A Depth-Bin-Based Graphical Model for Fast View Synthesis Distortion Estimation

Jian Jin, Jie Liang, *Senior Member, IEEE*, Yao Zhao, *Senior Member, IEEE*, Chunyu Lin, Chao Yao, and Anhong Wang

***Abstract*— During 3-D video communication, transmission errors, such as packet loss, could happen to the texture and depth sequences. View synthesis distortion will be generated when these sequences are used to synthesize virtual views according to the depth-image-based rendering method. A depth-value-based graphical model (DVGM) has been employed to achieve the accurate packet-loss-caused view synthesis distortion estimation (VSDE). However, the DVGM models the complicated view synthesis processes at depth-value level, which costs too much computation and is difficult to be applied in practice. In this paper, a depth-bin-based graphical model (DBGM) is developed, in which the complicated view synthesis processes are modeled at depth-bin level so that it can be used for the fast VSDE with 1-D parallel camera configuration. To this end, several depth values are fused into one depth bin, and a depth-bin-oriented rule is developed to handle the warping competition process. Then, the properties of the depth bin are analyzed and utilized to form the DBGM. Finally, a conversion algorithm is developed to convert the per-pixel input depth value probability distribution into the depth-bin format. Experimental results verify that our proposed method is 8–32 × faster and requires 17%–60% less memory than the DVGM, with exactly the same accuracy.**

***Index Terms*— 3-D video coding, depth-image-based rendering (DIBR), distortion estimation, graphical model.**

I. Introduction

A. Motivation

3-D VIDEO technologies have been widely studied recent years as they can provide immersive 3-D experience. 3-D videos are usually represented by Multi-view Videos plus Depth (MVD) [1] format, where the color videos of the 3-D scenarios are captured by several cameras at different locations, and the associated depth videos are obtained by estimation algorithms [2] or directly captured by depth cameras [3]. By transmitting the MVD data, a virtual view between any two captured views can be rendered in the decoder side utilizing the Depth Image-based Rendering (DIBR) technology [4], which has been adopted in the Moving Picture Experts Group (MPEG) View Synthesis Reference Software (VSRS) [5].

In contrast to the traditional 2-D video (single-view video), to ensure 3-D real-time visualization at the decoder side, MVD usually requires more transmission bandwidth. In other words, when the transmission bandwidth is restricted, network congestion will be more common in 3-D video transmission and will further cause transmission impairments, such as packet loss. Due to the predictive nature of the encoder, losing a part of a frame can cause error propagation to subsequent frames. This scenario not only introduces errors to the transmitted texture and depth images, but also affects the quality of the synthesized view when they are used as references. Besides, compared with the traditional packet-loss-induced distortion in single-view videos, where only color information is changed during 2-D video reconstruction, the distortion in depth videos may create excessive disparity error, which could lead to unacceptable geometric distortion in the synthesized virtual view.

Therefore, in 3-D video system, it is crucial to develop an accurate algorithm for the encoder to estimate the packet-loss-induced distortion of the synthesized view at the decoder. It may help designing various error-resilient tools at the encoder to improve the quality of 3-D video. In 2-D video transmission, the well known recursive optimal per-pixel estimate (ROPE) method [6] is capable of estimating packet-loss-induced distortion by estimating the first and second moments of each decoded pixel during encoding. Inspired by this, a ROPE-like scheme is developed in [7] to estimate the decoder-side distortion of the synthesized view from encoder, based on a depth-value-based graphical model (DVGM) that can handle the complex warping competition during view synthesis process. However, the main drawbacks of the DVGM are its high computational and memory costs, making it difficult to be used in real-time applications. Therefore, a more efficient view synthesis distortion estimation (VSDE) model is desired. This is the main motivation of the paper.

Manuscript received October 29, 2017; revised April 30, 2018; accepted May 22, 2018. Date of publication June 7, 2018; date of current version June 4, 2019. This work was supported by the National Key Research and Development of China under Grant 2016YFB0800404, in part by the National Natural Science Foundation of China under Grant 61532005, Grant 61772066, and Grant 61672373, in part by the China Scholarship Council, and the Engineering Research Council (NSERC) of Canada under Grant RGPAS478109. This paper was recommended by Associate Editor Z. Wang. *(Corresponding author: Yao Zhao.)*

J. Jin, Y. Zhao, and C. Lin are with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China, and also with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China (e-mail: jianjin@bjtu.edu.cn; yzhao@bjtu.edu.cn; cylin@bjtu.edu.cn).

J. Liang is with the School of Engineering Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: jiel@sfu.ca).

C. Yao is with the Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yaochao1986@gmail.com).

A. Wang is with the School of Electronic Information Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China (e-mail: wah_ty@163.com).

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/TCSVT.2018.2844743

B. Related Works

1) VSRS: Various view synthesis algorithms have been developed in [8]–[11]. In particular, in [11], a region-aware 3-D warping for the DIBR is proposed, which exploits the characteristics of different regions in the reference views and achieves significant computation saving with little degradation in the synthesis quality. However, the most widely used algorithms are those in the VSRS3.5 [5], which have superior performance on synthesis quality especially when the depth sequences are corrupted as studied in [11]. There are two modes in VSRS3.5. The 1-D parallel mode is designed for scenes captured by parallel camera array with only horizontal disparity, while the general mode has no restriction on the cameras array. Generally, 1-D parallel mode is more popular in 3-D video research and applications.

The framework of VSRS3.5 with 1-D parallel mode mainly contains the *warping* and *blending* stages. For the *warping* stage, pixels from the original views are projected to the virtual view, which includes the following steps: i) boundary detecting and boundary aware splatting are first executed in the depth map; ii) forward warping is then carried out. During the forward warping step, the mapping competition could happen, that is, several pixels in the reference views are mapped to the same point in the virtual view. To handle this problem, the depth-value-oriented rule is used in VSRS3.5, where the reference pixel with the largest depth value (closest to the camera) is selected as the winner. After that, the *blending* stage is carried out, which includes two steps: i) the two warped views from *warping* stage are firstly blended into one, which contains holes; ii) all the holes are filled by an inpainting operation. By utilizing VSRS3.5, the decoder can obtain the virtual view at arbitrary location between two neighboring reference views.

2) Source-Coding-Caused VSDE: Various methods have been developed to estimate the view synthesis distortion caused by source coding. In [12], a region-based synthesized view distortion estimation algorithm was proposed for depth map coding. In [13], a linear model-based virtual view distortion estimation method is developed and employed to optimally select the skipping mode for depth map coding. In [14], for the joint bit allocation between texture and depth sequences in 3-D video coding, a model-based view distortion estimation algorithm is developed. In [15], the structured similarity [16] is used to measure the subjective quality of the synthesized view, and to optimize the codec. In [17], the quantization-caused distortion is assumed to be a zero-mean white noise [14], and the distortion in the synthesized view is decomposed as the sum of texture image coding distortion and depth image coding distortion, which could estimate the virtual view distortion accurately. However, this method also has high computational complexity. To make a good trade-off between the actuary and efficiency, a method is developed in [18] to achieve a virtual view PSNR estimation directly without rendering virtual views. Besides, this method is also friendly for parallel implementation due to its row-by-row processing order.

3) Transmission-Error-Caused VSDE: The algorithms above only considered the impact of source coding on the synthesized view rather than the impact of transmission error. In [19], a recursive distortion model is developed for multi-view video transmission over lossy packet-switched networks, which estimates the expected channel-induced distortion at both the frame and sequence levels. However, this algorithm does not consider the MVD format. To relate the disparity errors caused by packet loss in the depth maps to the distortion contribution in the synthesized view, a quadratic model is proposed in [20]. In [21]–[23], to make the reference frame selection and optimize the quantization parameter, a quadratic-model-based distortion estimation is developed and used at the encoder. However, The overall transmission distortion estimation framework used in [21]–[23] is the block-based recursive approach and its estimation accuracy is quite limited. In [24], to improve error resilience of MVD, an end-to-end distortion model for MVD-based 3-D video transmission is proposed for rate-distortion optimized mode selection, where both the end-to-end distortions in the rendered view and the compressed texture video are characterized. Then, the view synthesis prediction is also considered in [25]. Note that it only focuses on modeling the right reference view. However, it still ignores some details in the complex view synthesis process. Therefore, its estimation accuracy is not optimal. Another drawback is its high computational complexity.

In [7], a depth-value-based graphical model (DVGM) is developed to capture the complicated warping competition during view synthesis process. Besides, a recursive optimal distribution estimation (RODE) approach is developed based on the well known ROPE to generate per-pixel texture and depth probability distributions. By integrating the RODE into the DVGM, this approach can estimate the packet-loss-induced 3-D video distortion accurately. However, the DVGM is formulated at depth-value level. Generally, depth value changes within a certain range may not lead to warping error due to the 3-D warping rounding operation in DIBR. Therefore, the DVGM is inefficient in describing the complicated view synthesis process, and can be sped up for a faster VSDE.

C. Contributions of This Paper

In this paper, a novel efficient depth-bin-based graphical model (DBGM) is presented for 1-D parallel mode, which can replace the DVGM in [7]. The main contributions are listed as follows.

- The concept of depth bin is firstly defined. At the same time, a depth-bin-oriented warping competition rule is developed.
- The DBGM is developed, which is the first work to formulate the complicated view synthesis process at depth-bin level to simplify the VSDE.
- The properties of depth bin are studied and utilized to optimize the DBGM further.
- A conversion of probability distribution between depth bin and depth value is developed so that it can be used to integrate the RODE method into the DBGM directly.

- The DBGM method is 8 to 32 times faster and requires 17% to 60% less memory than the DVGM, with exactly the same accuracy.

The rest of the paper is organized as follows. Section II reviews the related techniques on the packet-loss-caused VSDE algorithm proposed in [7]. Section III details the proposed model. Section IV presents the experimental results, and Section V concludes this paper.

II. VSDE Algorithm Overview

In this section, we briefly review the main ideas of the packet-loss-caused VSDE proposed in [7], which contains the overall framework of VSDE and the DVGM. The details are summarized in the appendices.

The framework of VSDE contains two main steps. The first step is to establish a function to generate the per-pixel expected distortion in the synthesized view, which is formulated by Eq. (25) in *Appendix*. Once the first and second moments of each synthesized pixel at the receiver are obtained, this function can be solved. The second step is to represent the required first and second moments of each pixel in synthesized view at the receiver with several components, which are formulated by Eq. (26) and Eq. (27) in *Appendix*, in which the distributions of the warped depth pixels are needed.

To generate the distributions of the warped depth pixels, a depth value-based graphical model (DVGM) was developed in [7], which contains three main steps. The first step is to calculate the winning probability of each edge emitting from reference vertex to warped vertex as formulated in Eq. (28) in *Appendix*. Besides, to handel the complex warping competition scenario, Eq. (28) is implemented with a condition that the depth value of the winner edge is the largest. The second step is to sum up all these winning probabilities of edges with the same starting and ending vertices in order to expresses the probability of one warped vertex taking the value of one reference vertex as expressed in Eq. (29) in *Appendix*. In the last step, the probability of one warped vertex taking no value from any reference vertices is calculated in Eq. (30) in *Appendix*.

As DVGM assumes that the distribution of random noise is known, the per-pixel distribution in the reference depth images can be derived. Finally, the distribution of the synthesized depth pixel is obtained according to Eq. (29) and Eq. (30). Hence, the view synthesis distortion is finally estimated. In fact, the initial per-pixel distribution in both reference texture and depth images can be derived by the RODE method. The distribution generated with the RODE depends on several factors, such as the slice mode selection, packet loss probability, the error concealment scheme and so on.

There are some drawbacks in the DVGM, which cause high complexity and large memory storage. In order to obtain the probability that a warped vertex will take the texture value from a reference vertex, the winning probability of each edge emitting from the reference vertex to the warped vertex is first calculated based on their depth values. Then, the winning probabilities of all the edges with the same starting and ending vertices are summed up together. However, all of these edges describe the same physical event that one reference vertex will be warped to one reference vertex. Hence, depth-values-based edge representation is inefficient. Besides, during winning probability calculation in Eq. (28), a large amount of information on depth values and locations is needed to be recorded in advance, which needs large memory. Meanwhile, the depth value of the winner edge is required to be the largest during warping competition. Hence, a sorting algorithm is needed in winning probability calculation in the DVGM, and these edges are sorted based on their associated depth values, which is time consuming. In summary, all these disadvantages in the DVGM are caused by modeling the complicated view synthesis at the depth-value level.

III. A Novel Efficient Depth-Bin-Based Graphical Model

An important fact in 3-D warping process is that several depth values could correspond to the same rounded disparity value due to the rounding operation. This can be represented in DVGM by multiple edges with different depth values emitting from reference vertex to warped vertex. This fact has also been observed and utilized in several video coding papers. For instance, [26] develops a depth no-synthesis-error (D-NOSE) model based on this fact, which is used to design depth video coding. In [27], this fact is utilized to design a quantizer to represent the physical depth distance with less bits. Different from [26] and [27], we are the first to use this fact to model the complicated view synthesis at depth-bin level so that the complexity and memory consumption of transmission-error-caused VSDE can be reduced.

In this section, we develop a novel efficient depth-bin-based graphical model (DBGM), and discuss several main techniques used in the DBGM. Similar to the DVGM, full pixel precision of view synthesis is considered in this paper.

A. Depth Bin

As reviewed above in the DVGM, each edge corresponds to a certain depth value with a floating-point disparity. In 1-D parallel view synthesis, given the depth value d of a point in the 3-D space, the disparity of its images in the reference view and virtual view can be obtained by first using the following equation in [11]:

$$\delta = \frac{f \cdot L \cdot d}{255} \cdot \left(\frac{1}{Z_{near}} - \frac{1}{Z_{far}}\right) + \frac{f \cdot L}{Z_{far}}, \tag{1}$$

where δ is the initial floating-point disparity. Z_{near} and Z_{far} denote the depth range of the physical scene. f is the camera focal length. L is the distance between virtual view and reference view. Eq. (1) can be rewritten as

$$\begin{cases} \delta = c_1 \cdot d + c_2 \triangleq D(d), \\ c_1 = \dfrac{f \cdot L}{255} \cdot \left(\dfrac{1}{Z_{near}} - \dfrac{1}{Z_{far}}\right) > 0, \\ c_2 = \dfrac{f \cdot L}{Z_{far}} > 0, \end{cases} \tag{2}$$

where c_1 and c_2 are positive constant. Besides, c_1 is usually smaller than 1. Therefore, δ could be regarded as a positive

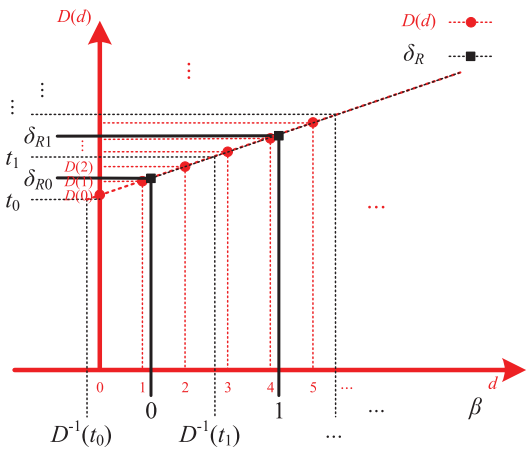


Fig. 1. Illustration of relationships among depth value d, associated floating-point disparity $D(d)$, depth bin β and rounded disparity δ_R. δ_{R0} and δ_{R1} are the neighbored integer rounded disparities.

linear function of d and represented by $D(d)$. Hence, its inverse function can be written as

$$d = D^{-1}(\delta). \tag{3}$$

Then, Eq. (2) is rounded to get the integer disparity:

$$\delta_R = [D(d)] = [c_1 \cdot d + c_2], \tag{4}$$

where $[\cdot]$ represents the rounding operation, and δ_R is the rounded disparity.

In this paper, we define the set of all the depth values with the same rounded disparity as a *depth bin*, which is denoted as β.

Assuming that the 256 depth values are finally mapped to N depth bins, N could be expressed as

$$\begin{cases} N = [D(d_{near})] - [D(d_{far})] + 1, \\ d_{near} = 255, \\ d_{far} = 0, \end{cases} \tag{5}$$

The index of depth bin β is from 0 to $N-1$.

Hence, there is an one-to-one relationship between the depth bin β and rounded disparity δ_R. The 3-D warping rounding calculation is reformulated as

$$\delta_R = \beta + [D(d_{far})] = \beta + [D(0)], \tag{6}$$

which will be explained latter in this part.

The relationship between the depth values and their corresponding depth bins can be described by Eq. (7), as shown at the bottom of the next page, where $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ are the ceiling and flooring operations. The details of derivation are given as follows.

The detailed relationship between the depth values, floating-point disparities, rounded disparity and depth bin are shown in Fig. 1, where depth values 0, 1, 2, ... are mapped to floating-point disparities $D(0)$, $D(1)$, $D(2)$, ... After rounding operation, some neighboring floating-point disparities will have the same rounded value. Their corresponding depth values will form a depth bin, e.g., $D(0)$, $D(1)$ and $D(2)$ will have the same rounded value δ_{R0}. Their corresponding depth values 0, 1 and 2 will form a depth bin with index 0.

Mathematically, to find all the depth values that are included in the i-th depth bin, we need to find its lower and upper bounds of depth values, which are denoted as $d_{L,i}$ and $d_{U,i}$, respectively.

We start from the first depth bin with index 0. The floating-point disparities will be rounded into δ_{R0} in the range of t_0 and t_1, where

$$\begin{cases} t_0 = [D(0)] - 0.5 \\ t_1 = [D(0)] + 0.5. \end{cases} \tag{8}$$

The corresponding integer boundaries of depth bin with index 0 can be founded from inverse function as follows

$$\begin{cases} d_{L,0} = \lceil D^{-1}(t_0)\rceil = \lceil D^{-1}([D(0)] - 0.5)\rceil = 0 \\ d_{U,0} = \lfloor D^{-1}(t_1)\rfloor = \lfloor D^{-1}([D(0)] + 0.5)\rfloor. \end{cases} \tag{9}$$

For the depth bin with index $i\,(0 \le i \le N-2)$, we have

$$\begin{cases} d_{L,i} = \lceil D^{-1}([D(0)] - 0.5 + i)\rceil \\ d_{U,i} = \lfloor D^{-1}([D(0)] + 0.5 + i)\rfloor. \end{cases} \tag{10}$$

For the last depth bin with index $N-1$, we get

$$\begin{cases} d_{L,N-1} = \lceil D^{-1}([D(0)] - 0.5 + N - 1)\rceil \\ d_{U,N-1} = \lfloor D^{-1}([D(0)] + 0.5 + N - 1)\rfloor = 255. \end{cases} \tag{11}$$

Combining Eq. (9) to (11), we can get Eq. (7). Based on this equation, we can easily find the depth bin index of any depth value.

B. Depth-Bin-Oriented Warping Competition Rule

During 1-D parallel view synthesis, if two vertices with different locations in reference are warped to the same vertex in the warped view, there will be a warping competition between these two vertices. In the traditional depth-value-oriented warping competition rule, the vertex with the largest depth value will be chosen as the winner, which is considered as belonging to the foreground.

In this subsection, we propose a depth-bin-oriented warping competition rule, from which the vertex with largest depth bin is chosen as the winner during warping competition. Essentially, this proposed rule still chooses the largest depth value as the winner, because the depth values within the largest depth bin are always larger than those within a smaller one. This fact is proved as follows.

Assume two vertices A and B with different locations in the reference view are warped to the same location in the warped view. Their depth bin are β_A and β_B, respectively. Let the indexes of β_A and β_B be i and j $(i > j)$. Let $d_{L,i}$ and $d_{U,j}$ respectively denote the smallest depth value (lower bound) within β_A and the largest depth value (upper bound) within β_B. Similarly, as discussed in Eq. (6), we have

$$\begin{cases} [D(d_{L,i})] = \beta_A + [D(0)], \\ [D(d_{U,j})] = \beta_B + [D(0)], \end{cases} \tag{12}$$

since $i > j$, we get

$$[D(d_{L,i})] - [D(d_{U,j})] = \beta_A - \beta_B = i - j > 0. \tag{13}$$

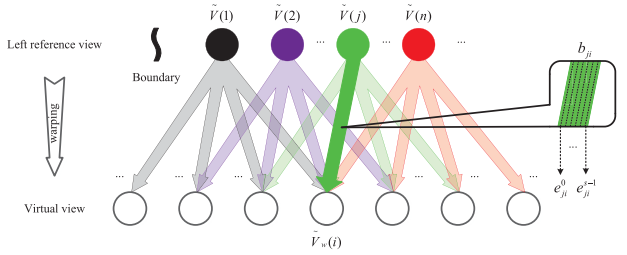


Fig. 2. Depth-bin-based graphical model for view synthesis. The vertexes on the first line are the pixels from left reference view, which will be warped to the vertexes on the second line of the virtual view. The big arrows are the bundles. To exhibit the relationship between the bundle and edge, we randomly highlight and zoom in bundle b_{ij}, which connects vertex $\tilde{V}(i)$ and $\tilde{V}_w(j)$. We can clearly find bundle b_{ij} includes edge e_{ij}^0 to edge e_{ij}^{s-1}, which have the same starting and ending and represented with black dotted line.

As $D(d)$ is positive linear function with respect to d, we always have $d_{L,i} > d_{U,j}$. Therefore, the depth values within a larger depth bin index are always larger than those in a depth bin with smaller index. Our depth-bin-oriented warping competition rule still chooses the largest depth value as the winner. However, in the worst case (warping competition), 256 depth value indexes will be compared in depth-value-oriented warping competition rule, while less number of depth bin indexes will be considered in our rule. Therefore, it is more efficient to use our rule during warping competition.

C. Depth-Bin-Based Graphical Model

In this part, we use the defined depth bin to build a fast graphical model, which can efficiently capture the interaction between the vertices and the warping competition operation during view synthesis.

Different from the DVGM, all the edges (each edge corresponds to a depth value) with the same starting and ending are grouped into a bundle (each bundle corresponds to a depth bin) in our model, as shown in Fig. 2. We use different colors to represent different vertices texture values. Therefore, all the edges emitting from each vertex in the reference view will be grouped into several bundles in our proposed model. The number of bundles used in our model is usually several times smaller than that of edges used in the DVGM. This can largely reduce the complexity of the model.

To obtain the probability of one vertex taking the value of another, instead of calculating the winning probability of each individual edge first and then summing up the probabilities of all the edges with the same starting and ending, we can directly calculate the winning probability of the bundle between these two vertices.

As shown in Fig. 2, we assume that only $\tilde{V}(1)$ to $\tilde{V}(n)$ are connected to $\tilde{V}_w(i)$. The bundle between $\tilde{V}(j)$ and $\tilde{V}_w(i)$ and its associated depth bin are denoted as b_{ji} and $\beta(b_{ji})$, respectively. Based on the depth-bin-oriented warping competition rule, when the bundle b_{ji} is the final winner, its $\beta(b_{ji})$ should be the largest. Therefore, all the bundles emitted from previous vertices $\tilde{V}(z)$ $(z = 1, \ldots, j-1)$ to $\tilde{V}_w(i)$ with condition $\beta(b_{zi}) \geq \beta(b_{ji})$, denoted by a set S_3 (which is smaller than $\bigcup_{z=1}^{j-1} S_{z,1}$ in Eq. (28) in *Appendix*), should be abandoned. Similarly, all the bundles emitted from subsequent vertices $\tilde{V}(z)$ $(z = j+1, \ldots, n)$ to $\tilde{V}_w(i)$ with condition $\beta(b_{zi}) > \beta(b_{ji})$, denoted by a set S_4 (which is smaller than $\bigcup_{z=j+1}^{n} S_{z,2}$ in Eq. (28)) should be abandoned as well. Let $P(b_{ji})$ denote the probability that $\tilde{V}(j)$ will be warped to $\tilde{V}_w(i)$ with bundle b_{ji}. The winning probability of bundle b_{ji} is defined as $P_{win}(b_{ji})$, which can be formulated as

$$P_{win}(b_{ji}) = P(b_{ji}) \times \prod_{z \in S_3} (1 - P(b_{zi})) \times \prod_{z \in S_4} (1 - P(b_{zi})). \tag{14}$$

In other words, the separated two-step depth-value-based operations in Eq. (28) and (29) in DVGM are replaced by just one-step depth-bin-based operations in Eq. (14) in our model. The complexity can thus be reduced.

As shown in Eq. (6), there is an one-to-one relationship between depth bin and disparity. Therefore, once the depth bin of $\tilde{V}(j)$ is set as $\beta(c_{ji})$, its associated disparity is confirmed uniquely and $\tilde{V}(j)$ will be warped to $\tilde{V}_w(i)$ undoubtedly. In view of this, the probability that $\tilde{V}(j)$ will be warped to $\tilde{V}_w(i)$ with bundle b_{ji} is equivalent to the probability that the depth bin of $\tilde{V}(j)$ is set as depth bin $\beta(c_{ji})$, which is expressed as

$$P(b_{ji}) \triangleq P(\beta_j = \beta(b_{ji})), \tag{15}$$

where β_j denotes the depth bin of $\tilde{V}(j)$. $P(\beta_j = \beta(b_{ji}))$ denotes the probability that the depth bin of $\tilde{V}(j)$ is set as $\beta(b_{ji})$. Similarly, we also have

$$P(b_{zi}) \triangleq P(\beta_z = \beta(b_{zi})). \tag{16}$$

Plugging Eq. (15) and (16) in Eq. (14), we can get

$$\begin{aligned} P_{win}(b_{ji}) \triangleq\ & P(\beta_j = \beta(b_{ji})) \\ & \times \prod_{z \in S_3} (1 - P(\beta_z = \beta(b_{zi}))) \\ & \times \prod_{z \in S_4} (1 - P(\beta_z = \beta(b_{zi}))), \end{aligned} \tag{17}$$

where the depth bin β_j should be the largest, when warping competition occurs. Hence, the sorting algorithm on depth bins is still needed in Eq. (17). In the next subsection, we will show that the sorting can be further eliminated.

$$\begin{cases} \text{The } 0^{th} \text{ index of } \beta, & \text{corresponds to } d \in [0, \lfloor D^{-1}([D(0)] + 0.5) \rfloor] \\ \text{The } i^{th} (1 \leq i \leq N-2) \text{ index of } \beta, & \text{corresponds to } d \in [\lceil D^{-1}([D(0)] - 0.5 + i) \rceil, \lfloor D^{-1}([D(0)] + 0.5 + i) \rfloor] \\ \text{The } (N-1)^{th} \text{ index of } \beta, & \text{corresponds to } d \in [\lceil D^{-1}([D(0)] - 0.5 + N - 1) \rceil, 255] \end{cases} \tag{7}$$

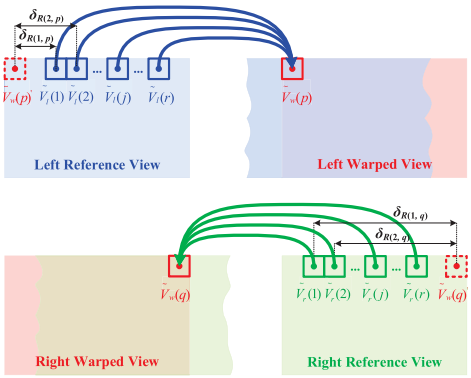


Fig. 3. Exhibition on warping competition in 1-D parallel mode. The blue and green canvases stand for the left and right reference views respectively, which will be warped to final warped view and represented with the red canvas.

To generate the probability distribution of the warped virtual view's depth bins, the probability of $\tilde{V}_w(i)$ taking no value from any edge is still necessary, which can be obtained by

$$P_{\tilde{V}_w(i)}(\phi) = \prod_{j=1}^{n} (1 - P_{win}(b_{ji})). \tag{18}$$

Based on the observations above, the probability distribution of the warped virtual view can be generated more easily by Eq. (17) and Eq. (18) instead of Eq. (28) to Eq. (30). Additionally, S_3 and S_4 are usually several times smaller than $\bigcup_{z=1}^{j-1} S_{z,1}$ and $\bigcup_{z=j+1}^{n} S_{z,2}$, which reduces the associated operations significantly. It should be noted that the results of the new method is equivalent to the old method. Next, we will show that the complexity of Eq. (17) can be further reduced utilizing the properties of depth bin.

D. Winning Probability Function Optimization According to the Properties of Depth Bin

Assume vertices $\tilde{V}_l(1)$, …, $\tilde{V}_l(m)$ in the left reference view are warped to the same target $\tilde{V}_w(p)$ in the left warped view. Similarly, vertices $\tilde{V}_r(1)$, …, $\tilde{V}_r(m)$ in right reference view are warped to the target $\tilde{V}_w(q)$ in the right warped view. $\tilde{V}_l(j)$ is the j-th left vertex, while $\tilde{V}_r(j)$ is the j-th right vertex. The disparities of vertexes $\tilde{V}_l(1)$ and $\tilde{V}_l(2)$ are denoted as $\delta_{R(1,p)}$ and $\delta_{R(2,p)}$, respectively. Based on the definition of disparity in [4], for the left reference view, we have $\delta_{R(j,p)} = \tilde{V}_l(j) - \tilde{V}_w(p)$. Therefore, $\delta_{R(1,p)}$ is always smaller than $\delta_{R(2,p)}$, as shown in Fig. 3. Since the depth bin and rounded disparity have an one-to-one positive correspondence as discussed in Eq. (6), we can derive that the depth bin of $\tilde{V}_l(1)$ is always smaller than that of $\tilde{V}_l(2)$.

For the general case of left reference view, we always have

$$\beta(b_{jp}) < \beta(b_{mp}), \quad \text{where } j < m. \tag{19}$$

Similarly, since the warp direction in the right reference view is opposite to that in left reference view, the disparity is defined as $\delta_{R(j,q)} = \tilde{V}_w(q) - \tilde{V}_r(j)$. In other words, the disparity of $\tilde{V}_r(1)$ is always larger than that of $\tilde{V}_r(2)$. Furthermore, we can derive that the depth bin of $\tilde{V}_r(1)$ is always larger than that of $\tilde{V}_r(2)$. Therefore, for the general case of right reference view, we have

$$\beta(b_{jq}) > \beta(b_{mq}), \quad \text{where } j < m. \tag{20}$$

To summarize, in 1-D parallel mode, if each vertex in the reference views is processed sequentially and independently in the raster scan order from left to right, and if several adjacent reference vertices are warped to the same location in the warped view, these adjacent vertices' depth bins satisfy the following property.

Property 1: For the left warped view, the depth bin of each warped vertex is always larger than those of the previously warped ones. Whereas, for the right warped view, the depth bin of each warped vertex is always smaller than those of the previously warped ones.

According to *Property* 1, Eq. (17) can be further simplified into Eq. (21), as shown at the top of the next page with the same result. Besides, the sorting algorithm used to select the largest depth bin can also be avoided, since *Property* 1 can be used to predict the largest depth bin by recording the last (or first) warped vertex in the left (or right) warped view, when warping competition occurs. To solve Eq. (21), the depth bins and the locations of vertices j and z ($z \in S_3 \bigcup S_4$) are needed to be recorded in advance, which incurs substantial memory requirment, especially with larger S_3 or S_4. To solve this problem, another property of depth bin is studied as follows.

As discussed above, the relationship between depth bin and its associated disparity can be quantitatively calculated by Eq. (6). For $\tilde{V}_l(1)$ and $\tilde{V}_l(2)$, which locate at adjacent integer coordinates, we can easily get $\beta(b_{2p}) - \beta(b_{1p}) = 1$ based on Eq. (6). In general, in the left reference views, the relationships of these depth bins and their associated locations can be expressed as follows,

$$\beta(b_{mp}) - \beta(b_{jp}) = m - j. \tag{22}$$

Similarly, for the right reference view, we have

$$\beta(b_{mq}) - \beta(b_{jq}) = j - m. \tag{23}$$

These two equations can be regarded as generalizations of *Property* 1, which is summarized as follows.

Property 2: In 1-D parallel view synthesis, with full-pixel precision selection, suppose several adjacent reference vertices are warped to the same location in the warped view. Given the location and depth bin of the current reference vertex (such as j and $\beta(b_{ji})$ in Eq. (21))*, and the locations of its adjacent reference vertices* (such as vertices z ($z \in S_3 \bigcup S_4$) in Eq. (21))*, we can exactly derive these adjacent reference vertices' depth bins* (such as $\beta(b_{zi})$ in Eq. (21)) *using Eq. (22) and Eq. (23).*

Then, Eq. (21) can be further simplified into Eq. (24), as shown at the top of the next page. To solve Eq. (24), only three components need to be recorded in memory in advance, namely, the location of current reference vertex j and its depth bin, and the locations of its adjacent reference vertices z. The depth bins of all z are not needed, which can further save memory.

$$P_{win}(b_{ji}) = \begin{cases} P(\beta_j = \beta(b_{ji})) \times \prod_{z \in S_4} (1 - P(\beta_z = \beta(b_{zi}))), & \text{for left reference view.} \\ P(\beta_j = \beta(b_{ji})) \times \prod_{z \in S_3} (1 - P(\beta_z = \beta(b_{zi}))), & \text{for right reference view.} \end{cases} \tag{21}$$

$$P_{win}(b_{ji}) = \begin{cases} P(\beta_j = \beta(b_{ji})) \times \prod_{z \in S_4} (1 - P(\beta_z = \beta(b_{ji}) + z - j)), & \text{for left reference view.} \\ P(\beta_j = \beta(b_{ji})) \times \prod_{z \in S_3} (1 - P(\beta_z = \beta(b_{ji}) + j - z)), & \text{for right reference view.} \end{cases} \tag{24}$$

E. Depth Bin Probability Distribution Conversion

Different from the DVGM, the DBGM is formulated based on depth bin. Therefore, the initial per-pixel depth value probability distribution in the reference depth maps should be converted into depth bin probability distribution. In this subsection, we will introduce a simple way to achieve this conversion.

For a general case, we assume that the original depth value (ground truth) of a reference pixel is d, then a discrete noise distributed in $[-a, b]$ is introduced. The corrupted depth value d_e is in the range $[d - a, d + b]$, each with its associated probability. Then, d_e will be grouped into depth bins based on Eq. (7). After that, the probability of d_e within the same depth bin will be summed up to generate the their corresponded depth bin probability distribution, which will be stored in the memory and further sent to the DBGM to generate the VSDE.

There are two advantages of this conversion. Firstly, it can reduce the memory cost. In the worst case 256 probabilities are needed to record the depth value probability distribution of each 8-bit pixel, while much less numbers are needed to record the depth bin probability distribution by this conversion. Secondly, if the packet-loss-caused view synthesis distortion is required to be estimated, the RODE method can be directly integrated with this conversion.

For the special case in [7] that a discrete noise signal which is uniformly distributed in the range of $[-\sigma, \sigma]$ is introduced to simulate errors in reference depth images, our conversion above can still fully handle it. Besides, some important theoretical analyses on time and memory consumptions can be conducted in this case. Generally, the time and memory consumptions in the DVGM and DBGM are affected by two factors, namely σ and *baseline* L in Eq. (1) (the distance between two reference view).

1) Fixing σ, and Increasing Baseline Distance: for instance, as the baseline distance increases the running time of both the DVGM and DBVM will increase. However, the increase in the DVGM will be less than that of in the DBGM. Similarly, The memory cost in both the DVGM and DBVM will increase as well. The reason are analyzed as follows.

For the DVGM, the main computation costs come from three equations: i) calculating winning probability of a certain edge as discussed in Eq. (28), ii) summing up the winning probability of all the edges within the set Ω in Eq. (29), and iii) generating the probability of hole as discussed in Eq. (30). Since σ is fixed, the distribution of the depth values of each pixel is confirmed and the number of edges of each vertex is constant. Therefore, the computation cost in Eq. (28) is constant due to its depth-value-based calculation in the DVGM. However, as the *baseline* becomes wide, N will increase according to Eq. (2) and Eq. (5), which means each set Ω will contain less edges and more set Ω are required to keep the total number of edges from each vertex unchanged. More calculation will be spent on Eq. (29) and Eq. (30), while calculation cost on Eq. (28) remain constant. Therefore, when *baseline* increases, the computation cost will be partially increased. In contrast with the DVGM, the main stage of computation cost of the DBGM focuses on two equations: i) calculating winning probability of a bundle as formulated in Eq. (14), and ii) obtaining the probability of hole as mentioned in Eq. (18). Both of these two equations are operated based on depth bins. As *baseline* gets wider, N will be increased, which will increase the computation cost in both equations. Therefore, when *baseline* increases, the computation cost will be fully increased.

Since many factors will cause memory cost change, such as the algorithm complexity, arguments storage and so on. As *baseline* increases, on the one hand, the complexity of both of these two methods will be increased undoubtedly. On the other hand, it also increases the number of variables used in the equations above and finally leads to increased memory cost.

2) Fixing Baseline Distance, and Increasing σ: generally, when σ increases, the running time and memory cost in both the DVGM and DBGM will increase.

In detail, as σ increases, the number of possible depth values for each vertex in the DVGM must be increased with the same scale, but the growth of σ may not bring the same growth for depth bins, because each depth bin in the DBGM represents several depth values. Therefore, our proposed model shows better performance in terms of efficiency, when larger distortion exists in reference depth images. Undoubtedly, the increase of σ will increase the algorithm complexity and storage in both the DVGM and DBVM.

IV. Experimental Results

To validate the proposed DBGM in this paper, three evaluations are first presented in this section. The first one is efficiency evaluation, which shows that the proposed DBGM is 8 to 32 times faster than the DVGM depending on the conditions. The second one is the memory cost evaluation, which shows that the DBGM achieves 17% to 60% of memory saving compared to the DVGM depending on the conditions. The last one is the accuracy evaluation, which confirms that the proposed DBGM can always achieve the same accuracy

TABLE I
RUNNING TIME COMPARISON BETWEEN DVGM AND DBGM

Sequences	Resolution	*baseline*	N	σ	DVGM (s/f)			DBGM (s/f)			Speedup Factor
					Time1	Time2	Total	Time1	Time2	Total	
Kendo	1024*768	2(1,3)	25	3	0.037	2.211	2.248	0.038	0.235	0.273	**8.22**
				5	0.039	4.570	4.610	0.038	0.245	0.283	**16.30**
				7	0.039	5.776	5.815	0.038	0.277	0.315	**18.48**
				9	0.038	7.634	7.673	0.037	0.288	0.325	**23.64**
		4(3,5)	25	3	0.037	2.211	2.248	0.036	0.238	0.274	**8.20**
				5	0.039	4.492	4.531	0.038	0.246	0.283	**15.99**
				7	0.039	5.978	6.017	0.038	0.275	0.313	**19.23**
				9	0.039	7.616	7.656	0.038	0.287	0.326	**23.52**
		3(1,5)	49	3	0.038	2.356	2.394	0.037	0.276	0.313	**7.65**
				5	0.039	4.668	4.707	0.038	0.307	0.345	**13.65**
				7	0.037	6.079	6.117	0.038	0.327	0.365	**16.76**
				9	0.039	8.111	8.150	0.037	0.425	0.463	**17.62**
Balloons	1024*768	2(1,3)	25	3	0.036	2.261	2.298	0.037	0.228	0.265	**8.67**
				5	0.038	4.500	4.538	0.038	0.236	0.274	**16.58**
				7	0.038	5.774	5.812	0.036	0.284	0.320	**18.15**
				9	0.037	7.793	7.830	0.038	0.292	0.330	**23.73**
		4(3,5)	25	3	0.037	2.263	2.300	0.036	0.233	0.269	**8.55**
				5	0.039	4.454	4.493	0.038	0.242	0.280	**16.04**
				7	0.038	6.066	6.104	0.039	0.282	0.321	**19.03**
				9	0.038	7.761	7.799	0.038	0.289	0.327	**23.83**
		3(1,5)	49	3	0.038	2.448	2.486	0.037	0.281	0.318	**7.82**
				5	0.037	4.660	4.696	0.036	0.305	0.342	**13.75**
				7	0.038	6.144	6.182	0.039	0.328	0.366	**16.87**
				9	0.037	8.242	8.279	0.037	0.429	0.466	**17.75**
Undo Dancer	1920*1088	2(1,3)	21	3	0.103	7.144	7.247	0.097	0.596	0.693	**10.45**
				5	0.099	15.847	15.946	0.099	0.659	0.758	**21.03**
				7	0.097	19.858	19.955	0.098	0.662	0.760	**26.27**
				9	0.098	25.877	25.974	0.102	0.722	0.824	**31.54**
		3(1,5)	41	3	0.099	7.262	7.361	0.102	0.688	0.790	**9.32**
				5	0.101	16.673	16.774	0.102	0.743	0.845	**19.85**
				7	0.102	20.707	20.808	0.099	0.832	0.931	**22.36**
				9	0.102	27.257	27.359	0.099	0.928	1.027	**26.64**
Newspaper	1024*768	3(2,4)	36	3	0.037	2.268	2.305	0.036	0.229	0.266	**8.67**
				5	0.037	4.738	4.775	0.036	0.276	0.312	**15.29**
				7	0.039	6.331	6.371	0.038	0.290	0.328	**19.40**
				9	0.038	8.066	8.104	0.038	0.325	0.363	**22.33**
*Lovebird*1	1024*768	5(4,6)	30	3	0.036	2.223	2.260	0.037	0.244	0.281	**8.03**
				5	0.038	4.428	4.467	0.039	0.266	0.305	**14.64**
				7	0.039	5.878	5.917	0.038	0.297	0.334	**17.69**
				9	0.038	7.725	7.763	0.038	0.340	0.379	**20.49**
Café	1920*1080	3(2,4)	61	3	0.099	7.590	7.689	0.095	0.724	0.819	**9.39**
				5	0.099	15.218	15.317	0.100	0.850	0.951	**16.11**
				7	0.098	19.717	19.815	0.099	1.057	1.156	**17.14**
				9	0.101	25.754	25.855	0.102	1.195	1.297	**19.94**

as the DVGM. Besides, we also integrate the DVGM and DBGM with the RODE and further present the accuracy evaluation, which shows that the integrated DBGM+RODE and DVGM+RODE can achieve the same accuracy performance during estimating packet-loss-caused view synthesis distortion.

It should be noted that this paper aims to optimize the graphical model in the VSDE. Hence, most of the tests in this section focus on the comparison between DBGM and DVGM. To get fair comparison, we firstly use the test setup in [7] that introduces a discrete noise signal which is uniformly distributed in the range of $[-\sigma, \sigma]$ to simulate errors in the reference depth images. Since the performances of both the DVGM and DBGM are associated with the distortion of depth images and the baseline distance, different values of σ and different baseline distances are chosen to be tested. Finally, we also record the number of depth bins N used in our model in each sequence. The results are exhibited in Sec IV.A, B, and C.1. Then, a setup on the integrated DBGM+RODE and DVGM+RODE test is elaborated as follows. Both the texture and depth images are independently encoded using H.264/AVC, where three rows of macroblocks are collected in each slice. The packet loss rates are 2%, 5%, and 8%, respectively. GOP sizes of 30 and 60 are used. The results are exhibited in Sec IV.C.2. Besides, as both the DVGM and DBGM are used to generate the view synthesis distortion rather than synthesizing virtual views, all the test records in this section do not include view synthesis operations in both of these two models.

In this section, all the testing MVD sequences are chosen from the Common Test Conditions (CTC) of 3DV Core Experiments [28], such as *Kendo* and *Balloons* [29] (provided by Nagoya University), *Undo Dancer* [30] (provided by Nokia), *Newspaper* and *Café* [31] (provided by Gwangju Institute of Science and Technology), *Lovebird*1 [32] (provided by ETRI and MPEG Korea). The resolutions of these sequences are listed in TABLE I and TABLE II. The first 100 frames from sequence *Kendo*, *Balloons*, *Newspaper*, *Lovebird*1 are selected. As the DVGM requires a lot of memory, for fair

TABLE II
MEMORY COST COMPARISON BETWEEN DVGM AND DBGM

Sequences	Resolution	*baseline*	*N*	σ	DVGM (kB)			DBGM (kB)			Saving Factor
					Sharable	Private	Working	Sharable	Private	Working	
Kendo	1024*768	2(1,3)	25	3	4320	459752	464072	4336	315356	319692	**0.31**
				5	4332	607700	612032	4336	315416	319752	**0.48**
				7	4332	685636	689968	4332	315572	319904	**0.54**
				9	4336	797976	802312	4336	315608	319944	**0.60**
		4(3,5)	25	3	4332	464488	468820	4332	314500	318832	**0.32**
				5	4336	614724	619060	4332	315548	319880	**0.48**
				7	4336	690744	695080	4336	316064	320400	**0.54**
				9	4332	805940	810272	4332	316160	320492	**0.60**
		3(1,5)	49	3	4336	525928	530264	4336	326596	330932	**0.38**
				5	4336	677620	681956	4332	332464	336796	**0.51**
				7	4332	788208	792540	4320	335372	339692	**0.57**
				9	4336	908548	912884	4324	362660	366984	**0.60**
Balloons	1024*768	2(1,3)	25	3	4332	453124	457456	4336	315732	320068	**0.30**
				5	4332	605692	610024	4332	316072	320404	**0.47**
				7	4336	678728	683064	4336	317356	321692	**0.53**
				9	4332	797008	801340	4336	317876	322212	**0.60**
		4(3,5)	25	3	4332	459860	464192	4332	315604	319936	**0.31**
				5	4312	613356	617668	4336	315848	320184	**0.48**
				7	4340	689488	693828	4316	317112	321428	**0.54**
				9	4332	806020	810352	4336	317628	321964	**0.60**
		3(1,5)	49	3	4332	500420	504752	4332	327436	331768	**0.34**
				5	4336	652396	656732	4320	331744	336064	**0.49**
				7	4332	747168	751500	4332	334136	338468	**0.55**
				9	4336	865504	869840	4336	352356	356692	**0.59**
Undo Dancer	1920*1088	2(1,3)	21	3	4336	1022952	1027288	4336	844592	848928	**0.17**
				5	4336	1465220	1469556	4336	844596	848932	**0.42**
				7	4332	1510300	1514632	4336	844600	848936	**0.44**
				9	4324	1905664	1909988	4340	844600	848940	**0.56**
		3(1,5)	41	3	4336	1032556	1036892	4336	844596	848932	**0.18**
				5	4336	1470652	1474988	4332	844640	848972	**0.42**
				7	4340	1531644	1535984	4340	844872	849212	**0.45**
				9	4332	1918956	1923288	4332	846288	850620	**0.56**
Newspaper	1024*768	3(2,4)	36	3	4336	447100	451436	4336	315592	319928	**0.29**
				5	4336	582244	586580	4332	316452	320784	**0.45**
				7	4340	658096	662436	4336	317056	321392	**0.51**
				9	4336	776560	780896	4336	320040	324376	**0.58**
*Lovebird*1	1024*768	5(4,6)	30	3	4336	386232	390568	4336	315320	319656	**0.18**
				5	4332	544612	548944	4336	315324	319660	**0.42**
				7	4336	567692	572028	4332	315384	319716	**0.44**
				9	4332	710272	714604	4336	315484	319820	**0.55**
Café	1920*1080	3(2,4)	61	3	4336	1060956	1065292	4336	836860	841196	**0.21**
				5	4336	1479304	1483640	4336	843480	847816	**0.43**
				7	4336	1571160	1575496	4332	879392	883724	**0.44**
				9	4340	1953598	1957938	4344	889782	894126	**0.54**

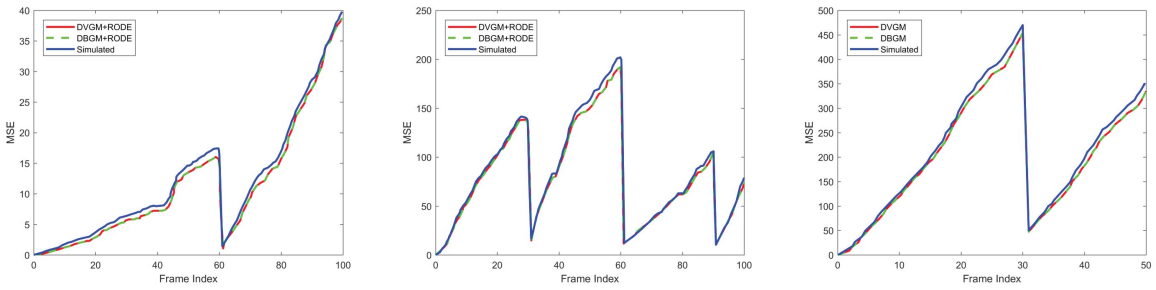
comparison, we have to choose the first 50 frames of *Undo Dancer* and *Café* in the test.

All the simulations in this paper are tested on a laptop, namely Dell inspiron 7559 Signature Edition with Intel(R) Core(TM) i7-6700HQ CPU, 16.00GB memory, and 64-bit Operating System.

A. Evaluation of Efficiency

In this subsection, we compare the running time between these two models without the RODE integration. The total running time contains two parts, namely the preparation-stage time cost and main-stage time cost, which are denoted as *Time1* and *Time2* in Table I, and their units are seconds per-frame. The *baseline* notation of $i(j,k)$ means that View j and View k are used to synthesize View i. It can be seen that our proposed DBGM is 8 to 32 times faster than the DVGM depending on σ and *baseline* configurations. The main reasons are as follows: i) To generate the per-pixel distribution in the synthesized view using the DVGM, each vertex in the reference depth images is required to be calculated through three equations (Eq. (28) to (30)), while only two equations (Eq. (18) and (24)) are needed in the DBGM. ii) Thanks to the properties of depth bins, the equations in the DBGM are much simpler than those in the DVGM, and less items are needed to be recorded in the memory. Besides, the sorting algorithm required in Eq. (28) in the DVGM is also avoided in Eq. (24) in the DBGM. iii) Compared with the redundant depth-value-oriented equations in the DVGM, the equations in the DBGM are implemented based on depth bin. In terms of the same σ, the DBGM needs considering less complexity.

Besides, based on the results in Table I, we can easily find: i) as we fix σ and increase *baseline*, the running time of both the DVGM and DBGM increases. However, the running time increase factor of the DVGM is less than that of the DBGM. ii) as we fix *baseline* and increase σ, the running time of both the DVGM and DBGM increases as well. All the experimental results in Table I exactly verifies our theoretical analyses on time consumption in Sec III.E.



(a) (b) (c)

Fig. 4. Distortion estimation performance of the DVGM+RODE and DBGM+RODE. (a) Balloons: GOP = 60, packet loss = 2%. (b) Kendo: GOP = 30, packet loss = 5%. (c) Dancer: GOP = 30, packet loss = 8%.

B. Evaluation of Memory Cost

In this part, we evaluate the memory cost in the DBGM and DVGM without the RODE integration. The *Resouce Monitor* [33] provided by Windows 10 system is used as our testing tool. *Resouce Monitor* can capture a data processing system's internal resource utilization, such as memory in real-time. With its assistance, we can directly obtain the memory consumption of a program of interest. We use the *Working Set* in Table II to represent the total memory of a program, which contains two components: 1) *Sharable*, which is the memory allocated to a program that can be shared to other programs, and 2) *Private*, which is the memory allocated to a program that can only be used by itself, and their units are all kilobyte (kB). As shown in TABLE II, our proposed DBGM can save 17-60% of memory compared to the DVGM depending on the σ and *baseline*. The main reasons are as follows: i) the complexity of the DVGM is higher than that of the DBGM, which leads to much more memory cost focusing on algorithm implement in the DVGM. ii) to implement the DVGM and DBGM, their corresponded variables are needed to be stored in the memory in advance. However, the variables of the DBGM is obviously less than that of the DVGM, since all the variables of the DBGM are based on depth bins, which is more sparse than that of the DVGM. Therefore, the memory cost for storage in the DVGM is larger than that in the DBGM. As a result, our proposed DBGM requires less memory cost compared with the DVGM.

Besides, from the experimental results in Table II, either increasing σ or *baseline* will cause the increase of memory cost in both the DVGM and DBGM, which verifies the our theoretical analyses on memory consumption in Sec III.E as well.

C. Evaluation of Accuracy

1) We First Evaluate the Accuracy of These Two Models Without the RODE Integration: The details results are shown in TABLE III, which are the average results of all the tested frames in different sequences. When the value of σ changes from 3 to 9, both methods have an MSE between 40 to 650, which covers the range of most practical scenarios. Besides, both these two models can achieve the same accuracy.

2) We Then Evaluate the Accuracy of These Two Models With the Rode Integration: Frame-by-frame results are exhibited in Fig. 4. We can obviously find that the integrated DBGM+RODE and DVGM+RODE still can achieve the same accuracy during estimating packet-loss-caused view synthesis

TABLE III
MSE COMPARISON BETWEEN THE DVGM AND DBGM

Sequences	Views	N	σ	MSE		Difference
				DVGM	**DBGM**	
Kendo	2(1,3)	25	3	71.375	71.375	0.000
			5	133.345	133.345	0.000
			7	295.422	295.422	0.000
			9	382.063	382.063	0.000
	4(3,5)	25	3	77.901	77.901	0.000
			5	138.862	138.862	0.000
			7	299.825	299.825	0.000
			9	387.310	387.310	0.000
	3(1,5)	49	3	274.479	274.479	0.000
			5	479.244	479.244	0.000
			7	542.914	542.914	0.000
			9	603.435	603.435	0.000
Balloons	2(1,3)	25	3	43.770	43.770	0.000
			5	57.793	57.793	0.000
			7	125.568	125.568	0.000
			9	159.660	159.660	0.000
	4(3,5)	25	3	51.963	51.963	0.000
			5	67.961	67.961	0.000
			7	128.424	128.424	0.000
			9	158.802	158.802	0.000
	3(1,5)	49	3	128.832	128.832	0.000
			5	219.285	219.285	0.000
			7	241.729	241.729	0.000
			9	273.290	273.290	0.000
Undo Dancer	2(1,3)	21	3	100.080	100.080	0.000
			5	149.696	149.696	0.000
			7	199.265	199.265	0.000
			9	262.346	262.346	0.000
	3(1,5)	41	3	200.683	200.683	0.000
			5	322.019	322.019	0.000
			7	407.312	407.312	0.000
			9	465.664	465.664	0.000
Newspaper	3(2,4)	36	3	138.716	138.716	0.000
			5	442.618	442.618	0.000
			7	557.020	557.020	0.000
			9	649.192	649.192	0.000
*Lovebird*1	5(4,6)	30	3	152.030	152.030	0.000
			5	257.702	257.702	0.000
			7	346.855	346.855	0.000
			9	409.659	409.659	0.000
Café	3(2,4)	61	3	259.315	259.315	0.000
			5	365.303	365.303	0.000
			7	422.483	422.483	0.000
			9	456.417	456.417	0.000

distortion. Besides, both methods can predict the simulated distortion trend very well.

From this subsection, we can conclude that both the DBGM and DVGM can achieve the same accuracy, no matter integrated with the RODE or not.

V. Conclusion

In this paper, we develop a novel depth-bin-based graphical model (DBGM), which models the complicated view synthesis process at depth-bin level so that it can be used for fast view synthesis distortion estimation. To this end, we first use depth bins to represent the redundant depth values. Then, the properties of depth bins are studied and used to optimize the winning probability function. Finally, we develop a conversion technique, which converts the depth value probability distribution in the reference depth maps into depth bin distribution. This conversion can also be used as a bridge to connect the RODE and DBGM in order to estimate the packet-loss-caused view synthesis distortion. Experimental results verify that the proposed DBGM is faster and consumes less memory than the DVGM, with exactly the same accuracy.

For the further work, we will try to apply the depth bin concept to simplify the process of RODE in depth image distortion estimation. Besides, the sub-pixel precision estimation will be taken into consideration in the DBGM to enhance its accuracy during the VSDE. Meanwhile, the region-based approach will also be considered for the DBGM to further reduce the complexity of the proposed scheme.

Appendix
Overview of the DVGM in [7]

In [7], to measure the packet-loss-caused view synthesis distortion, the mean squared error (MSE) is used as distortion metric. The expected per-pixel distortion in synthesized view can be written as

$$\begin{aligned} E\{D(i)\} &= E\{(T_s(i) - \tilde{T}_s(i))^2\} \\ &= T_s(i)^2 - 2T_s(i)E\{\tilde{T}_s(i)\} + E\{\tilde{T}_s(i)^2\}, \end{aligned} \tag{25}$$

where $T_s(i)$ is the correctly synthesized i-th texture pixel in the virtual view, $\tilde{T}_s(i)$ is the synthesized texture pixel at the receiver when the reference texture and depth images contain random errors. Therefore, once the first and second moments of $\tilde{T}_s(i)$ are achieved, synthesized view distortion can be estimated by this equation.

To represent the relationship between the synthesized view and the reference views, a weighted blending model is considered during formulation, where the weight is determinate and denoted as a. Then, the first and second moments of $\tilde{T}_s(i)$ are represented as

$$\begin{aligned} E\{\tilde{T}_s(i)\} = {} & (1-a)(1 - P_{\tilde{T}_{w_1}(i)}(\phi))E_{\setminus\phi}\{\tilde{T}_{w_0}(i)\} \\ & + a(1 - P_{\tilde{T}_{w_0}(i)}(\phi))E_{\setminus\phi}\{\tilde{T}_{w_1}(i)\} \\ & + P_{\tilde{T}_{w_1}(i)}(\phi)E_{\setminus\phi}\{\tilde{T}_{w_0}(i)\} \\ & + P_{\tilde{T}_{w_0}(i)}(\phi)E_{\setminus\phi}\{\tilde{T}_{w_1}(i)\} \\ & + P_{\tilde{T}_{w_0}(i)}(\phi)P_{\tilde{T}_{w_1}(i)}(\phi)E\{\tilde{T}_s^I(i)\}, \end{aligned} \tag{26}$$

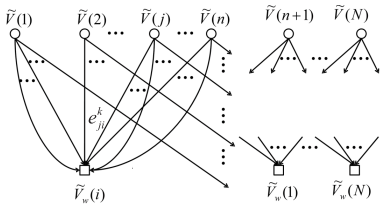


Fig. 5. Depth-value-based graphical model for the view synthesis algorithm.

$$\begin{aligned} E\{\tilde{T}_s(i)^2\} = {} & (1-a)^2(1 - P_{\tilde{T}_{w_1}(i)}(\phi))E_{\setminus\phi}\{\tilde{T}_{w_0}(i)^2\} \\ & + a^2(1 - P_{\tilde{T}_{w_0}(i)}(\phi))E_{\setminus\phi}\{\tilde{T}_{w_1}(i)^2\} \\ & + 2(1-a)aE_{\setminus\phi}\{\tilde{T}_{w_0}(i)\}E_{\setminus\phi}\{\tilde{T}_{w_1}(i)\} \\ & + P_{\tilde{T}_{w_1}(i)}(\phi)E_{\setminus\phi}\{\tilde{T}_{w_0}(i)^2\} \\ & + P_{\tilde{T}_{w_0}(i)}(\phi)E_{\setminus\phi}\{\tilde{T}_{w_1}(i)^2\} \\ & + P_{\tilde{T}_{w_0}(i)}(\phi)P_{\tilde{T}_{w_1}(i)}(\phi)E\{\tilde{T}_s^I(i)^2\}, \end{aligned} \tag{27}$$

where $\tilde{T}_{w_m}(i)$ $(m = 0, 1)$ is the i-th texture pixel in the warped view mapped from the left and right texture images respectively in the presence of errors. $E\{\tilde{T}_s^I(i)^k\}$ is the first and second moments of an inpainted hole, which is also determinate. $P_{\tilde{T}_{w_m}(i)}(\phi)$ represents the probability of $\tilde{T}_{w_m}(i)$ being a hole. $E_{\setminus\phi}\{\tilde{T}_{w_m}(i)^k\}$ denotes the partial k-th moment of $\tilde{T}_{w_m}(i)$ when it is not in a hole.

In order to calculate $E\{\tilde{T}_s(i)\}$ and $E\{\tilde{T}_s(i)^2\}$, both $P_{\tilde{T}_{w_m}(i)}(\phi)$ $(m = 0, 1)$ and $E_{\setminus\phi}\{\tilde{T}_{w_m}(i)^k\}$ are needed. In the 1-D view synthesis algorithm, since the distributions of the warped texture pixels are determined by those of the reference depth maps, once the distributions of the warped depth pixels are known, the $P_{\tilde{T}_{w_m}(i)}(\phi)$ $(m = 0, 1)$ and $E_{\setminus\phi}\{\tilde{T}_{w_m}(i)^k\}$ can be achieved.

In DVGM, for simplicity, assume that only the reference depth images are affected by random noises with a known distribution and the noises are independent from pixel to pixel. The relationship between a reference and the warped depth map is represented by a bipartite probabilistic graph as shown in Fig. 5, where vertices $\tilde{V}(j)$ represents the j-th vertex (or pixel) in the reference view, and $\tilde{V}_w(i)$ is the i-th vertex in the warped view. As random noise is added, each depth vertex may contains several different depth values with appropriate probabilities, which may correspond to several disparities and further lead to multiple possible warping targets in the warped view. This is represented by edges emitting from $\tilde{V}(j)$ to a number of $\tilde{V}_w(i)$ vertices. Each edge corresponds to one possible warping path with one depth value for $\tilde{V}(j)$ and the corresponding probability. For the k-th edge among all the edges between $\tilde{V}(j)$ and $\tilde{V}_w(i)$, the edge and its associated depth value are denoted as e_{ji}^k and $d(e_{ji}^k)$. Since vertices unconnected to $\tilde{V}_w(i)$ are irrelevant, only $\tilde{V}(1)$ to $\tilde{V}(n)$ are considered. Based on the depth-value-oriented warping competition rule, when e_{ji}^k is the final winner, $d(e_{ji}^k)$ should be the largest. Hence, all the edges emitted from previous vertices $\tilde{V}(z)$ to $\tilde{V}_w(i)$ with condition $d(e_{zi}^l) \geq d(e_{ji}^k)$ are denoted by a set $S_{z,1}$ $(z = 1, \ldots, j-1)$, which is required to be abandoned. All the edges emitted from subsequence vertices $\tilde{V}(z)$ to $\tilde{V}_w(i)$

with condition $d(e_{zi}^l) > d(e_{ji}^k)$ are denoted by another set $S_{z,2}$ $(z = j+1, \ldots, n)$, which should also be abandoned. Let $P(e_{ji}^k)$ denote the probability of edge e_{ji}^k. The winning probability of edge e_{ji}^k is defined as $P_{win}(e_{ji}^k)$, which could be formulated as

$$\begin{aligned} P_{win}(e_{ji}^k) = {} & P(e_{ji}^k) \\ & \times \prod_{z=1}^{j-1}(1 - \sum_{l \in S_{z,1}} P(e_{zi}^l)) \\ & \times \prod_{z=j+1}^{n}(1 - \sum_{l \in S_{z,2}} P(e_{zi}^l)), \end{aligned} \tag{28}$$

we define this function as the winning probability function.

Assume all these winning edges between $\tilde{V}(j)$ and $\tilde{V}_w(i)$ are collected into a set Ω. Then, the probability that $\tilde{V}(j)$ will be warped to $\tilde{V}_w(i)$ can be expressed as

$$P_{win}(e_{ji}) = \sum_{k \in \Omega} P_{win}(e_{ji}^k), \tag{29}$$

The probability of $\tilde{V}_w(i)$ taking no value from any edge, *i.e.*, $\tilde{V}_w(i)$ is in a hole, is denoted as $P_{\tilde{V}_w(i)}(\phi)$, which is expressed by

$$P_{\tilde{V}_w(i)}(\phi) = \prod_{j=1}^{n}(1 - P_{win}(e_{ji})). \tag{30}$$

Based on Eq. (29) and Eq. (30), the distributions of the synthesized depth pixels can be achieved, which is based on an assumption that only the reference depth images are affected by random noises with a known distribution. Therefore, the PMFs of both the depth and texture values of each reference pixel are needed, if DVGM is used to estimate the packet-loss-caused view synthesis distortion with DVGM.

To handle this, a RODE method is developed, which is a ROPE-like method. Instead of calculating the first and second moment of decoded pixel in ROPE, RODE is used to estimate the PMF of decoded pixel recursively. In RODE, for an intra-coded pixel, if its data are received, its PMF is simply a Kronecker delta function with a value of 1 at the location of the encoder reconstruction and 0 elsewhere. If the pixel is lost, the PMF from the previous frame will be propagated to the current frame due to the error concealment. For an inter-coded pixel, when the pixel is received, the PMF of current pixel is shifted from that of reference pixel by the residual value. If the pixel is lost, the PMF from the previous frame is propagated.

The DVGM assumes that only the reference depth has errors. When both the reference depth and texture have errors, the RODE method above and DVGM can be integrated.

ACKNOWLEDGMENT

The authors would like to thank Dr. Pan Gao from College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics for his valuable advice and kind help.

REFERENCES

[1] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," *Proc. SPIE*, vol. 5291, pp. 93–104, May 2004.

[2] A. Torralba and A. Oliva, "Depth estimation from image structure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1226–1238, Sep. 2002.

[3] S. Izadi *et al.*, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 559–568.

[4] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," *Proc. SPIE*, vol. 7443, pp. 74430T-1–74430T-11, Sep. 2009.

[5] *View Synthesis Reference Software (VSRS) Version 3.5*. Accessed: Mar. 2010. [Online]. Available: http://wg11.sc29.org/svn/repos/MPEG-4/test/tags/3D/viewsynthesis/VSRS-3-5

[6] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.

[7] D. Zhang and J. Liang, "View synthesis distortion estimation with a graphical model and recursive calculation of probability distribution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 5, pp. 827–840, May 2015.

[8] Y.-R. Horng, Y.-C. Tseng, and T.-S. Chang, "VLSI architecture for real-time HD1080p view synthesis engine," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1329–1340, Sep. 2011.

[9] P.-K. Tsung, P.-C. Lin, L.-F. Ding, S.-Y. Chien, and L.-G. Chen, "Single iteration view interpolation for multiview video applications," in *Proc. 3DTV Conf.*, May 2009, pp. 1–4.

[10] K. R. Vijayanagar, J. Kim, Y. Lee, and J.-B. Kim, "Efficient view synthesis for multi-view video plus depth," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 2197–2201.

[11] J. Jin, A. Wang, Y. Zhao, C. Lin, and B. Zeng, "Region-aware 3-D warping for DIBR," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 953–966, Jun. 2016.

[12] Y. Zhang, S. Kwong, L. Xu, S. Hu, G. Jiang, and C.-C. J. Kuo, "Regional bit allocation and rate distortion optimization for multiview depth video coding with view synthesis distortion model," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3497–3512, Sep. 2013.

[13] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Cairo, Egypt, Nov. 2009, pp. 721–724.

[14] H. Yuan, Y. Chang, J. Huo, F. Yang, and Z. Lu, "Model-based joint bit allocation between texture videos and depth maps for 3-D video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 485–497, Apr. 2011.

[15] H.-P. Deng, L. Yu, B. Feng, and Q. Liu, "Structural similarity-based synthesized view distortion estimation for depth map coding," *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1338–1344, Nov. 2012.

[16] 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.

[17] L. Fang, N.-M. Cheung, D. Tian, A. Vetro, H. Sun, and O. C. Au, "An analytical model for synthesis distortion estimation in 3D video," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 185–199, Jan. 2014.

[18] 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.

[19] Y. Zhou, C. Hou, W. Xiang, and F. Wu, "Channel distortion modeling for multi-view video transmission over packet-switched networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1679–1692, Nov. 2011.

[20] G. Cheung, J. Ishida, A. Kubota, and A. Ortega, "Transform domain sparsification of depth maps using iterative quadratic programming," in *Proc. 18th IEEE Int. Conf. Image Process.*, Brussels, Belgium, Sep. 2011, pp. 129–132.

[21] B. Macchiavello, C. Dorea, E. M. Hung, G. Cheung, and W.-T. Tan, "Reference frame selection for loss-resilient texture & depth map coding in multiview video conferencing," in *Proc. IEEE Conf. Image Process.*, Orlando, FL, USA, Sep./Oct. 2012, pp. 1653–1656.

[22] B. Macchiavello, C. Dorea, E. M. Hung, G. Cheung, and W.-T. Tan, "Reference frame selection for loss-resilient depth map coding in multiview video conferencing," *Proc. SPIE*, vol. 8305, pp. 83050C-1–83050C-11, Feb. 2012.

[23] B. Macchiavello, C. Dorea, E. M. Hung, G. Cheung, and W.-T. Tan, "Loss-resilient coding of texture and depth for free-viewpoint video conferencing," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 711–725, Apr. 2014.
[24] P. Gao and W. Xiang, "Rate-distortion optimized mode switching for error-resilient multi-view video plus depth based 3-D video coding," *IEEE Trans. Multimedia*, vol. 16, no. 7, pp. 1797–1808, Nov. 2014.
[25] P. Gao, Q. Peng, and W. Xiang, "Analysis of packet-loss-induced distortion in view synthesis prediction-based 3D video coding," in *Proc. IEEE Conf. Image Process.*, vol. 26, no. 6, pp. 2781–2796, Jun. 2017.
[26] Y. Zhao, C. Zhu, Z. Chen, and L. Yu, "Depth no-synthesis-error model for view synthesis in 3-D video," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2221–2228, Aug. 2011.
[27] J. Jin, Y. Zhao, C. Lin, and A. Wang, "An accurate and efficient nonlinear depth quantization scheme," in *Proc. Pacific Rim Conf. Multimedia*, vol. 9315, Sep. 2015, pp. 390–399.
[28] D. Rusanovskyy, K. Müller, and A. Vetro, *Common Test Conditions of 3DV Core Experiments*, document JCT3V-D1100, Incheon, South Korea, Apr. 2013.
[29] *3DV Sequences of Nagoya University*, Nagoya Univ., Nagoya, Japan, Mar. 2008. [Online]. Available: http://www.tanimoto.nuee.nagoya-u.ac.jp/mpeg/mpeg-ftv.html
[30] Nokia. *Multi-View Test Sequence Download Page*. Accessed: Feb. 19, 2013. [Online]. Available: ftp://mpeg3dv.research.nokia.com
[31] *3DV Sequences of ETRI and GIST*, Electron. Telecommun. Res. Inst. Gwangju Inst. Sci. Technol., Dæjeon, South Korea, Apr. 2008. [Online]. Available: ftp://203.253.130.48
[32] *Contribution for 3D Video Test Material of Outdoor Scene*, Standard ISO/IEC JTC1/SC29/WG11, Apr. 2008.
[33] D. A. Bishop *et al.*, "Real time internal resource monitor for data processing system," U.S. Patent 6 049 798, Apr. 11, 2000.

Jian Jin received the B.S. and M.E. degree in economics management and electronic engineering from Taiyuan University of Science and Technology, Taiyuan, Shanxi, China, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree with Beijing Jiaotong University, Beijing, China. He is currently a Visiting Ph.D. Student with Simon Fraser University, Burnaby, BC, Canada.

His research interests include image/video compression, 3D video synthesis, machine learning, and deep learning. He has served as a Reviewer for IEEE TRANSACTIONS ON MULTIMEDIA and *EURASIP Journal on Image and Video Processing*.

Jie Liang (S'99–M'04–SM'11) received the B.E. and M.E. degrees from Xi'an Jiaotong University, China, in 1992 and 1995, respectively, the M.E. degree from National University of Singapore (NUS) in 1998, and the Ph.D. degree from The Johns Hopkins University, Baltimore, MD, USA, in 2003. From 1997 to 1999, he was with Hewlett-Packard Singapore and the Center for Wireless Communications, NUS. From 2003 to 2004, he was with the Video Codec Group of Microsoft Digital Media Division. Since 2004, he has been with the School of Engineering Science, Simon Fraser University, Canada, where he is currently a Professor and the Associate Director. In 2012, he visited University of Erlangen-Nuremberg, Germany, as an Alexander von Humboldt Research Fellow.

His research interests include image and video coding, multimedia communications, sparse signal processing, computer vision, and machine learning. He received the 2014 IEEE TCSVT Best Associate Editor Award, the 2014 SFU Dean of Graduate Studies Award for Excellence in Leadership, and the 2015 Canada NSERC Discovery Accelerator Supplements Award. He has served as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE SIGNAL PROCESSING LETTERS, *Signal Processing: Image Communication*, and *EURASIP Journal on Image and Video Processing*.

Yao Zhao (M'06–SM'12) received the B.S. degree from the Radio Engineering Department, Fuzhou University, Fuzhou, China, in 1989, the M.E. degree from the Radio Engineering Department, Southeast University, Nanjing, China, in 1992, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1996,where he became an Associate Professor and a Professor in 1998 and 2001, respectively. From 2001 to 2002, he was a Senior Research Fellow with the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. In 2015, he visited the Swiss Federal Institute of Technology, Lausanne, Switzerland. From 2017 to 2018, he visited University of Southern California. He is currently the Director of the Institute of Information Science, BJTU. His current research interests include image/video coding, digital watermarking and forensics, video analysis and understanding, and artificial intelligence.

Dr. Zhao is a Fellow of the IET. He serves on the Editorial Boards of several international journals, including as an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, a Senior Associate Editor for IEEE SIGNAL PROCESSING LETTERS, and an Area Editor for *Signal Processing: Image Communication*. He was named a Distinguished Young Scholar by the National Science Foundation of China in 2010 and was elected as a Chang Jiang Scholar of Ministry of Education of China in 2013.

Chunyu Lin was born in Suizhong, Liaoning, China. He received the Ph.D. degree from Beijing Jiaotong University, Beijing, China, in 2011. From 2009 to 2010, he was a Visiting Researcher with the ICT Group, Delft University of Technology, Delft, The Netherlands. From 2011 to 2012, he was a Post-Doctoral Researcher with the Multimedia Laboratory, Gent University, Gent, Belgium. He is currently an Associate Professor with Beijing Jiaotong University. His research interests include image/video compression and robust transmission, 2D-to-3D conversion, 3D video coding, and virtual reality video processing.

Chao Yao received the B.S. degree in computer science from Beijing Jiaotong University (BJTU), Beijing, China, in 2009 and the Ph.D. degree from the Institute of Information Science, BJTU, in 2016. From 2014 to 2015, he was a Visiting Ph.D. Student with Ecole Polytechnique Federale de Lausanne, Switzerland. Since 2016, he held a post-doctoral position with Beijing University of Posts and Telecommunications, Beijing. His current research interests include image and video processing and computer vision.

Anhong Wang was born in Yuncheng, Shanxi, China, in 1972. She received the B.E. and M.E. degrees in electronic information engineering from Taiyuan University of Science and Technology (TYUST) in 1994 and 2002, respectively, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University, in 2009. She became an Associate Professor with TYUST in 2005, where she became a Full Professor in 2009. She is currently the Director with the Institute of Digital Media and Communication, TYUST. Her research interest includes image/video coding and transmission, compressed sensing, and secret image sharing. She has authored over 100 papers in international journals and conferences. She is also leading several research projects, including two National Science Foundations of China.