Abstract—This paper addresses high performance depth coding in 3D video by making good use of its coded texture video counterpart. The relationship between the depth and its associated texture video in terms of coding mode and motion vector is carefully examined. Our statistical study suggests that the skip coding mode and its associated motion vectors in the coded texture can be shared for depth coding by saving bit rate at the cost of little increase of distortion, which subsequently results in a non-sequential coding of the depth map. In this sense, coding/prediction of a block can be performed by utilizing the skip-coded blocks below and right which are not available in the conventional sequential coding, thus producing the so-called omnidirectionally predicted blocks (OD-Blocks) in the intra-coding by making the best use of (at most) four neighboring blocks. Moreover, in view of the depth-texture structure similarity, a depth-texture cooperative clustering (DTCC) based prediction method is proposed for cluster-based depth prediction in the intra-coding, which exploits the structure similarity for the current coding block and its neighboring pixels around the block. On the other hand, some large prediction errors may be present for the depth-texture misaligned pixels, which may greatly compromise the coding performance. To deal with these large residuals induced by the depth-texture misalignment, a simple yet effective detection and rectification approach is incorporated in the proposed depth coding scheme. Experimental results show that our proposed depth coding scheme achieves superior rate-distortion performance compared with other relevant coding methods.

Index Terms—Depth coding, 3D video, prediction, intra coding, depth-texture misalignment

I. INTRODUCTION

With the increasing desire for realistic media and the rapid development of computer graphics, computer vision, and multimedia technology, 3D video has made great progresses in the last few decades [1, 2]. Advances in multiview video technology have accelerated the development of new applications, such as three dimensional television (3DTV) [3] and free viewpoint television (FTV) [4]. To facilitate 3D rendering, the depth map has been introduced and widely used. A depth map is composed of depth samples, with each sample indicating the relative distance from an object in the 3D space to the camera plane. Each depth sample is represented by an 8 bit value corresponding to a pixel in the video frame [5]. Virtual views can be rendered with the depth image-based rendering (DIBR) technique [6, 7]. With the introduction of depth maps in the 3D video, depth maps need to be coded and transmitted in addition to the texture videos.

The Moving Picture Experts Group (MPEG) has explored technologies related to the Multiview Video plus Depth (MVD) format for several years. In March 2011, MPEG issued a Call for Proposals (CfP) for 3D video coding technology [8]. Since July 2012, the Joint Collaborative Team on 3D Video Coding (JCT-3V) has initiated two parallel H.264/AVC-based MVD coding developments, MVC+D and 3D-AVC. MVC+D [9, 10], as an MVC (Multiview Video Coding) extension for the inclusion of depth maps, specifies the encapsulation of MVC-coded texture and depth views into a single bitstream. The MVC+D specification was finalized in January 2013. Since the coding technology for texture and depth videos is identical to MVC, MVC+D is backward-compatible with MVC. On the other hand, 3D-AVC [11, 12], as a multiview video and depth extension of H.264/AVC, exploits the correlation between texture and depth, and includes several coding tools that improve the compression efficiency over MVC+D. The 3D-AVC specification was finalized in October 2013.

The depth map coding techniques in the literature can be classified into two categories depending on the relationship with the corresponding texture video: independent coding and texture-assisted depth coding. Independent depth coding techniques utilize the features of the depth maps which exhibit different characteristics from the conventional images, such as a large portion of smooth areas separated by sharp edges.
Merkle et al. [13, 14] proposed a platelet-based coding method that first divides the depth image into blocks of variable size employing the quadtree decomposition, and then approximates each block with one of the predefined piecewise polynomial modeling functions. In this way, a region of gradually changing depth can be represented efficiently with a single linear function. However, their method cannot efficiently deal with the blocks that sit on relatively complex object boundaries. In [15], Kang et al. proposed an adaptive geometry based intra prediction method for depth coding. Their method utilizes the information from the neighboring blocks to partition the current block, and each region of the block is then independently predicted. While saving the cost for the partition representation compared with their previous work in [16], the method also cannot efficiently deal with relatively complex edges in the depth block. In [17] and [18], a depth map coding scheme is proposed to achieve high scalability by using breakpoints to represent the geometry information. Oh et al. [19] proposed a depth intra skip prediction (DISP) method to skip the coding of the residual data in the Intra 16*16 mode with the prediction direction estimated from the neighboring blocks. In [20] and [21], an edge-aware intra prediction and an edge-adaptive transform (EAT) are used to code the edge blocks with an extra edge map. In [22], an edge prediction scheme is devised to code the edge map, and a sub-block motion prediction method is utilized to enhance the inter-frame coding efficiency. A plane segmentation based intra prediction (PSIP) method is proposed in [23] to efficiently code a prediction map instead of an edge map. In [24], Milani et al. proposed a depth image coder based on progressive silhouettes with a mask map coded with the JBIG binary coder. While such approaches improve the efficiency in the depth coding process, overhead bits are required to represent the edge map, which in turn reduces the overall coding efficiency.

The texture-assisted depth coding techniques make use of the corresponding texture information to enhance the depth coding efficiency. Based on the information used in the depth coding, the texture-assisted depth coding techniques can be further divided into two types, motion-sharing (based on temporal similarity) and structure-sharing (based on spatial similarity). The motion-sharing based depth coding takes advantage of the motion similarity between texture and depth by assuming the texture motion vector as a possible motion vector candidate in the coding of the depth map sequence. Grewntsch et al. [25] and Oh et al. [26] studied the similarity of motion vectors between the texture and depth sequences in terms of the correlation coefficient and the average distance. They both considered using all the motion vectors of different modes in the texture video as candidate motion vectors, for the depth map coding. A joint motion estimation process is used in [27] to estimate the common motion vectors for the texture video and the depth video by taking both of their distortions into consideration. An inside view motion prediction (IVMP) method is proposed in [28] to take advantage of the motion vectors of the texture video to predict those of the depth video. In [29], the skip mode in depth coding is automatically selected whenever the skip mode is adopted in the texture, thus leading to a reduction in both coding bitrate and coding complexity. Likewise, in [30], Lee et al. proposed a coding scheme to skip selected blocks of the depth map based on the temporal and inter-view correlations of texture images. Although these schemes can reduce the coding complexity by skipping the coding of selected blocks, the subsequent coding in these schemes does not use other available information to further improve the coding efficiency.

The second type of texture-assisted depth coding techniques is the structure-sharing based depth coding. This method exploits the structure similarity between texture and depth to form a better prediction for depth coding. Milani et al. [31] presented a depth coding scheme by exploiting the segmentation information of the corresponding texture image to predict the shape of the different surfaces in the depth map. Like the piecewise functions in [13], each of the segmentations in [31] is approximated with a parameterized plane. The complex regions of the depth map are still coded using conventional coding methods. In this way, the blocks with complex edges do not need to be further separated into smaller regions, thus leading to higher efficiency. However, there is no improvement for the complex regions. In [32], a sparse dyadic (SD) mode, together with a refinement procedure, is used to recover the edges of a depth block by utilizing the information from the corresponding texture block. An SD mode is determined first, and then the refinement procedure, which employs the corresponding texture information, segments the block into two partitions. Each partition is then predicted from neighboring pixels. The information used for the segmentation and the prediction is acquired separately which may corrupt the prediction for the block. With the corresponding texture video available, there are some works [29, 33-36] targeting at the view synthesis distortion optimized mode selection for depth coding. In [33], a synthesized view distortion metric is used to measure the distortion of the synthesized view with the coded depth block. In [29, 34-36], view synthesis distortion models are proposed to facilitate the bit allocation and the rate-distortion optimization. Such methods emphasize the distortion metric for depth coding instead of coding techniques.

The above structure-sharing based depth coding methods take advantage of the structure similarity between depth and texture to enhance the coding efficiency. However, the edge of the depth map is generally not well aligned with the texture edge [7, 37] due to the limitation of the depth generation techniques. Taking the most popular stereo matching algorithm for example, the generated depth values around the object boundaries are generally inaccurate due to the occlusion of background or similarities of textures. Without any regulation of the depth edges, large prediction errors may occur along the edges in the structure-sharing depth coding. Therefore, for block-based depth coding, a rectification procedure is needed before the coding of the residuals.

To make full use of the similarities of motion and structure, a

1051-8215 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
new framework is proposed in this paper for depth coding. The contributions of this paper are summarized as follows: 1) Based on the statistics, we propose to use the skip mode and its motion vector from the corresponding texture block to code the depth block. We also present statistics to justify the proposed approach; 2) We propose omnidirectional blocks which can be predicted better from the neighboring blocks; 3) We propose a depth-texture cooperative clustering (DTCC) based prediction method, which can give better predictions; 4) We propose a method which can detect and rectify misalignments between depth and texture edges to give better coding and virtual view synthesis results. Simulation results demonstrate the effectiveness of our proposed approaches.

The rest of this paper is organized as follows. In Section 2, new observations and analyses of the similarity between the depth and texture video are shown. In Section 3, the framework of our proposed coding scheme is presented based on these new observations. Section 4 introduces the proposed weighted averaging based omnidirectional prediction method for the omnidirectional blocks. Section 5 explains the proposed depth-texture cooperative clustering based prediction method. The detection and rectification technique for the misalignment between depth and texture is described in Section 6. Section 7 presents the experimental results. Finally, this paper is concluded in Section 8.

II. NEW OBSERVATIONS ON DEPTH-TEXTURE SIMILARITIES

Depth and the corresponding texture video both represent the same scene: the depth map sequence captures the 3D structure of a scene and depicts the surface of the objects, while the corresponding texture video represents the texture intensity of the objects. Therefore, motion and structure similarities exist between depth and texture videos.

A. Motion similarity

In the block-based video coding schemes, coding modes, which indicate the prediction mode and the size of the prediction units, reflect the structure of the video content to a certain degree. Inter modes, along with the corresponding motion vectors adopted by depth and texture coding, are similar to each other given the motion similarity between depth and texture videos. Intra modes, however, are not really similar between depth and texture coding due to the different characteristics of the depth and texture videos. To measure the motion similarity between depth and texture videos, we measure the mode similarity and then the motion vector similarity for each coding mode since the motion vector is derived on the basis of the mode.

1) Coding mode similarity

The percentage of the modes adopted in the depth and the corresponding texture video are shown in Fig. 1. In the figure, the results for four different QP (Quantization Parameter) values for texture and depth coding, respectively, as suggested in [38], are shown. From the figure, two conclusions can be obtained: first, most of the blocks are coded with the skip mode both in depth and texture videos. Second, the number of intra modes adopted in the depth video is much greater than that adopted in the texture video, which proves our statement in the previous paragraph about the significant differences of the intra modes between depth and texture.

Moreover, from the coding perspective it can be seen that intra modes in the depth coding are more important than that in the texture video coding. Therefore, a more efficient approach for intra mode for depth coding is needed.

Fig. 1. Mode distribution of texture and depth video when QP values for the texture video equal to 26, 31, 36 and 41, respectively, while QP values for the depth video equal to 30, 35, 40 and 45, respectively. Ten sequences (two views for each sequence), Ballet (100 frames), Breakdancers (100 frames), Lovebird1 (100 frames), Balloons (300 frames), Kendo (300 frames), Newspaper (300 frames), Champagne_tower (300 frames), Poznan_Hall2 (200 frames), Poznan_Street (250 frames) and Undo_dancer (250 frames), are used to derive the above figures and the figures in the following. The H.264/AVC JM version 18.2, is used as the codec with the IPPP... coding structure. Similar results can be obtained with other sequences and coding structures.

Fig. 2 further shows the mode distribution of the depth blocks where the corresponding texture blocks are coded with the skip mode. It can be seen that, for different QP values, most of the blocks which adopt the skip mode in the texture video are still coded with the skip mode in the depth video, while the remaining blocks are mostly coded with the Intra16 mode. In
Fig. 3, the blocks with white edges represent the depth blocks coded with the Intra16 mode while the corresponding texture blocks are also coded with the skip mode. From the figure, we can see that the blocks coded with Intra16 mode are mainly on smooth surfaces. In [29], Kim et al. noted that the depth information of the smooth regions is very likely to be noisy due to the lack of features to perform stereo matching, therefore it would be inefficient to spend more bits for the thorough coding of such smooth regions. To solve such problems and enhance the temporal consistency in the smooth regions, the skip mode from the texture video is ‘forced’ in depth coding. It needs to be noted that the skip mode is just a mode signaling that the motion vectors and residual block after prediction will not be coded. The motion vectors of the skip mode will be derived from the neighboring blocks in the decoder side. Therefore, for the block at the same location coded with the skip mode in the depth and texture videos, the motion vectors may not be the same. Since motion vectors are derived on the basis of the mode, the motion vector similarity of each mode is further investigated in the following.

![Mode distribution percentage of the depth blocks which corresponding blocks in texture video are coded with the skip mode.](image)

*Fig. 2.* Mode distribution percentage of the depth blocks which corresponding blocks in texture video are coded with the skip mode. The QP pairs for texture and depth are set to (26, 30), (31, 35), (36, 40) and (41, 45), respectively.

![Illustration of the blocks coded with I16MB mode in depth when the corresponding texture blocks are coded with the skip mode.](image)

*Fig. 3.* Illustration of the blocks coded with I16MB mode in depth when the corresponding texture blocks are coded with the skip mode.

2) **Motion vector similarity**

The motion vector in block-based coding indicates the displacement of the reference block from the current block, by minimizing a matching criterion such as SAD (Sum of Absolute Difference). When an object moves, besides the location changes, the depth may also change to a very different value. Since the motion estimation process is basically a pattern matching process, motion vectors in a texture frame and the corresponding depth frame indicate reference blocks with similar texture and depth values, respectively. Thus, the motion vectors for the depth and the texture videos may differ from each other in the same frame.

For a depth block, if we do not perform motion estimation to find its motion vector, but reusing the corresponding MV from the texture signal as its motion vector, the prediction residual (or the SAD) could increase. Fig. 4 shows the SAD increase introduced by using the motion vectors from the corresponding texture blocks for different modes. It can be seen that the increase of the prediction residual of the skip mode is almost negligible. Therefore, for the skip mode, the motion vectors of the texture blocks can be used for coding the corresponding depth blocks. Since the motion vectors point to reference depth blocks in the previously decoded depth frames, and the motion vectors of the neighboring texture blocks are available at both the encoder and the decoder, the skip blocks in the depth frame can be encoded and decoded first. Therefore, before encoding or decoding a depth block, the neighboring skipped depth blocks can be available for prediction or reconstruction at the encoder and the decoder.

Fig. 5 demonstrates an example of coding results of the depth blocks. Blocks with black color at the left and upper edges are the skipped blocks (S-Blocks) which are first coded by using the skip mode and the corresponding motion vectors from the corresponding texture blocks. The blocks with white color at the left and upper edges are the omnidirectional blocks (OD-Blocks) which are blocks with skipped blocks existing below or to the right, and the rest of the blocks with blue color at the left and upper edges are the normal blocks (N-Blocks). In this specific figure, there are 1566 S-Blocks (about 50.98%), 795 OD-Blocks (about 25.88%), and 711 N-Blocks (about 23.14%). It is clear that most of the blocks are S-Blocks which can be coded efficiently. For the remaining blocks, about 52.79% are OD-Blocks which can be predicted more accurately as will be discussed in the later sections.

![The increase of SAD for different modes by sharing the motion vectors from the texture video, when the QP pairs for texture and depth are set to (26, 30), (31, 35), (36, 40) and (41, 45), respectively.](image)

*Fig. 4.* The increase of SAD for different modes by sharing the motion vectors from the texture video, when the QP pairs for texture and depth are set to (26, 30), (31, 35), (36, 40) and (41, 45), respectively.

![Depth block coding results after sharing the skip mode and motion vectors from the corresponding texture block. Blocks in black, white, and blue at the left and upper edges represent the S-Blocks, OD-Blocks, and N-Blocks, respectively.](image)

*Fig. 5.* Depth block coding results after sharing the skip mode and motion vectors from the corresponding texture block. Blocks in black, white, and blue at the left and upper edges represent the S-Blocks, OD-Blocks, and N-Blocks, respectively.
B. Structure similarity

As the depth map represents the distance instead of the intensity of the object, it is composed of smooth areas bounded by sharp edges [13, 14, 24, 39, 40]. Each region can be regarded as an independent object. Considering that the texture image represents the same scene consisting of all the objects as the depth map, a region-based or object-based structure similarity can be expected between depth and texture videos.

However, misalignment along the edges [7, 37] exists between the depth and texture videos due to the limitation of depth map generation or capture techniques. Given that the depth map is used to render virtual views with the texture image, the misaligned pixels along the edges of the depth map need to be revised before synthesizing the view in the decoder side. Fig. 6 shows the edge of the depth map and the texture image using the Canny edge detector [41], and the corresponding misalignment between the depth map and the texture image when the texture edge is chosen as the trusted reference. From the figures, it can be seen that the misalignment is mainly one or two pixels around the depth edge. To deal with the misalignment between the texture and the depth maps, we propose a rectification technique which can be efficiently incorporated into the block-based coding process while maximizing the coding efficiency as will be discussed in Section VI.

(c) Misalignment between (a) and (b)

Fig. 6. Edges of texture and depth images and the misalignment along the edges (each edge pixel is marked with “+” for better illustration); (a) and (b) show the Canny edge detection results in texture and depth, respectively (the high and low threshold are set as 0.1 and 0.04); (c) illustrates the texture-depth misalignment with respect to their common edges.

III. OVERVIEW OF THE FRAMEWORK OF THE PROPOSED CODING SCHEME BASED ON THE NEW OBSERVATIONS

From the analysis of the motion similarity presented in the previous section, it is known that a large proportion of blocks are coded with the skip mode both in the depth and the texture videos, and the distortion introduced by sharing the motion vectors of the skip mode is negligible. Therefore, the skip mode and its corresponding motion vectors from the texture video can both be used in depth coding. With the corresponding texture coding information, these blocks can be coded first with the skip mode and the corresponding motion vectors, which can be decoded in the same way at the decoder side.

With information from below and/or to the right of the OD-Blocks (which are the blocks with skipped blocks existing below and/or to the right), available for prediction, new predicting techniques with higher prediction accuracy can be designed. Due to the characteristics of the depth frame which has a large portion of smooth regions with sharp edges, we further classify the OD-Blocks into two types, edge OD-Blocks (OD-Blocks with edges in the blocks) and non-edge OD-Blocks (OD-Blocks without edges in the blocks), in order to implement different methods according to their different characteristics. In this paper, a simple weighted averaging based omnidirectional (WA-OD) prediction mode is added for non-edge OD-Blocks to exploit the information from all available directions (the Canny edge detector is used to derive the edges for the depth map). This mode can more efficiently predict the blocks with gradual changes.

In order to obtain a higher coding efficiency for the blocks with sharp edges, each region within the edge block needs to be predicted separately. Given our previous analysis about the depth-texture region-based structure similarity, a region-based prediction method can be implemented with the aid of the texture image. To keep the algorithm compatible with the current block-based coding framework, the segmentation and the prediction processes used in the region-based prediction method both need to be implemented on the basis of the blocks, and the computation needs to be as simple as possible. In this paper, a depth-texture cooperative clustering (DTCC) based prediction mode is proposed and incorporated in the proposed coding scheme. This mode is available for both the OD-Blocks and the N-Blocks. Due to the misalignment between depth and
texture edges, large residuals may occur along the edges after the prediction. Therefore, a simple yet effective detection and rectification technique is developed to deal with the misalignment between the depth map and the texture image.

The structure of the proposed coding scheme is summarized in Fig. 7. It needs to be noted that with some adjustments, the proposed methods can be adapted to other coding schemes. In addition, since both the WA-OD prediction method and the DTCC based prediction method only utilize the information from the current depth frame and the corresponding texture frame, they can be incorporated as additional intra modes in the coding process. The proposed WA-OD mode, DTCC mode, and the detection and rectification of misalignments will be explained in section IV, V, and VI, respectively.

IV. WEIGHTED AVERAGING BASED OMNIDIRECTIONAL PREDICTION

In this section, a weighted averaging based omnidirectional (WA-OD) prediction mode is proposed for the OD-Blocks to exploit more information from the available neighboring blocks in all directions. For the OD-Blocks, more neighboring blocks are available for prediction, not only above and to the left of the OD-Block, but also below and/or to the right of them. Information to the right or below the OD-Blocks can improve the accuracy of the prediction. However, in conventional video coding methods, this information cannot be exploited. The proposed weighted averaging based omnidirectional prediction method is shown in Fig. 8. Fig. 8 (a) is for the OD-Blocks where neighboring blocks are available for prediction, not only from above and left, but also below and right, while Fig. 10 (b) represents the OD-Blocks where the neighboring blocks available for prediction are from above, left, and below, Fig. 10 (c) represents the neighboring blocks from above, left, and right. We take the first case as an example to explain the proposed WA-OD method. The prediction value for each pixel of the block can be obtained from the four neighboring pixels with the weighted averaging method. The weights used for prediction is determined based on the bilinear interpolation (i.e., the weights are inversely proportional to the distances) as shown in Eq. (1):

\[
p_{\text{value}}(i, j) = U(i) \frac{M + 1 - j}{M + N + 2} + D(i) \frac{j}{M + N + 2} + L(j) \frac{N + 1 - i}{M + N + 2} + R(j) \frac{i}{M + N + 2},
\]

where \(p_{\text{value}}(i, j)\) represents the prediction value for the pixel located at \((i, j)\), and \(U\), \(D\), \(L\), and \(R\) represent the values of the neighboring pixels from above, below, left, and right of the block, respectively. \(M\) and \(N\) represent the size of the block. The WA-OD method takes advantage of the pixel information from all neighboring blocks available for prediction. This method is suitable for the prediction of the region with gradual depth changes.

V. DEPTH-TEXTURE COOPERATIVE CLUSTERING FOR DEPTH PREDICTION

To obtain a precise intra prediction for the edge block is much more difficult than that for the non-edge block. The conventional predicting process cannot automatically identify which pixel in the neighboring block belongs to the same region as the current pixel, which leads to large prediction residuals and low coding efficiency. The depth map is composed of smooth regions bounded by sharp edges [13, 14, 23, 39, 40]. Even in an edge block, each part of the block segmented by the edge is still smooth. So, the prediction for the depth block can be efficient if each pixel in the region can be predicted from the neighboring pixels in the same region. Therefore, a region-based prediction is needed to efficiently predict the depth edge block. The region-based prediction of the depth block can be obtained through the help from the segmentation of the corresponding texture block in view of the region-based structure similarity between texture and depth videos.

The overall process of the proposed DTCC based prediction method is illustrated in Fig. 9 for the case of the OD-Block. The different colors represent the clusters of the block with F and B representing the clusters of foreground and background, respectively. The process starts from the neighboring pixels of the to-be-coded depth block. Taking the neighboring line of pixels DL1 for example, these pixels are first grouped into one or two clusters based on the following process.

Step 1) Obtain the maximum change location among the set of pixels as:

\[
P_n = \arg \max_{r \in \{P\}} (\text{abs}(P_r - P_n))
\]

where \(P_n\) represent the pixel with largest change between adjacent pixels, \(P\) represents the set of the neighboring pixels, and \(P_n\) represent the pixel at location \(n\).
Step 2) Classify the pixels according to the obtained maximum change location \( P_n \) as follows.

\[
\begin{align*}
    P_f & \sim P_n \in C_o, P_{n+1} \sim P_f \in C_1, \quad \text{if } \abs{P_{n+1} - P_n} > T \\
    P_f & \sim P_f \in C_o, \quad \text{otherwise.}
\end{align*}
\]

where \( P_f \) and \( P_l \) represent the first and the last pixel of the pixel set, respectively. \( C_o \) and \( C_1 \) represent the grouped clusters.

The same process is applied on the other neighboring lines of pixels, DL2 to DL4. The segmentation results from the depth are then mapped on the corresponding texture neighboring lines as shown in Fig. 9. To improve the classification accuracy, the texture pixels TL1 are further sub-clustered. Within each cluster from the depth partition, at most two texture sub-clusters are allowed. Likewise, TL2 to TL4 are clustered in the same way.

With the formed clusters of the neighboring pixels, the cluster center can be obtained as the representative value of each cluster. In this paper, the average value is used as the cluster center of each cluster.

![Fig. 9. The structure of the proposed DTCC prediction method. The different colors represent the clusters of the block with F and B representing the clusters of foreground and background, respectively.](image)

VI. DETECTION AND RECTIFICATION OF MISALIGNMENT BETWEEN DEPTH AND TEXTURE

The depth-texture misalignment generally exists in the boundary regions of the depth map due to the limitation of the current depth generation techniques. In the coding process, large residuals may occur at the locations of the misalignment after the prediction. This leads to a large number of high-frequency coefficients in the DCT-transform domain which will reduce the coding efficiency. In order to reduce such effect of the misalignment and improve the quality of the synthesized view, misaligned pixels need to be rectified before the residual coding.

As shown in the observation of the structure similarity, the misalignment is mainly one or two pixels around the depth edge. Therefore, a three-pixel-wide region around the depth edge is first drawn as an error-prone region within which misalignment is most likely to exist. The large residuals in this error-prone region after the prediction could be caused by either the inaccurate prediction or the misalignment, when the original value in the depth map is wrong. Two conditions are set to distinguish these two different cases. First, the depth value of the pixel with large residual is checked to see if it falls in the range of a cluster. This pixel will not be regarded as a misaligned pixel if it does not fall in the range of any clusters. Second, the residuals outside the error-prone region are further checked. If large residuals also exist outside the error-prone region, which means the region surrounding the large residual pixel cannot be well predicted, the large residual pixel in the error-prone region will also not be regarded as a misaligned pixel. Fig. 10 shows the detection principle for the misaligned pixels. The dots in the block are the pixels with large residuals and the regions with lines in different directions represent different objects in the block. In this figure, the residuals of the pixels in the reliable region are all small which means all the objects can be well predicted. Therefore, the misaligned pixels can be detected as large residuals.

For the detected misaligned pixels, a rectification procedure is applied before the entropy coding. As shown in Fig. 11, our rectification procedure takes the original depth values of the pixels around the misaligned pixel (but outside the error-prone region) and the predicted value of the misaligned pixel to determine the region that the misaligned pixel belongs to. The depth value of the misaligned pixel is set to the depth value of L or R in Fig. 11 depending on which one is closer to the predicted value, and the prediction residual of the misaligned pixel is revised accordingly. With the misaligned pixels rectified, no large residuals exist. Therefore, the coding efficiency can be improved in the DCT-based coding framework.

It should be noted that although there have been some works in the literature regarding the rectification of the misalignment such as those described in [7]. These works focus on the processing of the misalignment in the view synthesis process, which is generally done in the decoder side. To our best knowledge, our proposed misalignment detection and
rectification method is the first that takes the misalignment problem into consideration in the encoding process, aiming at improving the coding efficiency and the quality of the synthesized view.

The coding efficiency is measured by the rate-distortion (R-D) metrics. The PSNR value is calculated by using MSE between the rendered view using decoded depth/texture sequences and the rendered view using the original depth/texture sequences [46]. To evaluate the performance of the proposed depth coding scheme over the other depth coding schemes, the bitrate is represented by the total depth bitrate of the compressed depth maps as in [20-21, 29-32].

The R-D curves of the proposed DTCC intra prediction and misalignment rectification methods for Ballet sequence are shown in Fig. 12. In this figure, the “Proposed DTCC” indicates the proposed DTCC plus the misalignment rectification method. From the figure, it can be seen that the proposed DTCC method outperforms the plane segmentation intra prediction (PSIP) method and the depth intra skip prediction (DISP) method, and significantly improve the coding efficiency of the ATM. The BDPSNR and BDBR are computed according to [47]. It can be observed that, with the proposed DTCC and the misalignment rectification methods, we can achieve 34.56% bitrate savings for Ballet compared to ATM.

The weighted averaging based omnidirectional (WA-OD) prediction method is implemented together with the proposed forced skip method. The R-D curves for Kendo sequence are shown in Fig. 13. In this figure, the “Proposed WA-OD” indicates the proposed weighted averaging based omnidirectional prediction together with the proposed forced skip method. It can be seen that the proposed WA-OD method can achieve better performance than the forced skip method, inside-view motion prediction (IVMP) method, and the ATM. With the proposed WA-OD prediction and forced skip methods, we can achieve 29.57% bitrate savings for Kendo compared to ATM.

The R-D curves of the proposed depth coding scheme for Ballet and Kendo sequences are shown in Fig. 14. Table II shows coding results for eight sequences. In order to show the impact on total bitrate of the multiview video plus depth coding with the proposed depth coding scheme, the coding results of using the total bitrates including both the texture and depth video bitrates are also shown in Table II. It is observed that our proposed coding scheme is effective for improving the system performance. Compared with ATM, we can achieve 49.97% and 29.54% bitrate savings in terms of the depth video bitrates for Ballet and Kendo, respectively.

### Table I

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Image Property</th>
<th>Frames</th>
<th>Original Views</th>
<th>View to Synthesize</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ballet</td>
<td>1024x768, 15fps</td>
<td>100</td>
<td>3-5</td>
<td>4</td>
</tr>
<tr>
<td>Breakdancers</td>
<td>1024x768, 15fps</td>
<td>100</td>
<td>3-5</td>
<td>4</td>
</tr>
<tr>
<td>Kendo</td>
<td>1024x768, 30fps</td>
<td>300</td>
<td>3-5</td>
<td>4</td>
</tr>
<tr>
<td>Balloons</td>
<td>1024x768, 30fps</td>
<td>300</td>
<td>3-5</td>
<td>4</td>
</tr>
<tr>
<td>Newspaper</td>
<td>1024x768, 30fps</td>
<td>300</td>
<td>4-6</td>
<td>5</td>
</tr>
<tr>
<td>Undo_Dancer</td>
<td>1920x1088, 25fps</td>
<td>250</td>
<td>1-5</td>
<td>3</td>
</tr>
<tr>
<td>Pozan_Street</td>
<td>1920x1088, 25fps</td>
<td>250</td>
<td>3-5</td>
<td>4</td>
</tr>
<tr>
<td>Pozan_Hall2</td>
<td>1920x1088, 25fps</td>
<td>200</td>
<td>5-7</td>
<td>6</td>
</tr>
</tbody>
</table>
Fig. 12. RD curves of the proposed DTCC method for Ballet.

Fig. 13. RD curves of the proposed WA-OD prediction method for Kendo.

Fig. 14. RD curves for the proposed coding scheme. (a) Ballet. (b) Kendo.

Fig. 15. Snap shots of the reconstructed depth maps and rendered images with QP pair for texture and depth videos setting as (26, 30). (a) Reconstructed depth image by ATM. (b) Reconstructed depth image by proposed method. (c) Synthesized image by ATM. (d) Synthesized image by proposed method.

Fig. 16. Snap shots of the reconstructed depth maps and rendered images with QP pair for texture and depth videos setting as (31, 35). (a) Reconstructed depth image by ATM. (b) Reconstructed depth image by proposed method. (c) Synthesized image by ATM. (d) Synthesized image by proposed method.
Table II: Coding Performance Comparison

<table>
<thead>
<tr>
<th>Sequence</th>
<th>QP pair</th>
<th>Texture video bitrate</th>
<th>Depth video</th>
<th>Coding gain over ATM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rate / PSNR</td>
<td>Rate / PSNR</td>
<td>Rate / PSNR</td>
<td>Rate / PSNR</td>
</tr>
<tr>
<td><strong>Ballet</strong></td>
<td>(26, 30)</td>
<td>(36, 30)</td>
<td>(31, 35)</td>
<td>(36, 40)</td>
</tr>
<tr>
<td></td>
<td>106/36.33</td>
<td>349/33.82</td>
<td>482/37.25</td>
<td>389/34.37</td>
</tr>
<tr>
<td><strong>Breakdancers</strong></td>
<td>(26, 30)</td>
<td>(31, 35)</td>
<td>(31, 36)</td>
<td>(36, 40)</td>
</tr>
<tr>
<td></td>
<td>205/34.37</td>
<td>190/36.33</td>
<td>190/36.33</td>
<td>190/36.33</td>
</tr>
<tr>
<td><strong>Kendo</strong></td>
<td>(26, 30)</td>
<td>(31, 35)</td>
<td>(31, 36)</td>
<td>(36, 40)</td>
</tr>
<tr>
<td></td>
<td>205/34.37</td>
<td>190/36.33</td>
<td>190/36.33</td>
<td>190/36.33</td>
</tr>
<tr>
<td><strong>Balloons</strong></td>
<td>(26, 30)</td>
<td>(31, 35)</td>
<td>(31, 36)</td>
<td>(36, 40)</td>
</tr>
<tr>
<td></td>
<td>205/34.37</td>
<td>190/36.33</td>
<td>190/36.33</td>
<td>190/36.33</td>
</tr>
<tr>
<td><strong>Newspaper</strong></td>
<td>(26, 30)</td>
<td>(31, 35)</td>
<td>(31, 36)</td>
<td>(36, 40)</td>
</tr>
<tr>
<td></td>
<td>205/34.37</td>
<td>190/36.33</td>
<td>190/36.33</td>
<td>190/36.33</td>
</tr>
<tr>
<td><strong>Undo_Dancer</strong></td>
<td>(26, 30)</td>
<td>(31, 35)</td>
<td>(31, 36)</td>
<td>(36, 40)</td>
</tr>
<tr>
<td></td>
<td>205/34.37</td>
<td>190/36.33</td>
<td>190/36.33</td>
<td>190/36.33</td>
</tr>
<tr>
<td><strong>Pozan_Street</strong></td>
<td>(26, 30)</td>
<td>(31, 35)</td>
<td>(31, 36)</td>
<td>(36, 40)</td>
</tr>
<tr>
<td></td>
<td>205/34.37</td>
<td>190/36.33</td>
<td>190/36.33</td>
<td>190/36.33</td>
</tr>
<tr>
<td><strong>Pozan_Hall2</strong></td>
<td>(26, 30)</td>
<td>(31, 35)</td>
<td>(31, 36)</td>
<td>(36, 40)</td>
</tr>
<tr>
<td></td>
<td>205/34.37</td>
<td>190/36.33</td>
<td>190/36.33</td>
<td>190/36.33</td>
</tr>
</tbody>
</table>

Table III: Complexity Analysis Based on the Encoding Time

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Coding Time Per Frame (ms)</th>
<th>Time Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ballet</strong></td>
<td>384/34.9</td>
<td>379/34.8</td>
</tr>
<tr>
<td><strong>Breakdancers</strong></td>
<td>439/34.1</td>
<td>411/33.7</td>
</tr>
<tr>
<td><strong>Kendo</strong></td>
<td>401/34.8</td>
<td>397/34.8</td>
</tr>
<tr>
<td><strong>Balloons</strong></td>
<td>334/34.8</td>
<td>312/34.8</td>
</tr>
<tr>
<td><strong>Newspaper</strong></td>
<td>397/34.8</td>
<td>397/34.8</td>
</tr>
<tr>
<td><strong>Undo_Dancer</strong></td>
<td>1098/34.8</td>
<td>1074/34.8</td>
</tr>
<tr>
<td><strong>Pozan_Street</strong></td>
<td>958/34.8</td>
<td>976/34.8</td>
</tr>
</tbody>
</table>

Fig. 15 and Fig. 16 show snapshots of the reconstructed depth maps and rendered images of ATM 10.0 and the proposed depth coding scheme with the QP pair set as (26, 30) and (31, 35), respectively. It can be clearly seen that our proposed depth coding scheme can obtain a better quality than ATM, especially around the edges. From Fig. 15 (c), Fig. 15 (d), Fig. 16 (c), and Fig. 16 (d), it can be seen that the subjective quality of the synthesized view using the reconstructed depth
maps is greatly enhanced with our proposed depth coding scheme. Also with increasing QP, our coding method shows much better performance comparing with the conventional approaches. From Fig. 15 and Fig. 16, it can be seen that with the increasing QP, the edge of the depth map coded with ATM becomes obscured, while the edge coded with the proposed depth coding scheme is less affected.

We also perform an encoding complexity analysis based on the encoding time. Experimental results are illustrated in Table III. It shows that, the average coding time of the proposed depth coding scheme is less than ATM. Although some extra time are taken in the proposed DTCC and WA-OD methods, many selected depth blocks are skipped with the motion vectors of the corresponding texture video which significantly reduces the time used in the motion estimation.

In the decoder side, only the skip blocks of current row and next row need to be decoded beforehand. Therefore, the extra coding delay and memory requirements needed for the proposed depth coding scheme are not significant.

VIII. CONCLUSION

This paper presents a novel depth coding scheme based on the depth-texture similarities. The motion similarity between depth and texture is thoroughly studied, and the skip mode and its corresponding motion vectors of the texture video are first utilized for coding the depth maps directly. With these skip-coded blocks, a new type of block – omnidirectional block (OD-Block), is present in the depth map, of which prediction information may be obtained from the four immediate neighboring blocks, thus increasing the prediction gain. Accordingly, a more efficient intra coding scheme is developed for the purpose. Furthermore, in view of the structure similarity between depth and texture videos, a depth-texture cooperative clustering based prediction method is proposed to enhance the coding efficiency of the edge blocks. The proposed approach exploits the structure similarity of both the current block and its neighboring pixels to facilitate a better prediction of the current depth block. In spite of the above efforts to improve the prediction gain, some large prediction errors may still be present for the depth-texture misaligned pixels, which may greatly compromise the coding efficiency. To deal with these large residuals due to the depth-texture misalignment, a simple yet effective detection and rectification method is incorporated in the proposed depth coding scheme. Experimental results show that our proposed depth coding scheme can reduce the coding rate and improve the rendering quality compared with other existing coding approaches.

REFERENCES

[23] B. T. Oh, H. C. Wey and D. S. Park, “Plane segmentation based intra prediction for depth map coding,” In Picture Coding Symposium (PCS), May, 2012, pp. 41-44.
Jianjun Lei (M’11) received the Ph.D. degree in electronic engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2007. He is currently an Associate Professor with the School of Electronic Information Engineering, Tianjin University, Tianjin, China. He is also a visiting researcher at the Information Processing Lab, Department of Electrical Engineering, University of Washington, Seattle, WA, from August 2012. His research interests include 3D video coding, 3D display, and computer vision.

Shuai Li received the B.S. degree in electronic engineering from Yantai University, Yantai, China, in 2011. He is currently pursuing the M.S. degree at the School of Electronic Information Engineering, Tianjin University, Tianjin, China.

His research interests include 3D video coding and digital image processing.

Ce Zhu (M’03–SM’04) received the B.S. degree from Sichuan University, Chengdu, China, and the M.Eng and Ph.D. degrees from Southeast University, Nanjing, China, in 1989, 1992, and 1994, respectively, all in electronic and information engineering.

Dr. Zhu is currently with the School of Electronic Engineering, University of Electronic Science and Technology of China, China. He had been with Nanyang Technological University, Singapore, from 1998 to 2012, where he was promoted to Associate Professor in 2005. Before that, he pursued postdoctoral research at Chinese University of Hong Kong in 1995, City University of Hong Kong, and University of Melbourne, Australia, from 1996 to 1998. He has held visiting positions at Queen Mary, University of London (UK), and Nagoya University (Japan). His research interests include image/video coding, streaming and processing, 3D video, joint source-channel coding, multimedia systems and applications.

Dr. Zhu serves on the editorial boards of seven international journals, including as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON BROADCASTING, IEEE SIGNAL PROCESSING LETTERS, Editor of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and Area Editor of SIGNAL PROCESSING: IMAGE COMMUNICATION (Elsevier). He has served on technical/program committees, organizing committees and as track/area/session chairs for about 60 international conferences. He received 2010 Special Service Award from IEEE Broadcast Technology Society, and is an IEEE BTS Distinguished Lecturer (2012-2014).
Ming-Ting Sun (S’79–M’81–SM’89–F’96) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1976, and the Ph.D. degree from the University of California at Los Angeles, Los Angeles, in 1985, all in electrical engineering. He joined the University of Washington, Seattle, in 1996, where he is currently a Professor. He was the Director of the Video Signal Processing Research Group, Bellcore, Morristown, NJ.

He is a Chaired or Visiting Professor with Tsinghua University, Beijing, China, Tokyo University, Tokyo, Japan, National Taiwan University, National Cheng Kung University, Tainan, Taiwan, National Chung Cheng University, Chiayi, Taiwan, National Sun Yat-sen University, Kaohsiung, Taiwan, the Hong Kong University of Science and Technology, Kowloon, Hong Kong, and National Tsing Hua University, Hsinchu, Taiwan. He holds 13 patents and has published over 200 technical papers, including 17 book chapters in video and multimedia technologies. He is the co-editor of the book Compressed Video Over Networks. His current research interests include video and multimedia signal processing, transport of video over networks, and very large-scale integration architecture and implementation.

Dr. Sun is currently the Editor-in-Chief of JOURNAL OF VISUAL COMMUNICATIONS AND IMAGE REPRESENTATION. He was the Guest Editor of 11 special issues for various journals. He has given keynotes for several international conferences. He was a member of many prestigious award committees. He was a Technical Program Co-Chair of several conferences including International Conference on Multimedia and Expo 2010. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA and a Distinguished Lecturer of the Circuits and Systems Society from 2000 to 2001. He received the IEEE CASS Golden Jubilee Medal in 2000. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 1995 to 1997. He received the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Best Paper Award in 1993. From 1988 to 1991, he was the Chairman of the IEEE CAS Standards Committee, and established the IEEE Inverse Discrete Cosine Transform Standard. He received the Award of Excellence from Bellcore for his work on the digital subscriber line in 1987.

Chunping Hou received the M.Eng. and Ph.D. degrees, both in electronic engineering, from Tianjin University, Tianjin, China, in 1986 and 1998, respectively.

Since 1986, she has been the faculty of the School of Electronic and Information Engineering, Tianjin University, where she is currently a Full Professor and the Director of the Broadband Wireless Communications and 3D Imaging Institute.

Her current research interests include 3D image processing, 3D display, wireless communication, and the design and applications of communication systems.