Scalable Coding of Depth Images with Synthesis-Guided Edge Detection

Lijun Zhao[1], Anhong Wang[1], Bing Zeng[2, 3], and Jian Jin[1]
[1] Institute of Digital Media & Communication, Taiyuan University of Science and Technology
Taiyuan, China, 030024
[e-mail: wah_ty@163.com]
[2] Institute of Image Processing, University of Electronic Science and Technology of China
Chengdu, China, 611731
[e-mail: eezeng@uestc.edu.cn]
[3] Dept. of Electronic and Computer Eng., The Hong Kong University of Electronic Science and Technology
Kowloon, Hong Kong
[e-mail: eezeng@uestc.edu.cn]
*Corresponding author: Anhong Wang

Received February 28, 2015; revised May 13, 2015; revised June 27, 2015; accepted August 17, 2015; published October 31, 2015

Abstract

This paper presents a scalable coding method for depth images by considering the quality of synthesized images in virtual views. First, we design a new edge detection algorithm that is based on calculating the depth difference between two neighboring pixels within the depth map. By choosing different thresholds, this algorithm generates a scalable bit stream that puts larger depth differences in front, followed by smaller depth differences. A scalable scheme is also designed for coding depth pixels through a layered sampling structure. At the receiver side, the full-resolution depth image is reconstructed from the received bits by solving a partial-differential-equation (PDE). Experimental results show that the proposed method improves the rate-distortion performance of synthesized images at virtual views and achieves better visual quality.

Keywords: depth image, scalable coding, inter-layer prediction, depth difference synthesis-guided edge detection, partial differential equation

1. Introduction

Multi-view video plus depth (MVD) has recently emerged as an efficient format of 3D video [1] in which 3D scenes can be better described with the help of depth information. With a few pairs of depth and color images from neighboring views in MVD, arbitrary intermediate views can be synthesized via the depth-image-based rendering (DIBR) technique [2]. As a result, MVD greatly reduces the data volume required by the auto-stereoscopic 3D display in which many views are commonly needed for immersive experiences. Depth image is usually an 8-bit gray image whose pixels describe distances between the camera and spatial points in the real scene. This differs from a 24-bit color image whose pixels represent the color/texture information of a scene. The significant characteristics of depth images are the homogeneous regions divided by sharp edges. Additionally, depth images are only used to render virtual views rather than being displayed directly. Therefore, the coding of depth images should not use traditional methods, such as the quantization-based coding methods for texture images.

Many approaches have been proposed to compress depth images. For instance, Krishnamurthy et al. proposed a depth coding method based on an improved standard JPEG2000 encoder [3]. Nevertheless, the quantization-based methods usually cause large distortions on edge pixels, which may lead to a wrong 3D-warping and deformation of the virtual view during the 3D-synthesis process. Morvan et al. employed a quad-tree decomposition to divide the depth image into blocks of variable sizes, with each block approximated by one platelet [4]. This method needs to decompose the image into a full quad-tree up to the pixel level to achieve a more accurate description of the discontinuous regions of the depth image, which consequently results in increasing bit-rate representing the tree structure. Since the edges in a depth image are especially important for the synthesis quality of 3D video, some researches proposed to exploit near-lossless or even lossless coding methods for the edge pixels, e.g., Gautier et al. proposed a new method for lossless edge depth coding based on an optimized path and fast homogeneous diffusion [5]. To best of our knowledge, however, most of the existing approaches pay little attention to the edge detection of depth images, or they just use some traditional edge detection methods that are tailored for texture images. Additionally, those approaches do not consider the bandwidth's variability when a depth image is transmitted over heterogeneous networks.

Based on the above analysis, we propose a scalable depth coding with synthesis-guided edge detection. Firstly, a novel edge-detection method is proposed to obtain the significant edges for a depth image from the point of synthesized quality, which seriously affect the synthesized image of the virtual view. In our proposed scheme, we first generate a base layer that combines fewer edges with fewer sampled-pixels to reconstruct a low-quality image for preserving important edges that have greater depth difference along the discontinuous regions. Then, more edges and sampled pixels would be sent stage by stage to provide a gradually improved quality. This allows for scalable coding for depth images and progressive quality improvement of virtual images in the synthesized views rendered by the DIBR technique.

The rest of this paper is organized as follows. In Section 2, we review some related work, including the scalable coding, edge detection methods and the bit-rate allocation for edges and sampled images. We then describe the proposed scheme in Section 3. After experimental results are presented in Section 4, we draw a conclusion in Section 5.

2. Related Work

2.1 Scalable Coding for Multi-view Videos

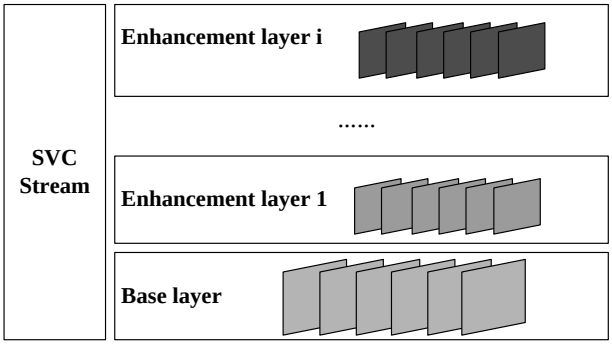


Fig. 1. The illustration of stream for SVC

Scalable video coding (SVC) encodes video with multiple layers (a base layer and several enhancement layers), as shown in **Fig. 1**, so that only the base layer is decoded when the bandwidth is not enough and the decoded enhancement layers are used to improve video quality when bandwidth gradually becomes wider [6]. Compared with traditional one-time coding with fixed video stream, SVC provides a possibility of a single multi-layer embedded bit-stream. In general, sub-bit-stream can satisfy the transmission rate and the requirement for different video quality. Although SVC has many advantages, it dose not suit to depth image and the coding of depth image with scalability is still a hot issues for 3D video coding.

For the scalable coding of depth images, some papers [7-8] have developed very useful research, though the performances are required to be further improved. Therefore, in this paper, we try to explore a scalable coding method to efficiently compress depth images. Our works employ lose-less coding to down-sampled depth image and edge information considering the features of smooth areas with sharp edges. The down-sampling can greatly reduce the amount of data need to be transmitted, while lossless coding for significant edges and pixels is required by high quality reconstruction of depth image, where the missing pixels can be interpolated by these lossless pixels. Besides, we explore inter-layered prediction to further remove the redundancy between down-sampled images of different enhancement layers.

2.2 Edge Detection for Depth Image

Because the edge information is important for the synthesis of 3D video, the detection of edges in depth image is a key problem for depth coding. However, in most studies, the edge detection of depth images is based on the methods used for texture images, such as the Canny and Sobel edge detection method, which inevitably causes some problems. Hence, in this paper, we propose an edge detection method based on the synthesis view and the depth difference, which is denoted as “DDED” and introduced in the following Section 3. As a motivation of our method, in this section, we just compare our “DDED” with the Canny and Sobel edge detection method.

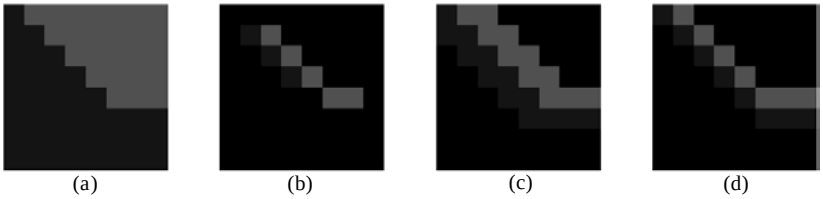


Fig. 2. Comparison of different edge detection methods: (a) depth image, (b) detected by Canny, (c) detected by Sobel, (d) detected by our method.

A snapshot is presented in **Fig. 2** to highlight the difference between our edge detection and other edge detection methods. It can be seen that the edges detected by the traditional Canny method are not appropriate for the reconstruction of depth image because it uses a Gause filter prior to its edge detection. This Gause filter would smooth the sharp edges and consequently, the depth image will be blurred to some degree, which will affect the final quality of the synthesized view. Additionally, the correlations of edge-pixels after Canny detection are weak, which affects the efficiency of depth coding. For Sobel edge detection, it is difficult to decide the threshold, which usually leads to unreasonably wide edges, as shown in **Fig. 2 (c)**, while such wide edges will require a higher bit-rate upon encoding. Our DDED scheme pays more attention to the depth/disparity difference and bigger pixel position shift (disparity) for foreground objects relative to background objects during the synthesis of the virtual view, while other methods only detect obvious illumination changes as edges.

2.3 The Bit-rate Allocation for Edges and Sampled Images

High quality reconstruction of depth image strongly depends on the bit-rate allocation for edges and sampled image. Ref [5] uses a fixed sampling-rate and adjusts the bit-rate allocation by changing the thresholds of Sobel detection. It did not distinguish the importance of edges to the synthesis quality. In order to improve the coding efficiency, in this paper, a efficient stream structure is proposed to provide scalability by sampling, inter-layered prediction, and changing thresholds for edge detection, which distinguishes edges' importance to the synthesis quality.

3. Proposed Scheme

4. In this section, we describe the proposed scalable coding scheme of depth images. Our scheme includes four parts: 1) scalable coding of edges, 2) scalable coding of sampled pixels, 3) inter-layered prediction, and 4) PDE-based reconstruction, as shown in **Fig. 3**.

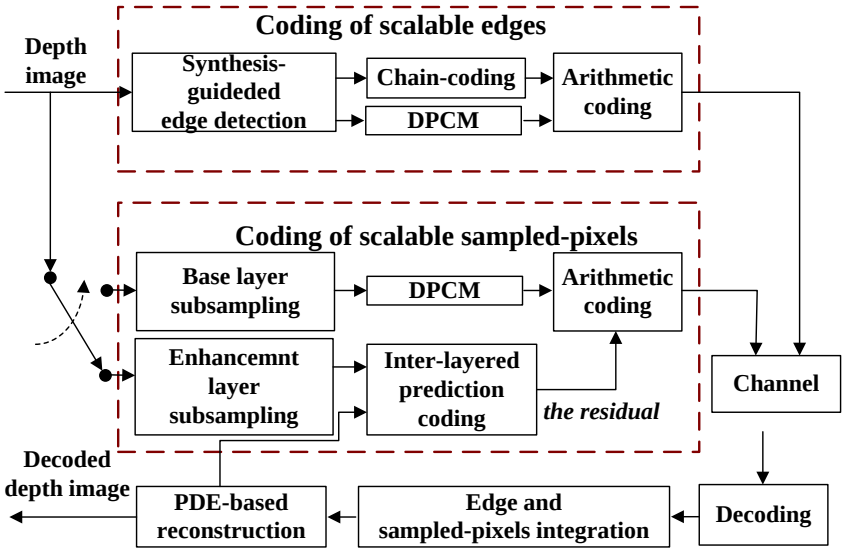


Fig. 3. The diagram of our proposed scheme

3.1 Scalable Coding of Edges

After depth images are captured or estimated by the stereo matching algorithm, they are always recorded as an 8-bit value for each pixel according to the nonlinear equation applied in paper [8]. Therefore, a nearby object has more precise depth value than an object that is farther from the camera. With the fact that, relative to the camera in the real scene, the nearer points (forming the foreground) have larger disparity than farther points (forming the background), even a small depth error for these foreground objects will lead to an apparent position shift and the deformation of synthesized images in the process of rendering. In fact, the occlusion and dis-occlusion holes usually occur around the depth boundary regions. Also, because the discontinuous regions of the depth image are more susceptible to coding, the distortion of boundary regions will badly affect the orders of occlusion in the rendering. As a result, during the process of depth coding, not only the discontinuous regions of the depth image should be well protected, but the greater depth differences around boundary regions should also be preserved initially. For the purpose of detecting edges, in our DDED, we define the significant edges as what will cause occlusion or dis-occlusion holes in the synthesized virtual view. Considering that both occlusion and dis-occlusion holes originate from depth difference in the depth image, for simplicity, we just deduce a threshold at the case of dis-occlusion holes in this paper. In our work, we assume a 1D parallel configuration for camera, which has also been adopted in 3D-HEVC [11]. Our DDED will be described in detail below and the efficiency of DDED will be demonstrated in the experimental results.

3.1.1 Threshold Calculation for Edge Detection

In our method, the significant edges of a depth image are detected with a threshold deduced from the point of synthesized quality. Specifically, during the 3D-synthesis process, when the depth difference between two neighbor pixels is greater than a threshold, there will be a dis-occlusion hole between their warped-points in the virtual view, otherwise it is generally reckoned no holes are generated. With the increase of the depth difference, hole's size increases. Based on this observation, we deduce a threshold for edge-detection as follows.

As shown in **Fig. 4**, assuming a point x in the real world, two points x_L and x_R are respectively the coordinates projected onto left and right image planes. Here, we consider a parallel arrangement of multi-view cameras, meaning the vertical disparity is zero. Notice that x_l and d_2 are respectively the horizontal coordinates for the point x_L relative to the left first column and center column at the same row in the depth image of the left view. Similarly, x_r and d_1 are respectively the horizontal coordinates for the point x_R in the depth image of the right view.

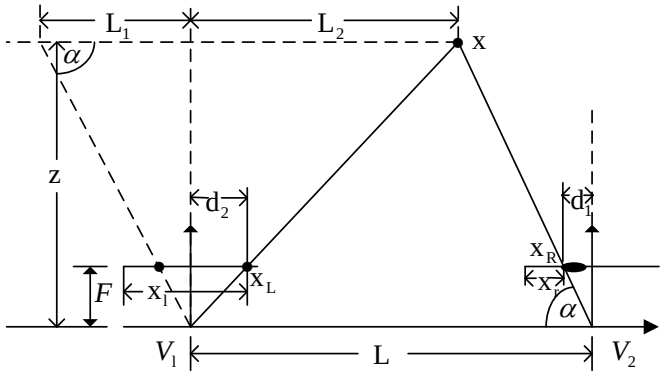


Fig. 4. The model of 1D parallel arrangement for multi-view cameras.

In this case, the horizontal disparity d can be calculated by Eq.(1), where F, L and z respectively represent the focal length of the camera, the baseline length between two nearest neighbor views, and the real depth value of point x [9].

$$d = x_l - x_r = d_1 + d_2 = \frac{F \cdot L_1}{z} + \frac{F \cdot L_2}{z} = \frac{F \cdot (L_1 + L_2)}{z} = \frac{F \cdot L}{z}, \tag{1}$$

Meanwhile, we consider a commonly used non-linear 8-bit quantization function for depth value [10] in Eq.(2), where D, z, Z_{near} and Z_{far} represent the depth value in the depth image, the real depth value and the nearest and farthest depth values of a scene. Considering that the near object has a greater disparity than the far object, it can be seen that using the nonlinear quantization of Eq.(2) will produce a more accurate depth for near objects than for those far away. Combining Eq.(1) and Eq.(2), we can get the relation between depth value D and horizontal disparity d to Eq.(3).

$$D = Q(z) = \left\lfloor 255 \cdot \frac{Z_{near}}{z} \cdot \frac{Z_{far} - z}{Z_{far} - Z_{near}} + 0.5 \right\rfloor, \tag{2}$$

$$d = \text{Disparity}(D) \approx F \cdot L \cdot ((D/255) \cdot (1/Z_{near} - 1/Z_{far}) + 1/Z_{far}). \tag{3}$$

Here, for the given two neighbor pixels (x_1^l, y^l) and (x_2^l, y^l) (assume that $x_2^l - x_1^l = 1$) in the left view with respective depth values $D(x_1^l, y^l)$ and $D(x_2^l, y^l)$, assuming they are

warped respectively to (x_1^r, y^r) and (x_2^r, y^r) after left-to-right 3D-warping, the disparity d_1 of point (x_1^l, y^l) between left and right views and the disparity d_2 of point (x_2^l, y^l) can be described as: $d_1 = x_1^l - x_1^r, d_2 = x_2^l - x_2^r$, and the disparity difference Δd between these two points can be obtained:

$$\Delta d = d_1 - d_2 = F \cdot L \cdot \frac{D(x_1^l, y^l) - D(x_2^l, y^l)}{255} \cdot \left(\frac{1}{Z_{near}} - \frac{1}{Z_{far}}\right) \tag{4}$$

Considering $x_2^l - x_1^l = 1$ and $\Delta d = (x_1^l - x_2^l) - (x_1^r - x_2^r)$, we get Eq.(5):

$$x_2^r - x_1^r = F \cdot L \cdot \frac{D(x_1^l, y^l) - D(x_2^l, y^l)}{255} \cdot \left(\frac{1}{Z_{near}} - \frac{1}{Z_{far}}\right) + 1. \tag{5}$$

It is apparent that because $x_2^l - x_1^l = 1$, we can see that when $x_2^r - x_1^r = 2$, a dis-occlusion hole with the size of one-pixel will appear between the virtual (x_1^r, y^r) and (x_2^r, y^r), while if $x_2^r - x_1^r > 2$, then a dis-occlusion hole with a size more than one pixel will appear. Additionally, the holes' size increases with the increase of $x_2^r - x_1^r$. Therefore, based on this observation, in order to detect dis-occlusion holes more than one pixel in width, we obtain a threshold K for depth difference by setting $x_2^r - x_1^r = 2$ in Eq.(5), so we obtain

$$K = \Delta D = D(x_1^l, y^l) - D(x_2^l, y^l) = \frac{255}{L \cdot F \cdot (1/Z_{near} - 1/Z_{far})}. \tag{6}$$

3.1.2 Edge Detection

After obtaining the threshold K, edge detection is performed on depth images by judging if the absolute depth difference ΔD is greater than or equal to the threshold K. Although the threshold is deduced for horizontal disparity, it is used in both horizontal and vertical depth difference in order to keep the continuity of edges. Typically, if the absolute depth difference between the current pixel and anyone of its neighbor (including its up, down, left, and right adjacent pixels) is greater than or equal to this threshold, the bigger pixel is detected as a foreground edge pixel, and meanwhile, the smaller value is selected to be a background edge pixels. A special case is that a pixel may be detected as both foreground edge pixels and background edge pixels. In this case, we define it as a foreground edge pixel in order to give it more priority. In this way, all significant edges are detected and the corresponding foreground edge pixels and background edge pixels are distinguished.

3.1.3 Scalable Coding of Edges

First, it is easy to understand that the foreground edge-pixels are more important than the background edge pixels. Thus, we will only encode edge information and corresponding foreground edge-pixels. We also arrange a scalable edge stream by changing the threshold for the edge detection (e.g., K, 2K can also be set as thresholds). When the small threshold is used, more edges will be detected and transmitted so that a better reconstruction quality can be obtained; with a larger threshold, fewer edges are detected, resulting in a lower bit-rate but also a lower reconstruction quality at the receiver end.

The edge information of depth images consists of edge positions and its corresponding foreground edge pixels. We firstly employ chain-code to describe edge positions, and then compress the edge pixels by a forward difference predictive coding without quantization (a simplest Differential Pulse Code Modulation (DPCM)). Finally, the output of both chain code and DPCM are encoded by arithmetic coding [7].

3.2 Scalable Coding of Sampled Pixels

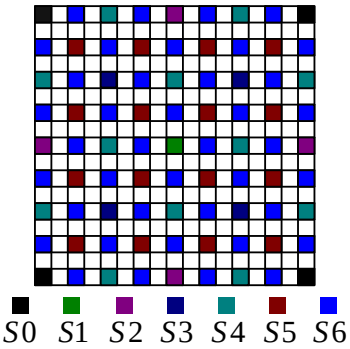


Fig. 5. To illustrate the scalable down-sampling mode

Besides the scalable coding of edges, we also exploit a scalable down sampling to obtain the sampled pixels. The depth image is first divided into some non-overlapped blocks, and then for each block, a progressive down sampling as shown in **Fig. 5** is implemented to generate the scalable stream: one pixel is first sampled per 16×16 block to form the base layer (BL), denoted as S0 model; then an additional pixel is sampled according to S1 mode to generate the pixels of enhancement layer, denoted as 'Layer1', etc. The advantage of this sampling pattern is that the reconstruction quality would increase with the received layers increase. The reconstruction of lower layer is used as the prediction of a higher layer and then the residuals are encoded by arithmetic coding, which will be introduced in the following Subsection 3.4. The reconstructed quality of smooth regions depends on the quantity of sampled-pixels, so the depth image's quality can be better reconstructed, when the more sampled pixels are received.

4.3 The Scalable Stream Structure of Depth Image

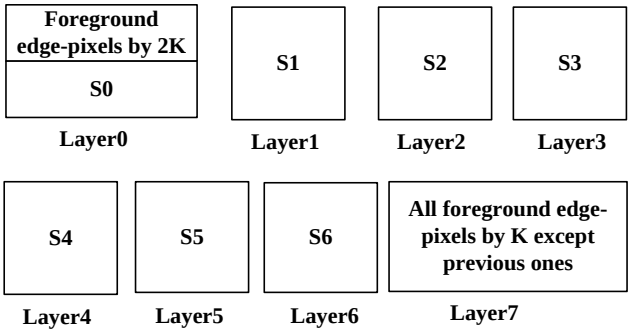


Fig. 6. To illustrate the scalable stream structure

Combining the edge information and the sampled pixels of each layer, we can achieve a scalable stream, with which a low-capability user can decode a low-quality depth image while the users with wide bandwidth can get more bits so as to restore high quality depth images. The structure of the scalable stream of a depth image is illustrated in **Fig. 6**. Note that in our scheme, only the edge position and its foreground edge pixels are coded and transmitted in order to save bit rate. This design is based on the fact that human eyes are more sensitive to foreground objects than to the background.

3.4 Inter-layered Prediction Coding

In order to remove the redundancy between inter-layers, we carry out an inter-layered prediction for the coding of the enhancement layers. Specifically, the decoded lower layers are used as the predictions of the original depth image. In the coding of high layers, the residuals between the original depth image and its prediction are encoded by arithmetic coding. We conduct this prediction in all adjacent layers for the sampled images, which can efficiently improve the performances of coding.

3.5 PDE-based Reconstruction

A PDE-based approach has been applied in [12] to reconstruct cartoon-like images. We chose this PDE approach in our work due to similar characteristics between cartoon-like images and depth images. PDE comes from the heat equation where both Dirichlet boundary condition and Neumann boundary condition should be satisfied.

In practice, such a PDE approach can be discretized in a straightforward way by finite difference [13]. Specially, in our work, we first integrate the decoded edge information and the sampled pixels to obtain an initial sparse depth image, which constitutes Neumann and Dirichlet's boundaries conditions. Then, the missed pixels in this sparse depth image are reconstructed by the smooth diffusion from the decoded edge pixels and the sampled pixels. To guarantee the smoothness of depth's surface, each pixel's neighbor up, down, left and right pixels can be iteratively used to reconstruct the missed value. The pseudo-code for the full-resolution depth image's reconstruction process with edges, corresponding foreground edge-pixels and sampled pixels is presented below.

Initialization

Merge the foreground edge-pixels and sampled-pixels to get an initial depth image D_0^f; $\hat{D}^f$ refers to pixels that are missed in D_0^f; Ω^f refers to the foreground edge-pixels; (u, v) is the coordinate; m and n are respectively the width and height of depth image; i is the iteration number.

Iteration: start from i=0 with $\hat{D}_0^f = D_0^f$.

While 1
  for u=1:m
    for v=1:n
      if ($D_0^f(u,v) \in \hat{D}^f$)
        if ($\hat{D}_i^f(u-1,v) \in \Omega^f$ or $\in \hat{D}^f$) **then** N_1=0 **else** N_1 =1 **end**
        if ($\hat{D}_i^f(u+1,v) \in \Omega^f$ or $\in \hat{D}^f$) **then** N_2=0 **else** N_2 =1 **end**
        if ($\hat{D}_i^f(u,v-1) \in \Omega^f$ or $\in \hat{D}^f$) **then** N_3=0 **else** N_3 =1 **end**

if ($\hat{D}_i^f(u,v+1) \in \Omega^f$ or $\in \hat{D}^f$) **then** N_4=0 **else** N_4 =1 **end**

if ($N_1 = N_2 = N_3 = N_4$=0) **then** $\hat{D}_{i+1}^f(u,v) = \hat{D}_i^f(u,v)$ **else**

$$\hat{D}_{i+1}^f(u,v) = 1/(N_1+N_2+N_3+N_4)\times[\hat{D}_i^f(u\text{-}1,v)\cdot N_1 + \hat{D}_i^f(u+1,v)\cdot N_2 + \hat{D}_i^f(u,v\text{-}1)\cdot N_3 + \hat{D}_i^f(u,v+1)\cdot N_4]$$

end

end

end

end

if(max[max(| $\hat{D}_i^f - \hat{D}_{i+1}^f$ |)]< ε)

break;

end

i=i+1;

end

4. Experimental Results and Analysis

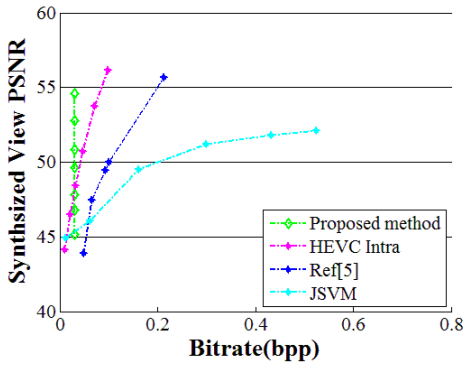
To demonstrate the efficiency of our method, we conducted experiments to compare our scheme with depth images compression based on lossless edge coding [5], HEVC 16.5 intra-coding (with QP set to be 16, 20, 26, 31, 36 and 41) [14-16], SVC with medium grain scalable of JSVM_19_5 software and group of picture (GOP) setting to 1. Five sequences with MVD format ("Undo_Dancer" from Nokia, "Book_Arrival" sequence from HHI, "Kendo" and "Champagne_Tower" provided by Nagoya University, "Love_Bird" from ETRI) are tested over the first 30 frames [17]. In the simulation, 1D-fast mode of 3D-HEVC (HTM-DEV-2.0-dev3 version) [18] is used to synthesize the virtual view image with uncompressed or compressed depth images and uncompressed texture images. We set $\varepsilon = 0.0001$ in the process of depth reconstruction of PDE.

4.1 Rate-distortion Performance Comparison of Depth and Synthesized Image

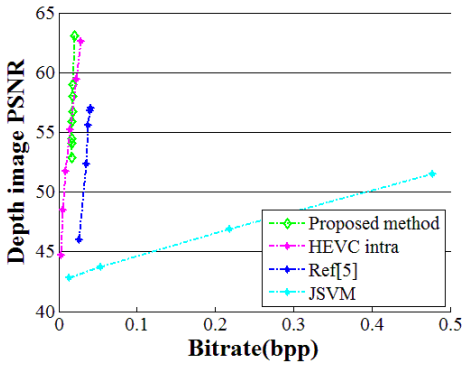
The Peak Signal to Noise Ratio (PSNR) is employed to measure the performance of depth image coding, while the average PSNR of Y, U, V components for virtual view is calculated for the virtual view as the objective evaluation of a synthesized image. In **Fig. 7**, 'V-i to V-j' represents that View-i is the reference view with View-j being the virtual view, and 'bpp' refers to bit-per-pixel. It gives the comparisons of rate distortion performances for depth images and synthesis quality and the comparison of bit-rate allocation for edges and sampled images between proposed method and the Ref [5], where a fixed sampling rate is used and the bit rate is adjusted according to the threshold of Sobel edge detection, following the configuration of Ref [5].

It is obvious that the rate-distortion performances of the virtual color image and depth map by our scheme are better than that using SVC of JSVM software [6] and the Ref [5] method, and are comparable with HEVC intra for all tested sequences. This comes from that we use an edge detection based on synthesized quality and a scalable coding structure of edge and sampled pixels. Meanwhile, the bit-rate allocation for edge and sampled images is the biggest difference between our scheme and the scheme in the previous study [5], as shown in **Fig. 7 (a2-e2)**. From the results, it can been seen that, at lower layers, the proposed allocation scheme gives more bits to the sampled images under the condition of keeping the

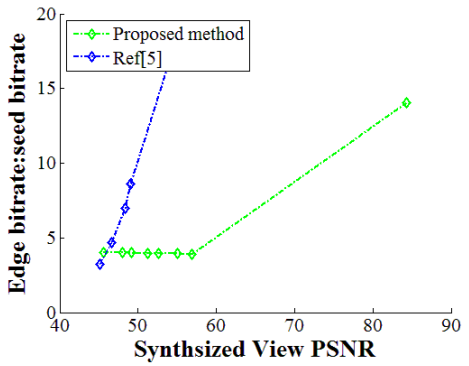
significant edges rather than ceaselessly increases the bits of edges. At the same time, because the information of those insignificant edges does not greatly affect the quality of synthesized view, we do not consume any bits for them. Benefiting from this bit-rate allocation, the proposed scheme has better performances of rate-distortion than other schemes. In addition, compared with these non-scalable methods our proposed scalable coding has many advantages, such as flexible storage management and the flexible bandwidth applicability with a multi-layer embedded bit stream, which enables it to support multiple devices and network access simultaneously. Meanwhile, although SVC can also provide the scalability with one-time coding, it cannot achieve high efficiency for depth images' coding, because it does not provide enough protection for the edges of depth images.



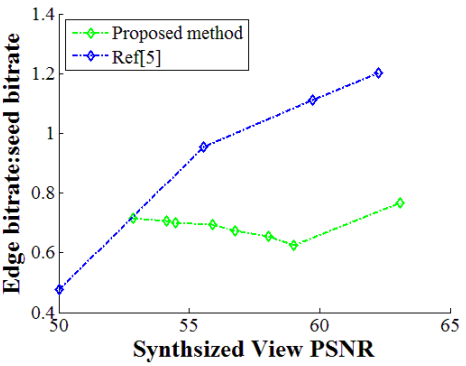
（a1）



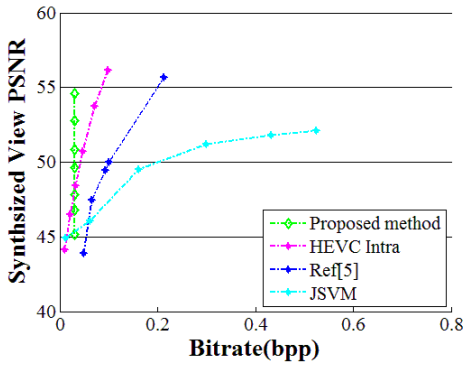
（b1）



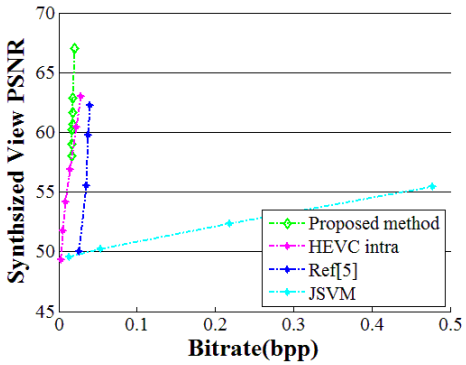
（a2）



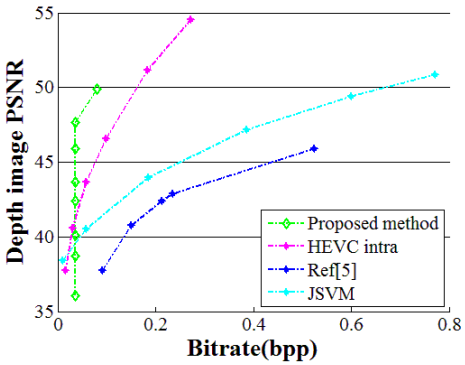
（b2）



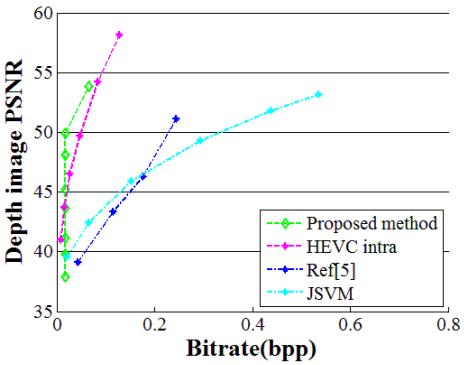
（a3）



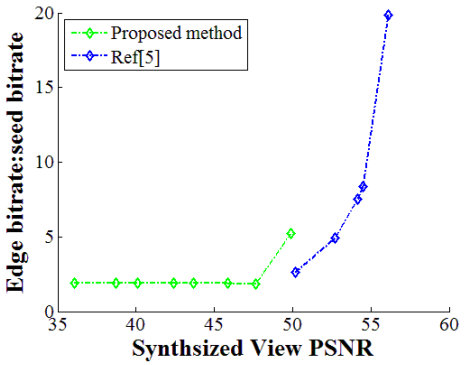
（b3）



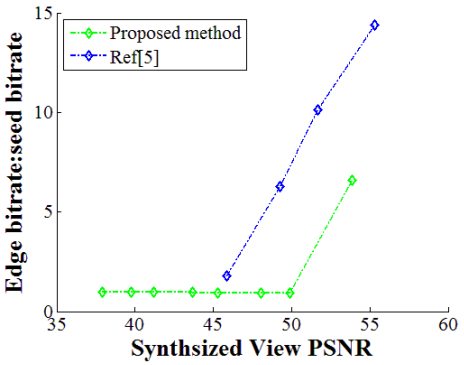
（c1）



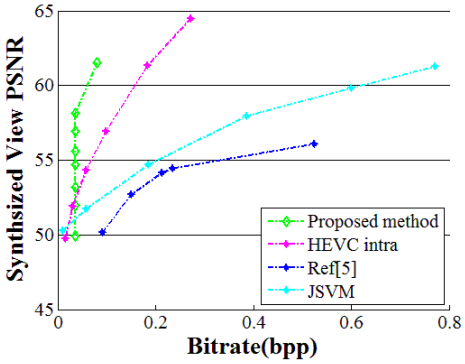
（d1）



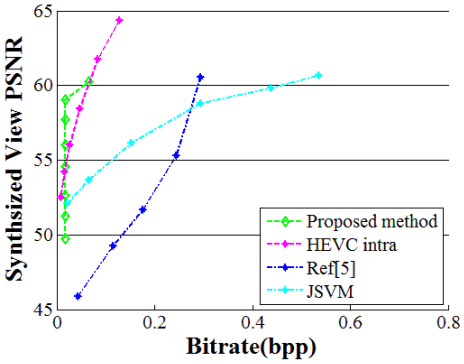
（c2）



（d2）



（c3）



（d3）

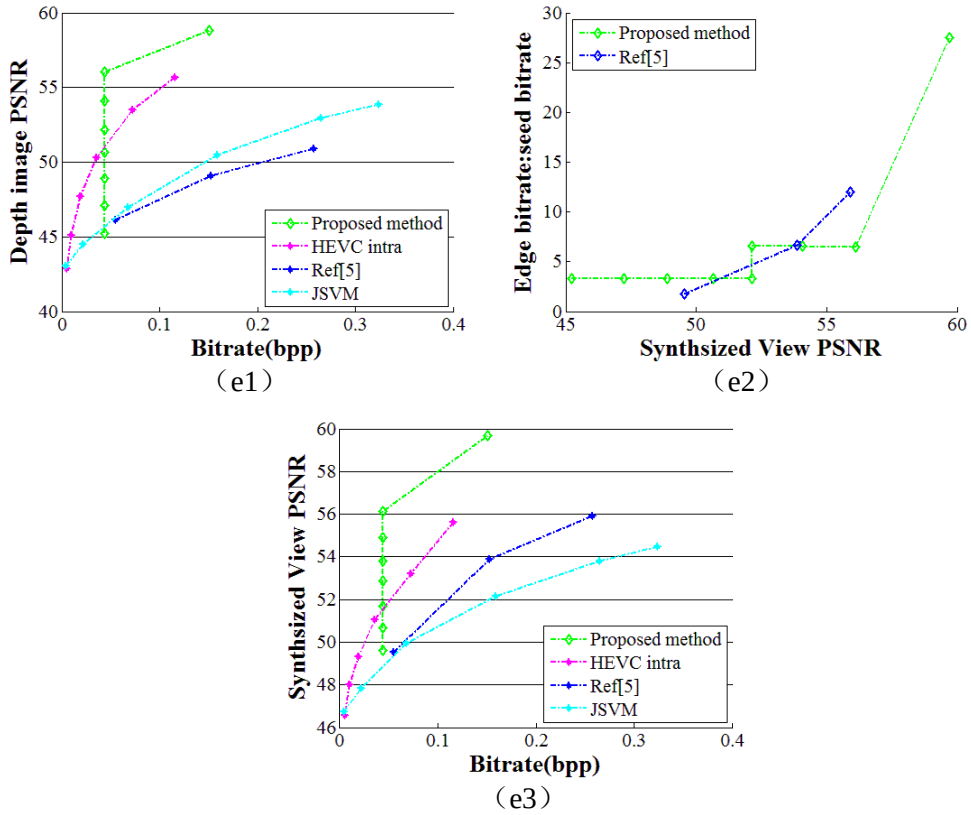


Fig. 7. The rate-distortion performance comparison of depth image ((a1)-(e1)) and the synthesized color image ((a3)-(e3)); and the bit-rate allocation for edge and sampled images ((a2)-(e2)). (a) "Champagne_Tower" (V-37 to V-38), (b) "Undo_Dancer" (V-1 to V-3), (c) "Book-Arrival" (V-8 to V-9), (d) "Kendo" (V-1 to V-2), (e) "Love_Bird" (V-6 to V-7).

4.2 Visual Comparison of Synthesized Image

We compare the visuals of synthesized images of a depth frame from "Book_Arrival" as shown in **Figs. 8-Fig. 11**. Apparently our subjective qualities are better than HEVC intra coding. This is a result of the block-wise DCT and quantization methods of HEVC making the AC components of DCT coefficients mostly quantized to zero. Because the AC components are corresponding to the boundary regions of depth images, this quantization will cripple the sharp edges and make it noisy with blurring, and will consequently lead to the artifacts around the edges in the depth image and the deformation of objects and ringing artifacts in the synthesized view, even if the color image is uncompressed. On the contrary, our scheme puts emphasis on preserving significant edge information, so the quality of the proposed method provide good performances not only at a low bit-rate, but also at a higher bit rate.

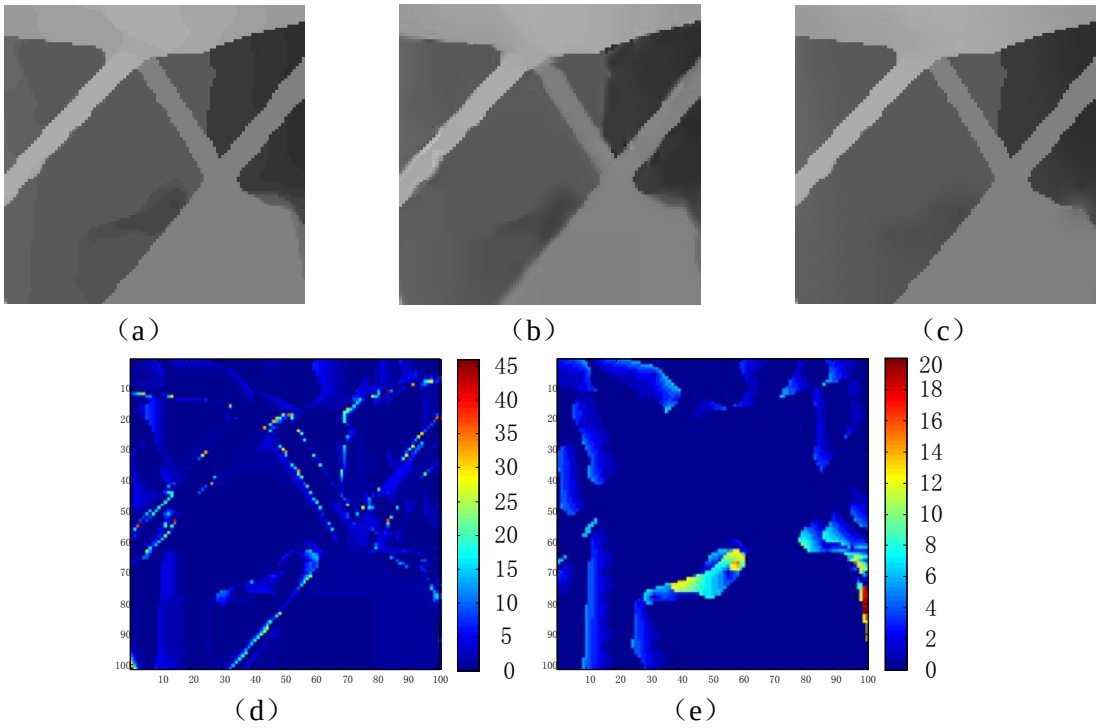


Fig. 8. The comparison of part depth image (a-e), (a)original, (b) HEVC intra at 0.031 bpp, (c) proposed method at 0.036 bpp, (d)difference between (a) and (b), (e)difference between (a) and (c).

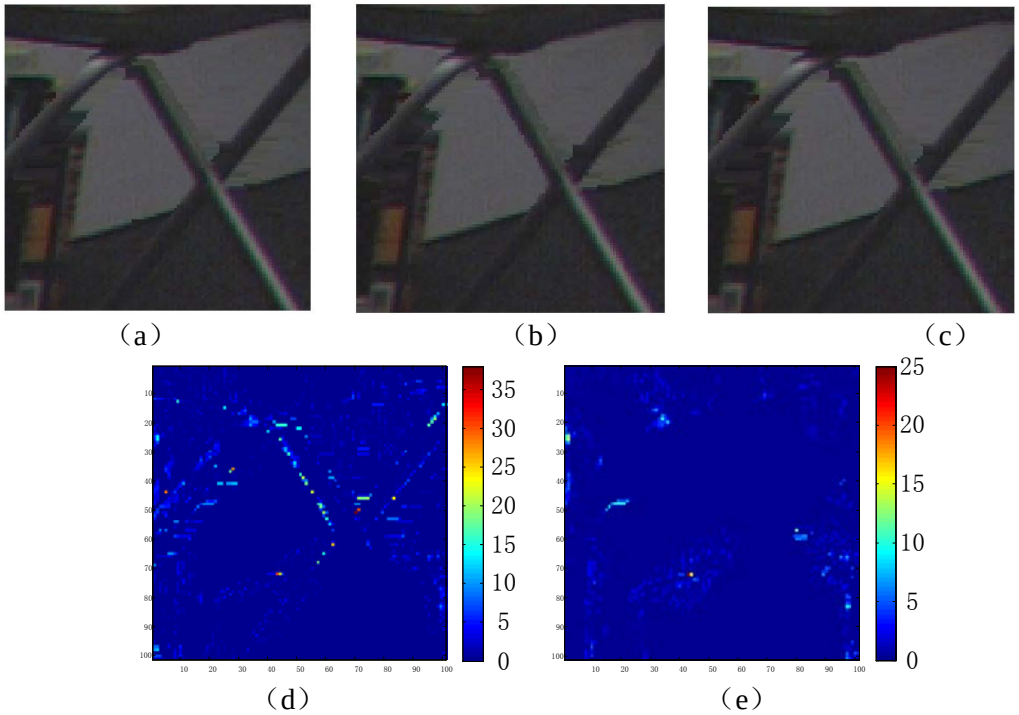


Fig. 9. The comparison of part synthesized image (a-e), (a)original, (b) HEVC intra at 0.031 bpp, (c) proposed method at 0.036 bpp, (d)difference between (a) and (b), (e)difference between (a) and (c).

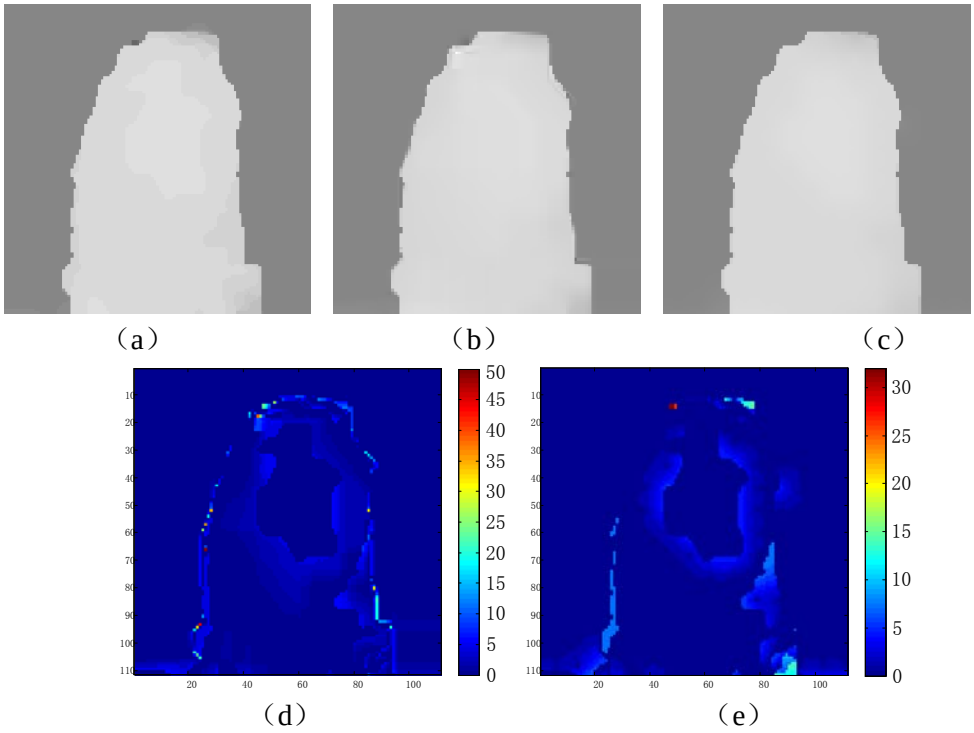


Fig. 10. The comparison of part depth image (a-e), (a)original, (b) HEVC intra at 0.031 bpp, (c)proposed method at 0.036 bpp, (d)difference between (a) and (b), (e)difference between (a) and (c).

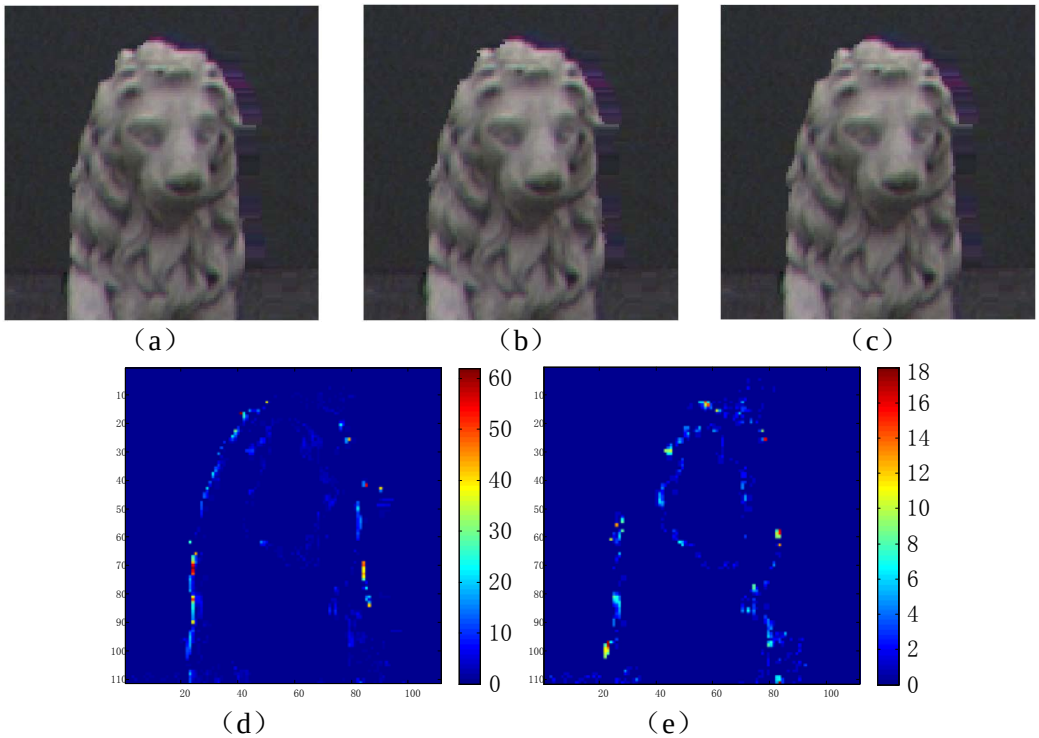


Fig. 11. The comparison of part synthesized image (a-e), (a)original, (b) HEVC intra at 0.031bpp, (c)proposed method at 0.036 bpp, (d)difference between (a) and (b), (e)difference between (a) and (c).

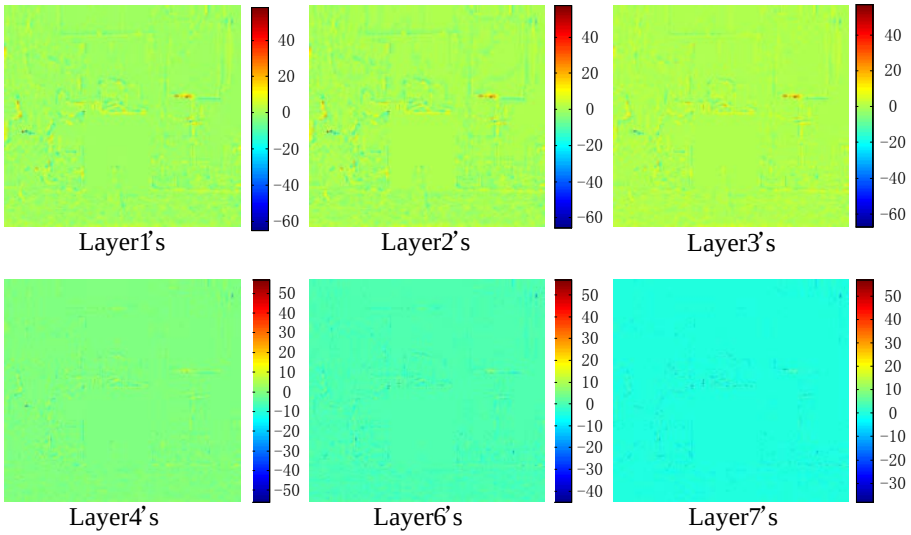


Fig. 12. The example of proposed inter-layer prediction residuals without sampling for "Book_Arrival" depth video's 1st frame

4.3 Illustration of Prediction Residuals from Inter-layers Coding

Fig. 12 shows the residuals between the original image and the inter-layered prediction, which follows the principle of Subsection 3.4. We can see that the redundancy of inter-layers can be removed so that the rate distortion performance can be improved.

5. Conclusions

In this paper, a scalable depth coding scheme based on the synthesis-directed edge detection and scalable stream structure is proposed. Because the significant edges are firstly detected considering the processing of synthesis, and because the lossless coding of significant edges offers better edges, the synthesized quality, especially along the edges are better. Also due to the scalable stream structure, the bit steam of the depth image is suitable to be transmitted over heterogeneous networks. The reconstructions of lower layers are utilized as the inter-layered prediction for higher layers, which can efficiently encode the sampled pixels. Experimental results show that our proposed method achieves a better visual quality than the referenced edge lossless coding method and the rate distortion performances of our proposed method is comparable with HEVC intra coding. However, depth coding does not utilize the correlations between depth and texture images as well as the temporal correlations, which will be left for the future work.

References

[1] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," *IEEE International Conference on Image Processing*, vol. 1, pp. 201-204, September, 2007. Article (CrossRef Link)

[2] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Proc. of SPIE 5291, Stereoscopic Displays and Virtual Reality Systems*

XI, 93, May 21, 2004. Article (CrossRef Link)
[3] R. Krishnamurthy, B. Chai and H. Tao, "Compression and transmission of depth maps for image-based rendering," in *Proc. of IEEE International Conference on Image Processing*, vol. 3, pp. 828-831, 2001. Article (CrossRef Link)
[4] Y. Morvan, P. H.N.With, et al., "Platelet-based coding of depth maps for the transmission of multiview images," *Electronic Imaging 2006*. International Society for Optics and Photonics, 2006. Article (CrossRef Link)
[5] G. Josselin, O. Meur, et al., "Efficient depth map compression based on lossless edge coding and diffusion," in *Proc. of IEEE Picture Coding Symposium (PCS)*, pp. 81-84, Krakow, 7-9 May 2012. Article (CrossRef Link)
[6] H. Schwarz, D. Marpe, et al., "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, 2007. Article (CrossRef Link)
[7] Y. Li, M. Sjostrom, U. Jennehag, et al. "A scalable coding approach for high quality depth image compression," in *Proc. of IEEE 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2012: 1-4. Article (CrossRef Link)
[8] V. Velisavljevic, V. Stankovic, et al. "View and rate scalable multi-view image coding with depth-image-based rendering," in *Proc. of IEEE 17th International Conference on Digital Signal Processing (DSP)*, 2011. Article (CrossRef Link)
[9] D. Tian, P.L. Lai, et al., "View synthesis techniques for 3D video," *Proc. SPIE 7443, Applications of Digital Image Processing XXXII, 74430T*, September, 2009. Article (CrossRef Link)
[10] L. Zhang, and W.J. Tam, "Stereoscopic image generation based on depth images for 3DTV," *IEEE Transactions on Broadcasting*, vol. 51, no. 2, pp. 191-199, 2005. Article (CrossRef Link)
[11] K. Muller, H. Schwarz, et al., "3D High-Efficiency Video Coding for Multi-View Video and Depth Data," *IEEE Transactions on Image Processing*, vol.22, no.9, pp. 3366-3378, 2013. Article (CrossRef Link)
[12] M. Mainberger, A. Bruhn, et al., "Edge-based compression of cartoon-like images with homogeneous diffusion," *Pattern Recognition*, vol. 44, no. 9, pp. 1859-1873, 2011. Article (CrossRef Link)
[13] K.W. Morton and D.F. Mayers, "Numerical solution of partial differential equations," *Journal of Fluid Mechanics*, vol. 363, pp. 349-349, 1998. Article (CrossRef Link)
[14] HEVC Test Software [online]. Available:https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.5/ Article (CrossRef Link)
[15] C.G. Yan, et al., "Efficient parallel framework for HEVC motion estimation on many-core processors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 12, pp. 2077-2089, 2014. Article (CrossRef Link)
[16] C.G. Yan, et al. "A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors," *IEEE Signal Processing Letters*, vol. 21, no. 5, pp. 573-576, 2014. Article (CrossRef Link)
[17] ISO/IEC JTC1/SC29/WG11, "Draft call for proposals on 3D video coding technology," *MPEG2011/N11830*, Daegu, Korea, January 2011. Article (CrossRef Link)
[18] 3D-HEVC Test Software (HTM) [online]. Available:http://hevc.kw.bbc.co.uk/git/w/jctvc-3de.git/shortlog/refs/heads/HTM-DEV-2.0-dev3-Zhejiang Article (CrossRef Link)

Lijun Zhao received his B.S. and M.E. degrees in Taiyuan University of Technology, Taiyuan University of Science and Technology (TYUST) respectively in 2011 and 2015 respectively. And now he is pursuing his PHD degree in Institute of Information Science, Beijing Jiaotong University. His research interests include video coding, image processing, pattern recognition, and computer vision.

Anhong Wang received B.E. and M.E. degrees from Taiyuan University of Science and Technology (TYUST) respectively in 1994 and 2002, and PHD degree in Institute of Information Science, Beijing Jiaotong University in 2009. She became an associate professor with TYUST in 2005 and became a professor in 2009. She is now the director of Institute of Digital Media and Communication, Taiyuan University of Science and Technology. Her research interests include image and video coding, compressed sensing, and secret image sharing. She has published more than 70 papers. Now she is leading two national research projects from National Science Foundation of China.

Bing Zeng was born in Neijiang, a small city in Sichuan, China, and grew up in a village full of tangerine and orange trees. He enjoyed very much the college life that earned him two degrees (B.E. and M.E.) in 1983 and 1986, respectively, both in electronic engineering. After two years of hanging around in Chengdu, he determined to make another change. He got his Ph.D. degree in electrical engineering from Tampere University of Technology, Tampere, Finland, in 1991. After that, he moved to another cold country to continue his postdoctoral research at the University of Toronto and Concordia University. He joined the Hong Kong University of Science and Technology in early 1993 where he is currently an associate professor in the Department of Electrical and Electronic Engineering. He has been affiliated with IEEE for more than 10 years and particularly has had the pleasure to serve as an Associate Editor for its Transactions on Circuits and Systems for Video Technology (during 1995-99). In the meantime, in order to show the loyalty, he has been trying hard to make contributions to various journals and conferences (organized or sponsored by IEEE).

Jian Jin was born in Hebei Province. He received B.E. and M.E. degrees from Taiyuan University of Science and Technology (TYUST) respectively in 2011 and 2014, and now he is pursuing his PHD degree in Institute of Information Science, Beijing Jiaotong University. His research interests include video processing and 3D image rendering.