to be efficient. Budagavi *et al.* [88] proposed compressing projected 2D videos derived by sorting points in a 3D point cloud with HEVC. However, this work cannot exploit the mature interprediction, as the generated videos do not have high spatial and temporal correlations. To alleviate this drawback, He *et al.* [89] employed the cubic projection method to convert a 3D dynamic point cloud to 2D videos. Although this work promotes video coding performance, this algorithm leads to the loss of many points due to occlusion. Lasserre *et al.* [90] proposed combining an octree and a projection to decrease the number of occluded points. Mammou *et al.* [91] considered projecting a 3D dynamic point cloud onto 2D videos with a patch-based algorithm. Compared to other proposals, the patch-based algorithm [93] shows better compression efficiency. The MPEG immersive (MPEG-I) media working group adopts a patch-based algotithm as the base of the V-PCC standard. In addition, Li *et al.* [113] proposed using occupancy-map-based rate-distortion optimization and partitioning to improve the performance of V-PCC. Although V-PCC has been proven to be efficient due to its astonishing performance, the downsampled occupancy map video, which intrinsically guides the reconstruction of the geometry and texture information, leads to severe objective and subjective quality degradation of the reconstructed point cloud.

3.2.2 Recent advances in occupancy map video improvement

Through the V-PCC standardization process, many occupancy map refinement methods were proposed to improve occupancy map accuracy. Vosoughi *et al.* [106] proposed a scalable locally adaptive erosion filter that first classified the current pixel of the full-resolution decoded occupancy map into a set of intuitively well-defined classes. Then, different erosion patterns were applied to various classes in the neighborhood of the current pixel. Due to the coarse occupancy resolution, some noisy points are added to the reconstructed point cloud. Oh *et al.* [107] proposed a combination of upsampling and 2D filtering to remove the added points in the occupancy map video. To smooth the jaggy patch boundaries and reduce redundant points, Lee *et al.* [108] proposed an occupancy map refinement method with corner-based boundary estimation. This work primarily addressed the oblique lines. Cai *et al.* [109] proposed an adaptive occupancy map upsampling method for reconstructing a high-resolution occupancy map video. However, there is no guarantee that it can be as close as possible to the original full-resolution occupancy map video. Najaf-Zadeh *et al.* [110] proposed signaling a ternary occupancy map to the decoder if a boundary block in the occupancy map is allowed to be trimmed. Wang *et al.* [112] proposed shifting the position of the occupancy map bounding box during patch generation. However, it can only partially reduce the number of noisy points. These methods can partially solve the problem of inaccurate occupancy map videos. However, none of them are significant enough to be adopted by V-PCC.

There are some methods adopted by the V-PCC encoder during the V-PCC standardization process. Andrivon *et al.* [4] proposed a patch border filtering (PBF) method to manipulate the occupancy map and geometry videos to reduce the distance between contours of patches. However, this method can still insert some noisy pixels on the contours. Guede *et al.* [3] proposed a method to iteratively refine an occupancy map video. This method is proposed to modify the occupancy flags of the blocks with fewer pixels to avoid inserting error flags in an occupancy map video. However, this method may still introduce some noisy points while removing some real points. As a result, they are disabled in the V-PCC encoder by default and are not part of the V-PCC common test condition. Therefore, there is still considerable space to devise a better occupancy map video accuracy improvement method to boost the dynamic point cloud compression efficiency.

3.3 Occupancy map video in V-PCC

In this section, we first give a clear definition of the resolution of an occupancy map video. Then, the influences of the occupancy map resolution on distortions and bitrates are introduced in detail.

3.3.1 Occupancy map video resolution

Ideally, an occupancy map video should be coded at full resolution to indicate exactly whether pixels in the geometry and attribute videos correspond to any points. Nevertheless, a full-resolution occupancy map would cost too many bits. To save bit cost, the V-PCC downscales the full-resolution occupancy map video by P times. Correspondingly, a $P \times P$ block b_p of the full-resolution occupancy map, consisting of P^2 pixels, is downscaled to a single pixel s_p in the downsampled video. When P equals 2 and 4, the downscaled video is called a half-resolution and quarter-resolution occupancy map video, respectively. The V-PCC then upscales the downsampled occupancy map video back to a full-resolution video. Correspondingly, s_p is upscaled to a $P \times P$ block $b_p^{'}$. The reconstructed full-resolution occupancy map video is finally used for reconstructing the geometry and attributes.

In the following, to further analyze the influence of the occupancy map video on the reconstructed quality of the geometry and attributes, we call the pixels indicating that there are corresponding points in 3D space occupied pixels, while we name the pixels indicating that there are no corresponding points in 3D space unoccupied pixels. Suppose a $P \times P$ block b_p in the original full-resolution occupancy map video includes X_o occupied

Table 17: V-PCC anchor [2] performance comparison of the quarter-resolution, half-resolution, and full-resolution occupancy maps within the first 32 frames

Class	Sequence	Quarter vs. Full		Quarter vs. Half	
		Geom.BD-GeomRate		Geom.BD-GeomRate	
		D1	D2	D1	D2
A	Loot	-50.3%	-45.5%	-21.9%	-14.9%
	Redandblack	-52.9%	-43.3%	-23.8%	-11.8%
	Soldier	-52.1%	-44.2%	-23.8%	-13.9%
	Queen	-61.1%	-52.7%	-25.4%	-15.3%
B	Longdress	-50.1%	-41.1%	-23.2%	-12.8%
	Class A	-54.1%	-46.4%	-23.7%	-14.0%
	Class B	-50.1%	-41.1%	-23.2%	-12.8%
Avg.	All	-53.3%	-45.3%	-23.6%	-13.8%

pixels. If X_o is less than P^2, then b_p is partially occupied. Even though b_p is partially occupied, V-PCC marks its corresponding s_p as occupied to avoid losing points. The occupied pixel s_p indicates that all P^2 pixels in the reconstructed full-resolution occupancy map video are occupied. The downsampling and upsampling processes would increase $P^2 - X_o$ occupied pixels. Fig. 18 gives a typical example to compare the full-resolution occupancy map video with the quarter-resolution occupancy map video. The red pixels indicate the occupied pixels in the full-resolution occupancy map video, while the blue pixels indicate the occupied pixels in the reconstructed full-resolution occupancy map video. In an extreme case, in the original full-resolution occupancy map, as shown in the top left subfigure of Fig. 18, only one pixel is occupied in a 4×4 block. However, in the quarter-resolution occupancy map, as shown in the top right subfigure of Fig. 18, all the pixels in the corresponding 4×4 block are considered occupied. In this way, 15 noisy pixels are generated in the restored full-resolution occupancy map.

3.3.2 The impact of the occupancy map resolution on the quality of the geometry and attributes

An increase in the number of noisy pixels in the full-resolution occupancy map video can lead to noisy pixels in the geometry and attribute videos. As illustrated in Fig. 19, the reconstructions of the occupancy maps, geometry, and attributes in the first and second rows are derived from the configurations of the quarter resolution and full resolution, respectively. We can see from the enlarged areas of the occupancy map videos ((a) and (b)) that the edge of the body shows a more severe block artifact in the quarter-resolution case than in the full-resolution case. Compared with the quarter resolution, the full resolution provides more accurate representations of the occupancy map. Moreover, the impact of the occupancy accuracy can be propagated into the geometry and attributes. We can see from the enlarged areas of the geometry ((c) and (d)) and attributes ((e) and (f)) that the block distortions are more severe in the quarter-resolution case than in the full-resolution case.

3.3.3 The impact of the occupancy map on the bitrates

As mentioned in Section 3.3.1, an occupancy map with higher resolution may lead to smaller distortions. However, it also brings a much higher bitrate cost. According to our observations, the bit cost of the full-resolution occupancy map is approximately four times greater than that of the quarter-resolution map. As shown in Table 17, we compare the BD-rates [60] of the point-to-point error (D1) and point-to-plane error (D2) [104] among the quarter-resolution, half-resolution and full-resolution occupancy maps in the V-PCC anchor version 11 [2]. Compared to the full-resolution occupancy map, the quarter-resolution occupancy map achieves a -53.3% BD-rate savings on D1 and a -45.3% BD-rate savings on D2. Compared to the half-resolution occupancy map, the quarter-resolution occupancy map achieves a -23.6% and -13.8% BD-rate savings on D1 and D2, respectively. The main reason for these results is that, compared to the half-resolution and full-resolution occupancy map videos, the quarter-resolution videos are downscaled two and four times, respectively; hence, they cost much fewer bits.

3.4 Occupancy map refinement OGCNN with two inputs

In this section, we introduce the proposed OGCNN scheme in detail, including a detailed discussion on the design of the OGCNN, loss function, dataset, and training process.

3.4.1 Architecture of the proposed OGCNN

As shown in Table 17, the quarter-resolution occupancy map video leads to a better performance compared with the half-resolution and full-resolution occupancy map videos. However, we also know that the higher the occupancy map video resolution is, the better the quality of the reconstructed geometry and attributes. Therefore, we use the quarter-resolution occupancy map video as the base and try to design an algorithm to improve its accuracy and to improve the reconstructed point cloud geometry and attribute quality. As CNNs have been demonstrated to be powerful in both low-level and high-level vision tasks [82], we propose using a CNN to make the accuracy of the quarter-resolution occupancy map as close as possible to that of a higher-resolution target. The higher-precision target can be the full-resolution or half-resolution occupancy map.

When we design the proposed architecture, we mainly consider the following two aspects to optimize its performance. First, as the occupancy map is a particular type of video that incorporates only binary values, we formulate the problem of improving the occupancy map precision as a binary segmentation problem. In other words, we try to devise a segmentation CNN that can discriminate the occupied (value 1) and unoccupied (value 0) statuses per pixel in the inputted occupancy map. Second, in addition to the quarter-precision occupancy map video, geometry reconstruction is introduced as the other input to provide the network with more useful information. In the V-PCC encoder, as the geometry values of the unoccupied pixels are padded from their neighbors [114], they can better reflect the real occupancy distribution than the binary occupancy map. For example, if the geometry value of a specific position is not the same as that of its neighbors, it is almost impossible for it to be an unoccupied pixel. However, we cannot obtain this information from the quarter-resolution occupancy map itself. Therefore, we consider the geometry reconstruction to be an important supplement to the quarter-resolution occupancy map.

Fig. 20 shows the overall architecture of the proposed OGCNN scheme with both the quarter-resolution occupancy map video and reconstructed geometry video as inputs. The scheme consists of two subnetworks: the Occupancy Network and the Geometry Network. The Occupancy Network uses the quarter-resolution occupancy map video as input. It derives occupancy segmentation feature maps from the occupancy map. The

Geometry Network uses the reconstructed geometry video as input and derives geometry segmentation feature maps from the geometry. The occupancy map and geometry segmentation features are then concatenated together and used as the input of the remaining convolutional layers.

Algorithm 2 The flow of the OGCNN approach in V-PCC

```
Input: $x$ is the Occupancy Network input, and the geometry reconstruction $z$ is the
       Geometry Network input.
Output: The enhanced occupancy map $F_{out}(O(x), G(z))$.

if Initialization succeeds then
    Input the occupancy map $x$ into the Occupancy Network;
    Extract the geometry reconstruction $z$ as the Geometry Network input;
    Compute the Occupancy Network segmentation features $O(x)$;
    Compute the Geometry network segmentation features $G(z)$;
    Concatenate the segmentation features of $O(x)$ and $G(z)$;
    Obtain the enhanced occupancy map $F_{out}(O(x), G(z))$;
end
```

In addition to the dual inputs, as shown in Fig. 20, we develop different subnetworks for the quarter-resolution occupancy map video and the reconstructed geometry video. As mentioned above, the characteristics of the occupancy map and geometry reconstruction are different. The occupancy map is binary, while the information in the geometry is more sensitive. We design different subnetworks to optimize the features derived from the occupancy map and geometry. Detailed introductions of the two subnetworks are described in Section 3.4.2 and Section 3.4.3, respectively.

Algorithm 2 shows the algorithm flow of the proposed OGCNN. We first extract the occupancy map and geometry reconstructions from the bitstream. Then, both of them

Table 18: Occupancy Network Parameters of the Conv and Transposed Conv Layers

Layer	Conv1	Conv2	Transposed Conv1	Conv3	Conv4
Kernel Size	3×3	3×3	2×2	3×3	3×3
Feature Map Number	4	8	4	4	4
Stride	1	1	2	1	1
Padding	1	1	0	1	1

are fed into the OGCNN to generate the occupancy map video with a higher accuracy. The occupancy map video with a higher accuracy is finally used in loop for reconstructing the geometry, attributes, and point cloud.

3.4.2 Design of the Occupancy Network

The Occupancy Network uses the unsampled quarter-resolution occupancy map video as the input. It adopts the classic autoencoder architecture [51] [52] with a skip connection concatenating the encoder and decoder [53]. The Occupancy Network contains a downsampling and upsampling pair to segment the occupancy map. In this way, the Occupancy Network can collect the global information as much as possible.

The lower branch of Fig. 20 shows the detailed architecture of our proposed Occupancy Network. We adopt the max pooling plus convolutional layer and transposed convolutional layer [54] to perform downsampling and upsampling, respectively. At the encoder, downsampling reduces the occupancy map redundancy and keeps the most distinctive features for segmentation. At the decoder, upsampling increases the spatial resolution of the features to the target resolution for accurate segmentation. However, the downsampling-upsampling process may lead to a loss of global information. To provide accurate global information for segmentation, a skip connection, which concatenates the features in the encoder and decoder, is added to the network structure.

Table 18 shows the detailed configurations of the Occupancy Network. For the convolutional layers, we set the kernel size to 3×3, the stride to 1, the padding size to 1, and the feature map number to 4 or 8. For the transposed convolutional layers, we set the kernel size to 2×2, the stride to 2, the padding size to 0, and the feature map number to 4. We use the rectified linear unit (ReLU) as the activation function.

3.4.3 Design of the Geometry Network

As analyzed in Section 3.4.1 above, we consider the reconstructed geometry video as the other input of the proposed OGCNN to improve the precision of the occupancy map video. Accordingly, we develop a specific Geometry Network to derive distinctive features. In the Geometry Network, the residual block [43] is employed to derive the geometry features for segmentation. The residual block also has the benefit of preventing the vanishing of the gradient.

The upper branch of Fig. 20 describes the detailed structure of the proposed Geometry Network. The Geometry Network includes a residual block and three convolutional layers. Considering the complexity, we only use a total of five convolutional layers to derive the geometry features. For each convolutional layer, we set the kernel size to 3×3, the padding size to 1, the stride to 1, and the feature map number to 4.

3.4.4 Loss function

To train our proposed segmentation network effectively, we adopt the binary cross-entropy loss [115] to supervise the training of the proposed OGCNN.

$$\begin{aligned} L(\Theta) = \tfrac{1}{N} \textstyle\sum_{i=1}^{N} (\log \Upsilon\left((O_i, G_i)|\Theta\right) \cdot X_i - \\ \log(1 - \Upsilon\left((O_i, G_i)|\Theta\right)) \cdot (1 - X_i)) \end{aligned} \tag{3.1}$$

where Θ encapsulates the whole parameter set of the OGCNN, including the weights and bias, and $\Upsilon\left(Y_i|\Theta\right)$ denotes the OGCNN module. X_i denotes the labels of a half-resolution or full-resolution occupancy map, where i indexes each label. O_i and G_i are the corresponding dual inputs of the upsampled quarter-resolution occupancy map and the reconstructed geometry video, respectively. N is the number of samples. Under the supervision of the binary cross-entropy loss, the output of the occupancy map video is close to that of the target half-resolution or full-resolution occupancy map video.

3.4.5 Dataset and training

Dataset. There are currently no widely used datasets to train the proposed OGCNNN for improving V-PCC. The only dataset we can have access to is the dynamic point cloud dataset provided by $8i$ and defined in the V-PCC CTC [105]. We divide the five dynamic point clouds from $8i$ into training, validating, and testing datasets. More specifically, we use the dynamic point cloud called Soldier for training and validation. We use the other four dynamic point clouds, Loot, Redandblack, Queen, and Longdress, for

Table 19: Training parameters

Parameters	Value
Batch size	16
Total Epochs	60
Base Learning Rate	$1e^{-4}$
γ Adjusting Coefficient	0.1
Adjusting Epoch Intervals	50
Weight Decay	$1e^{-4}$
Momentum	0.9

testing. With Soldier, we first derive 300 frames of the quarter-resolution occupancy map video and reconstructed geometry video, both of which have spatial resolutions of 1280×1280, from the V-PCC reference software. Among them, 224 and 76 frames are used for training and validation, respectively. Then, we generate the same number of full-resolution and half-resolution occupancy map videos as labels. Finally, we extract 64×64 blocks from the Luma component of the occupancy map videos and the reconstructed geometry videos and use them for training the proposed OGCNN. In total, there are $89,600$ pairs of 64×64 inputs and labels for training and $30,400$ pairs for validation.

Training. Table 19 shows the detailed parameters of the training process. The batch size and total number of epochs are set as 16 and 60, respectively. For training, we set the base learning rate to $1e^{-4}$. After 50 epochs, we decrease the learning rate by multiplying by 0.1. We adopt the adaptive moment estimation (Adam) [59] algorithm as the gradient optimizer. The momentum and weight decay are set to 0.9 and $1e^{-4}$, respectively.

Table 20: Performance comparison of the full-resolution OGCNN, half-resolution OGCNN and quarter-resolution V-PCC [2] under the all intra case

Class	Sequence	Full OGCNN vs. V-PCC [2]					Half OGCNN vs. V-PCC [2]				
		Geom.BD-TotalRate		Attr.BD-TotalRate			Geom.BD-TotalRate		Attr.BD-TotalRate		
		D1	D2	Luma	Cb	Cr	D1	D2	Luma	Cb	Cr
A	Loot	6.3%	**-16.9**%	2.9%	1.2%	4.6%	−0.2%	**-13.5**%	0.8%	0.4%	2.4%
	Redandblack	16.2%	**-20.4**%	3.4%	−0.2%	2.2%	4.1%	**-15.2**%	1.1%	−0.2%	0.5%
	Queen	18.4%	**-26.1**%	23.9%	24.3%	48.1%	−1.5%	**-18.5**%	5.0%	7.3%	14.4%
B	Longdress	22.0%	**-18.5**%	3.2%	0.9%	2.0%	7.7%	**-14.2**%	0.9%	0.1%	0.6%
	Class A	13.6%	**-21.1**%	10.1%	8.4%	18.3%	0.8%	**-15.7**%	2.3%	2.5%	5.8%
	Class B	22.0%	**-18.5**%	3.2%	0.9%	2.0%	7.7%	**-14.2**%	0.9%	0.1%	0.6%
Avg.	All	15.7%	**-20.5**%	8.4%	6.5%	14.2%	2.5%	**-15.3**%	1.9%	1.9%	4.5%

Table 21: Occupancy accuracy comparison of the V-PCC anchor and OGCNNs on boundary blocks

	$N = 16$			$N = 32$			$N = 64$		
	V-PCC Anchor	Half OGCNN	Full OGCNN	V-PCC Anchor	Half OGCNN	Full OGCNN	V-PCC Anchor	Half OGCNN	Full OGCNN
Loot	89.27%	94.61%	96.24%	94.03%	97.00%	97.89%	96.08%	98.03%	98.61%
RedandBlack	88.27%	93.52%	95.38%	92.70%	95.96%	97.10%	94.59%	97.00%	97.84%
Queen	87.35%	92.26%	94.09%	91.86%	95.01%	96.16%	93.58%	96.06%	96.95%
Longdress	88.96%	93.69%	95.06%	93.42%	96.23%	97.00%	95.53%	97.43%	97.95%
Avg. All	88.47%	93.52%	95.19%	93.01%	96.05%	97.04%	94.94%	97.13%	97.84%

3.5 Experimental results

3.5.1 Experimental settings and metrics

To test the performance of the proposed OGCNN, we implement the proposed OGCNN in version 11 of the V-PCC reference software [2] to compare it with the V-PCC version 11 anchor, PBF [4], and OR [3]. Two OGCNNs are trained depending on whether we use the full-resolution occupancy map video or the half-resolution occupancy map video as the label. The OGCNN trained with the full-resolution occupancy map video as the label is named the Full OGCNN. The OGCNN trained with the half-resolution occupancy map is called the Half OGCNN. We test the performance of the proposed algorithms in both the all intra and random access cases, as defined in the V-PCC CTC [105]. We test the five rate points from a low bitrate $r1$ to a high bitrate $r5$, as defined in the V-PCC CTC [105]. As the dynamic point cloud Soldier is used in the training process, we use the other four dynamic point clouds from $8i$ to show the performance of the proposed OGCNN. To save some encoding time, we only test the first 32 frames of each point cloud, which are a good representation of all frames.

To evaluate the geometry distortions, we use the point-to-point error (D1) and point-to-plane error (D2) as the metrics [105]. Both D1 and D2 are calculated in a symmetrical way with both the original point cloud and reconstructed point cloud as the anchors. The one with a larger distortion is used as the final distortion. O and R denote the original point cloud and its reconstruction. For each point $r \in R$, we identify its corresponding point $o \in O$ by searching the nearest neighbor with a KD-tree in O. Then, D1

$d^{'}_{R,O}$ from R to O is calculated as follows:

$$d^{'}_{R,O} = \frac{1}{N_R} \sum_{\forall r \in R} ||D(r,o)||_2^2 \tag{3.2}$$

where N_R is the number of points in point cloud R. $D(r,o)$ is the error vector connecting r to o. D1 $d^{'}_{O,R}$ from O to R can be computed in a similar manner.

Similarly, $d^{''}_{R,O}$ denotes D2 from R to O, which is calculated as

$$d^{''}_{R,O} = \frac{1}{N_R} \sum_{\forall r \in R} (D(r,o) \cdot V_r)^2 \tag{3.3}$$

where V_r is the normal vector on point r. D2 $d^{''}_{O,R}$ from O to R can be computed in a similar manner.

The attribute distortion also employs the symmetric computation method. The attribute distortion [105] $d_{R,O}$ from R to O uses the mean square error (MSE)

$$d_{R,O} = \frac{1}{N_R} \sum_{\forall r \in R} ||y(o) - x(r)||_2^2 \tag{3.4}$$

where $y(o)$ and $x(r)$ are the attribute values of the original and reconstruction point cloud points, respectively. The attribute distortion $d_{O,R}$ from O to R can be computed in a similar manner.

To better show the performance of the proposed OGCNN in improving occupancy map accuracy, we provide a new quality metric to measure the occupancy map accuracy. As the occupancy map accuracy is meaningful only at the boundary block between the patches and the empty space, we first give a clear definition of the boundary block. As shown in Fig. 21, a boundary block is an $N \times N$ square that consists of both occupied and unoccupied pixels. In Fig. 21, the white grids represent the occupied pixels, while the black grids represent the unoccupied pixels. The red, blue, yellow squares indicate the 16×16, 32×32, and 64×64 boundary blocks, respectively. Then, our proposed occupancy accuracy α_N is defined as

$$\alpha_N = \frac{\sum_{i=1}^{\xi} \Phi_N(i)}{\Psi_N}, \tag{3.5}$$

where N is the boundary block size and ξ is the total number of boundary blocks. $\Phi_N(i)$ indicates the number of correctly identified pixels in the ith boundary block between the reconstructed occupancy map video and the label. Ψ_N is the total number of pixels in all ξ boundary blocks. As indicated by (3.5), when we measure the occupancy map accuracy, we restrict the statistical area to the boundary blocks to avoid counting large amounts of successive occupied or unoccupied pixels, as they are identical in the reconstructed and original occupancy map videos. Therefore, our proposed occupancy accuracy measure can better reflect the benefits of the proposed algorithms for improving the occupancy map accuracy.

3.5.2 Performances of the proposed OGCNN algorithm under the all intra case

Table 20 shows the BD-rate comparison results of the proposed OGCNN and the quarter-resolution V-PCC anchor under the all intra case. We can see that the proposed half and Full OGCNNs achieve an average of 15.3% and 20.5% BD-rate savings when D2 is used as the quality metric. The performance improvements are consistent for all tested dynamic point clouds, as the proposed OGCNN achieves over 10% rate-distortion (R-D) performance improvements for all dynamic point clouds. The peak difference reaches

18.5% and 26.1% for the dynamic point cloud Queen. The experimental results demonstrate the effectiveness of the proposed OGCNN.

In addition, we can see from Table 20 that both the half and Full OGCNNs lead to some performance losses on the geometry if measured by D1 and the attributes. As stated in Section 3.4.1, the OGCNN aims to remove some noisy points. Therefore, the numbers of points N_R of the proposed half and Full OGCNNs are less than that of the V-PCC anchor. According to (3.2) and (3.4), the smaller the number of points N_R is, the larger D1 and the attribute distortion are since they are the average of all points. That is why the proposed OGCNN suffers some performance losses in terms of geometry if measured by D1 or the attributes. In addition, as explained by [104], D1 has the disadvantage of ignoring the fact that point clouds represent surfaces of objects.

To better show the performance of the proposed OGCNNs for improving occupancy map accuracy, we compare the occupancy accuracies on the boundary blocks between the OGCNNs and the V-PCC anchor in Table 21. In Table 21, N represents the boundary block size. We test three configurations with N set to 16, 32 and 64 for evaluation. We can see that both the Full OGCNN and Half OGCNN perform much better than the V-PCC anchor. For example, when N equals 16, the Full OGCNN and the Half OGCNN improve the occupancy map accuracy by 6.72% and 5.05% compared with the V-PCC anchor, respectively. These results further demonstrate that the proposed OGCNN can lead to a better occupancy map video than the V-PCC anchor.

To measure the complexities of the proposed algorithm, we use the same environment to test both the V-PCC anchor and the proposed algorithm. More specifically, the

CPU configuration is an Intel(R) Core i5-8400 CPU @ 2.80 GHz, and the GPU configuration is a GTX 1080ti. In the all intra case, both the full and Half OGCNNs lead to almost the same encoding time compared with the V-PCC anchor. In addition, the decoding time is increased by 2% on average. The time complexities of the proposed algorithms are similar to that of the the V-PCC anchor.

Table 22: Performance comparison of the Half OGCNN and V-PCC anchor [2] under Random Access

Sequence	Geom.BD-TotalRate		Attr.BD-TotalRate		
	D1	D2	Luma	Cb	Cr
A.Loot	−1.1%	**-12.0**%	1.6%	0.5%	5.4%
A.Red&black	4.2%	**-14.1**%	1.1%	−0.1%	0.5%
A.Queen	−0.6%	**-18.8**%	7.0%	9.5%	17.6%
B.Longdress	9.2%	**-14.8**%	1.2%	1.1%	1.6%
Class A	0.8%	**-15.0**%	3.2%	3.3%	7.8%
Class B	9.2%	**-14.8**%	1.2%	1.1%	1.6%
Avg. All	2.9%	**-14.9**%	2.7%	2.7%	6.3%

3.5.3 Performance of the proposed OGCNN algorithm under the random access case

In the random access case, as shown in Table 22, we can see that compared to the V-PCC anchor, the proposed Half OGCNN can achieve an average 14.9% R-D performance improvement when D2 is used as the quality metric. The peak difference between the OGCNN and the V-PCC anchor reaches 18.8% on the dynamic point cloud Queen.

This result demonstrates that, in addition to the all intra case, the OGCNN can bring significant benefits to the random access case. As explained in Section 3.5.2, compared to the anchor, the Half OGCNN also suffers a few performance losses in terms of the attributes.

Table 23: Performance comparison of the Half OGCNN, OR [3], and PBF [4] under the all intra case

Sequence	Half OGCNN vs. PBF [4]		Half OGCNN vs. OR [3]	
	Geom.BD-TotalRate		Geom.BD-TotalRate	
	D1	D2	D1	D2
A.Loot	0.9%	**-4.5**%	1.2%	**-9.5**%
A.Red&black	3.3%	**-9.5**%	5.7%	**-8.9**%
A.Queen	4.2%	**-10.5**%	6.5%	**-11.3**%
B.Longdress	4.6%	**-12.8**%	7.4%	**-8.8**%
Class A	2.8%	**-8.1**%	4.5%	**-9.9**%
Class B	4.6%	**-12.8**%	7.4%	**-8.8**%
Avg. All	3.3%	**-9.3**%	5.2%	**-9.6**%

3.5.4 Comparison of the OGCNN and SOTAs

Table 23 shows the BD-rate comparison of the Half OGCNN, PBF [4], and OR [3]. The Half OGCNN performs better than the PBF and OR by an average of 9.3% and 9.6% when D2 is used as the quality metric, respectively. These performance results demonstrate that the proposed OGCNN significantly outperforms the SOTAs. In addition, the time complexities of the proposed algorithms are comparable to that of the SOTAs.

3.5.5 Ablation analysis of introducing the geometry

Table 24: Performance comparison of the Half OGCNN and Occupancy Network under the all intra case

Sequence	Geom.BD-TotalRate		Attr.BD-TotalRate		
	D1	D2	Luma	Cb	Cr
A.Loot	1.4%	**-4.1**%	0.4%	0.5%	1.0%
A.Red&black	0.2%	**-2.8**%	0.2%	0.3%	0.2%
A.Queen	1.8%	**-3.6**%	0.8%	3.0%	5.8%
B.Longdress	2.0%	**-2.6**%	0.5%	0.0%	0.2%
Class A	1.1%	**-3.5**%	0.5%	1.2%	2.3%
Class B	2.0%	**-2.6**%	0.5%	0.0%	0.2%
Avg. All	1.4%	**-3.3**%	0.5%	0.9%	1.8%

To evaluate the effect of introducing the geometry as an additional input, we compare the proposed Half OGCNN with the Occupancy Network illustrated in Fig. 20. The Occupancy Network uses only the quarter-resolution occupancy map video as input. Note that to ensure fairness, the Half OGCNN and the Occupancy Network use the same network configurations. Table 24 shows the comparison of the Half OGCNN with and without the Geometry Network. Compared to the Occupancy Network, the Half OGCNN saves an average of 3.3% BD-rate when D2 is used as the quality metric, while suffering a few performance losses of attributes. These performance results demonstrate that the geometry, as an additional input, can lead to clear benefits.

3.5.6 Number of points

To further demonstrate that the proposed OGCNN can reduce the number of noisy points, we count the numbers of points N_Rs in the reconstructed 3D point clouds under different algorithms in Fig. 22. The Y axis is the bitrate, which gradually increases from low bitrate $r1$ to high bitrate $r5$. We can see that for all dynamic point clouds, the number of points N_R of our proposed Half OGCNN and Full OGCNN are less than those of the V-PCC anchor, SOTAs, and Occupancy Network. These statistics fully demonstrate that the proposed OGCNN removes noisy points to improve the R-D performance.

3.5.7 Rate-Distortion Curves

Fig. 23 shows some representative geometry R-D curves from the all intra case. We can see that the D2 PSNRs of the proposed OGCNN at all five rate points are higher than those of the V-PCC anchor, SOTAs and Occupancy Network. These experimental results demonstrate that the proposed OGCNN is significantly superior to the V-PCC anchor, SOTAs, and Occupancy Network.

3.5.8 Visual results of the 2D occupancy maps

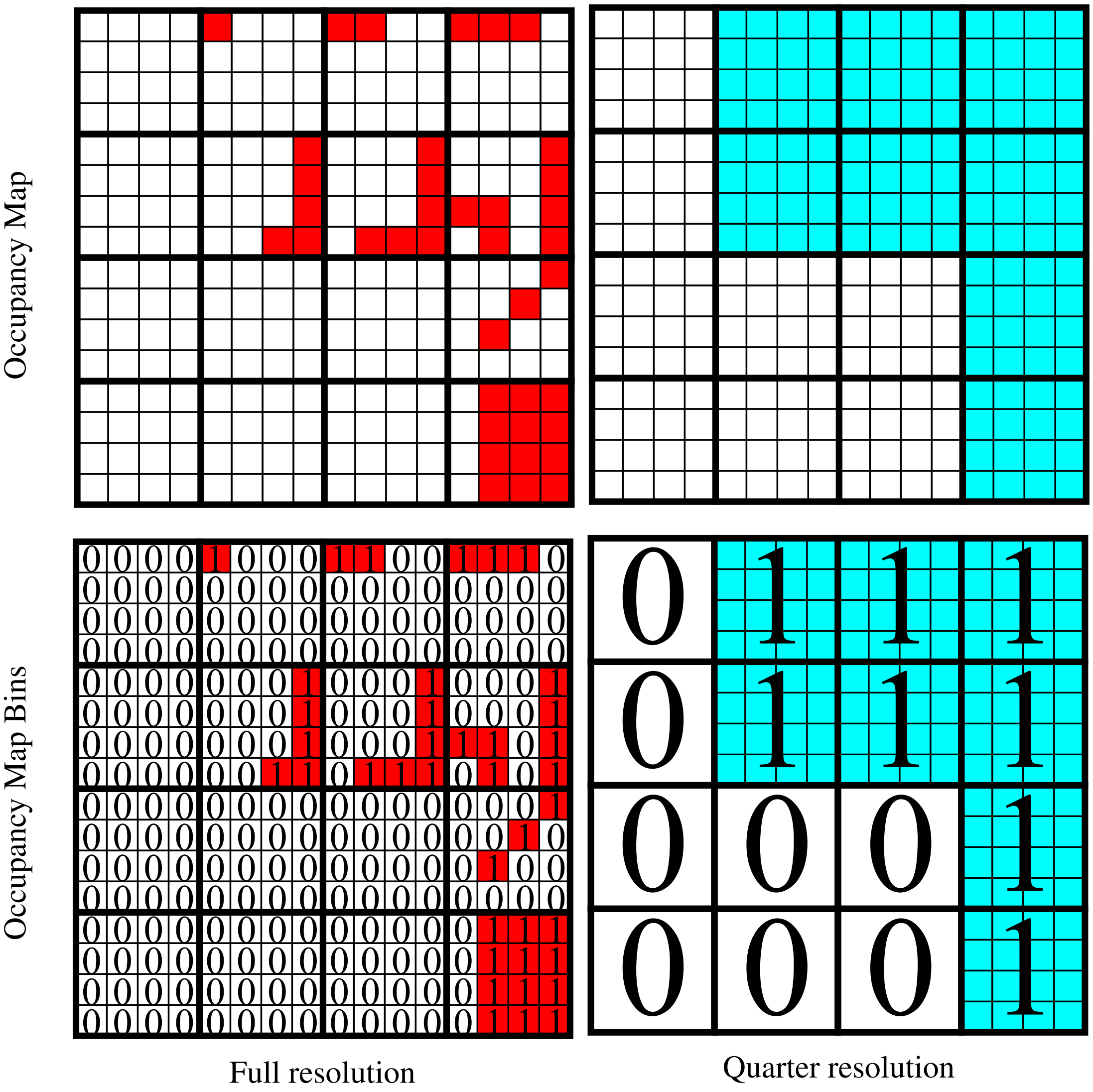
Fig. 24 shows the 2D occupancy map video comparison of the ground truth, V-PCC anchor, and proposed Full OGCNN. The reconstructed occupancy map videos are derived from the first frames of Loot and Longdress. For Loot, (a), (b), (c) and (d) are the occupancy map reconstructions of the Full OGCNN and the V-PCC anchor, the difference between the two, and the ground truth, respectively. (e) and (f) are the enlarged areas of the gold and blue blocks in (c). For Longdress, the same order is followed. In (c) and (i), the green pixels denote the unoccupied pixels of the V-PCC anchor correctly removed by the Full OGCNN. The red pixels denote the occupied pixels of the V-PCC anchor wrongly removed by the Full OGCNN. We can see from (c) and (i) that the number of green pixels is much greater than the number of red pixels. The 2D occupancy map results demonstrate that the proposed OGCNN can remove many noisy points and very few original points.

3.5.9 Visual results of the 3D point clouds

Fig. 25 shows a visual comparison of the original point clouds and the point clouds reconstructed by the V-PCC anchor and the proposed Half OGCNN. The zoomed figures are derived from the first frame of Loot, the first frame of RedandBlack, the first frame of Longdress, and the 300th frame of Longdress. From these frames, we can clearly see from the red and green rectangles that there are many noisy points in the reconstructions of the V-PCC anchor. However, the reconstructions of the Half OGCNN are much smoother and closer to the original point clouds. The visual results demonstrate that the proposed OGCNN can achieve a much better subjective quality.

3.6 Summary

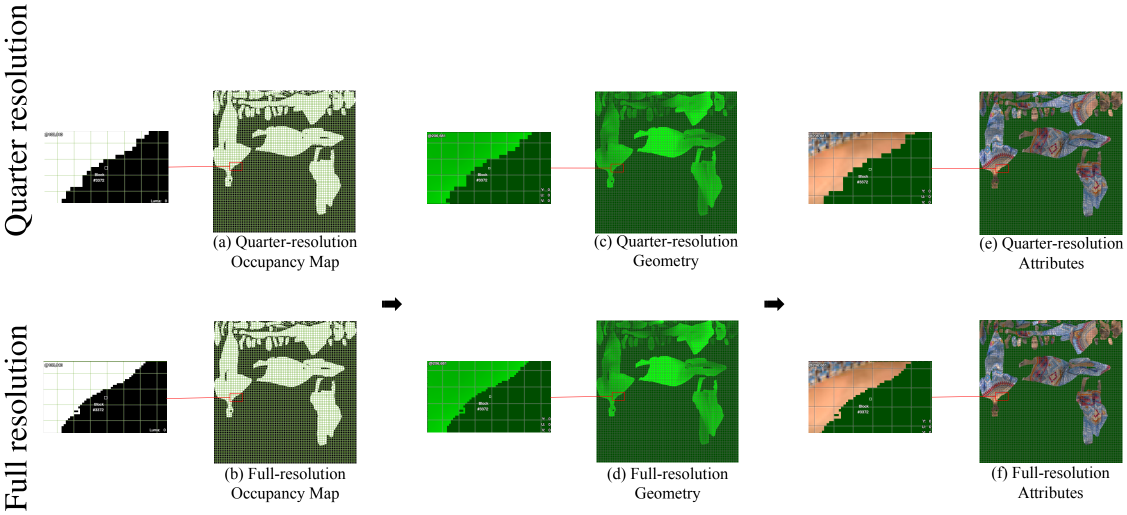
In this paper, we first point out that the accuracy of the occupancy map video is important to the quality of reconstructed point clouds under video-based point cloud compression (V-PCC). Then, we propose an occupancy-geometry-based convolutional neural network (OGCNN) to improve the occupancy map accuracy. We formulate the problem of improving occupancy map accuracy as a binary segmentation problem. In addition to the quarter-resolution occupancy map video, we use the reconstructed geometry video as the other input. The experimental results show that our proposed OGCNN approach presents clear accuracy improvements in the occupancy map video and leads to significant BD-rate savings compared to the state-of-the-art schemes. To the best of our knowledge, this is the first CNN-based work on improving the performance of V-PCC. We will consider more CNN-based algorithms to improve the performance of V-PCC in the future.



0	0	0	0	1	0	0	0	1	1	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	0	0	0	1	1	0	1	1	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

0	1	1	1
0	1	1	1
0	0	0	1
0	0	0	1

Figure 18: Occupancy map comparison of the full resolution and quarter resolution with the same occupancy distribution. A grid represents a pixel in the occupancy map video. The bold border square is denoted as a 4×4 block. The red pixels indicate the occupied pixels in the full-resolution occupancy map video, while the blue pixels indicate the occupied pixels in the reconstructed full-resolution occupancy map video.



(a) Quarter-resolution Occupancy Map

(c) Quarter-resolution Geometry

(e) Quarter-resolution Attributes

(b) Full-resolution Occupancy Map

(d) Full-resolution Geometry

(f) Full-resolution Attributes

Figure 19: Comparison of the occupancy maps, geometry, and attributes of the quarter-resolution and full-resolution videos. The reconstructions of the occupancy maps, geometry, and attributes in the first and second rows are derived from the configurations of the quarter-resolution and full-resolution videos, respectively. We can see from the enlarged areas that, compared to the full resolution, the edges of the body in the quarter-resolution occupancy map, geometry and attribute videos show a more serious zigzag artifact.

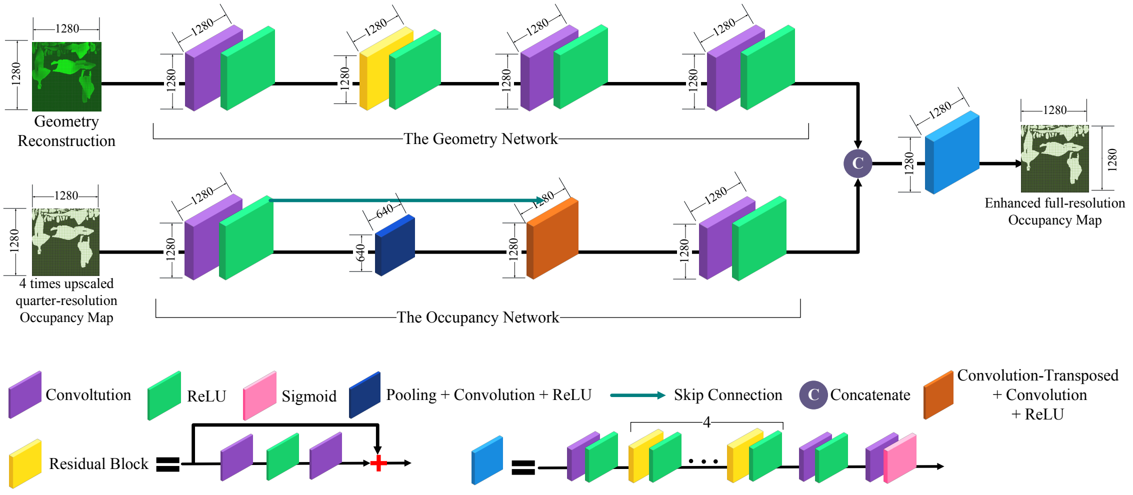

1280
Geometry Reconstruction
The Geometry Network
4 times upscaled quarter-resolution Occupancy Map
640
The Occupancy Network
C
Enhanced full-resolution Occupancy Map
Convoltution
ReLU
Sigmoid
Pooling + Convolution + ReLU
Skip Connection
Concatenate
Convolution-Transposed + Convolution + ReLU
Residual Block
4

Figure 20: The proposed OGCNN framework includes two subnetworks: the Occupancy Network and the Geometry Network. The Occupancy Network uses the quarter-resolution occupancy map video as input. It derives occupancy segmentation feature maps from the occupancy map. The Geometry Network uses the reconstructed geometry video as input and derives geometry segmentation feature maps from the geometry. The occupancy map and geometry segmentation features are then concatenated together and used as the input of the remaining convolutional layers.

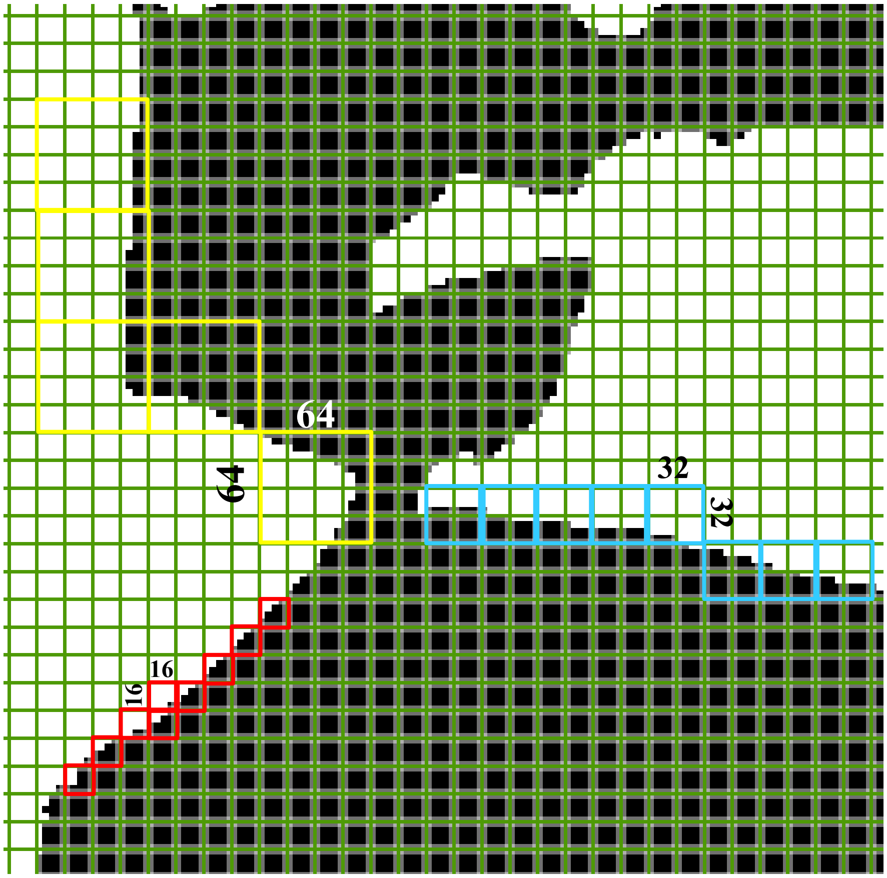
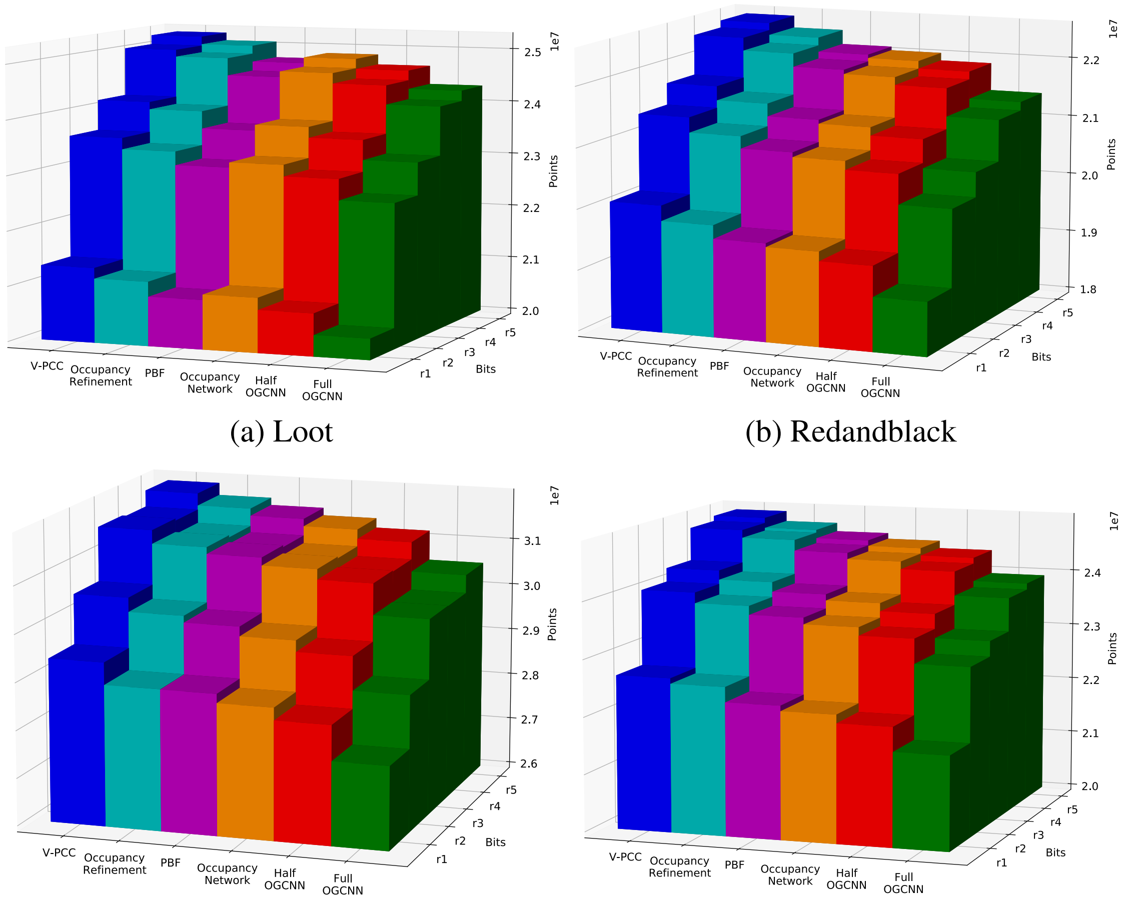


Figure 21: Occupancy map boundary blocks. A boundary block is an $N \times N$ square that consists of both occupied and unoccupied pixels. The white grids represent the occupied pixels, while the black grids represent the unoccupied pixels. The red, blue, yellow squares indicate the 16×16, 32×32, and 64×64 boundary blocks, respectively.



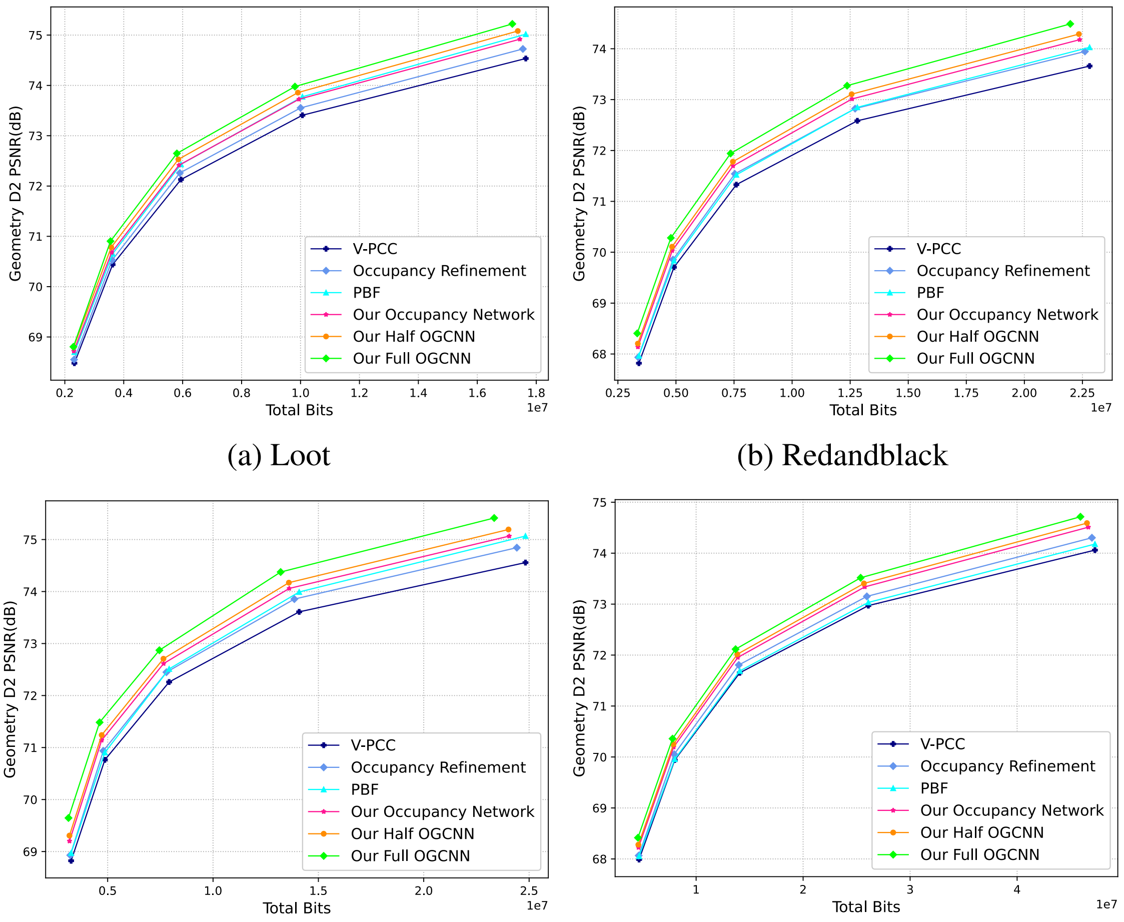
(a) Loot

(b) Redandblack

(c) Queen

(d) Longdress

Figure 22: Comparison of the numbers of points N_R in the reconstructed 3D point clouds of the V-PCC [2], OR [3], PBF [4], Occupancy Network and OGCNN. The Y axis is the bitrate, which gradually increases from low bitrate $r1$ to high bitrate $r5$. We can see that for all dynamic point clouds, the number of points N_R in our proposed Half OGCNN and Full OGCNN are less than those in the V-PCC anchor, SOTAs, and the Occupancy Network.



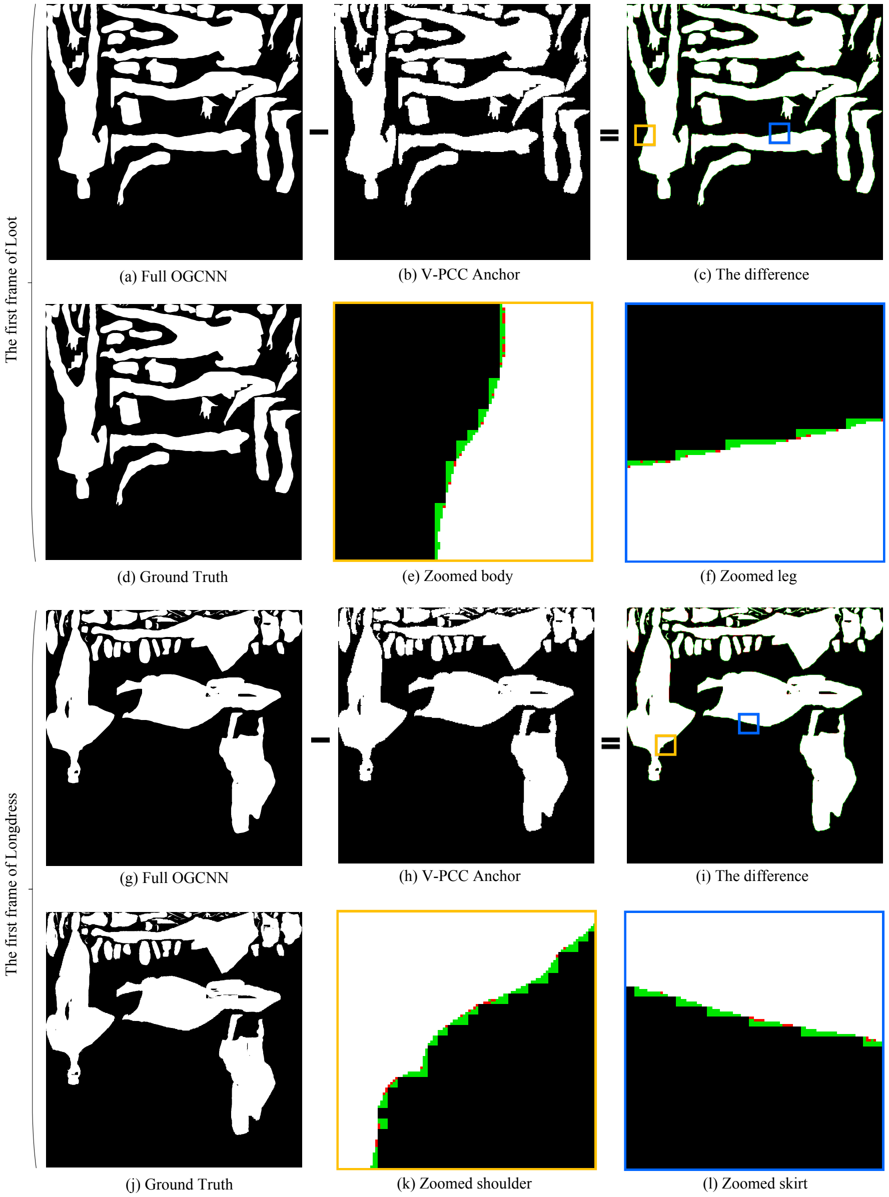
(a) Loot

(b) Redandblack

(c) Queen

(d) Longdress

Figure 23: Geometry R-D curve comparison of the V-PCC [2], OR [3], PBF [4], Occupancy Network and OGCNN for the all intra case. We can see that the D2 PSNRs of the proposed OGCNN at all five rate points are higher than those of the V-PCC anchor, SOTAs and Occupancy Network.



(a) Full OGCNN

(b) V-PCC Anchor

(c) The difference

(d) Ground Truth

(e) Zoomed body

(f) Zoomed leg

(g) Full OGCNN

(h) V-PCC Anchor

(i) The difference

(j) Ground Truth

(k) Zoomed shoulder

(l) Zoomed skirt

Figure 24: 2D occupancy map comparison of the ground truth, V-PCC anchor [2] and proposed Full OGCNN. For Loot, (a), (b), (c) and (d) are the occupancy map reconstructions of the Full OGCNN and the V-PCC anchor, the difference between the two, and the ground truth, respectively. (e) and (f) are the enlarged areas of the gold and blue blocks in (c). For Longdress, the same order is followed. In (c) and (i), the green pixels denote the unoccupied pixels of the V-PCC anchor correctly removed by the Full OGCNN. The red pixels denote the occupied pixels of the V-PCC anchor wrongly removed by the Full OGCNN.

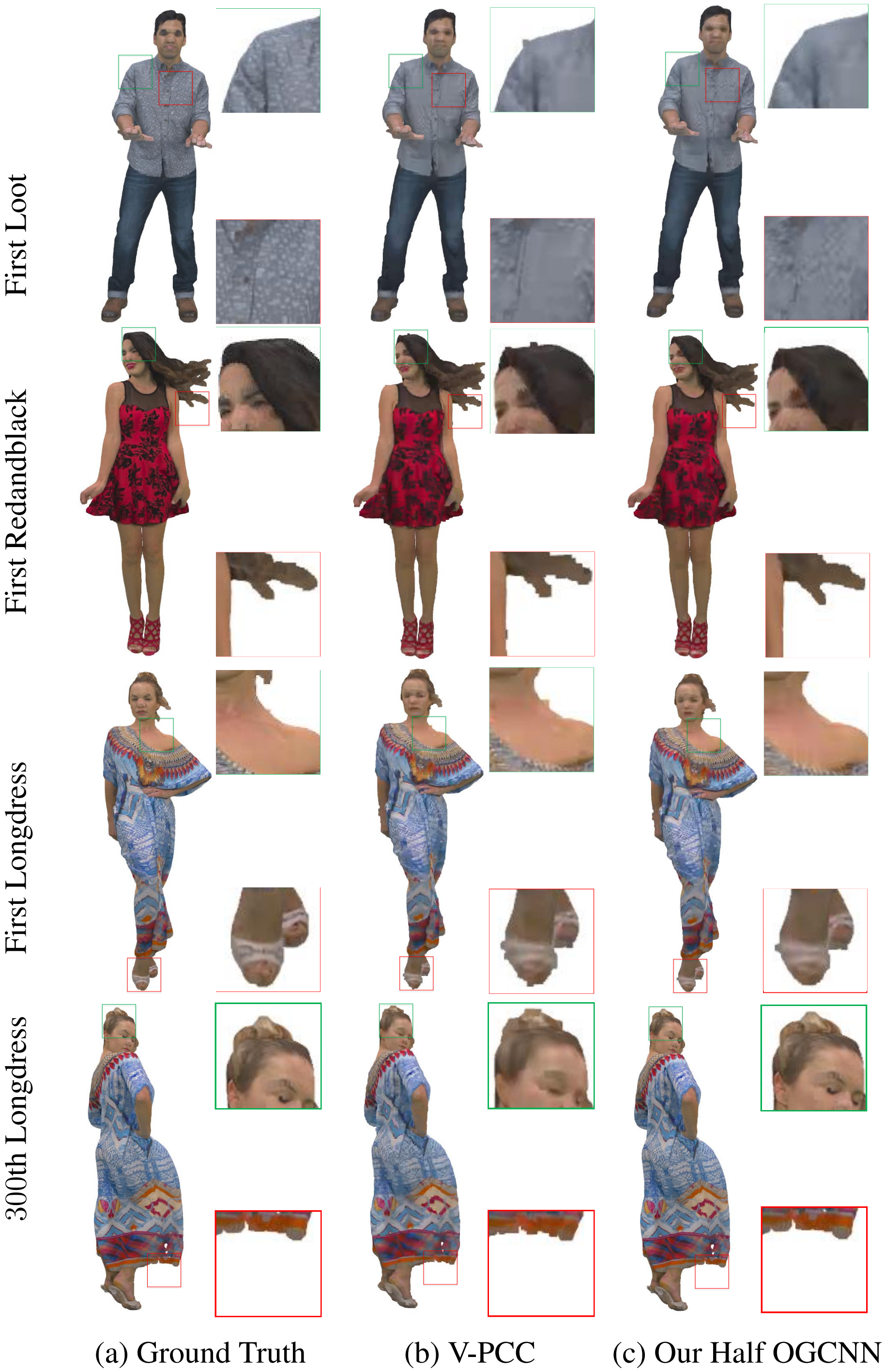


Figure 25: 3D visual comparison of the original point clouds, the point clouds reconstructed by the V-PCC anchor and the proposed Half OGCNN.

REFERENCE LIST

[1] "V-Pcc Codec Description," *Document ISO/IEC JTC1/SC29/WG11 W19526, Italy*, Sep. 2020.

[2] Point Cloud Compression Category 2 Reference Software Tmc2-11.0. [Online]. Available: Http://MPEGx.Int-Evry.Fr/Software/MPEG/Pcc/Tm/MPEG-Pcc-Tmc2

[3] C. Guede, J. Ricard, J. Llach, J.-C. Chevet, Y. Olivier, and D. Gendron, "Improve point cloud compression through occupancy map refinement," *Document ISO/IEC JTC1/SC29/WG11 MPEG2018/ M44779, Macao, China*, 2018.

[4] P. Andrivon, J. Ricard, C. Guede, O. Nakagami, D. Graziosi, and A. Tabatabai, "Patch Border Filtering Specification In V-Pcc," *Document ISO/IEC JTC1/SC29/WG11 M51501, Geneva, Ch*, Oct. 2020.

[5] W. Lin, X. He, X. Han, D. Liu, J. See, J. Zou, H. Xiong, and F. Wu, "Partition-aware adaptive switching neural networks for post-processing in hevc," *IEEE Transactions On Multimedia*, vol. 22, no. 11, pp. 2749–2763, 2019.

[6] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition Workshops*, 2017, pp. 136–144.

[7] MPEG. (2019) Point Cloud Compression Category 2 Reference Software Tmc2-8.0. [Online]. Available: Http://MPEGx.Int-Evry.Fr/Software/MPEG/Pcc/Tm/MPEG-Pcc-Tmc2

[8] D. Graziosi and A. Tabatabai, "[V-Pcc] New Contribution On Geometry Padding," *Document ISO/IEC JTC1/SC29/WG11 M47496, Geneva, Ch*, Mar. 2019.

[9] O. Nakagami, "Pcc Tmc2 Low Complexity Geometry Smoothing," *Document ISO/IEC JTC1/SC29/WG11 M43501, Ljubjana, Si*, Jul. 2018.

[10] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview Of The H. 264/Avc Video Coding Standard," *IEEE Transactions On Circuits And Systems For Video Technology*, pp. 560–576, 2003.

[11] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview Of The High Efficiency Video Coding (Hevc) Standard," *IEEE Transactions On Circuits And Systems For Video Technology*, pp. 1649–1668, 2012.

[12] B. Bross, J. Chen, and S. Liu, "Versatile Video Coding (Draft 4)," *Document ITU-T SG 16 Wp 3 And ISO/IEC Jtc 1/Sc 29/Wg 11 Jvet-M1001-V6, Marrakech, Ma*, Jan. 2019.

[13] H. Lim and H. Park, "A Ringing-Artifact Reduction Method For Block-Dct-Based Image Resizing," *IEEE Transactions On Circuits And Systems For Video Technology*, pp. 879–889, 2011.

[14] T.-M. Liu, W.-P. Lee, and C.-Y. Lee, “An In/Post-Loop Deblocking Filter With Hybrid Filtering Schedule,” *IEEE Transactions On Circuits And Systems For Video Technology*, pp. 937–943, 2007.

[15] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Van Der Auwera, “Hevc deblocking filter,” *IEEE Transactions On Circuits And Systems For Video Technology*, pp. 1746–1754, 2012.

[16] C.-M. Fu, C.-Y. Chen, and Y.-W. Huang, “Te10 Subtest 3: Quadtree-Based Adaptive Offset,” *ITU-T/ISO/IEC JCT-VC Document JCTVC-C147*, Oct. 2010.

[17] ——, “Ce8 Subset 3: Picture Quadtree Adaptive Offset,” *ITU-T/ISO/IEC JCT-VC Document JCTVC-D122*, Jan. 2011.

[18] C.-M. Fu, C.-Y. Chen, and C.-Y. Tsai, “Ce13: Sample Adaptive Offset With Lcu-Independent Decoding,” *ITU-T/ISO/IEC JCT-VC Document JCTVC-E049*, Mar. 2011.

[19] C.-Y. Tsai, C.-Y. Chen, T. Yamakage, I. S. Chong, Y.-W. Huang, C.-M. Fu, T. Itoh, T. Watanabe, T. Chujoh, M. Karczewicz, and Others, “Adaptive Loop Filtering For Video Coding,” *IEEE Journal Of Selected Topics In Signal Processing*, pp. 934–945, 2013.

[20] S. Ma, X. Zhang, J. Zhang, C. Jia, S. Wang, and W. Gao, “Nonlocal In-Loop Filter: The Way Toward Next-Generation Video Coding?” *IEEE Multimedia*, pp. 16–26, 2016.

[21] Y. Zhang, T. Shen, X. Ji, Y. Zhang, R. Xiong, and Q. Dai, “Residual Highway Convolutional Neural Networks For In-Loop Filtering In Hevc,” *IEEE Transactions On Image Processing*, pp. 3827–3841, 2018.

[22] G. Lu, W. Ouyang, D. Xu, X. Zhang, Z. Gao, and M.-T. Sun, “Deep Kalman Filtering Network For Video Compression Artifact Reduction,” in *Proceedings Of The European Conference On Computer Vision (Eccv)*, 2018, pp. 568–584.

[23] C. Jia, S. Wang, X. Zhang, S. Wang, J. Liu, S. Pu, and S. Ma, “Content-Aware Convolutional Neural Network For In-Loop Filtering In High Efficiency Video Coding,” *IEEE Transactions On Image Processing*, 2019.

[24] X. He, Q. Hu, X. Zhang, C. Zhang, W. Lin, and X. Han, “Enhancing Hevc Compressed Videos With A Partition-Masked Convolutional Neural Network,” in *2018 25th IEEE International Conference On Image Processing (Icip)*, 2018, pp. 216–220.

[25] W. Jia, L. Li, Z. Li, X. Zhang, and S. Liu, “Residual Guided Deblocking With Deep Learning,” in *Icip 2020 IEEE International Conference On Image Processing (Icip)*, 2020.

[26] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, “Adaptive Deblocking Filter,” *IEEE Transactions On Circuits And Systems For Video Technology*, pp. 614–619, 2003.

[27] Y. Zhang, C. Yan, F. Dai, and Y. Ma, “Efficient parallel framework for h.264/avc deblocking filter on many-core platform,” *IEEE Transactions On Multimedia*, pp. 510–524, 2012.

[28] T. Li, X. He, L. Qing, Q. Teng, and H. Chen, “An iterative framework of cascaded deblocking and superresolution for compressed images,” *IEEE Transactions On Multimedia*, pp. 1305–1320, 2018.

[29] C.-M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsai, C.-W. Hsu, S.-M. Lei, J.-H. Park, and W.-J. Han, “Sample Adaptive Offset In The Hevc Standard,” *IEEE Transactions On Circuits And Systems For Video Technology*, pp. 1755–1764, 2012.

[30] W.-J. Chien and M.Karczewicz, “Adaptive Filter Based On Combination Of Sum-Modified Laplacian Filter Indexing And Quadtree Partitioning,” *ITU-T/ISO/IEC JCT-VC Document VCEG-Al27*, Jul. 2009.

[31] K. Mccann, W.-J. Han, and I.-K. Kim, “Samsung’s Response To The Call For Proposals On Video Compression Technology,” *ITU-T SG16 Wp3 And ISO/IEC JTC1/SC29/WG11 Jct-Vc Document JCTVC-A124, Dresden, De*, Apr. 2010.

[32] Y.-W. Huang, C.-M. Fu, and C.-Y. Chen, “In-Loop Adaptive Restoration,” *ITU-T/ISO/IEC JCT-VC Document JCTVC-B077*, Jul. 2010.

[33] Q. Han, R. Zhang, W.-K. Cham, and Y. Liu, “Quadtree-Based Non-Local Kuan’s Filtering In Video Compression,” *Journal Of Visual Communication And Image Representation*, pp. 1044–1055, 2014.

[34] X. Zhang, R. Xiong, W. Lin, J. Zhang, S. Wang, S. Ma, and W. Gao, “Low-Rank-Based Nonlocal Adaptive Loop Filter For High-Efficiency Video Compression,” *IEEE Transactions On Circuits And Systems For Video Technology*, pp. 2177–2188, 2016.

[35] C. Dong, Y. Deng, C. Change Loy, and X. Tang, “Compression Artifacts Reduction By A Deep Convolutional Network,” in *Proceedings Of The IEEE International Conference On Computer Vision*, 2015, pp. 576–584.

[36] L. Kang, C. Hsu, B. Zhuang, C. Lin, and C. Yeh, “Learning-based joint super-resolution and deblocking for a highly compressed image,” *IEEE Transactions On Multimedia*, pp. 921–934, 2015.

[37] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, “D3: Deep Dual-Domain Based Fast Restoration Of Jpeg-Compressed Images,” in *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*, 2016, pp. 2764–2772.

[38] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, “Video Enhancement With Task-Oriented Flow,” *International Journal Of Computer Vision*, pp. 1–20, 2017.

[39] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, “Detail-Revealing Deep Video Super-Resolution,” in *Proceedings Of The IEEE International Conference On Computer Vision*, 2017, pp. 4472–4480.

[40] Y. Dai, D. Liu, and F. Wu, “A Convolutional Neural Network Approach For Post-Processing In Hevc Intra Coding,” in *International Conference On Multimedia Modeling*, 2017, pp. 28–39.

[41] R. Yang, M. Xu, and Z. Wang, “Decoder-Side Hevc Quality Enhancement With Scalable Convolutional Neural Network,” in *2017 IEEE International Conference On Multimedia And Expo (Icme)*, 2017, pp. 817–822.

[42] R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, “Enhancing Quality For Hevc Compressed Videos,” *IEEE Transactions On Circuits And Systems For Video Technology*, 2018.

[43] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning For Image Recognition,” in *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*, 2016, pp. 770–778.

[44] Y. Wang, H. Zhu, Y. Li, Z. Chen, and S. Liu, “Dense Residual Convolutional Neural Network Based In-Loop Filter For Hevc,” in *2018 IEEE Visual Communications And Image Processing (Vcip)*, 2018, pp. 1–4.

[45] C. Jia, S. Wang, X. Zhang, S. Wang, and S. Ma, "Spatial-Temporal Residue Network Based In-Loop Filter For Video Coding," in *2017 IEEE Visual Communications And Image Processing (Vcip)*, 2017, pp. 1–4.

[46] W.-S. Park and M. Kim, "Cnn-Based In-Loop Filtering For Coding Efficiency Improvement," in *2016 IEEE 12th Image, Video, And Multidimensional Signal Processing Workshop (Ivmsp)*, 2016, pp. 1–5.

[47] T. Wang, M. Chen, and H. Chao, "A Novel Deep Learning-Based Method Of Improving Coding Efficiency From The Decoder-End For Hevc," in *2017 Data Compression Conference (Dcc)*, 2017, pp. 410–419.

[48] Z. Jin, M. Z. Iqbal, D. Bobkov, W. Zou, X. Li, and E. Steinbach, "A flexible deep cnn framework for image restoration," *IEEE Transactions On Multimedia*, pp. 1055–1068, 2020.

[49] A. Veit, M. J. Wilber, and S. Belongie, "Residual Networks Behave Like Ensembles Of Relatively Shallow Networks," in *Advances In Neural Information Processing Systems*, 2016, pp. 550–558.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep Into Rectifiers: Surpassing Human-Level Performance On Imagenet Classification," in *Proceedings Of The IEEE International Conference On Computer Vision*, 2015, pp. 1026–1034.

[51] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, “Variational Lossy Autoencoder,” *Arxiv Preprint Arxiv:1611.02731*, 2016.

[52] G. E. Hinton and R. R. Salakhutdinov, “Reducing The Dimensionality Of Data With Neural Networks,” *Science*, pp. 504–507, 2006.

[53] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks For Biomedical Image Segmentation,” in *International Conference On Medical Image Computing And Computer-Assisted Intervention*, 2015, pp. 234–241.

[54] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional Networks.” in *Cvpr*, 2010.

[55] D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics With Applications*. Cengage Learning, 2014.

[56] A. Ignatov, R. Timofte, and Others, “Pirm Challenge On Perceptual Image Enhancement On Smartphones: Report,” in *European Conference On Computer Vision (Eccv) Workshops*, January 2019.

[57] E. Agustsson and R. Timofte, “Ntire 2017 Challenge On Single Image Super-Resolution: Dataset And Study,” in *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition Workshops*, 2017, pp. 126–135.

[58] F. Bellard. (2019) Ffmpeg Software, A Complete, Cross-Platform Solution To Record, Convert And Stream Audio And Video. [Online]. Available: Http://Ffmpeg.Org/

[59] D. P. Kingma and J. Ba, “Adam: A Method For Stochastic Optimization,” *Arxiv Preprint Arxiv:1412.6980*, 2014.

[60] G. Bjontegaard, “Calculation Of Average Psnr Differences Between Rd-Curves,” *Document VCEG-M33, Austin, Texas, Usa*, April 2001.

[61] K. Sharman and K. Suehring, “Common Test Conditions,” *Document ITU-T SG 16 Wp 3 And ISO/IEC Jtc 1/Sc 29/Wg 11 JCTVC-Ae1100, San Diego, Us*, Apr. 2018.

[62] Y. Li, S. Liu, and K. Kawamura, “Methodology And Reporting Template For Neural Network Coding Tool Testing,” *Document ITU-T SG 16 Wp 3 And ISO/IEC Jtc 1/Sc 29/Wg 11 Jvet-M1006-V1, Marrakech, Ma*, Jan. 2019.

[63] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, “A survey of model compression and acceleration for deep neural networks,” *Arxiv Preprint Arxiv:1710.09282*, 2017.

[64] ——, “Model compression and acceleration for deep neural networks: The principles, progress, and challenges,” *IEEE Signal Processing Magazine*, pp. 126–136, 2018.

[65] J. Cheng, P.-S. Wang, G. Li, Q.-H. Hu, and H.-Q. Lu, “Recent advances in efficient computation of deep convolutional neural networks,” *Frontiers Of Information Technology & Electronic Engineering*, pp. 64–77, 2018.

[66] E. S. Jang, M. Preda, K. Mammou, A. M. Tourapis, J. Kim, D. B. Graziosi, S. Rhyu, and M. Budagavi, “Video-based point-cloud-compression standard in mpeg: From evidence collection to committee draft [standards in a nutshell],” *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 118–123, 2019.

[67] E. D’eon, B. Harrison, T. Myers, and P. Chou, “Input to ad hoc groups on mpeg point cloud compression and jpeg pleno,” *Document ISO/IEC JTC1/SC29/WG11 M40059, Geneva, Switzerland*, 2017.

[68] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu, “Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera,” *Acm Transactions On Graphics (Tog)*, vol. 36, no. 4, p. 1, 2017.

[69] G. Bruder, F. Steinicke, and A. NÜChter, “Poster: Immersive point cloud virtual environments,” in *2014 IEEE Symposium On 3d User Interfaces (3dui)*. IEEE, 2014, pp. 161–162.

[70] C. Tulvan, R. Mekuria, Z. Li, and S. Laserre, “Use cases for point cloud compression (pcc),” *Document ISO/IEC JTC1/SC29/WG11 MPEG2015/ N16331, Geneva, Switzerland*, 2016.

[71] J. Chen, C. Lin, P. Hsu, and C. Chen, “Point cloud encoding for 3d building model retrieval,” *IEEE Transactions On Multimedia*, vol. 16, no. 2, pp. 337–345, 2014.

[72] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3d object detection network for autonomous driving,” in *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*, 2017, pp. 1907–1915.

[73] Mitsubishielectric. (2019) Mobile Mapping System. [Online]. Available: Http://Www.Mitsubishielectric.Com/Bu/Mms/Index.Html

[74] D. Sportillo, A. Paljic, M. Boukhris, P. Fuchs, L. Ojeda, and V. Roussarie, “An immersive virtual reality system for semi-autonomous driving simulation: A comparison between realistic and 6-dof controller-based interaction,” in *Proceedings Of The 9th International Conference On Computer And Automation Engineering*, 2017, pp. 6–10.

[75] H. Fuchs, A. State, and J.-C. Bazin, “Immersive 3d telepresence,” *Computer*, vol. 47, no. 7, pp. 46–52, 2014.

[76] C3dc. (2019) Culture 3d Cloud. [Online]. Available: Http://C3dc.Fr/

[77] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. KrivokuĆA, S. Lasserre, Z. Li, and Others, “Emerging mpeg standards for point cloud compression,” *IEEE Journal On Emerging And Selected Topics In Circuits And Systems*, vol. 9, no. 1, pp. 133–148, 2018.

[78] Y. Sun, M. Liu, and M. Q.-H. Meng, “Improving rgb-d slam in dynamic environments: A motion removal approach,” *Robotics And Autonomous Systems*, vol. 89, pp. 110–122, 2017.

[79] C. Guede, K. Cai, J.Ricard, J. Llach, and J.-C. Chevet, “Geometry Image Coding Improvements,” *Document ISO/IEC JTC1/SC29/WG11 M42111, Gwangju, Korea*, Jan. 2018.

[80] N. Dawar, H. Najaf-Zadeh, R. Joshi, and M. Budagavi, “Pcc Tmc2 Interleaving In Geometry And Texture Layers,” *Document ISO/IEC JTC1/SC29/WG11 M43723, Ljubljana, Slovenia*, Jul. 2018.

[81] S. Rhyu, Y. Oh, and J. Woo, “Pcc Ce2.13 Report On Texture And Depth Padding Improvement,” *Document ISO/IEC JTC1/SC29/WG11 M43667, Ljubjana, Si*, Jul. 2018.

[82] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” *Computational Intelligence And Neuroscience*, vol. 2018, 2018.

[83] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, “Real-time compression of point cloud streams,” in *2012 IEEE International Conference On Robotics And Automation*. IEEE, 2012, pp. 778–785.

[84] D. Thanou, P. A. Chou, and P. Frossard, “Graph-based compression of dynamic 3d point cloud sequences,” *IEEE Transactions On Image Processing*, vol. 25, no. 4, pp. 1765–1778, 2016.

[85] R. L. De Queiroz and P. A. Chou, “Motion-compensated compression of dynamic voxelized point clouds,” *IEEE Transactions On Image Processing*, vol. 26, no. 8, pp. 3886–3895, 2017.

[86] R. Mekuria, K. Blom, and P. Cesar, “Design, implementation, and evaluation of a point cloud codec for tele-immersive video,” *IEEE Transactions On Circuits And Systems For Video Technology*, vol. 27, no. 4, pp. 828–842, 2016.

[87] X. Sun, H. Ma, Y. Sun, and M. Liu, “A novel point cloud compression algorithm based on clustering,” *IEEE Robotics And Automation Letters*, vol. 4, no. 2, pp. 2132–2139, 2019.

[88] M. Budagavi, E. Faramarzi, T. Ho, H. Najaf-Zadeh, and I. Sinharoy, “Samsungs response to cfp for point cloud compression (category 2),” *Document ISO/IEC JTC1/SC29/WG11 M41808, Macau, China*, 2017.

[89] L. He, W. Zhu, and Y. Xu, “Best-effort projection based attribute compression for 3d point cloud,” in *2017 23rd Asia-Pacific Conference On Communications (Apcc)*. IEEE, 2017, pp. 1–6.

[90] S. Lasserre, J. Llach, C. Guede, and J. Ricard, “Technicolor’s response to the cfpp for point cloud compression,” *Document ISO/IEC JTC1/SC29/WG11 M41822, Macau, China*, 2017.

[91] K. Mammou, A. M. Tourapis, D. Singer, and Y. Su, “Video-based and hierarchical approaches point cloud compression,” *Document ISO/IEC JTC1/SC29/WG11 M41649, Macau, China*, 2017.

[92] L. Li, Z. Li, V. Zakharchenko, J. Chen, and H. Li, “Advanced 3d motion prediction for video-based dynamic point cloud compression,” *IEEE Transactions On Image Processing*, vol. 29, pp. 289–302, 2019.

[93] M. Preda, “Report on pcc cfp answers,” *Document ISO/IEC JTC1/SC29/WG11 W17251, Macau, China*, 2017.

[94] C. Tu, E. Takeuchi, C. Miyajima, and K. Takeda, “Compressing continuous point cloud data using image compression methods,” in *2016 IEEE 19th International Conference On Intelligent Transportation Systems (Itsc)*. IEEE, 2016, pp. 1712–1719.

[95] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*, 2017, pp. 652–660.

[96] C. Choy, J. Gwak, and S. Savarese, “4d spatio-temporal convnets: Minkowski convolutional neural networks,” in *Proceedings Of The IEEE/Cvf Conference On Computer Vision And Pattern Recognition*, 2019, pp. 3075–3084.

[97] Z. Gojcic, C. Zhou, J. D. Wegner, L. J. Guibas, and T. Birdal, “Learning multi-view 3d point cloud registration,” in *Proceedings Of The IEEE/Cvf Conference On Computer Vision And Pattern Recognition*, 2020, pp. 1759–1769.

[98] C. Tu, E. Takeuchi, A. Carballo, and K. Takeda, “Point cloud compression for 3d lidar sensor using recurrent neural network with residual blocks,” in *2019 International Conference On Robotics And Automation (Icra)*. IEEE, 2019, pp. 3274–3280.

[99] M. Quach, G. Valenzise, and F. Dufaux, “Learning convolutional transforms for lossy point cloud geometry compression,” in *2019 IEEE International Conference On Image Processing (Icip)*. IEEE, 2019, pp. 4320–4324.

[100] T. Huang and Y. Liu, “3d point cloud geometry compression on deep learning,” in *Proceedings Of The 27th Acm International Conference On Multimedia*, 2019, pp. 890–898.

[101] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, “Octsqueeze: Octree-structured entropy model for lidar compression,” in *Proceedings Of The IEEE/Cvf Conference On Computer Vision And Pattern Recognition*, 2020, pp. 1313–1323.

[102] S. Biswas, J. Liu, K. Wong, S. Wang, and R. Urtasun, “Muscle: Multi sweep compression of lidar using deep entropy models,” in *Advances In Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp.

22 170–22 181. [Online]. Available: Https://Proceedings.Neurips.Cc/Paper/2020/File/\\Fc152e73692bc3c934d248f639d9e963-Paper.Pdf

[103] Y. Olivier and J. Llach, "Per Patch Projection Optimization For Tmc2," *Document ISO/IEC JTC1/SC29/WG11 M43723, San Diego, Ca, Us*, Apr. 2018.

[104] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in *2017 IEEE International Conference On Image Processing (Icip)*. IEEE, 2017, pp. 3460–3464.

[105] S. Schwarz, G. Martin-Cocher, D. Flynn, and M. Budagavi, "Common test conditions for point cloud compression," *Document ISO/IEC JTC1/SC29/WG11 W17766, Ljubljana, Slovenia*, 2018.

[106] A. Vosoughi, S. Yea, and S. Liu, "New proposal on occupancy map recovery using scalable locally adaptive erosion filter," *Document ISO/IEC JTC1/SC29/WG11 MPEG2019/ M46347, Marrakesh, Morocco*, 2019.

[107] R. J. Youngho Oh and M. Budagavi, "Improved point cloud compression through filtering of occupancy map," *Document ISO/IEC JTC1/SC29/WG11 MPEG2019/ M46370, Marrakesh, Morocco*, 2019.

[108] Y.-H. Lee, J.-L. Lin, Y.-C. Chang, C.-C. Ju, Y.-T. Tsai, C.-C. Lin, C.-L. Lin, Y. Oh, R. Joshi, and M. Budagavi, "New proposal on occupancy map refinement using corner-based boundary estimation," *Document ISO/IEC JTC1/SC29/WG11 MPEG2019/ M46389, Marrakesh, Morocco*, 2019.

[109] K. Cai, D. Zhang, V. Zakharcchenko, and J. Chen, “Adaptive occupancy map up-sampling,” *Document ISO/IEC JTC1/SC29/WG11 MPEG2019/ M46455, Marrakesh, Morocco*, 2019.

[110] H. Najaf-Zadeh, M. Budagavi, R. Joshi, and Y. Oh, “Constrained occupancy map trimming using a ternary occupancy map,” *Document ISO/IEC JTC1/SC29/WG11 MPEG2019/ M47593, Geneva, Ch*, 2019.

[111] L. Li, Z. Li, S. Liu, and H. Li, “Efficient projected frame padding for video-based point cloud compression,” *IEEE Transactions Multimedia*, 2020.

[112] S.-P. Wang, Y.-T. Tsai, C.-C. Lin, C.-L. Lin, Y.-H. Lee, J.-L. Lin, Y.-C. Chang, and C.-C. Ju, “Bounding box shifting for occupancy map generation,” *Document ISO/IEC JTC1/SC29/WG11 MPEG2019/ M47766, Geneva, Ch*, 2019.

[113] L. Li, Z. Li, S. Liu, and H. Li, “Occupancy-map-based rate distortion optimization and partition for video-based point cloud compression,” *IEEE Transactions On Circuits And Systems For Video Technology*, 2020.

[114] “V-Pcc Codec Description,” *Document ISO/IEC JTC1/SC29/WG11 W19526, Virtual, Italy*, Sep. 2020.

[115] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A tutorial on the cross-entropy method,” *Annals Of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.

VITA

Wei Jia was born on November 14, 1986 in Xi An, Shaanxi. He received the B.S. and M.S. degrees in Electronic Engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2009 and 2012, respectively.

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