# Correspondences

## Low Complexity Adaptive View Synthesis Optimization in HEVC Based 3D Video Coding

Siwei Ma, Shiqi Wang, and Wen Gao, *Fellow, IEEE*

***Abstract*—In this correspondence, we explore a low-complexity adaptive view synthesis optimization (VSO) scheme in the upcoming high-efficiency video coding (HEVC)-based 3-D video coding standard. We first devise a novel zero-synthesized view difference (ZSVD) model which jointly accounts for the distortion of the synthesized view induced by the compound impact of depth-disparity mapping, texture adaptation, and occlusion in the view synthesis process. This model can efficiently estimate the maximum allowable depth distortion in synthesizing a virtual view without introducing any geometry distortion. Then, an adaptive ZSVD-aware VSO scheme is proposed by incorporating the ZSVD model into the rate-distortion optimization process, which is developed by pruning the conventional view synthesis algorithm. Extensive experimental results confirm that the proposed model is capable of accurately predicting the zero distortion of the synthesized view and exhibit that the proposed ZSVD-aware VSO scheme can remarkably reduce the coding computational complexity with negligible performance loss.**

***Index Terms*—3D video coding, computational complexity, view synthesis optimization, zero synthesized view difference.**

### I. Introduction

Recently, we have witnessed a striking rise in the popularity of the three-dimension video (3DV) system, as it can provide viewers with the immersive perception of the real world scene. To address the demand of broad 3DV applications, the Moving Picture Experts Group (MPEG) of ISO/IEC launched the 3D video coding project in March 2011 [1], covering solutions based both on H.264/AVC and HEVC. The HEVC based 3D video coding standard employs the multiview video plus depth format [2], named "video-plus-depth" in the paper. It can provide large view-angle scene with fewer amounts of data by using the depth image-based rendering (DIBR) technique, which refers to the generation of new views of different perspectives in the same scene with corresponding videos and depth maps [4], [8].

Depth map is a grey scale image/video which plays a key role in obtaining high quality virtual synthesized view. Since it has different statistics and characteristics compared with the color video, many advanced coding tools are incorporated into the 3DV codec to improve the performance of depth compression [2], [3], [6], [7]. One key principle in depth compression lies in that it is not the depth map but the synthesized view to be displayed in the 3DV system. Therefore, different from the traditional 2D or multi-view video coding, the quality measurement of the "video-plus-depth" 3D video codec is dependent on the synthesized view rendered with the compressed depth. Inspired by this, on top of these advanced depth compression tools, view synthesis guided rate distortion optimization (RDO) coding techniques are employed in the 3DV codec. As a result, the encoding tools are optimized in accordance with the quality of the synthesized view instead of the depth map.

Generally, distorted depth may cause some pixels being warped to different positions in the synthesized view, which leads to geometry changes [10]. Moreover, the occlusion order may even change due to the depth distortion. Therefore, though it is generally true that better depth leads to better quality of synthesized video [11], the quality of depth map still cannot be linearly mapped to the quality of the synthesized view. How to efficiently model the distortion of the synthesized view from the depth distortion is then a key issue in the view synthesis guided RDO.

Recently, many synthesized view distortion calculation algorithms are proposed for 3DV encoder optimization. These algorithms can be mainly classified into two categories: rendering and non-rendering based. In the first category, the view rendering process is actually performed in the encoder for distortion calculation [13]–[15]. In [13], a view synthesis optimization (VSO) scheme for the exact synthesized view distortion calculation was proposed by employing a measure called synthesis view distortion change (SVDC). To compute the rate distortion (RD) cost, the view rendering is actually performed iteratively in the encoding process. Though this scheme achieves high compression efficiency, it imposes a heavy computation burden to the encoder. In [14], [15], simplified RDO schemes were proposed based on the estimated distortion models. In these distortion models, the synthesized view generated by the original texture and depth need to be computed before coding the current depth map. However, they are based on the integer position grid in the virtual view, which may not be accurate enough in general. The non-rendering RDO originates from the observation that the sensitivity of depth distortion is highly correlated with the properties of texture image [16]–[20]. Therefore, some simplified view synthesis distortion models based on the texture information and depth distortion are proposed. These techniques estimate the distortion of the synthesized view without actually performing view rendering, which results in little computational resource consumption. However, the accuracy may not be as high as the rendering based schemes.

In general, after depth compression, a significant number of distorted depth pixels will not cause distortion of the synthesized view. For example, if two distorted depth values are mapped to one same disparity value, there will not be any difference in the warping process. Inspired by this observation, a depth no-synthesis-error model is proposed based on the relationship between depth and disparity [12]. However, this model didn't consider the properties of input texture sequences and depth map, which may not provide a tight solution.

In this work, we propose a novel zero-synthesized view difference (ZSVD) model and employ it in the VSO scheme for depth compression. The main contributions of this work are:

1) We develop the ZSVD model which indicates whether there is difference or change of the synthesized view after the view rendering with the distorted depth. Specifically, in this model, three kinds of

Manuscript received February 07, 2013; revised May 30, 2013; accepted July 31, 2013. Date of publication October 07, 2013; date of current version December 12, 2013. This work was supported in part by the National High-tech R&D Program of China (863 Program) under Grant SS2012AA010805 and the National Science Foundation under Grant 61210005 and Grant 61121002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shahram Shirani.

The authors are with the Institute of Digital Media, School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: swma@pku.edu.cn; sqwang@pku.edu.cn; wgao@pku.edu.cn).

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

Digital Object Identifier 10.1109/TMM.2013.2284751

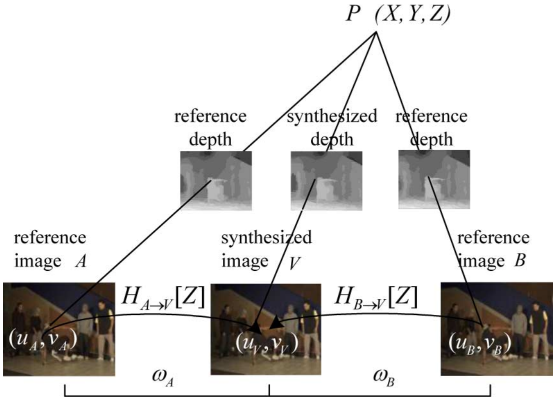

Fig. 1. Depth-image-based view synthesis.

effect are considered to derive the conditions of ZSVD in depth compression, which are depth-disparity mapping, texture adaptation and occlusion.

2) The ZSVD model is applied in the view synthesis optimization process by pruning the traditional view rendering. With the proposed ZSVD aware VSO scheme, the distortion of the synthesized view can be accurately computed with low computational cost.

Due to its superior performance, the proposed adaptive VSO scheme has been proposed to 3DV standard and partially adopted [27].

## II. View Synthesis Optimization

In this section, we will give a brief review of the SVDC based VSO [2], [13]. The motivation behind this technique is that lossy coding of depth image causes the geometry distortion in the synthesized intermediate views; hence, the distortion measure used for depth RDO coding should be in line with the distortion of the synthesized view.

### A. View Rendering

Following the HEVC based 3D video coding standard specification [2]; in this work we assume that the virtual view is positioned at the horizontal axis of the original view, which is commonly known as the 1-D parallel arrangement. Under this assumption, the vertical disparity is zero. As shown in Fig. 1, the view rendering supports the synthesis of a virtual view $V$ from a left view $A$ and right view $B$ with corresponding depth maps. Assume $H_{A\rightarrow V}[Z]$ and $H_{B\rightarrow V}[Z]$ are homography matrix at depth $Z$ from view $A$ and $B$ to the virtual view $V$, respectively. Therefore, the relation among the pixels $(u_V, v_V)$, $(u_A, v_A)$ and $(u_{B,}v_{\rm B})$ can be described as

$$(u_V, v_V) = \mathbf{H}_{A\rightarrow V}[z](u_A, v_A)^T = \mathbf{H}_{B\rightarrow V}[z](u_B, v_B)^T. \quad (1)$$

As the value of depth corresponds to fractional-pixel precision of disparity, in the view rendering, the upsampling process is firstly applied on the input texture to generate an upsampled version of the input view. Subsequently, two special cases in view rendering are detected: disocclusion and occlusion. A critical problem in view rendering is that, the image information which is occluded in the original view may become visible, and the missing information should be estimated. This is typically referred as disocclusion. The occlusion effect refers to that some of the pixels in the background of the input view are not visible after view rendering, as the foreground and background overlap in the synthesized view.

In the following, we briefly review the view rendering process in VSO by introducing two neighboring pixels $p$ and $c$ in original view as an example. We assume that $p$ is on the left of $c$. The horizontal coordinates of $p$ and $c$ are denoted to be $x_p$ and $x_c$, and the warped positions of $p$ and $c$ are denoted to be $\tilde{x}_p$ and $\tilde{x}_c$, respectively. Assume that the rendering is from right view to left. That is to say, all the rendering steps for the left pixel $p$ are finished before rendering $c$.

*1) Occlusion Detection:* Occlusion is detected by checking whether the internal boundary is reversed, for example, $\tilde{x}_p > \tilde{x}_c$. It indicates that the foreground pixel $p$ occludes the background pixel $c$. Moreover, the foreground edge is stored to determine whether next pixels are still occluded. If occlusion is detected, and provided that the pixel is still rendered into the occluded region from the distorted depth data, the view synthesis process won't be affected, no matter how great the depth pixel is distorted.

*2) Disocclusion Detection:* Disocclusion is detected by checking the distance between $\tilde{x}_p$ and $\tilde{x}_c$, and if the distance is greater than two times of full-pixel sample intervals, the current interval is set to be disoccluded and thus hole filling is carried out. In this case, $p$ corresponds to the foreground pixel and $c$ corresponds to the background. Generally, the samples are set to be the background value of pixel $c$. A special case is that if the leftmost full sample position within the interval is closer to the left interval border, it is set equal to the value of the left interval boundary $p$, which indicates that it belongs to the foreground.

*3) Interpolation:* The interpolation in view rendering is performed by mapping the synthesized integer pixel between $\tilde{x}_p$ and $\tilde{x}_c$ to the up-sampling version of the captured texture. In the current test model, DCT based interpolation filter [26] is employed to generate the quarter-accuracy version of the input texture. Assume the synthesized integer pixel between $\tilde{x}_p$ and $\tilde{x}_c$ to be $x_{FP}$, and the position in the up-sampled texture corresponding to $x_{FP}$ is derived as

$$\hat{x} = 4 \cdot \left( \frac{x_{FP} - \tilde{x}_p}{\tilde{x}_c - \tilde{x}_p} + x_p \right). \quad (2)$$

Finally, after these three operations, the final virtual view is generated by blending the left and right views together, and more details of this part can be found in [2].

### B. RDO Process for Depth Compression

To obtain an accurate distortion measure for RDO, the actual view synthesis is carried out at the encoder to obtain the overall distortion of the synthesized view. The distortion model of VSO is defined to be [13],

$$\begin{aligned} \Delta D = & \sum_{(u_V, v_V)\in I} \left[\tilde{I}_V(u_V, v_V) - I_V(u_V, v_V)\right]^2 \\ & - \sum_{(u_V, v_V)\in I} \left[I'_V(u_V, v_V) - I_V(u_V, v_V)\right]^2 \end{aligned} \quad (3)$$

where $I_V(u_V, v_V)$ denotes the virtual view rendered by the original texture and original depth. Both $\tilde{I}_V(u_V, v_V)$ and $I'_V(u_V, v_V)$ are synthesized by the reconstructed texture. $I'_V(u_V, v_V)$ denotes the virtual view rendered by the reconstructed depth for already encoded blocks, original depth for the current depth block and original depth for others. In contrast, $\tilde{I}_V(u_V, v_V)$ differs from $I'_V(u_V, v_V)$ in that $\tilde{I}_V(u_V, v_V)$ is rendered by the distorted depth for the current test block.

The RDO process with VSO in depth coding aims at optimizing its overall fidelity on synthesized view: minimizing the distortion $\Delta D$, with the number of used bits $R$ subjected to a bit constraint $R_c$; and this can be expressed as [9]

$$\min\{\Delta D\} \; subject \; to \; R \leq R_c \quad (4)$$

which can be converted to an unconstrained problem by:

$$\min\{J\} \; where \; J = \Delta D + \lambda \cdot R \quad (5)$$

where $J$ is called the Rate Distortion (RD) cost and $\lambda$ is known as the Lagrange multiplier. In the RDO process, the coding mode with the minimal RD cost is selected as the final one.

## III. ZSVD Aware VSO for Depth Compression

For virtual view synthesis, it has been generally recognized that depth distortions may not always cause distortion of the synthesized view. This implies that not all of the distortion computations in VSO are necessary as the explicitly changed depth values may not lead to signal degradation in the synthesized view. Assume the original and distorted depth for the pixel $p$ in the captured view $A$ to be $z_p$ and $z'_p$. The synthesized views rendered with $z_p$ and $z'_p$ are denoted to be $V(z_p)$ and $V(z'_p)$, respectively. Then the sufficient condition of ZSVD can be formulated as

$$C_{ZSVD}(p) = \{z'_p | V(z_p) = V(z'_p)\}. \tag{6}$$

In this section, three ZSVD conditions are introduced. Specifically, the relationship between depth and disparity is firstly established. Then the texture adaptation and occlusion information are employed to derive the ZSVD condition in terms of disparity. Based on them, the ZSVD aware VSO approach is proposed to accelerate depth encoding.

### A. ZSVD Conditions

*1) Depth-Disparity Mapping:* The relationship between depth $z$ and disparity $d$ can be expressed as

$$d = \pm f \cdot \frac{b}{z} \tag{7}$$

where $f$ indicates the focal length of the camera and $b$ represents the baseline between two views. For simplicity of the description, signed disparity vector is used, which indicates that when the current view is on the left/right side of the synthesized view, the disparity vector is negative/positive.

Generally, the disparity is rounded to $1/N$ sub-pixel position as

$$DP_N(d) = sign(d)\frac{\lceil (|d| - o) \cdot N \rceil}{N} \tag{8}$$

where $o$ represents the offset error and determines the decision level of the rounding process.

Assume $d_p$ and $d'_p$ represent the original and distorted disparity of the pixel $p$, respectively. Then the disparity distortion at the pixel $p$ can be defined as

$$D_N(d_p, d'_p) = \left| DP_N(d_p) - DP_N(d'_p) \right|. \tag{9}$$

In the current implementation of HEVC based 3D video coding, as the interpolation is performed with quarter-accuracy, $N$ is set to be 4.

In view synthesis, the warping process is directly determined by the sub-pixel disparity, which implies that the depth distortion is reflected in terms of the disparity distortion. Therefore, as long as the original and distorted depth map to the same disparity value, the warping process won't be affected, which corresponds to the following condition [12]:

$$D_N(d_p, d'_p) = 0. \tag{10}$$

Based on this, we detail the ZSVD condition equivalently with (6) in terms of the lower bound $B_l$ and upper bound $B_r$, which determine the range of the distorted disparity with the constraint of not changing the synthesized view. This can be formulated as

$$C_{ZSVD}(p) = \left\{ d'_p | B_l(p) \le DP_N(d'_p) \le B_r(p) \right\}. \tag{11}$$

The lower and upper bounds $B_l$ and $B_r$ are derived by texture adaptation and occlusion information in the following.

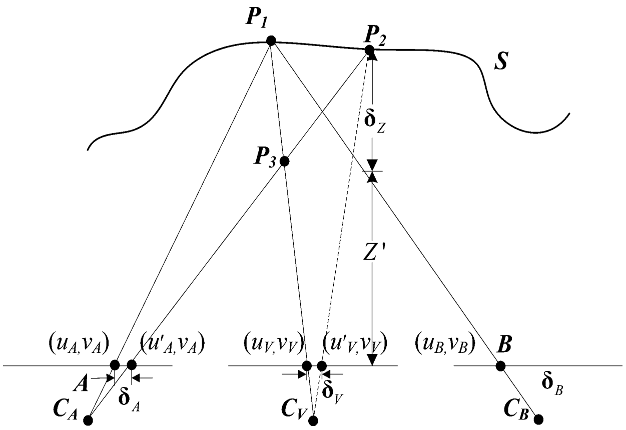

Fig. 2. The pixel mapping with distorted disparity (parallel camera setup).

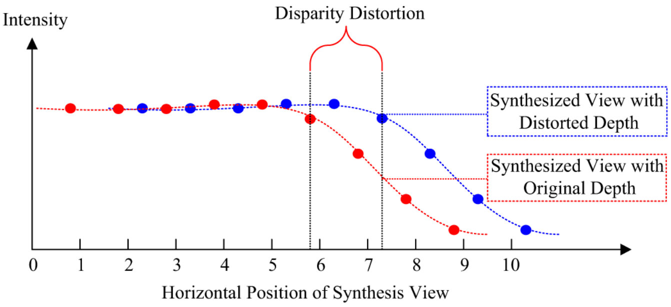

Fig. 3. Synthesized views with the original and distorted depth.

*2) Texture Adaptation:* In depth-disparity mapping analysis, we deal with the case when there is no disparity distortion for view warping. However, to meet the bandwidth, the bit rate of depth is usually less than twenty percent of the texture bit rate [21], and thus the quantization in compression usually introduces distortion of the disparity. Here we assume that the current pixel is neither occluded nor occluding other pixels (foreground edge). The pixel mapping with distorted disparity is demonstrated in Fig. 2. Points $P_1$ and $P_2$ are the actual points in the surface $S$. In the ideal warping, the pixel at position $(u_V, v_V)$ is interpolated by the pixel $(u_A, v_A)$ in view $A$ and the pixel $(u_B, v_B)$ in $B$, which can be formulated as

$$I_V(u_V, v_V) = \omega_A \cdot I_A(u_A, v_A) + \omega_B \cdot I_B(u_B, v_B). \tag{12}$$

where $\omega_A$ and $\omega_B$ are the blending weights. However, in the actual warping, point $P_2$ in the 3D world would change into $P_3$ due to depth distortion, and $P_3$ will be projected to the position $(u_V, v_V)$ in the virtual view. The interpolated pixel with the distorted depth is expressed as

$$\widehat{I}_V(u_V, v_V) = \omega_A \cdot I_A(u_A + \delta_A, v_A) + \omega_B \cdot I_B(u_B, v_B). \tag{13}$$

This leads to the distortion in the virtual view as follows

$$\widehat{I}_V(u_V, v_V) - I_V(u_V, v_V) = \omega_A \cdot (I_A(u_A + \delta_A, v_A) - I_A(u_A, v_A)). \tag{14}$$

According to the two-value Taylor expansion, we further obtain

$$I_A(u_A + \delta_A, v_A) = I_A(u_A, v_A) + \delta_A \cdot \nabla I_A(\xi_A). \tag{15}$$

Finally, we have

$$\widehat{I}_V(u_V, v_V) - I_V(u_V, v_V) = \omega_A \cdot \delta_A \cdot \nabla I_A(\xi_A) \tag{16}$$

which indicates that the same disparity distortions in complex and homogeneous region will cause different synthesis distortion. Fig. 3

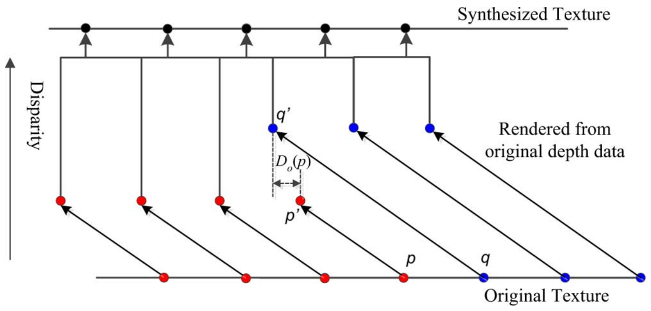

Fig. 4. Illustration of the occluded pixels in view rendering.

presents the synthesized view with the original and distorted depth, respectively. We can also observe that the smooth texture can mask the geometry distortion in view synthesis. Specifically, when the gradient approaches zero, the synthesized view distortion would also tend to be zero.

Assume that the synthesis distortion is zero, from (16) we can have:

$$I_A(u_A, v_A) = I_A(u_A + \delta_A, v_A) \tag{17}$$

which indicates that the number of identical texture pixels left and right to $I_A(u_A, v_A)$ directly determines the maximum allowable distortion of disparity. Based on this observation, we define $G_l(p)$ and $G_r(p)$ to represent the horizontal smoothness of the pixel $p$ in the interpolated high resolution picture. More specifically, $G_l(p)$ is the leftmost position which has identical pixel value with all the pixels in the interval $[G_l(p), p]$, which is called smoothness interval. $G_l(p)$ can be formulated as

$$G_l(p) = \min\{p_l\} \ subject \ to \forall m \in [p_l, p], I_H(m) = I_H(p) \tag{18}$$

where $I_H$ represents the current row in the interpolated high resolution frame. $G_r(p)$ is defined in a same way but it is on the right side of the pixel $p$, which can be formulated as

$$G_r(p) = \max\{p_r\} \ subject \ to \forall m \in [p, p_r], I_H(m) = I_H(p). \tag{19}$$

As illustrated in Fig. 2, the maximum allowable disparity distortion can be approximated by the width of the smoothness interval, which motivated us to examine the smoothness interval to determine the maximum allowable disparity distortion. Therefore, when a pixel is not occluded, only texture adaptation need to be considered to derive the ZSVD condition, which implies that in this case, the smooth interval $[G_l(p), G_r(p)]$ can be employed to determine $B_l$ and $B_r$ in (11) as

$$\begin{aligned} B_l(p) &= DP_N(d_p) + G_l(p) - p \\ B_r(p) &= DP_N(d_p) + G_r(p) - p. \end{aligned} \tag{20}$$

*3) Occlusion Information:* In the virtual view rendering, a pixel will be occluded if its neighbouring pixel has much larger disparity value, as shown in Fig. 4 (pixel $p$). Generally, when a pixel is occluded in the synthesized view, distortion of its depth data does not affect the rendering process as long as it is still rendered into the occluded region. That is to say, the depth pixels which lie in the occlusion region can tolerate a certain degree of distortion. Inspired by this observation, we define $D_o(p)$ as

$$D_o(p) = |(x_q + DP_N(d_q)) - (x_p + DP_N(d_p))| \tag{21}$$

where $q$ represents the foreground edge pixel that occludes the current pixel $p$. As shown in Fig. 4, the two pixels $p$ and $q$ in the current view are rendered to $p'$ and $q'$ respectively in the synthesized view. $D_o(p)$ is the distance between $q'$ and $p'$, which represents the maximum allowable disparity distortion for the pixel $p$. As such, with $D_o(p)$ we can decide whether one pixel is occluded in the synthesized view, and for this kind of pixels we can skip VSO to reduce the computation complexity.

In summary, by considering $G_l$, $G_r$ and $D_o$ jointly, we can define the lower bound $B_l$ and upper bound $B_r$ in (11). If the current pixel is occluded by other pixels, both the occlusion information and smoothness of the texture are considered. Specifically, if the current view is on the left, the lower bound $B_l$ is determined by the occlusion information as it determines where the minimum disparity goes, and $B_l$ and $B_r$ are calculated as follows

$$\begin{aligned} B_l(p) &= DP_N(d_p) - D_o(p) \\ B_r(p) &= DP_N(d_p) + G_r(p) - p. \end{aligned} \tag{22}$$

In contrast, the situation is on the contrary when the current view is on the right, since in this case the occlusion information determines the maximum allowable distorted disparity. Therefore, the $B_l$ and $B_r$ for left view is derived as

$$\begin{aligned} B_l(p) &= DP_N(d_p) + G_l(p) - p \\ B_r(p) &= DP_N(d_p) + D_o(p). \end{aligned} \tag{23}$$

### *B. ZSVD Aware VSO for Depth Compression*

In this subsection, we propose a ZSVD aware VSO approach to achieve fast depth compression. The whole process of the ZSVD condition is summarized as follows. We classify the ZSVD condition into three cases: occluding other pixels, occluded by other pixels and neither occluding nor occluded by other pixels. Specifically, if the current pixel occludes other pixels, $B_l$ and $B_r$ are both set to original disparity. If the current pixel is occluded, the occlusion condition is directly employed. Otherwise, smooth interval $[G_l(p), G_r(p)]$ is employed to determine $B_l$ and $B_r$.

The ZSVD model is incorporated into SVDC calculation and view rendering of the VSO process. The purpose of SVDC calculation and view rendering is to update the current synthesized view with the changed depth block by real view rendering. In this work, we aim to prune the view rendering process with the help of the ZSVD prediction. Based on the assumption of the 1-D parallel arrangement, the view rendering for each row is independent. In contrast, within each row, the distorted depth pixel may affect the synthesis result of the undistorted depth pixel. Therefore, the basic unit for the ZSVD aware VSO is defined to be one row in the current depth block.

The whole process is described as follows. In the encoder, for each depth block of size $M \times N$ to be rendered, we check each row of the block by the condition $C_{ZSVD}$. Assume the pixel set of the $j$-th row in current block is $L_j$. If all pixels in the current row satisfy the condition $C_{ZSVD}$, view rendering of this row can be completely skipped.

## IV. Validations

To validate the accuracy and efficiency of the proposed ZSVD aware VSO, we integrate our scheme into the 3D-HTM reference software 4.0 [29]. The common test configurations are employed in our experiment, as shown in Table I[5]. Eight 3DV sequences with resolutions 1024×768 and 1920×1088 are tested in the experiments. It is noted that since the proposed scheme is partially adopted [27], the anchor is generated by disabling the proposed scheme in HTM software [28].

### *A. Performance Comparison*

Firstly, we evaluate the performance of the scheme by the RD performance and the coding complexity. Specifically, the RD per-

TABLE I
TEST CONFIGURATIONS IN THE EXPERIMENTS.

| | |
|---|---|
| Inter-view coding structure | center-left-right<br>P-I-P inter-view prediction |
| Temporal prediction structure | GOP size 8, intra period 24 |
| Texture QP | 25, 30, 35, 40 |
| Depth QP | 34, 39, 42, 45 |
| Virtual views in VSO | 0.25, 0.5 and 0.75 |
| View synthesis algorithm | VSRS 1d fast mode |
| Configuration | Random Access |

TABLE II
R-D PERFORMANCE OF THE PROPOSED SCHEME.

| Sequence | 3 View | | | 2 View | | |
|---|---|---|---|---|---|---|
| | $\Delta R_v$ | $\Delta R_s$ | $\Delta R_{v,s}$ | $\Delta R_v$ | $\Delta R_s$ | $\Delta R_{v,s}$ |
| Balloons | 0.00% | -0.03% | 0.01% | 0.00% | -0.17% | -0.08% |
| Kendo | 0.00% | 0.06% | 0.07% | 0.00% | 0.06% | 0.07% |
| Newspaper | 0.00% | 0.12% | 0.13% | 0.00% | 0.11% | 0.13% |
| Lovebird1 | 0.00% | -0.07% | -0.03% | 0.00% | -0.04% | 0.00% |
| GTFly | 0.00% | 0.01% | 0.03% | 0.00% | 0.12% | 0.04% |
| PoznanHall2 | 0.00% | 0.47% | 0.45% | 0.00% | 0.39% | 0.35% |
| PoznanStreet | 0.00% | 0.16% | 0.17% | 0.00% | 0.16% | 0.15% |
| UndoDancer | 0.00% | 0.26% | 0.20% | 0.00% | 0.16% | 0.12% |
| **Average** | **0.00%** | **0.12%** | **0.13%** | **0.00%** | **0.10%** | **0.10%** |

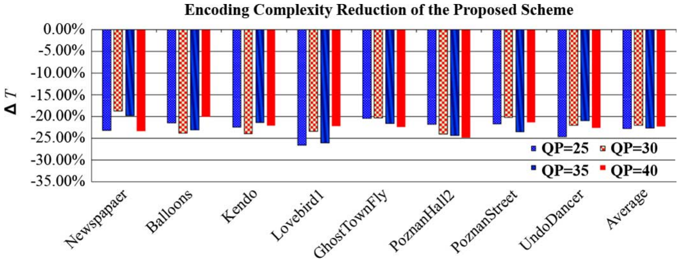

Fig. 5. Encoding complexity reduction of the proposed scheme.

formance is evaluated by the BD-rate [32] and the time reduction is evaluated as

$$\Delta T = \frac{T_{pro} - T_{org}}{T_{org}} \times 100\% \tag{24}$$

where $T_{org}$ and $T_{pro}$ indicate the depth encoding time of the original RDO and the proposed RDO scheme, respectively. The test platform is Intel Xeon(R) CPU E5–2650L@1.8 GHz with Microsoft Visual Studio $C++$ compiler.

Table II tabulates the RD performance of the proposed scheme for 2 view and 3 view cases, where the 2 view test case is extracted from the 3 view data. $\Delta R_v$, $\Delta R_s$ and $\Delta R_{v,s}$ indicate the bit rate reduction in terms of the original video, the synthesized view, and both the original and synthesized views, respectively. The experimental results show that on average the proposed scheme brings negligible performance loss (0.1%).

To verify the robustness and efficiency of the proposed scheme, the encoding complexity reduction computed by (24) is shown in Fig. 5. From this figure, it is observed that the proposed algorithm can achieve 18% to 26% of total encoding time reduction depending on texture QP value ranging from 25 to 40, and on average about 22% of complexity reduction is achieved. It indicates that our scheme can efficiently reduce the encoding complexity in terms of the view synthesis computations.

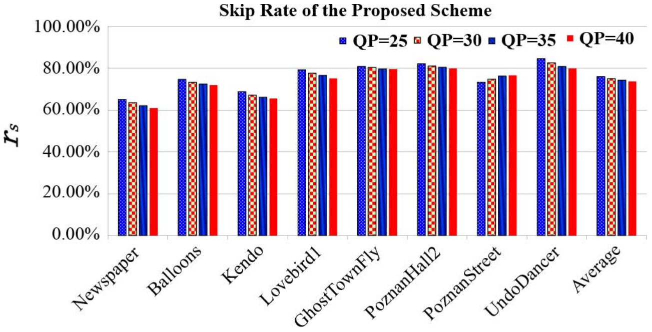

Fig. 6. Skip rate of VSO by the proposed scheme.

TABLE III
HIT RATE OF THE PROPOSED ZSVD MODEL TO THE ACTUAL VIEW RENDERING.

| Sequence | QP=25 | QP=30 | QP=35 | QP=40 |
|---|---|---|---|---|
| Balloons | 94.5% | 94.6% | 94.7% | 94.7% |
| Kendo | 92.1% | 92.2% | 92.5% | 92.7% |
| Newspaper | 92.7% | 92.8% | 92.9% | 92.9% |
| Lovebird1 | 96.7% | 96.6% | 96.3% | 95.8% |
| GhostTownFly | 94.9% | 94.1% | 93.4% | 93.2% |
| PoznanHall2 | 93.4% | 93.1% | 92.9% | 92.5% |
| PoznanStreet | 94.4% | 94.4% | 94.3% | 94.3% |
| UndoDancer | 96.3% | 95.7% | 95.4% | 94.7% |
| **Average** | **94.4%** | **94.2%** | **94.1%** | **93.9%** |

### *B. Analysis of Hit Rate and False Alarm*

To further verify the effectiveness of the proposed ZSVD aware VSO, we conduct a series of experiments on these test sequences. Firstly, we measure the "skip rate" of our proposed algorithm, which is defined as follows

$$r_s = \frac{N_{skip}}{N_t} \tag{25}$$

where $N_{skip}$ denotes the number of rows that are skipped in view synthesis in the proposed algorithm and $N_t$ denotes the total number of rows to be synthesized in the original VSO. The "skip rate" for each sequence is shown in Fig. 6, and it indicates that more than 70% of the total view synthesis computations are skipped, which demonstrates the superior performance of the proposed scheme. Moreover, it is observed that different sequences have different skip rate. This is because that the depth-disparity mapping, texture smoothness and occlusion are all dependent on the properties of the sequences. The skip rate also varies with QP values, as at a higher QP value more distortions are introduced.

In the second experiment, we measured the hit rate and false alarm by comparing the forecast result of the proposed ZSVD model with the actual view rendering approach. The hit rate measures the fraction of true ZSVD that are correctly forecasted by the proposed model, which are provided in Table III. It is observed that on average 94% of the true ZSVD can be correctly forecasted. It is noted that when the depth distortion is within the ZSVD range, the change of depth map may introduce difference in the synthesized view because of the depth dependent view merging or hole filling. For example, the depth change may alter the selection of pixels in view merging, which may lead to slight difference in the final synthesized view. Therefore, we evaluate the false alarm of the proposed scheme, which is tabulated in Table IV. It indicates that the fraction of the forecast ZSVD that were not true is no more than 4% on average. This proves that the proposed model can provide an efficient and accurate determination of ZSVD for the VSO.

TABLE IV
FALSE ALARM OF THE PROPOSED ZSVD MODEL TO THE ACTUAL VIEW RENDERING.

| Sequence | QP=25 | QP=30 | QP=35 | QP=40 |
|---|---|---|---|---|
| Balloons | 3.3% | 3.5% | 3.9% | 4.1% |
| Kendo | 3.6% | 3.3% | 3.1% | 3.1% |
| Newspaper | 2.9% | 3.0% | 2.9% | 2.9% |
| Lovebird1 | 2.1% | 2.6% | 3.2% | 4.1% |
| GhostTownFly | 2.7% | 3.1% | 3.6% | 3.7% |
| PoznanHall2 | 3.3% | 4.1% | 4.2% | 4.1% |
| PoznanStreet | 2.9% | 3.4% | 3.6% | 3.6% |
| UndoDancer | 2.1% | 2.6% | 2.9% | 3.7% |
| **Average** | **2.9%** | **3.2%** | **3.4%** | **3.7%** |

## V. CONCLUSION

In this correspondence, we propose a low complexity adaptive view synthesis optimization scheme to achieve fast and efficient depth compression. The novelty of this scheme lies in defining a novel ZSVD model and demonstrating its applications in high efficiency depth compression. Specifically, three kinds of zero synthesized view error conditions incurred by depth distortion are revealed by ZSVD modeling, which can provide precise conditions for the early termination of VSO. The proposed ZSVD model has been incorporated into the VSO process of HEVC based 3DV codec, in which the view rendering algorithm is pruned by the ZSVD estimator. The proposed scheme demonstrates superior performance as compared to the state-of-the-art HEVC 3DV codec by offering significant complexity reduction, while keeping the comparable RD performance.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Huifang Sun of MERL for his valuable suggestions. The authors also would like to thank the anonymous reviewers for their valuable comments that significantly helped us in improving the presentation of the paper.

## REFERENCES

[1] "Call for proposals on 3D video coding technology," ISO/IEC JTC1/SC29/WG11, MPEG, , Geneva, Switzerland, Doc. N12036, 2011.

[2] G. Tech, K. Wegner, Y. Chen, and S. Yea, "3D-HEVC test Model3," ISO/IEC JTC1/SC29/WG11 MPEG, Switzerland, Switzerland, Doc. JCT3V-C1005, Jan. 2013.

[3] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, D. Marpe, P. Merkle, K. Müller, H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D video coding using advanced prediction, depth modeling, and encoder control methods," in *Proc. Picture Coding Symp.*, May 2012, pp. 1–4.

[4] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Multiview video plus depth representation and coding," in *Proc. IEEE Int. Conf. Image Process.*, 2007, pp. 201–204.

[5] D. Rusanovskyy, K. Müller, and A. Vetro, "Common test conditions of 3DV core experiments," ISO/IEC JTC1/SC29/WG11, MPEG, , Stockholm, Sweden, Doc. JCT3V-A1100, Jul. 16-20, 2012.

[6] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, D. Marpe, P. Merkle, K. Müller, H. Rhee, G. Tech, M. Winken, and T. Wiegand, "Description of 3D video coding technology proposal by Fraunhofer HHI," ISO/IEC JTC1/SC29/WG11 MPEG, Doc, M22570.

[7] M. Domanski, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Luczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski, and K. Wegner, "Technical description of Poznan University of Technology proposal for Call on 3D video coding technology," ISO/IEC JTC1/SC29/WG11 (MPEG),, Geneva, Switzerland, Doc. M22697, Nov. 2011.

[8] P. Ndjiki-Nya, M. Köppel, D. Doshkov, H. Lakshman, P. Merkle, K. Müller, and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-D video," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 551–562, Jun. 2011.

[9] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video Compression," *IEEE Signal Process. Mag.*, vol. 15, pp. 74–90, Nov. 1998.

[10] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, P. H. N. d. With, and T. Wiegand, "The effect of depth compression on multiview rendering quality," in *Proc. IEEE Conf. 3D-TV*, May 2008, pp. 245–248.

[11] P. Merkle, Y. Morvan, A. Smolic, and D. Farin, "The effects of multiview depth video compression on multiview rendering," *Signal Process.: Image Commun.*, vol. 24, no. 1–2, pp. 73–88, Jan. 2009.

[12] 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.

[13] G. Tech, H. Schwarz, K. Müller, and T. Wiegand, "3D video coding using the synthesized view distortion change," in *Proc. Picture Coding Symp.*, May 2012, pp. 25–29.

[14] D. V. S. X. D. Silva, W. A. C. Fernando, and S. T. Worrall, "Intra mode selection method for depth maps of 3D video based on rendering distortion modeling," *IEEE Trans. Consumer Electron.*, vol. 56, no. 4, pp. 2735–2740, Nov. 2010.

[15] J. Jung, E. Son, and S. Yea, "3D-HEVC-CE10 results on modified depth distortion measure by LG," ISO/IEC JTC1/SC29/WG11 MPEG, Doc. M23856, Feb. 2012.

[16] W. 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.*, 2009, pp. 721–724.

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

[18] "3D-HEVC-CE10 results on joint RDO for depth coding of 3D video by ZJU," ISO/IEC JTC1/SC29 WG11 MPEG, San Jose, CA, USA, Doc. M23829, Feb. 2012.

[19] B. Oh, J. Lee, and D. Park, "3D-HEVC-CE10 results on samsung's view synthesis optimization using distortion in synthesized views," ISO/IEC JTC1/SC29 WG11, MPEG, San Jose, CA, USA, Doc. M23668, Feb. 2012.

[20] W.-S Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," in *IS&T/SPIE Electron. Imaging*, 2010, pp. 7543–10.

[21] P. Kauff, N. Atzpadin, C. Fehn, M. Muller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process.: Image Commun.*, vol. 22, no. 2, pp. 217–234, 2007.

[22] C. Chen, Y. Chen, F. Yang, and W. Peng, "A synthesis-quality-oriented depth refinement scheme for MPEG free viewpoint television (FTV)," in *Proc. IEEE Int. Symp. Multimedia*, 2009, pp. 171–178.

[23] H. Schwarz and D. Rusanovskyy, "Common test conditions for AVC and HEVC-based 3DV," ISO/IEC JTC1/SC29/WG11 MPEG, San Jose, USA, Doc. N12560, Feb. 2012.

[24] A. Secker and D. Taubman, "Highly scalable video compression with scalable motion coding," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1029–1041, Aug. 2004.

[25] H. Schwarz and D. Rusanovskyy, "Common test conditions for 3DV experimentation," ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, Doc. w12745, May 2012.

[26] W. Han, J. Min, I. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. Hong, M. Cheon, N. Shlyakhov, K. McCann, T. Davies, and J. Park, "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1709–1720, Dec. 2010.

[27] B. Oh, J. Lee, D. Park, G. Tech, K. Müller, T. Wiegand, S. Wang, S. Ma, H. Liu, J. Jia, and J. Jung, "3D-CE8.h results on view synthesis optimization by Samsung, HHI and LG-PKU," ISO/IEC JTC1/SC29/WG11 MPEG2012, , Stockholm, Sweden, JCT2-A0093, Jul. 2012.

[28] B. Oh, J. Lee, D. Park, G. Tech, K. Müller, and T. Wiegand, "3D-CE8.h results on view synthesis optimization," ISO/IEC JTC1/SC29/WG11 MPEG2012, , Stockholm, Sweden, Doc. JCT2-A0033, Jul. 2012.

[29] [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVC-Software/tags/HTM-4.0

[30] H. Yuan, Y. Chang, 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.

[31] Q. Wang, X. Ji, Q. Dai, and N. Zhang, "Free viewpoint video coding with rate-distortion analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 6, Jun. 2012.

[32] G. Bjontegaard, "Improvements of the BD-PSNR model," ITU-T SG16, Doc. VCEG-AI11, Jul. 2008.