Convolutional Neural Network-Based Synthesized View Quality Enhancement for 3D Video Coding

Linwei Zhu, *Student Member, IEEE*, Yun Zhang, *Senior Member, IEEE*, Shiqi Wang, *Member, IEEE*, Hui Yuan, *Senior Member, IEEE*, Sam Kwong, *Fellow, IEEE*, and Horace H.-S. Ip

Abstract—**The quality of synthesized view plays an important role in the 3D video system. In this paper, to further improve the coding efficiency, a convolutional neural network (CNN)-based synthesized view quality enhancement method for 3D high efficiency video coding (HEVC) is proposed. First, the distortion elimination in synthesized view is formulated as an image restoration task with the aim to reconstruct the latent distortion free synthesized image. Second, the learned CNN models are incorporated into 3D HEVC codec to improve the view synthesis performance for both view synthesis optimization (VSO) and the final synthesized view, where the geometric and compression distortions are considered according to the specific characteristics of synthesized view. Third, a new Lagrange multiplier in the rate-distortion cost function is derived to adapt the CNN-based VSO process to embrace a better 3D video coding performance. Extensive experimental results show that the proposed scheme can efficiently eliminate the artifacts in the synthesized image, and reduce 25.9% and 11.7% bit rate in terms of peak-signal-to-noise ratio and structural similarity index, which significantly outperforms the state-of-the-art methods.**

Index Terms—**Convolutional neural network, view synthesis, depth coding, 3D high efficiency video coding, Lagrange multiplier.**

I. Introduction

RECENT years, the demand of the Three Dimensional (3D) videos that are able to offer the immersion feeling to the users has dramatically increased. In contrast to the traditional 2D video, the multiple viewpoints double or redouble the data volume in the 3D video system, which result in vast proliferation of video data and bring great challenges to transmission and storage. Fortunately, the representation of Multi-view plus Depth (MVD) [1] provides an efficient solution to 3D video system. With the help of depth information, various virtual viewpoints between two reference viewpoints can be generated by the technique of Depth Image Based Rendering (DIBR) [2], [3], such that the video data at the positions of the synthesized views are not required to be transmitted. To further improve the coding efficiency, the standard of 3D extensions of High Efficiency Video Coding (3D HEVC) [1] has been issued [4], [5] for joint texture and depth encoding. Due to the fact that depth map is not eventually watched by the users, the quality of the depth should not be straightforwardly evaluated by the Mean Squared Error (MSE) between the original and distorted versions. Instead, the Synthesized View Distortion Change (SVDC) [6] is measured and utilized in the Rate Distortion (RD) optimization for depth coding, and the View Synthesis Optimization (VSO) is performed to optimize the depth coding performance [6]. Moreover, to reduce the computational complexity from view synthesis operation while maintaining the coding efficiency simultaneously, the View Synthesis Distortion (VSD) [7] is estimated based on the depth map fidelity and texture characteristics.

To further improve the quality of depth as well as the synthesized image, numerous algorithms have been proposed in literatures, which can be divided into three categories. The first category preprocesses the depth map because of its imperfect quality. In particular, in the depth estimation or depth camera acquisition process, inevitable artifacts may be generated, leading to distortions in the synthesized image even without the compression of the depth map. In [8], a depth video pre-processing algorithm was presented to enhance the consistency with a low pass filter in the temporal domain. Lee and Effendi [9] devised an adaptive smoothing filter for the depth map according to the characteristics of the hole region, where the geometric distortion and hole occurrence were efficiently reduced. Methods in the second category improve the view synthesis algorithm, and most of them focus on the hole filling process. In [10], a view-spatial-temporal

Manuscript received November 16, 2017; revised May 24, 2018 and July 1, 2018; accepted July 5, 2018. Date of publication July 20, 2018; date of current version August 14, 2018. This work was supported in part by the Natural Science Foundation of China under Grant 61672443, Grant 61471348, and Grant 61571274, in part by the Hong Kong RGC General Research Fund under Grant 9042322 (CityU 11200116) and Grant 9042489 (CityU 11206317), in part by the Hong Kong RGC Early Career Scheme under Grant 9048122 (CityU 21211018), in part by the City University of Hong Kong under Grant 7200539/CS, in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant 2016A030306022, in part by the Project for Shenzhen Science and Technology Development under Grant JSGG20160229202345378, in part by the Shenzhen International Collaborative Research Project under Grant GJHZ20170314155404913, in part by the Guangdong Provincial Science and Technology Development under Grant 2017B010110014, in part by the Guangdong Special Support Program for Youth Science and Technology Innovation Talents under Grant 2014TQ01X345, in part by the Membership of Youth Innovation Promotion Association, Chinese Academy of Sciences, under Grant 2018392, and in part by the Shandong Natural Science Funds for Distinguished Young Scholar under Grant JQ201614. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Catarina Brites. *(Corresponding author: Sam Kwong.)*

L. Zhu, S. Wang, S. Kwong, and H. H.-S. Ip are with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the Shenzhen Research Institute, City University of Hong Kong, Shenzhen 518057, China (e-mail: lwzhu2-c@my.cityu.edu.hk; shiqwang@cityu.edu.hk; cssamk@cityu.edu.hk; cship@cityu.edu.hk).

Y. Zhang is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: yun.zhang@siat.ac.cn).

H. Yuan is with the School of Information Science and Engineering, Shandong University, Jinan 250100, China (e-mail: huiyuan@sdu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2858022

post-refinement method was proposed to fill the holes and remove the boundary artifacts. Zhu and Li [11] provided a fundamental analysis of holes generation mechanism, and took advantages of the visible and invisible background information together to perform hole filling. To exploit the temporal correlation for hole filling in view synthesis, Rahaman and Paul [12] proposed a new view synthesis technique, where various models in the Gaussian Mixture Modeling (GMM) were adopted to separate background and foreground pixels. Then the missing part could be filled in with the adaptive weighted average of the pixel values from the associated models of the GMM and the warped images. The final one is the synthesized image post-processing. Due to the distortion introduced by texture and depth coding, various types of distortions are mixed together in the synthesized image. To eliminate these distortions, the wiener filter was utilized for synthesized image quality improvement [13]. Furthermore, an in-loop filter [14] was devised to minimize VSD at the cost of transmitting extra filter parameters which are utilized as supplementary information to improve the quality of synthesized image. However, the existing approaches improve the quality of the synthesized view by regarding the synthesized view from pristine texture and depth as the reference. This has largely ignored the distortions introduced in depth generation and view synthesis.

Due to the substantial success of deep learning in signal processing tasks, Convolutional Neural Network (CNN) has been widely applied in the field of image restoration, especially for the compression distortion reduction. The compression Artifacts Reduction CNN (AR-CNN) [15] model was firstly proposed to reduce the distortion in the compressed image caused by JPEG compression, and approximately 1 dB gain in terms of Peak-Signal-to-Noise Ratio (PSNR) had been achieved. Zhang *et al.* [16] constructed a feed-forward CNN for image denoising with blind Gaussian noise, where the residual learning and batch normalization were both adopted to accelerate the CNN training. Park and Kim [17] proposed an in-loop filter using CNN to improve the coding efficiency and subjective visual quality. Furthermore, the Variable-filter-size Residue-learning CNN (VR-CNN) [18] was utilized to replace the Sample Adaptive Offset (SAO) [19] for post-processing in HEVC intra coding, which achieved 4.6% bit rate reduction on average. In [20], the compression artifacts were removed and the details of HEVC-compressed videos were enhanced by a CNN equipped with a fully end-to-end feed forward architecture. Basically, the above mentioned image denoising and artifact reduction are all CNN based schemes, which have been used for general Gaussian noise removal as well as JPEG and HEVC compression artifacts elimination.

Due to the fact that the depth map is not perceived eventually, a number of depth distortion evaluation schemes have been proposed based on the VSD estimation. The VSD was estimated as a function of depth coding error in [21], which was further applied in the RD optimization process. An analytical model in [22] was proposed to estimate the virtual VSD caused by depth error, where the distance between reference and virtual viewpoints were both taken into account. A fine VSD estimation approach was presented in [23], where the depth distortion as well as the texture gradient of the co-located texture were considered. Besides the VSD estimation, a virtual view PSNR estimation method was presented in [24]. In [25], through theoretical analysis and practical view synthesis simulation, VSD was proved to be non-monotonically related with the texture distortion while monotonically related with the depth distortion. These derivations were utilized to improve the viewing performance of 3D video. Tech *et al.* [26] proposed a partial depth image based re-rendering scheme for VSD calculation, which was adopted in the 3D HEVC reference software. Oh *et al.* [27] presented an efficient depth coding approach with the VSD, which emphasized on the quality of synthesized image and around 10% bit rate was reduced for overall multi-view texture plus depth videos. In [28], a solution to the optimal depth map down-sampling problem was derived, in which the target was to minimize the depth-caused distortion in the synthesized image. Liu *et al.* [29] provided a synthesized video quality database with distortions from encoded texture and encoded depth, then a full reference objective video quality assessment metric was proposed. Based on the full reference synthesized video quality assessment metric, an improved RD Optimization (RDO) algorithm [30] was devised aiming at minimizing the perceptual distortion of synthesized view under a given bit rate budget. Generally speaking, these aforementioned methods mainly focus on the distortion derivation of the synthesized view. However, as the quality of the reference synthesized view in VSO is imperfect due to the view synthesis process, the 3D-HEVC coding performance can be further boosted from the perspective of reference synthesized view quality improvement.

In view of these limitations, we concentrate on improving the quality of synthesized image based on the physically acquired image, and propose a novel CNN based synthesized view quality enhancement approach for 3D HEVC. The contributions of this paper are listed as follows:

1) We formulate the distortion elimination in synthesized image as an image restoration task, and the learned CNN models based on the captured video at the synthesized view are applied to improve the synthesized view quality.
2) We incorporate the CNN models into 3D HEVC codec to improve the view synthesis performance for both VSO and final synthesized view.
3) The Lagrange multiplier in the RD cost function is derived to adapt to the CNN based VSO process, which further improves the 3D video coding performance.

The remainder of this paper is organized as follows. The motivations are presented in Section II. Section III proposes the scheme of CNN based synthesized view quality enhancement in 3D HEVC and the CNN training is discussed in detail in Section IV. Experimental results are demonstrated in Section V. Section VI concludes this paper.

II. Problems and Motivations

In the MVD data format, the virtual viewpoints are typically generated with DIBR. One of the key techniques in DIBR is 3D warping [3], which maps the pixel $\mathbf{p}_1 = \{x_1, y_1, 1\}^T$

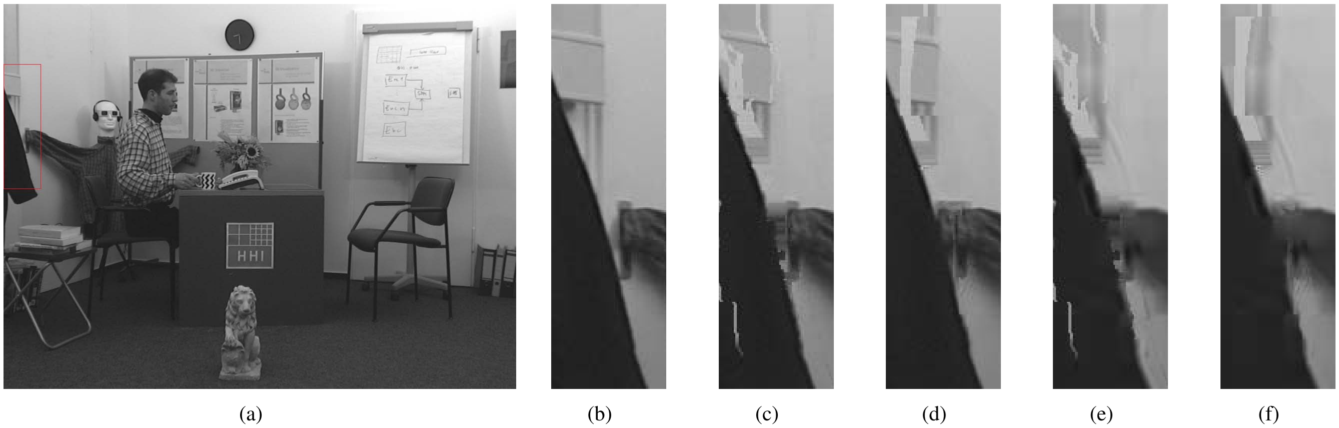


Fig. 1. Comparisons of the view 8 in Bookarrival sequence (enlarged for better visualization). (a) 1^{st} frame of Bookarrival sequence; (b) The image captured by camera at view 8; (c) The synthesized image from views 6, 10 with original texture and original depth; (d) The synthesized image from views 6, 10 with original texture and encoded depth ($QP_d = 49$); (e) The synthesized image from views 6, 10 with encoded texture and original depth ($QP_t = 45$); (f) The synthesized image from views 6, 10 with encoded texture and encoded depth $\{(QP_t, QP_d) = (45, 49)\}$.

in reference viewpoint to the pixel $\mathbf{p}_2 = \{x_2, y_2, 1\}^T$ in the virtual viewpoint,

$$s_2\mathbf{p}_2 = s_1\mathbf{A}_2\mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{A}_1^{-1}\mathbf{p}_1 - \mathbf{A}_1\mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{t}_1 + \mathbf{A}_2\mathbf{t}_2, \qquad (1)$$

where s_1 is the depth value of reference viewpoint at position (x_1,y_1) and s_2 is a scaling factor. Moreover, $\mathbf{A}_1$ and $\mathbf{A}_2$ are intrinsic parameters, $\mathbf{R}_1$ and $\mathbf{R}_2$ are rotation matrices, and $\mathbf{t}_1$ and $\mathbf{t}_2$ are translation vectors of the reference and virtual cameras, respectively.

In essence, the parameters of $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{t}_1$ and $\mathbf{t}_2$ remain constants when the positions of the reference and virtual viewpoints are fixed. Therefore, the distortion of the synthesized view originates from both the depth and texture degradations. In particular, if the quality of the depth map is degraded, the location of the pixel $\mathbf{p}_2$ will be shifted, which can be regarded as the warping distortion in view synthesis. Moreover, the compression artifacts induced in the texture image of the reference viewpoint will also directly influence the quality of the synthesized image. Experiments of view synthesis are further conducted to vividly prove the above hypotheses. As shown in Fig. 1, the synthesized images under different settings are illustrated, where (QP_t, QP_d) indicate the Quantization Parameters (QPs) for texture and depth encoding. It can be observed that there still exist artifacts even in the image synthesized from original texture and original depth. Moreover, when both the warping distortion and the texture coding distortion are encountered, the quality of the synthesized view is severely degraded.

In 3D HEVC, the measurement of the distortion from depth coding is not only determined by depth map distortion between the original and the reconstructed depth blocks D_d, but also by the SVDC D_s. As such, the RD cost function is formulated as [1]

$$\min\{J\} \quad \text{where } J = \eta_s D_s + \eta_d D_d + \lambda R, \qquad (2)$$

where η_s and η_d are weighting factors, λ is the Lagrange multiplier for mode decision and R is the depth coding bit. Here, D_d can be obtained by the Sum of Squared Differences (SSD) between the original depth map $\mathbf{S}_d$ and encoded depth map $\tilde{\mathbf{S}}_d$,

$$D_d = \sum_{(x,y)} \mathbf{D}_d(x, y) = \sum_{(x,y)} [\mathbf{S}_d(x, y) - \tilde{\mathbf{S}}_d(x, y)]^2. \qquad (3)$$

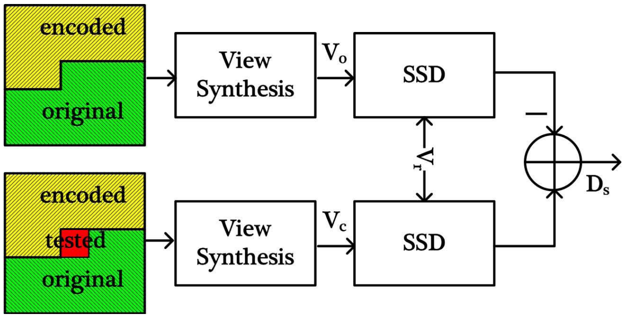


Fig. 2. Illustration of synthesized view distortion change [6].

D_s can be calculated by the variation of a synthesized block when the depth map is modified from original to distorted values. More specifically, the definition of SVDC D_s [6] is shown in Fig. 2, and it can be formulated by,

$$D_s = \sum_{(x,y)} [\mathbf{V}_c(x, y) - \mathbf{V}_r(x, y)]^2 - \sum_{(x,y)} [\mathbf{V}_o(x, y) - \mathbf{V}_r(x, y)]^2, \qquad (4)$$

where $\mathbf{V}_o$ is a synthesized view from depth maps consisting of encoded depth data in already encoded blocks and original depth data in the to-be-encoded block, $\mathbf{V}_c$ is a synthesized view from the depth map containing the distorted depth data for the current block, and $\mathbf{V}_r$ is the reference synthesized view rendered from the original texture and depth.

It is worth mentioning that in the real application scenario the image at the virtual viewpoint is not always available in the case of MVD coding. Instead, the image synthesized from original texture and depth is used for reference. However, artifacts may be inevitably introduced, as shown in Fig. 1(c), and this may make the RDO process less efficient. In view of this, we aim at improving the quality of the

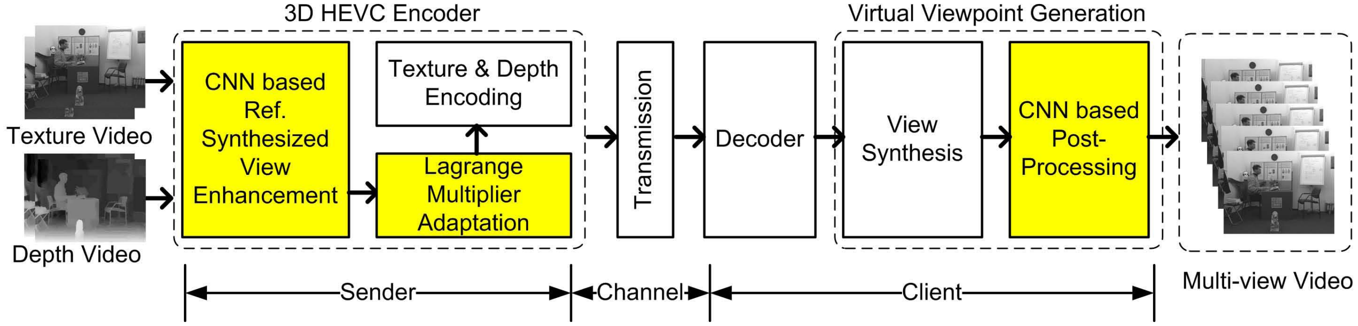


Fig. 3. Proposed framework of the CNN based synthesized view quality enhancement in 3D video coding.

reference synthesized image from the perspective of approaching the captured view as the reference with CNN. Accordingly, new Lagrange multiplier adaptation is mandatory while applying this reference enhancement to 3D-HEVC, since the distortion D_s has been changed. Otherwise, the RD performance may not be optimal. In addition, from the experimental results shown in Fig. 1(f), it can be found that the warping distortion and coding distortion make the quality of the final synthesized view bad. A post-processing method is also needed to eliminate these distortions.

III. Proposed CNN Based Synthesized View Quality Enhancement in 3D HEVC

According to the above analysis, a CNN based synthesized view quality enhancement method is presented in this paper to eliminate the artifacts in the synthesized view. The proposed framework is illustrated in Fig. 3, and the proposed schemes are highlighted, including CNN based reference synthesized view enhancement, Lagrange multiplier adaptation and CNN based post-processing.

A. CNN Based Reference Synthesized View Enhancement

As discussed in Section II, the synthesized image from original texture and original depth is regarded as $\mathbf{V}_r$ to measure the depth coding distortion in VSO. The inevitably introduced artifacts in $\mathbf{V}_r$ motivates us to further improve its quality. Let $\mathbf{Y}$ denote the ground truth which is captured by camera, the optimal filter that aims to restore $\mathbf{V}_r$ to the perfect quality image $\mathbf{Y}$ can be obtained as follows,

$$\Theta_n^* = \arg\min_{\Theta_n} \|\mathbf{Y} - \mathbf{V}_n\|^2, \quad \mathbf{V}_n = \Psi_n(\mathbf{V}_r, \mathbf{L}_n, \mathbf{R}_n|\Theta_n), \quad (5)$$

where $\mathbf{V}_n$ is the enhanced result from $\mathbf{V}_r$, Θ_n is the filtering parameter, and Ψ_n is a filtering function. Due to the mechanism of view synthesis [2], the pristine texture images of left and right reference viewpoints, $\mathbf{L}_n$ and $\mathbf{R}_n$, are introduced as the inter-view information. The wiener filter, which has been widely used in video coding, can be regarded as an instance of Ψ_n. However, the parameter Θ_n cannot be generally applied to a wide range of sequences, because it relays on the content of the frame.

Therefore, a more sophisticated filtering algorithm is desired. Inspired by the image denoising [16] with a CNN model, we propose to enhance $\mathbf{V}_r$ by a CNN model due to its substantial performance improvement in many signal processing tasks. Moreover, for the learning based approach, the benefit is that it can learn the features adaptively with large amounts of training data. In Eq. (5), Ψ_n is regarded as a CNN model, and Θ_n is the whole parameter of CNN, including weight and bias. As such, we can have the following conclusion according to [16],

$$\|\mathbf{Y} - \mathbf{V}_n\|^2 < \|\mathbf{Y} - \mathbf{V}_r\|^2. \quad (6)$$

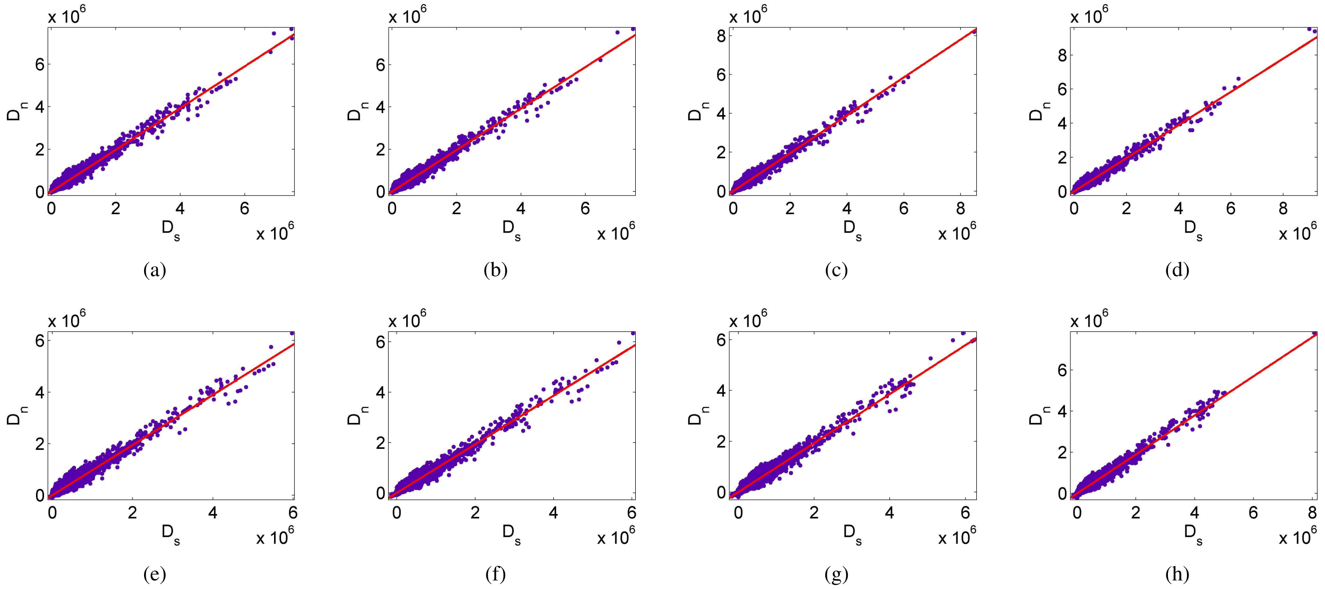
This indicates that $\mathbf{V}_n$ is closer to the ground truth $\mathbf{Y}$ than $\mathbf{V}_r$. In this manner, based on the inspiration that better reference in distortion calculation will lead to better RD optimization, the SVDC in the Eq. (4) is redefined as

$$D_n = \sum_{(x,y)} [\mathbf{V}_c(x,y) - \mathbf{V}_n(x,y)]^2 - \sum_{(x,y)} [\mathbf{V}_o(x,y) - \mathbf{V}_n(x,y)]^2. \quad (7)$$

In 3D video system, the number and the position of synthesized views are not available in the encoding process. As such, the positions of synthesized views are always fixed, and the assumption of 3 synthesized views is typically adopted in the VSO process. In other words, three synthesized views with the same interval will be generated by the associated depth (to be encoded), and the VSD is the average value of the differences between the generated synthesized views and reference synthesized views. Different from traditional filters, the CNN model for enhancing reference synthesized view can be used for any given sequences, not just for a specific one. In addition, the position of synthesized view is not limited any more. It means that it is applicable to cross synthesized view case.

B. Lagrange Multiplier Adaptation in 3D HEVC

Due to the fact that reference synthesized view has been enhanced in Section III-A, the trade-off between synthesized view distortion and the coding bit in the RD cost function should be further adjusted. In [31], since various pattern modes were included for video coding, the Lagrange multiplier was adjusted for better coding performance. As the pattern modes only require fewer bits when compared with other modes, the adjusted Lagrange multiplier suggested by [31] signifies less importance in bits compared to the distortion, which is



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

(e) (f) (g) (h)

Fig. 4. The relationship between D_n and D_s under different QP settings. (a) $(QP_t, QP_d) = (27, 36)$, $\omega = 0.9828$; (b) $(QP_t, QP_d) = (30, 39)$, $\omega = 0.9609$; (c) $(QP_t, QP_d) = (32, 41)$, $\omega = 0.9555$; (d) $(QP_t, QP_d) = (35, 42)$, $\omega = 0.9537$; (e) $(QP_t, QP_d) = (37, 43)$, $\omega = 0.9534$; (f) $(QP_t, QP_d) = (40, 45)$, $\omega = 0.9457$; (g) $(QP_t, QP_d) = (42, 46)$, $\omega = 0.9439$; (h) $(QP_t, QP_d) = (45, 49)$, $\omega = 0.9285$.

able to maintain the similar quality. This motivates us to derive the Lagrange multiplier accordingly,

$$\min\{J\} \quad \text{where } J = \eta_s D_n + \eta_d D_d + \lambda_{new} R, \tag{8}$$

where λ_{new} is the new Lagrange multiplier when the distortion D_n is adopted. To derive λ_{new}, the relationship of D_n and D_s is firstly analyzed. According to Eqs. (4) and (7), we build the relationship between D_n and D_s with a parameter of ω,

$$\frac{D_n}{D_s} = \frac{\sum_{(x,y)} \mathbf{M}(x,y)[\mathbf{V}_c(x,y)+\mathbf{V}_o(x,y)-2\mathbf{V}_n(x,y)]}{\sum_{(x,y)} \mathbf{M}(x,y)[\mathbf{V}_c(x,y)+\mathbf{V}_o(x,y)-2\mathbf{V}_r(x,y)]} = \omega, \tag{9}$$

where,

$$\mathbf{M}(x,y) = \mathbf{V}_c(x,y) - \mathbf{V}_o(x,y). \tag{10}$$

In particular, the experiments are also conducted. The relationship between D_n and D_s is shown in Fig. 4, in which the data can be collected from the sequences of *Kendo*, *Lovebird1*, *Pantomime*, *Poznan_Hall2* and *Poznan_Carpark*. For simplicity, the relationship is fitted in a linear way to derive λ_{new} in global. The fitting accuracies reach 0.9744, 0.9690, 0.9743, 0.9881, 0.9723, 0.9793, 0.9700 and 0.9977 under (QP_t,QP_d) of (27, 36), (30, 39), (32, 41), (35, 42), (37, 43), (40, 45), (42, 46), and (45, 49) for texture and depth coding, respectively. The QP pairs for texture and depth coding are recommended by the Common Test Conditions (CTC) [32].

With the relationship between D_n and D_s in Eq. (9), the new Lagrange multiplier λ_{new} can be achieved by taking the derivative of Eq. (8) with respect to R and setting to 0,

$$\begin{aligned}\lambda_{new} &= -\frac{\partial(\eta_s D_n + \eta_d D_d)}{\partial R} = -\frac{\partial(\eta_s \omega D_s + \eta_d D_d)}{\partial R} \\ &= -[\omega\eta_s \frac{\partial D_s}{\partial R} + \eta_d \frac{\partial D_d}{\partial R}].\end{aligned} \tag{11}$$

Similarly, the derivative of Eq. (2) with respect to R is calculated and set to 0,

$$\lambda = -\frac{\partial(\eta_s D_s + \eta_d D_d)}{\partial R} = -[\eta_s \frac{\partial D_s}{\partial R} + \eta_d \frac{\partial D_d}{\partial R}]. \tag{12}$$

From Eqs. (11) and (12), it can be observed that the only difference lies in the parameter of ω.

In [7], the VSD was estimated by depth map fidelity and horizontal gradient of texture of reference viewpoint. For simplicity, here D_s is represented by VSD to obtain the relationship between λ and λ_{new},

$$D_s \approx \sum_{(x,y)} \frac{1}{4}\alpha^2 \mathbf{D}_d(x,y)(\nabla_{\mathbf{T}(x,y)})^2, \tag{13}$$

where α is a constant when the reference and virtual viewpoints are selected. $\mathbf{D}_d(x,y)$ can be calculated by Eq.(3), and ∇_T is the gradient of the encoded texture of reference view. More specifically, α is calculated by [7]

$$\alpha = \frac{fL}{255}(\frac{1}{Z_{near}} - \frac{1}{Z_{far}}), \tag{14}$$

where f is the focal length, L is the distance between reference and virtual viewpoints, Z_{near} and Z_{far} are the nearest and farthest depth value, respectively. The horizontal gradient $\nabla_{\mathbf{T}(x,y)}$ is calculated as [7]

$$\nabla_{\mathbf{T}(x,y)} = |\tilde{\mathbf{S}}_t(x,y)-\tilde{\mathbf{S}}_t(x-1,y)|+|\tilde{\mathbf{S}}_t(x,y)-\tilde{\mathbf{S}}_t(x+1,y)|, \tag{15}$$

where $\tilde{\mathbf{S}}_t$ is the encoded texture of reference viewpoint.

Analogous to [30], with the relationship between D_s and D_d in Eq. (9), Mathematical Expectation is applied to Eqs. (11) and (12), then the new Lagrange multiplier λ_{new} can be achieved as follows,

$$\lambda_{new} = \lambda\frac{\omega\mu + \eta_d}{\mu + \eta_d}, \tag{16}$$

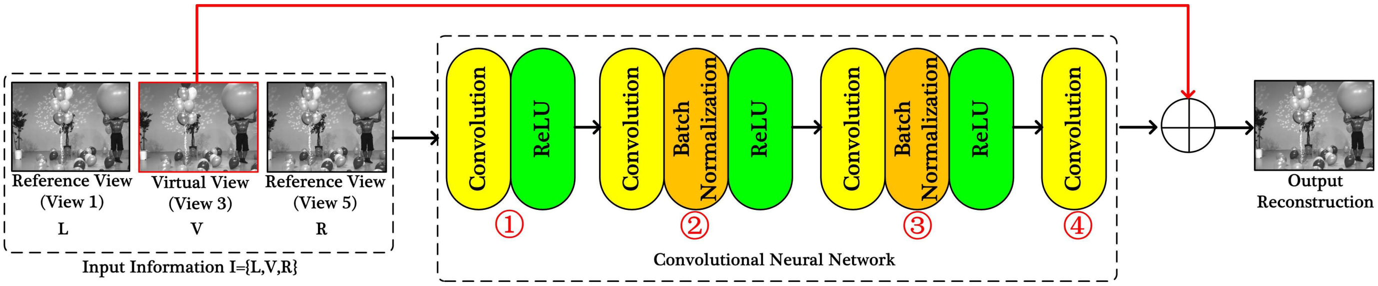


Fig. 5. Proposed architecture of CNN model.

where μ can be represented as

$$\mu = \frac{1}{W \times H} \sum_{(x,y)} \frac{1}{4} \eta_s \alpha^2 (\nabla_{\mathbf{T}(x,y)})^2. \tag{17}$$

Here W and H indicate the width and height of the block.

According to Eq. (16), it can be found that the new Lagrange multiplier λ_{new} equals to the original λ if ω is 1. The parameters of η_d, η_s, W, H, and α can be regarded as constants. The horizontal gradient $\nabla_{\mathbf{T}(x,y)}$ plays an important role in the relationship between λ_{new} and λ. If the value of $\nabla_{\mathbf{T}(x,y)}$ is very large, μ will be much greater than η_d, then $\lambda_{new} = \omega\lambda$. While if the value of $\nabla_{\mathbf{T}(x,y)}$ is approaching zero, μ will be much less than η_d, then $\lambda_{new} = \lambda$. Here, the calculation of $\nabla_{\mathbf{T}(x,y)}$ can be directly extracted from the 3D HEVC encoder because of the calculation of VSD, leading to ignorable computational complexity.

C. CNN Based Post-Processing

In analogies to reference synthesized view enhancement, the post-processing of the synthesized view at the decoder side is also able to reduce the artifacts. The difference lies in that mixed distortions with both warping distortion and compression distortion are introduced in the synthesized view at the decoder side. As such, different CNN training strategies should be adopted for the post-processing process.

Here, the CNN model is used for post-processing of the synthesized view. Suppose $\mathbf{V}_e$ is the synthesized result from encoded texture and encoded depth, and the synthesized result after post-processing $\mathbf{V}_p$ can be represented as

$$\mathbf{V}_p = \Psi_p(\mathbf{V}_e, \mathbf{L}_e, \mathbf{R}_e | \Theta_p), \tag{18}$$

where Θ_p is the parameter of CNN and Ψ_p is the CNN model for post-processing. Different from reference synthesized view enhancement, the original texture image of reference viewpoint is unavailable, and $\mathbf{L}_e$ and $\mathbf{R}_e$ represent the encoded texture images of left and right reference viewpoints. The parameter of CNN is achieved as follows,

$$\Theta_p^* = \arg\min_{\Theta_p} \|\mathbf{Y} - \mathbf{V}_p\|^2, \tag{19}$$

where $\mathbf{Y}$ is the ground truth captured by camera. In this paper, different CNN models for different distortion levels are trained, which will be discussed in Section IV.

IV. Convolutional Neural Networks Training

The architecture of proposed CNN model is designed by four convolutional layers, as shown in Fig. 5. In the first layer, there are three images of input and 64 feature maps of output with filtering window size of 3×3. Due to the mechanism of DIBR, the left and right reference views are added as input to the CNN model for providing useful pixel information from inter-view domain. The input $\mathbf{I}$ includes the distorted synthesized image $\mathbf{V}$ as well as the texture images of the left and right reference viewpoints, $\mathbf{L}$ and $\mathbf{R}$, *i.e.*, $\mathbf{I} = \{\mathbf{L}, \mathbf{V}, \mathbf{R}\}$. The outputs are non-linear mapped by activation function of Rectified Linear Unit (ReLU). As such, the processing of the first layer is represented by,

$$\Psi_1(\mathbf{I}|\mathbf{W}_1, \mathbf{B}_1) = ReLU(\mathbf{W}_1 * f(\mathbf{I}) + \mathbf{B}_1), \tag{20}$$

where $\mathbf{W}_1$ and $\mathbf{B}_1$ are the weight and bias in the first layer. The symbol "*" indicates convolution operation, and the activation function is given by,

$$ReLU(x) = \max(0, x), \tag{21}$$

where $\max(\cdot)$ returns the maximum value. Moreover, $\mathbf{I}$ is normalized to [0, 1] as follows,

$$f(\mathbf{I}) = \mathbf{I}/(2^n - 1), \tag{22}$$

where n represents the bit-depth of $\mathbf{I}$.

For the second and third layers, the inputs are the outputs of prior layers, and the outputs are 64 feature maps after batch normalization [33] and ReLU. Again, the filtering window sizes of these two layers are both set as 3×3, and the process can be formulated as follows,

$$\Psi_i(\mathbf{I}|\mathbf{W}_i, \mathbf{B}_i) = ReLU(BN(\mathbf{W}_i * \Psi_{i-1}(\mathbf{I}|\mathbf{W}_{i-1}, \mathbf{B}_{i-1}) + \mathbf{B}_i)), \tag{23}$$

where $\mathbf{W}_i$ and $\mathbf{B}_i$ are the weight and bias in the second and third layers, $(i \in \{2, 3\})$. $BN(\cdot)$ indicates the operation of batch normalization.

In contrast with the other layers, in the last layer only the convolution operation is performed without batch normalization and ReLU. Moreover, the residual learning is considered, and the synthesized image with distortion is added to the output from the last layer as the final enhanced view. The process of the last layer is formulated as

$$\Psi_4(\mathbf{I}|\mathbf{W}_4, \mathbf{B}_4) = \mathbf{W}_4 * \Psi_3(\mathbf{I}|\mathbf{W}_3, \mathbf{B}_3) + \mathbf{B}_4 + f(\mathbf{V}), \tag{24}$$

TABLE I
SEQUENCES FOR TRAINING AND TESTING

ID	Set 1	Set 2	Sequence	Resolution	Frame Rate	Reference Viewpoints	Virtual Viewpoints
1	Test	Train	Bookarrival	1024×768	16.67	6, 10	8
2	Train	Test	Kendo		30	1, 5	3
3	Test	Train	Balloons				
4	Train	Test	Lovebird1			4, 8	6
5	Test	Train	Newspaper			2, 6	4
6	Train	Test	Pantomime	1280×960		37, 41	39
7	Test	Train	Champagne_Tower				
8	Train	Test	Poznan_Hall2	1920×1088	25	5, 7	6
9	Test	Train	Poznan_Street			3, 5	4
10	Train	Test	Poznan_Carpark				

where $\mathbf{W}_4$ and $\mathbf{B}_4$ are the weight and bias in the fourth layers. We also clip the final pixel value into the valid range as follows

$$\Psi(\mathbf{I}|\Theta) = \min(\max(0, f^{-1}(\Psi_4(\mathbf{I}|\mathbf{W}_4, \mathbf{B}_4))), 2^n - 1), \quad (25)$$

where Θ denotes the whole parameter set of CNN, Ψ is the CNN model, $\min(\cdot)$ returns the minimum value, and the function $f^{-1}(\mathbf{I})$ is the inverse operation of $f(\mathbf{I})$,

$$f^{-1}(\mathbf{I}) = \mathbf{I} \times (2^n - 1). \quad (26)$$

The objective of training is to minimize MSE between ground truth and the predicted one using the Stochastic Gradient Descent (SGD) with error backpropagation algorithm,

$$\ell(\Theta) = \frac{1}{N}\sum_{m=1}^{N} \|\Psi_4(\mathbf{I}_m|\mathbf{W}_4, \mathbf{B}_4) - f(\mathbf{Y}_m)\|^2, \quad (27)$$

where N is the number of batch sizes.

As described in Section III, the synthesized view quality enhancement is implemented into both VSO and post-processing modules. For the training data of CNN based reference synthesized view enhancement in VSO, the images are synthesized by the original texture and depth. As such, one synthesized image ($\mathbf{V}_r$), two original texture images of left and right reference viewpoints ($\mathbf{L}_n, \mathbf{R}_n$) and associated ground truth are formed as a training pair. For the training data of CNN based post-processing, the texture and depth videos of reference viewpoints are jointly encoded by the 3D HEVC Test Model version 16.2 (HTM 16.2) [34] under four QP pairs of (QP_t,QP_d), (30, 39), (35, 42), (40, 45), and (45, 49), following the CTC [32]. Four distorted levels of synthesized videos are generated by the encoded texture and depth. One synthesized image ($\mathbf{V}_e$), two encoded texture images of left and right reference viewpoints ($\mathbf{L}_e, \mathbf{R}_e$) and associated ground truth are incorporated as a training pair. For each distortion level, the CNN model is trained individually.

As shown in Table I, ten multi-view sequences with different contents and resolutions are adopted. The synthesized images of training and testing are generated by the 1D-FAST view synthesis software [34] and the ground truth images are physically captured at the same viewpoint. More specifically, two sets are defined for cross-validation, *i.e.*, Set 1 and Set 2. In each set, five multi-view sequences are used for training and the remaining ones are used for testing. Ten frames (the 1^{th}, 11^{th}, …, and 91^{th} frames) of each training sequence are selected. We set the patch size as 32 × 32 with the stride of 16, and the batch size is set to be 128. Therefore, in total there are 1662 batches with 212736 patches. The Tensorflow package is utilized for CNN training on Tesla K80 GPU with 100 epoches (1662 × 100 = 166200 iterations). The learning rate is set as 1×10^{-4}. It takes about 5 hours to train each CNN model. The training loss converge curves are shown in Figs. 6 and 7. It can be found that when the iteration is greater than 15×10^4, the training of CNN models all converge.

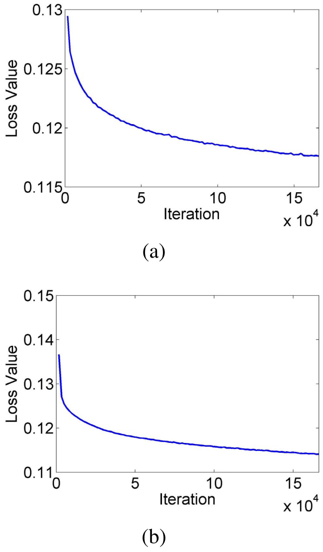
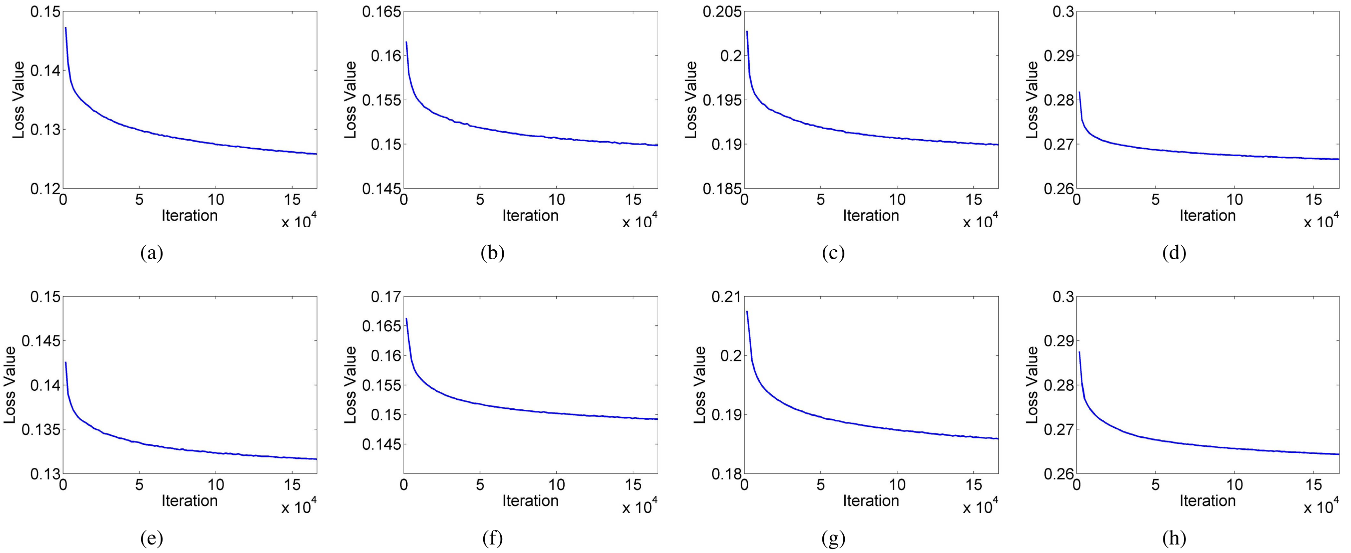


Fig. 6. Illustration of the training loss converge for the synthesized view quality improvement (reference synthesized view enhancement). (a) Training Set1; (b) Training Set2.

V. EXPERIMENTAL RESULTS AND ANALYSES

In this section, experiments are conducted on the platform of HTM 16.2 [34], based on which the proposed CNN model has been implemented to improve the quality of the synthesized view. The CNN based post-processing is also implemented in the 1D-FAST view synthesis software [34]. The multi-view sequences, as listed in Table I, are encoded with four QP pairs of (QP_t,QP_d) for texture and depth, including (30, 39), (35, 42), (40, 45) and (45, 49), under CTC [32]. The VSO and VSD are both enabled under default configuration. All the encoding experiments are performed on the computer equipped with the Intel Core i7-4790 CPU @ 3.60GHz, 8GB memory, Windows 7 Enterprise 64-bit operating system. The original



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

(e) (f) (g) (h)

Fig. 7. Illustration of the training loss converge for the synthesized view quality improvement (post-processing). (a) $(QP_t,\ QP_d) = (30, 39)$, training Set1; (b) $(QP_t,QP_d) = (35, 42)$, training Set1; (c) $(QP_t,\ QP_d) = (40, 45)$, training Set1; (d) $(QP_t,QP_d) = (45, 49)$, training Set1; (e) $(QP_t,\ QP_d) = (30, 39)$, training Set2; (f) $(QP_t,\ QP_d) = (35, 42)$, training Set2; (g) $(QP_t,\ QP_d) = (40, 45)$, training Set2; (h) $(QP_t,\ QP_d) = (45, 49)$, training Set2.

HTM 16.2 is utilized as the anchor for RD performance comparison, in which the baseline model with synthesized view V_r has been equipped. The values of PSNR and Structural SIMilarity (SSIM) index [35] are both calculated between synthesized image and the original one captured by camera for luma component. The RD performance is measured by Bjøntegaard Delta Bit Rate (BD-BR) [36], and a positive value implies the RD performance degradation and vice versa. It should be noted that the CNN models in the proposed method are not compressed and sent to decoder when they are implemented in the 3D HEVC codec. The CNN models used for enhancing reference synthesis view and post-processing are embedded in the encoder and decoder, respectively.

TABLE II

PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART CNN MODELS IN TERMS OF PSNR [UNIT: dB]

Set	Seq.	Original	AR-CNN	VR-CNN	Proposed
Set1	1	35.34	35.47	**35.86**	35.65
	3	33.16	33.31	33.25	**33.88**
	5	27.86	27.91	27.91	**28.35**
	7	27.59	27.61	**28.69**	28.52
	9	35.15	35.25	35.50	**35.72**
	AVG	31.82	31.91	32.24	**32.42**
Set2	2	35.72	35.82	35.98	**36.02**
	4	29.33	29.86	**30.43**	30.05
	6	34.80	36.10	**36.68**	35.46
	8	35.80	35.70	35.85	**36.21**
	10	31.63	31.80	31.77	**33.00**
	AVG	33.46	33.86	34.14	**34.15**
AVERAGE		32.64	32.89	33.19	**33.29**

TABLE III

PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART CNN MODELS IN TERMS OF SSIM

Set	Seq.	Original	AR-CNN	VR-CNN	Proposed
Set1	1	0.9264	0.9309	0.9360	**0.9367**
	3	0.9535	0.9544	0.9563	**0.9616**
	5	0.8859	0.8876	0.8912	**0.8942**
	7	0.9282	0.9391	**0.9442**	0.9409
	9	0.9233	0.9266	0.9300	**0.9323**
	AVG	0.9235	0.9277	0.9315	**0.9331**
Set2	2	0.9619	0.9629	0.9652	**0.9696**
	4	0.9141	0.9196	**0.9252**	0.9214
	6	0.9667	0.9734	**0.9761**	0.9726
	8	0.9216	0.9260	0.9290	**0.9377**
	10	0.9027	0.9059	0.9080	**0.9185**
	AVG	0.9334	0.9376	0.9407	**0.9440**
AVERAGE		0.9285	0.9327	0.9361	**0.9386**

A. CNN Model Validation for Intermediate Synthesized View

Firstly, the proposed CNN model for synthesized view quality enhancement is validated by the comparisons with the state-of-the-art compression artifact reduction and image denoising CNN models, *i.e.*, AR-CNN [15] and VR-CNN [18]. Here the input texture and depth for view synthesis are not compressed. The detailed architectures of AR-CNN and VR-CNN can be found in [15] and [18]. AR-CNN is proposed to reduce the compression distortion from JPEG codec, while VR-CNN is used as an in-loop filter to replace the de-blocking and SAO in HEVC intra coding. Compared with AR-CNN and VR-CNN, the proposed CNN model has more input information from left and right reference views, and the kernel sizes of different convolutional layers are always set as 3 × 3. The multi-view sequences, listed in Table I, are adopted for testing. The results are shown in Tables II and III, respectively. In Table II, it can be found that the proposed scheme achieves better quality improvement results compared to AR-CNN and VR-CNN on average. Similar results can be observed in Table III in terms of SSIM. For the special video sequences, *i.e.*, *Pantomime* and *Champagne_tower*, there is no texture information in the background. The proposed CNN does not perform better than VR-CNN in this case. The reason is that the proposed CNN is equipped with fixed kernel of convolutional layers while VR-CNN has various kernels of convolutional layers. In addition, the visual quality comparisons are provided in Figs. 8 and 9, where it can be observed that the artifacts

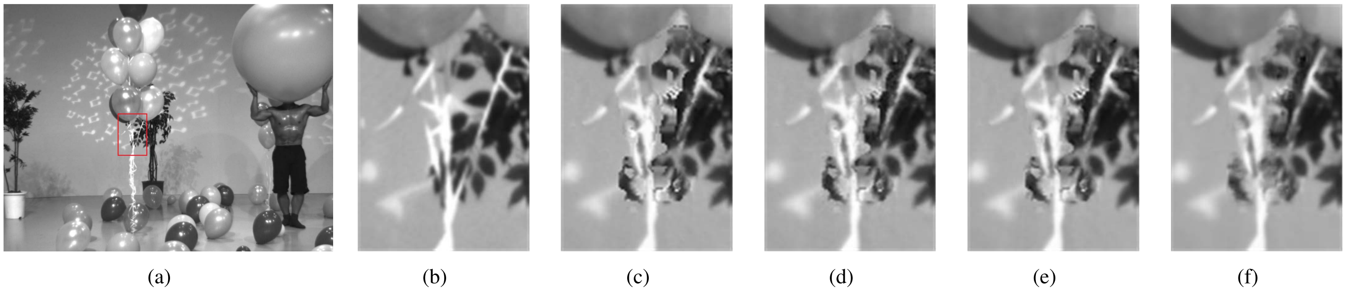


Fig. 8. Visual quality comparisons for the synthesized view (Balloons). (a) The 1^{st} frame of Balloons sequence; (b) The image captured by camera; (c) The original synthesized image; (d) The synthesized image (processed by AR-CNN); (e) The synthesized image (processed by VR-CNN); (f) The synthesized image processed by the proposed CNN.

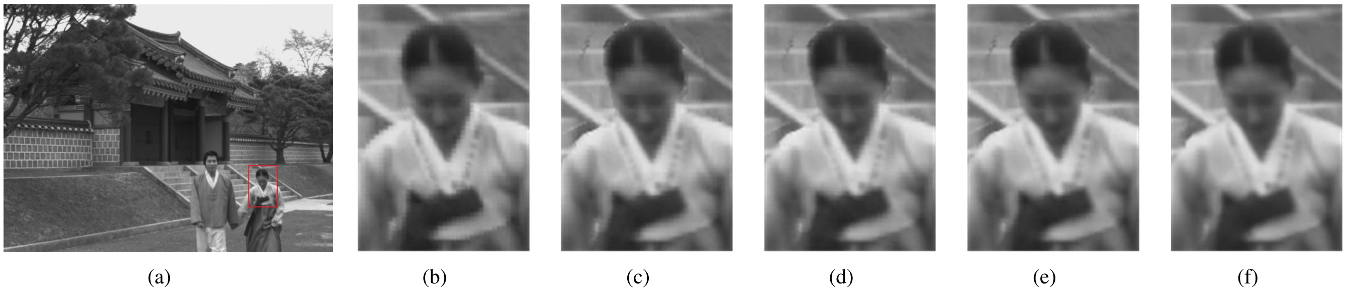


Fig. 9. Visual quality comparisons for the synthesized view (Lovebird1). (a) The 4^{th} frame of Lovebird1 sequence; (b) The image captured by camera; (c) The original synthesized image; (d) The synthesized image (processed by AR-CNN); (e) The synthesized image (processed by VR-CNN); (f) The synthesized image processed by the proposed CNN.

in the synthesized view from original texture and original depth can be significantly eliminated by the proposed CNN. Moreover, although the synthesized view quality have been improved by AR-CNN and VR-CNN, strict observers may find that certain parts of the synthesized view are still distorted, and the visual quality of the synthesized view processed by the proposed CNN is much better than those processed by AR-CNN and VR-CNN. For example, there are some boundary artifacts below balloons in Figs. 8(c), (d) and (e), and to the left of the girl's head in Figs. 9(c), (d) and (e). By contrast, these artifacts are not apparent in Figs. 8(f) and 9(f).

Secondly, the CNN model is also evaluated and compared with AR-CNN and VR-CNN as the module of post-processing. In particular, they are re-trained with the same training data as the proposed CNN model. More specifically, the training data are obtained by the distorted texture and depth under (QP_t,QP_d) of (30, 39). The results of *Balloons* and *Poznan_Carpark* sequences are shown in Figs. 10(a) and (b), from which it can be found that all these CNN models can improve the quality of synthesized image. Moreover, the proposed CNN model has better performance gain than AR-CNN and VR-CNN.

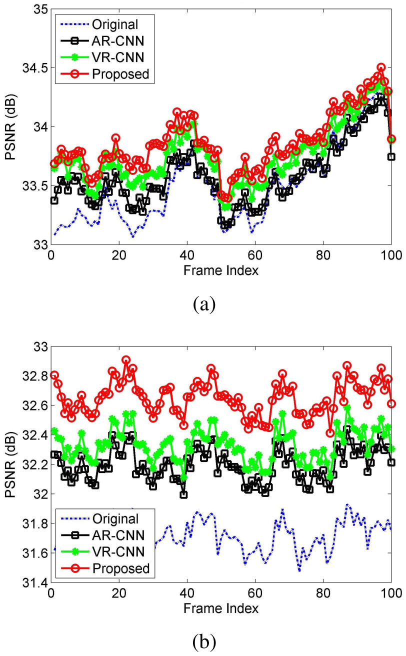


Fig. 10. Performance comparisons with the state-of-the-art CNN models in terms of PSNR, where the image was synthesized by encoded texture and depth under $(QP_t, QP_d) = (30, 39)$. (a) Balloons; (b) Poznan_Carpark.

B. CNN Model Validation for Cross View Case

To evaluate that the proposed CNN model is applicable to cross synthesized view case, more experiments are conducted for reference synthesized view enhancement. The synthesized view positions are illustrated by *BookArrival* sequence with views from 06 to 10 in Fig. 11. Views 06 and 10 are reference views while views 07 to 09 are synthesized views at positions 0.25 to 0.75. As we know, with two reference views, the synthesized views at any positions between them can be generated. However, we do not know where the synthesized view is at the encoder side in practice. Three fixed synthesized views (0.25, 0.50, and 0.75) are adopted for

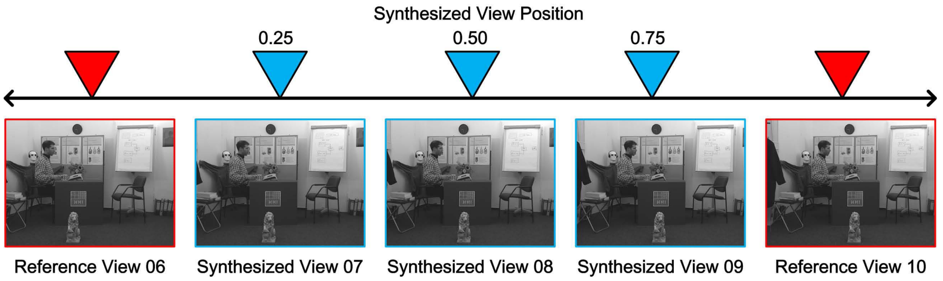


Fig. 11. Synthesized view position illustration.

TABLE IV

PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART CNN MODELS IN TERMS OF PSNR FOR CROSS VIEW CASE [UNIT: dB]

Set	Seq.	Synthesized View Position 0.25				Synthesized View Position 0.75			
		Original	AR-CNN	VR-CNN	Proposed	Original	AR-CNN	VR-CNN	Proposed
Set1	1	34.92	34.99	35.18	**35.23**	36.25	36.33	36.56	**36.66**
	3	34.73	34.81	**34.90**	34.87	34.82	34.93	**34.43**	35.39
	5	29.29	29.37	29.15	**29.66**	29.70	29.76	29.62	**29.92**
Set2	2	35.91	35.85	35.92	**35.97**	36.82	36.79	**36.97**	36.93
	4	30.86	31.03	31.64	**31.84**	28.58	28.61	28.17	**28.84**
	6	37.47	37.24	**37.56**	37.52	34.49	34.64	**35.89**	34.64
AVERAGE		33.86	33.88	34.06	**34.18**	33.44	33.51	33.60	**33.73**

TABLE V

PERFORMANCE EVALUATION OF INDIVIDUAL ALGORITHMS (BDBR IN TERMS OF PSNR) [UNIT: %]

Seq.	SET 1		Seq.	SET 2	
	RSVE [1]	POST [2]		RSVE [1]	POST [2]
1	-10.12	-13.69	2	-9.745	-10.87
3	-11.36	-17.22	4	-24.19	-23.26
5	-14.63	-38.20	6	-6.076	-6.936
7	-43.47	-47.45	8	-7.402	-12.47
9	-1.976	-6.095	10	-7.774	-25.48
AVG	-16.31	-24.53	AVG	-11.04	-15.80

TABLE VI

PERFORMANCE EVALUATION OF INDIVIDUAL ALGORITHMS (BDBR IN TERMS OF SSIM) [UNIT: %]

Seq.	SET 1		Seq.	SET 2	
	RSVE [1]	POST [2]		RSVE [1]	POST [2]
1	-3.229	-8.303	2	-1.916	-15.70
3	-0.561	-7.444	4	-4.150	-7.035
5	-3.055	-15.48	6	-0.328	-3.588
7	-9.398	-19.34	8	-2.434	-2.215
9	-1.248	-4.548	10	-1.710	-8.802
AVG	-3.498	-11.02	AVG	-2.108	-7.468

validation in the 3D HEVC encoder. In this paper, only the intermediate synthesized view is considered for CNN model training. The experimental results of cross synthesized view case are shown in Table IV. It can be found that the proposed CNN model still achieves 0.32dB and 0.29dB gain on average when synthesized view positions are 0.25 and 0.75, and it is also better than that of AR-CNN and VR-CNN on average. It indicates that although the proposed CNN model is trained from intermediate view, it can be applied to cross synthesized view case as well.

C. Coding Performance of Individual Algorithm

Here, the performances of the proposed schemes are evaluated in terms of BD-BR under different metrics (PSNR and SSIM). The results are shown in Tables V and VI. In Table V, the schemes of RSVE and POST can reduce 16.31% and 24.53% bit rate on average under Set 1, and reduce 11.04% and 15.80% bit rate on average under Set 2 in terms of PSNR. In Table VI, the schemes of RSVE and POST can reduce 3.498% and 11.02% bit rate on average under Set 1, and reduce 2.108% and 7.468% bit rate on average under Set 2 in terms of SSIM. From these results, it can be found that the post-processing process has more contributions of the performance improvement than encoder optimization, and there is more bit rate reduction in terms of PSNR than that of SSIM.

Moreover, for the encoder optimization, the sequence *Champagne_Tower* has the most significant bit rate reduction, which reaches -43.47% in terms of PSNR. For the post-processing process, the sequences *Newspaper* and *Champagne_Tower* have the most significant bit rate reduction. For the sequence of *Champagne_Tower*, most of the background is low illumination and relatively smooth. Moreover, the quality of the synthesized view from the pristine texture and depth is relatively lower than the other sequences (27.59dB), as shown in Table II. This provides more room for the quality improvement of the synthesized view, such that better performance improvement can be achieved.

D. Coding Performance Comparison With the State-of-the-Art Algorithm

In this subsection, we compare the proposed scheme with the traditional wiener filter based quality enhancement scheme Yuan *et al*'s method [13]. In Yuan *et al.*'s method, the wiener filtering parameters for post-processing are derived in the

[1] RSVE indicates CNN based Reference Synthesized View Enhancement plus Lagrange multiplier adaption.

[2] POST indicates CNN based POST-processing.

TABLE VII
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART ALGORITHMS

Set	Seq	QP	HTM 16.2			Yuan *et al.*'s method [13]					Proposed method				
			Bit Rate	PSNR	SSIM	Bit Rate	PSNR	SSIM	BDBR		Bit Rate	PSNR	SSIM	BDBR	
									PSNR	SSIM				PSNR	SSIM
Set1	1	30,39	579.4	34.81	0.9227	580.3	34.96	0.9243	-5.53%	-5.20%	571.4	35.14	0.9238	-16.4%	-10.9%
		35,42	312.7	34.06	0.9108	313.3	34.19	0.9129			308.0	34.46	0.9127		
		40,45	172.6	32.65	0.8895	172.8	32.79	0.8915			170.9	33.06	0.8943		
		45,49	90.85	30.76	0.8569	91.10	30.85	0.8590			90.71	31.07	0.8636		
	3	30,39	1105	33.18	0.9549	1111	33.44	0.9573	-14.8%	-12.4%	1084	33.75	0.9572	-22.6%	-11.7%
		35,42	632.0	32.84	0.9480	634.5	33.12	0.9509			622.7	33.36	0.9491		
		40,45	379.2	32.14	0.9339	380.3	32.41	0.9383			374.4	32.59	0.9379		
		45,49	221.8	30.76	0.9098	222.3	31.01	0.9146			219.5	31.05	0.9153		
	5	30,39	1556	27.94	0.8837	1558	28.09	0.8883	-23.5%	-17.7%	1517	28.46	0.8887	-47.0%	-16.5%
		35,42	809.1	27.90	0.8749	810.4	28.04	0.8799			797.6	28.39	0.8775		
		40,45	451.1	27.73	0.8589	451.9	27.89	0.8651			445.6	28.06	0.8646		
		45,49	247.7	27.13	0.8312	248.2	27.31	0.8379			248.8	27.64	0.8392		
	7	30,39	1972	27.68	0.9334	1975	27.92	0.9369	-37.9%	-23.4%	1936	28.46	0.9360	-60.8%	-22.1%
		35,42	1139	27.65	0.9300	1141	27.90	0.9338			1104	28.35	0.9336		
		40,45	653.6	27.53	0.9232	654.9	27.77	0.9275			641.7	28.20	0.9278		
		45,49	363.4	27.11	0.9065	364.0	27.33	0.9119			358.9	27.53	0.9093		
	9	30,39	3030	34.70	0.9104	3033	34.88	0.9123	-5.57%	-4.82%	2977	34.88	0.9111	-8.10%	-5.62%
		35,42	1434	33.82	0.8891	1436	33.95	0.8910			1417	33.95	0.8898		
		40,45	728.8	32.43	0.8615	729.6	32.54	0.8638			725.2	32.60	0.8646		
		45,49	369.9	30.48	0.8212	370.6	30.57	0.8239			369.3	30.66	0.8261		
	AVERAGE					——			**-17.4%**	**-12.7%**	——			**-30.9%**	**-13.4%**
Set2	2	30,39	979.5	35.47	0.9644	983.4	35.61	0.9664	-8.02%	-15.3%	968.0	35.65	0.9656	-13.3%	-16.1%
		35,42	544.5	34.95	0.9593	546.0	35.11	0.9619			543.4	35.03	0.9609		
		40,45	323.3	33.89	0.9502	324.4	34.07	0.9536			321.2	34.29	0.9537		
		45,49	189.1	32.39	0.9355	189.6	32.55	0.9384			188.9	32.63	0.9415		
	4	30,39	1778	29.33	0.9042	1779	29.40	0.9065	-13.7%	-5.67%	1758	29.81	0.9075	-34.6%	-9.88%
		35,42	876.5	29.13	0.8864	877.2	29.20	0.8885			866.2	29.62	0.8902		
		40,45	447.6	28.74	0.8578	448.1	28.85	0.8603			445.8	29.07	0.8619		
		45,49	213.8	27.99	0.8175	214.3	28.08	0.8205			213.4	28.24	0.8223		
	6	30,39	1806	34.88	0.9704	1811	34.98	0.9710	-4.42%	-4.30%	1711	35.09	0.9712	-10.4%	-7.38%
		35,42	984.8	34.22	0.9658	986.1	34.36	0.9665			949.5	34.58	0.9668		
		40,45	561.3	33.14	0.9578	562.1	33.21	0.9584			555.5	33.35	0.9586		
		45,49	308.1	31.31	0.9444	308.8	31.40	0.9451			308.1	30.93	0.9455		
	8	30,39	769.4	36.00	0.9362	801.1	36.06	0.9371	-3.82%	-8.55%	793.3	36.10	0.9368	-16.4%	-7.11%
		35,42	416.6	35.78	0.9330	418.9	35.84	0.9340			412.8	35.92	0.9340		
		40,45	237.1	35.19	0.9269	238.4	35.23	0.9280			235.9	35.40	0.9275		
		45,49	140.2	34.20	0.9181	140.5	34.23	0.9189			138.7	34.44	0.9187		
	10	30,39	3378	31.65	0.8999	3385	31.99	0.9035	-14.9%	-5.94%	3142	32.41	0.9031	-30.1%	-9.91%
		35,42	1650	31.28	0.8785	1653	31.55	0.8820			1583	31.80	0.8811		
		40,45	786.2	30.41	0.8391	787.4	30.61	0.8422			768.6	30.87	0.8436		
		45,49	356.1	29.07	0.7857	356.7	29.18	0.7882			355.2	29.42	0.7927		
	AVERAGE					——			**-8.99%**	**-7.96%**	——			**-20.9%**	**-10.1%**
AVERAGE						——			**-13.2%**	**-10.3%**	——			**-25.9%**	**-11.7%**

encoder and signalled in the bitstream to improve the quality of synthesized image. However, the position of virtual viewpoint is required to be defined before filtering parameters calculation, which is impractical in real-world system. The proposed method can address these problems because the trained CNN models can be employed to any sequences, and the position of virtual viewpoint is not limited, which means that it is applicable to cross synthesized view case. In the experiment, three default interpolated viewpoints (0.25, 0.5, and 0.75) are adopted to evaluate the performance at different positions illustrated in Fig. 11, and the results are shown in Table VII.

For Yuan *et al.*'s method, it can achieve 17.4% and 8.99% bit rate reduction under Sets 1 and 2 in terms of PSNR, and achieve 12.7% and 7.96% bit rate reduction under Sets 1 and 2 in terms of SSIM. Moreover, it is observed that the bit rate of Yuan *et al.*'s method increases when compared with HTM 16.2 due to the wiener filtering parameters signalled in the bitstream, which results in significant bit rate overhead. The proposed method can reduce 30.9% and 20.9% bit rate under Sets 1 and 2 in terms of PSNR, and reduce 13.4% and 10.1% bit rate under Sets 1 and 2 in terms of SSIM. The sequence of *Champagne_Tower* has the most significant bit rate reduction, which reaches 60.8% and 22.1% in terms of PSNR and SSIM.

E. Complexity of Proposed CNN Model in 3D Video System

In this subsection, the computational complexity of the 3D HEVC encoder and the 1D-FAST view synthesis software equipped with the proposed CNN models are recorded and compared with the original HTM 16.2 and 1D-FAST view synthesis software. The computational complexity increment is measured by,

$$\Delta T = \frac{T_{Pro} - T_{Org}}{T_{Org}} \times 100\%, \qquad (28)$$

where T_{Pro} and T_{Org} indicate the running time of proposed method and the original codec, respectively. The results are shown in Table VIII. According to the results, it can be observed that the computational complexity increases 294% and 369% on average for the module of encoder optimization under Sets 1 and 2, respectively. For the post-processing with the proposed CNN model, the computational complexity increases significantly, *i.e.*, 1711% and 2098% under

TABLE VIII

COMPLEXITY COMPARISON WITH HTM 16.2

Set	Seq.	RSVE ^{1}	POST ^{2}
Set1	1	233%	1292%
	3	176%	1577%
	5	216%	1456%
	7	314%	2033%
	9	529%	2198%
	AVG	**294%**	**1711%**
Set2	2	152%	1119%
	4	216%	1323%
	6	272%	1988%
	8	662%	3135%
	10	545%	2923%
	AVG	**369%**	**2098%**
AVERAGE		**331%**	**1904%**

TABLE IX

CROSS-VALIDATION OF CNN MODELS IN TERMS OF PSNR [UNIT: dB]

Test Set	Anchor	CNN1	CNN2	CNN3	CNN4
(QP_t,QP_d)=(30,39)	32.80	**33.22**	33.11	33.13	32.92
(QP_t,QP_d)=(35,42)	32.34	32.70	**32.73**	32.67	32.51
(QP_t,QP_d)=(40,45)	31.46	31.73	31.69	**31.78**	31.70
(QP_t,QP_d)=(45,49)	30.09	30.26	30.25	30.34	**30.37**

Sets 1 and 2, respectively. Most of computation time is spend on the convolution operation in CNN model. It should be noted that these encoding and view synthesis experiments are carried out on the CPU platform instead of GPU. Although the proposed method equipped with CNN models is time consuming, this is the first attempt of applying CNN to optimize the 3D HEVC and it achieves better RD performance than the traditional method.

F. Cross-Validation of CNNs Under Different QP Settings

In this subsection, we conduct an experiment to further investigate whether the CNN models trained under different QP settings can be shared in the post-processing module. In this paper, there are four CNN models for post-processing based on four QP pairs of (QP_t,QP_d) from (30, 39) to (45, 49). For simplicity, these CNN models are denoted as CNN1 to CNN4, and the training of them has been mentioned in Section IV. The average results of Sets 1 and 2 are shown in Table IX. From these results, we can observe that these CNN models are able to be shared to some extent because all of them can bring performance gains when compared with the anchor. Moreover, it is not surprising to see that these synthesized images which have been processed by the specific CNN model trained with the corresponding QP value can obtain the best performance, and the quality decreases with the QP distance between training and testing.

VI. CONCLUSION

In this paper, we present a CNN based synthesized view quality enhancement method for 3D HEVC to further improve the coding efficiency, where the proposed CNN models are incorporated into the 3D HEVC encoding and post-processing process. The novelty of this paper lies in that the learning based restoration model is applied to push the synthesized view towards the physically captured one, which benefits both the encoder and post-processing process in 3D video coding. Accordingly, the new Lagrange multiplier is further adjusted to adapt to such synthesized view quality improvement. Compared with the state-of-the-art CNN models, the proposed scheme achieves better performance improvement, and significantly improves the 3D video coding performance. In the future work, the computational complexity issue will be further investigated.

REFERENCES

[1] K. Müller *et al.*, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3366–3378, Sep. 2013.

[2] C. Fehn, "Depth image based rendering (DIBR), compression, and transmission for a new approach on 3DTV," in *Proc. SPIE Conf. Stereoscopic Displays Virtual Reality Syst. XI*, San Jose, CA, USA, vol. 5291, Jan. 2004, pp. 93–104.

[3] C. Lipski, F. Klose, and M. Magnor, "Correspondence and depth-image based rendering a hybrid approach for free-viewpoint video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 942–951, Jun. 2014.

[4] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, Jan. 2016.

[5] G. J. Sullivan, J. M. Boyce, Y. Chen, J.-R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (HEVC)," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

[6] G. Tech, H. Schwarz, K. Müller, and T. Wiegand, "3D video coding using the synthesized view distortion change," in *Proc. Picture Coding Symp.*, Krakow, Poland, May 2012, pp. 25–28.

[7] B. T. Oh and K.-J. Oh, "View synthesis distortion estimation for AVC- and HEVC-compatible 3-D video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 1006–1015, Jun. 2014.

[8] Z. Peng, G. Jiang, M. Yu, S. Pi, and F. Chen, "Temporal pixel classification and smoothing for higher depth video compression performance," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1815–1822, Nov. 2011.

[9] P.-J. Lee and Effendi, "Nongeometric distortion smoothing approach for depth map preprocessing," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 246–254, Apr. 2011.

[10] L. Zhu, Y. Zhang, M. Yu, G. Jiang, and S. Kwong, "View-spatial–temporal post-refinement for view synthesis in 3D video systems," *Signal Process., Image Commun.*, vol. 28, no. 10, pp. 1342–1357, Nov. 2013.

[11] C. Zhu and S. Li, "Depth image based view synthesis: New insights and perspectives on hole generation and filling," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 82–93, Mar. 2016.

[12] D. M. M. Rahaman and M. Paul, "Virtual view synthesis for free viewpoint video and multiview video compression using Gaussian mixture modelling," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1190–1201, Mar. 2018.

[13] H. Yuan, J. Liu, Z. Li, and W. Liu, "Coding distortion elimination of virtual view synthesis for 3D video system: Theoretical analyses and implementation," *IEEE Trans. Broadcast.*, vol. 58, no. 4, pp. 558–568, Dec. 2012.

[14] L. Zhu, Y. Zhang, X. Wang, and S. Kwong, "View synthesis distortion elimination filter for depth video coding in 3D video broadcasting," *Multimedia Tools Appl.*, vol. 74, no. 15, pp. 5935–5954, Jul. 2015.

[15] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 576–584.

[16] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[17] W.-S. Park and M. Kim, "CNN-based in-loop filtering for coding efficiency improvement,"in *Proc. IEEE 12th Image, Video, Multidimensional Signal Process. Workshop (IVMSP)*, Bordeaux, France, Jul. 2016, pp. 1–5.

[18] Y. Dai, D. Liu, and F. Wu, "A convolutional neural network approach for post-processing in HEVC intra coding," in *Proc. Int. Conf. Multimedia Modeling (MMM)*, Reykjavik, Iceland, Jan. 2017, pp. 28–39.

[19] C.-M. Fu *et al.*, "Sample adaptive offset in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1755–1764, Dec. 2012.

[20] T. Wang, M. Chen, and H. Chao, "A novel deep learning-based method of improving coding efficiency from the decoder-end for HEVC," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Apr. 2017, pp. 410–419.

[21] W.-S. Kim, A. Ortega, P. Lai, and D. Tian, "Depth map coding optimization using rendered view distortion for 3D video coding," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3534–3545, Nov. 2015.

[22] L. Fang, Y. Xiang, N.-M. Cheung, and F. Wu, "Estimation of virtual view synthesis distortion toward virtual view position," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 1961–1976, May 2016.

[23] Z. Zheng, J. Huo, B. Li, and H. Yuan, "Fine virtual view distortion estimation method for depth map coding," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 417–421, Mar. 2018.

[24] H. Yuan, S. Kwong, X. Wang, Y. Zhang, and F. Li, "A virtual view PSNR estimation method for 3-D videos," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 134–140, Mar. 2016.

[25] M. Yang, N. Zheng, C. Zhu, and F. Wang, "A novel method of minimizing view synthesis distortion based on its non-monotonicity in 3D video," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5122–5137, Nov. 2017.

[26] G. Tech, K. Müller, H. Schwarz, and T. Wiegand, "Partial depth image based re-rendering for synthesized view distortion computation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1273–1287, Jun. 2018.

[27] B. T. Oh, J. Lee, and D.-S. Park, "Depth map coding based on synthesized view distortion function," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 7, pp. 1344–1352, Nov. 2011.

[28] C. Yao, J. Xiao, T. Tillo, Y. Zhao, C. Lin, and H. Bai, "Depth map down-sampling and coding based on synthesized view distortion," *IEEE Trans. Multimedia*, vol. 18, no. 10, pp. 2015–2022, Oct. 2016.

[29] X. Liu, Y. Zhang, S. Hu, S. Kwong, C.-C. J. Kuo, and Q. Peng, "Subjective and objective video quality assessment of 3D synthesized views with texture/depth compression distortion," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4847–4861, Dec. 2015.

[30] Y. Zhang, X. Yang, X. Liu, G. Jiang, and S. Kwong, "High-efficiency 3D depth coding based on perceptual quality of synthesized video," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5877–5891, Dec. 2016.

[31] M. Paul and M. Murshed, "Video coding focusing on block partitioning and occlusion," *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 691–701, Mar. 2010.

[32] K. Müller and A. Vetro, *Common Test Conditions of 3DV Core Experiments*, document JCT3V-G1100, JCTVC ITU-T SG16 WP3 ISO/IEC JTC1/SC29/WG11, San Jose, CA, USA, Jan. 2014.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, vol. 37, Jul. 2015, pp. 448–456.

[34] G. Tech, *JCT-3V AHG Report: MV-HEVC and 3D-HEVC Software Integration (AHG2)*, document JCT3V-O0002, JCT-VC ITU-T SG16 WP3 ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, May 2016. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-16.2

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[36] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document M33, ITU-T Video Coding Experts Group, Austin, TX, USA, 2001.

Linwei Zhu (S'16) received the B.S. degree in applied physics from the Tianjin University of Technology, China, in 2010, and the M.S. degree in signal and information processing from Ningbo University, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong. After graduation from Ningbo University, he joined the Shenzhen Institutes of Advanced Technology (SIAT) as a Research Assistant. From 2011 to 2013, he was a Visiting Student with SIAT, Chinese Academy of Science. His research interests mainly include depth image-based rendering, depth estimation, and video coding/transcoding.

Yun Zhang (M'12–SM'16) received the B.S. and M.S. degrees in electrical engineering from Ningbo University, Ningbo, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, China, in 2010. From 2009 to 2014, he was a Post-Doctoral Researcher with the Department of Computer Science, City University of Hong Kong, Hong Kong. He is currently a Full Professor with the Shenzhen Institutes of Advanced Technology, CAS. His research interests are video compression, 3D video processing, and visual perception.

Shiqi Wang (M'15) received the B.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2008, and the Ph.D. degree in computer application technology from Peking University, Beijing, China, in 2014. From 2014 to 2016, he was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. From 2016 to 2017, he was with the Rapid-Rich Object Search Laboratory, Nanyang Technological University, Singapore, as a Research Fellow. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has proposed over 30 technical proposals to ISO/MPEG, ITU-T, and AVS standards. His research interests include image/video compression, and analysis and quality assessment.

Hui Yuan (S'08–M'12–SM'17) received the B.E. and Ph.D. degrees in telecommunication engineering from Xidian University, Xi'an, China, in 2006 and 2011, respectively. From 2013 to2014, he was a Post-Doctoral Fellow with the Department of Computer Science, City University of Hong Kong, Hong Kong. Since 2011, he has been a Lecturer, an Associate Professor, and a Full Professor with the School of Information Science and Engineering, Shandong University, Jinan, China. His current research interests include video/image compression, adaptive video streaming, and computer vision.

Sam Kwong (F'13) received the B.S. in electrical engineering from The State University of New York at Buffalo in 1983, the M.S. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1985, and the Ph.D. degree from the University of Hagen, Germany, in 1996. From 1985 to 1987, he was a Diagnostic Engineer with Control Data Canada. He joined Bell Northern Research, Canada, as a member of Scientific Staff. In 1990, he became a Lecturer with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, where he is currently a Professor with the Department of Computer Science. His research interests are video and image coding, and evolutionary algorithms.

Horace H.-S. Ip received the B.Sc. degree (Hons.) in applied physics and the Ph.D. degree in image processing from University College London, London, U.K., in 1980 and 1983, respectively. He is currently the Chair Professor of computer science, the Founding Director of the Centre for Innovative Applications of Internet and Multimedia Technologies, and the Vice-President of the City University of Hong Kong, Hong Kong. He has published over 200 papers in international journals and conference proceedings. His research interests include pattern recognition, multimedia content analysis and retrieval, virtual reality, and technologies for education. He is a fellow of the Hong Kong Institution of Engineers, the U.K. Institution of Electrical Engineers, and the International Association for Pattern Recognition.