Boundary Artifact Reduction in View Synthesis of 3D Video: From Perspective of Texture-Depth Alignment

Yin Zhao, Ce Zhu, *Senior Member, IEEE*, Zhenzhong Chen, *Member, IEEE*, Dong Tian, and Lu Yu

***Abstract*—3D Video (3DV) with depth-image-based view synthesis is a promising candidate of next generation broadcasting applications. However, the synthesized views in 3DV are often contaminated by annoying artifacts, particularly notably around object boundaries, due to imperfect depth maps (e.g., produced by state-of-the-art stereo matching algorithms or compressed lossily). In this paper, we first review some representative methods for boundary artifact reduction in view synthesis, and make an in-depth investigation into the underlying mechanisms of boundary artifact generation from a new perspective of texture-depth alignment in boundary regions. Three forms of texture-depth misalignment are identified as the causes for different boundary artifacts, which mainly present themselves as scattered noises on the background and object erosion on the foreground. Based on the insights gained from the analysis, we propose a novel solution of *s*uppression of *m*isalignment and *a*lignment enfo*r*cemen*t* (denoted as SMART) between texture and depth to reduce background noises and foreground erosion, respectively, among different types of boundary artifacts. The SMART is developed as a three-step pre-processing in view synthesis. Experiments on view synthesis with original and compressed texture/depth data consistently demonstrate the superior performance of the proposed method as compared with other relevant boundary artifact reduction schemes.**

***Index Terms*—Background noise, boundary artifact, foreground erosion, texture-depth alignment, view synthesis, 3D video.**

I. Introduction

In THE past two decades, many types of 3D displays [1] have been developed, among which stereoscopic displays with shutter or polarized glasses are the most accepted in the current consumer market. Conventional stereoscopic video systems present two fixed views on stereoscopic displays. However, the baseline between the two views captured by professional stereo cameras is usually wider than our pupil distance (typically 65 mm), which may lead to an unsuitably large disparity that induces severe eye strain and visual discomfort [2]. Besides, viewers may require different degrees of binocular parallax based on their preference to the intensity of 3D perception, but the stereoscopic video systems fail to meet this requirement. Therefore, 3D Video (3DV) [3] systems are proposed to address these problems. Employing view synthesis with Depth-Image-Based Rendering (DIBR) [4] technique, 3DV systems can generate arbitrary virtual views between two camera views, thus enabling baseline adjustment desirable for each viewer. On the other hand, multiview autostereoscopic displays are also stepping into the niche market including advertisement and exhibition, which simultaneously present a series of discrete views without the need of glasses in viewing. The 3DV framework appears more efficient and economical to support the multiview displays, in view that it can render multiple views based on typically one or two camera views with depth maps, which can significantly reduce bandwidth compared with the conventional scheme of transmitting all the displayed views. Owing to these benefits, 3DV has been attracting increasing interest from both academia and industry, and is in the spotlight as a promising candidate of next generation broadcasting applications.

However, 3DV systems still encounter some tough challenges. First, 3DV explicitly involves depth information of the scene which is usually obtained from automatic depth estimation (DE) based on stereo matching. Depth maps generated by state-of-the-art DE algorithms generally have poor quality at object boundaries due to occlusion of background and color mixture between foreground and background which compromise the performance of stereo matching. Moreover, temporal and inter-view inconsistency of depth data is another common problem, since most DE algorithms are image-oriented by calculating depth map of each frame individually. On the other hand, perfect depth maps are generally smooth within objects but exhibit sharp discontinuities at object boundaries. Then, lossy coding (e.g., H.264/AVC) on depth data may inevitably distort the depth structure, especially the depth edges with abundant high- frequency components. As we know, view synthesis quality highly depends on the accuracy of depth maps. Estimation errors or compression loss in depth data will yield some pixels to be projected to wrong positions in the virtual view, resulting in geometry distortions [5]. In addition, most view synthesis methods select the front-most pixels to cope with occlusions in the virtual view. Changes of depth values may also alter the occlusion order of the overlapping foreground and background objects, making the background object visible unexpectedly, which leads to ghosting artifacts. The transient but frequent occurrences of depth errors also produce annoying flickering artifacts that significantly impair

Manuscript received July 15, 2010; revised January 14, 2011; accepted January 20, 2011. Date of publication March 28, 2011; date of current version May 25, 2011. This work was supported in part by Singapore Ministry of Education Academic Research Fund Tier 1 (AcRF Tier 1 RG7/09) and the National Basic Research Program of China (973) under Grant 2009CB320903.

Y. Zhao is with the School of Electrical and Electronic Engineering, Nanyang Technological University and the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: zhaoyin@zju.edu.cn).

C. Zhu and Z. Chen are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore (e-mail: eczhu@ntu.edu.sg; zzchen@ntu.edu.sg).

D. Tian is with the Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: tian@merl.com).

L. Yu is with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: yul@zju.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/TBC.2011.2120730

visual quality of the synthesized views [6]. In view that the depth maps can hardly be perfect due to either the limitations of current DE algorithms or the lossy data compression for storage and transmission, advanced view synthesis techniques are expected to take error correction or mitigation measures to minimize or reduce synthesis artifacts for better rendering performance. In this paper, we will investigate several pre- or post-processing approaches in view synthesis to improve visual quality of boundary regions in the synthesized virtual views. We only consider techniques that directly deal with spatial distortions, and skip those suppressing flickering artifacts [7] by enforcing temporal consistency of the depth maps.

Generally, most noticeable synthesis artifacts appear at object boundaries. As discussed above, the accuracy of depth data around object boundaries is generally worse than that inside objects. Moreover, human is more sensitive to structural degradation of salient texture patterns, e.g., artifacts surrounding object boundaries or contours presenting strong visual distortions. In this sense, good quality of boundary regions is highly desirable and even critical in the synthesized images. In DIBR based view synthesis, boundary artifacts manifest themselves with two major visual effects: 1) slim silhouettes of foreground objects are scattered into the background, denoted as background noises, and 2) the foreground boundaries are eroded by background texture, termed as foreground erosion. Apart from the two major distortions, some other (minor) boundary artifacts may appear in view synthesis, e.g., the effect of unreliable transition and neighborhood misplacement, which will be discussed in the following sections. Several different methods have been proposed to reduce the boundary artifacts [8]–[12], which will be reviewed in Section II. To gain a better understanding and more insights into underlying mechanism of the boundary artifact generation in view synthesis, we make an in-depth analysis from a new prospective. Our finding is that inaccurate texture and depth information at object boundaries introduce texture-depth misalignment, which consequently yields the boundary artifacts, as to be evidenced in our analytical results. Based on the analysis from the perspective of texture-depth alignment in boundary regions, we develop a novel method to cope with the two major types of artifacts, specifically, to eliminate the background noises by refraining the unreliable boundary pixels with misaligned texture-depth values from warping in view synthesis, and to prevent foreground erosion by enforcing the foreground texture-depth alignment along object boundaries. In short, the proposed method targets at ***s***uppression of ***m***isalignment as well as ***a***lignment enfo***r***cemen***t** (coined as SMART) between texture and depth, which can be realized with a three-step pre-processing in view synthesis. Apart from addressing the major problems of background noises and foreground erosion, the proposed method is also effective to deal with the minor boundary distortions, e.g., unreliable foreground-background transition due to possible background changes in a synthesized view. Experimental results demonstrate that the proposed SMART method effectively decreases the boundary artifacts and improves visual quality of boundary regions.

The rest of the paper is organized as follows. Section II first introduces the conventional view synthesis process, and then summarizes three representative methods of boundary artifact reduction. In Section III, we explore and identify the underlying mechanisms of boundary artifact generation in view synthesis, and accordingly propose a novel solution of SMART scheme in Section IV. Section V describes the experiments on view synthesis with different methods of boundary artifact reduction for comparison, and Section VI concludes the paper.

II. Representative Methods of Boundary Artifact Reduction in View Synthesis

To better understand various methods to reduce boundary artifacts in synthesized views, we first conduct a review of conventional view synthesis process.

A. View Synthesis

View synthesis in 3DV employs Depth-Image-Based Rendering (DIBR) technology [4] to generate virtual views, which basically consists of three steps: 1) forward warp all pixels in an original view Ω (usually captured by a camera) to a virtual view Φ based on depth information, 2) merge all warped views into one image if multiple (typically two) original views are warped to the same virtual view, in which most dis-occlusion holes in one warped view are filled complementarily by pixels from the other views, and 3) fill the remaining holes in the merged image.

For simplicity and without loss of generality, we only discuss a basic and common scenario of synthesizing a virtual view from two original camera views in *1D parallel arrangement* [13], where all views are rectified to be parallel in a horizontal line. In this setup, an object keeps the same distance to each camera. Besides, the vertical disparity is zero, and the horizontal disparity can be obtained by [14]

$$d = \frac{f \times l}{z}, \tag{1}$$

where f, l, and z represent the camera focal length, baseline length between two views, and the depth value of the object, respectively. In this arrangement, pixel warping is limited within a horizontal line, and the 3D-image projection can be decomposed into simple line-based 1D pixel shifting. Treating d as a positive magnitude, pixel $\alpha(x_\Omega, y_\Omega)$ in the original view will be mapped to $\beta(x_\Phi, y_\Phi)$ in the virtual view via (2) (or (3)) when the virtual view is at the left side (or the right side) of the original view, that is, for a left virtual view

$$x_\Phi = x_\Omega + d, \quad y_\Phi = y_\Omega, \tag{2}$$

and for a right virtual view

$$x_\Phi = x_\Omega - d, \quad y_\Phi = y_\Omega. \tag{3}$$

Then, each pixel in an original view is mapped into the virtual view. If several pixels are warped to the same position, which means occlusion occurs in the virtual view, the one that is closest to the camera will be selected to occlude the other pixels at that position, known as the z-buffer method [15] or the front-most scheme [16]. After forward warping, most pixels in a warped view are determined, and the remaining pixels (holes) will be dealt with view merging and hole filling algorithms.

When two original views are warped to the same virtual view, a merging algorithm is applied to fuse pixels from different

warped views. View merging algorithms are mainly based on one of or a combination of the three strategies below:

1) Blend available pixels from two warped views with a linear weighting function [9], [16]. As a result, artifacts in the two warped views will appear in the merged view, although the distortion intensities are generally decreased by averaging all the candidates (some of which may not be distorted).
2) Take one warped view as a dominant view. Pixels from the other warped view are only used to fill the holes in the dominant view [17]. Compared with the blending scheme, this dominance-complement strategy can provide a higher-quality synthesized view if the dominant view has fewer artifacts than the complementary view.
3) Select the closest pixel based on the z-buffer method [15]. This strategy works well with perfect depth maps, but increases flickering artifacts with temporally inconsistent depth data, which encourages pixel competition between the two views that may slightly differ in color.

To handle the remaining holes in the virtual view after view merging, hole filling algorithms are employed, which are generally based on linear interpolation using neighboring pixels or more sophisticated inpainting techniques. In addition, some directional hole filling methods [18] first detect foreground and background areas around holes and fill the holes by extending the background texture, which often produces more natural boundary regions.

With imperfect depth maps, the conventional view synthesis suffers from rendering artifacts, especially at object boundaries. Generally, incorrect depth values will render the associated pixels to wrong positions in the virtual view. An example is shown in Fig. 1(a), where the pixels at the left (or right) side of the depth edge have foreground (or background) depth values. In that case, the pixels in area a (or c) are wrongly projected into the area b (or d) in the warped view due to the incorrect depth values, while the positions that the pixels are supposed to be with correct depth values will become holes (i.e., area a and c). In view merging, the hole area is usually filled by background pixels from the other view (as described in detail in Section III-C). Therefore, we can see that, on one hand, some pieces of foreground texture are scattered into the background (e.g., area b and d), causing the visual artifacts of background noises, while on the other hand, the background texture is punched into the foreground object, yielding the foreground erosion artifacts. More detailed analysis on the boundary artifact generation will be presented in Section III.

Several prior art methods have been proposed to reduce the boundary artifacts in the synthesized images. In essence, these methods attempt to detect unreliable pixels (which are prone to yield synthesis artifacts) and prevent them from being warped to a virtual view through different mechanisms. We summarize these methods into three major categories with representative algorithms, respectively, as follows.

B. Background Contour Region Replacement (BCRR) Method

On the observation that background noises usually exist on the background side of dis-occlusion holes, Lee & Ho [8] proposed a method to clean background noises in the warped views. First, contours around holes in the warped views are detected and categorized into *foreground contours* (on the foreground neighboring to the holes) and *background contours* (on the background neighboring to the holes) by simply checking the depth values around the holes, as shown in Fig. 1(b). Empirically, the background contour regions are probably spotted by some noises like fractions of the foreground object (also known as the color bleeding artifacts mentioned in [28]), whereas the corresponding areas in the other warped view are usually free from distortions or with much fewer artifacts (a brief explanation can be found in [28]). Thus, the background contour regions are intentionally replaced by more reliable pixels from the other view, and most background noises are removed [e.g., area b and e in Fig. 1(b)].

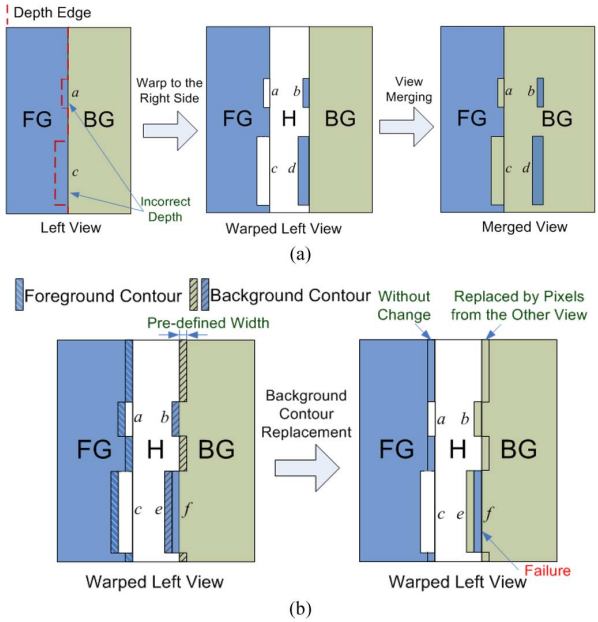

Fig. 1. (a) Example of boundary artifacts on the warped view due to incorrect depth values associated with some foreground pixels. (FG: Foreground, BG: Background, and H: Hole.) Pixels at area a and c are misaligned with background depth values. After warping, foreground area a (or c) is separated from foreground and projected to the background area b (or d) due to the incorrect background depth values, which yields background noises as well as foreground erosion. (b) Schematic of the ***b***ackground ***c***ontour ***r***egion ***r***eplacement (BCRR) method proposed by Lee and Ho [8]. Background contour region with a pre-determined width is replaced by pixels from the other view, and background noises in b and e are eliminated. Failures (e.g., area f) occur when the background noises are beyond the background contour region.

However, this BCRR method may not accurately predict the location of boundary artifacts, and then fails to clean the background noises that are distributed outside the background contour regions with a fixed width (e.g., area f). Besides, this method neglects the foreground erosion artifacts, since no processing is done around the foreground contours to make up the eroded areas. Moreover, any possible artifacts on the foreground side of the hole (e.g., on the foreground contours) will not be cleaned, as the method always believes the foreground regions are free from distortions.

C. Prioritized Multi-Layer Projection (PMLP) Scheme

Müller *et al.* [9] proposed a multi-layer projection scheme to reduce rendering artifacts. Depth edges are located by a Canny edge detector [19], and 7-sample-wide area along the edge are marked as unreliable pixels. The unreliable region is split into

a *foreground boundary layer* and a *background boundary layer*, and the rest of the image is a *base layer*. The base layer in the two original views are first warped to the virtual view and merged into a *common main layer*. The foreground boundary layers are projected with the second priority, and merged with the common main layer based on the front-most scheme. The background boundary layers are treated as the most unreliable pixels, and only used to fill the remaining holes in the merged image. The PMLP scheme is a variant of the two-layer representation by Zitnick *et al.* [10], in which the image is segmented into a main layer and a boundary layer. The boundary layer is treated differently from that in [9], which is considered as composition of foreground and background colors. A Bayesian image matting technique [20] is applied to estimate the foreground and background colors along with the opacities (alpha values). Depth values in the boundary layer are modified by alpha-weighted average of nearby foreground and background depth values. The two layers in both views are warped to the virtual view, and fused based on the opacity by a view merging algorithm involving the blending (for warped pixels in the same position with similar depth values) and z-buffer method.

Although the PMLP scheme is not explicitly targeted at dealing with specific boundary artifacts like the background noises in the BCRR, it generally reduces the usage of color-distorted pixels at boundaries owing to an empirical observation that boundary artifacts likely result from these color-mixed pixels. In fact, the background boundary layer is generally warped to the positions of the background contour defined in BCRR, if the background contour region is of the same width as the background boundary layer. Thus, the two methods essentially share the same idea of reducing unreliable pixels in warped views via either post- or pre-processing of forward warping. More specifically, background boundary pixels are treated as zero reliability in the BCRR, while they are still used occasionally in the PMLP.

D. Inter-View Cross Check (IVCC) Approach

Both the BCRR and PMLP only use depth information to check pixel reliability. Yang *et al.* [11] proposed an alternative scheme of unreliability reasoning based on inter-view cross check, in which texture and depth information are jointly utilized. Specifically, each pixel in the left (or the right) camera view is warped to the right (or the left) camera view for checking the color difference between the projected pixel and the corresponding original pixel in the other camera view. If the color difference is larger than a threshold, the pixel is determined as unreliable (e.g., pixel $1R$ in Fig. 2), because the color mismatch is probably induced by a wrong depth value; otherwise, it is a trusted pixel (e.g., pixel $1L$) with a reliable depth value. Pixels in the virtual view that result from unreliable pixels in the two original views are discarded, which means withdrawing all unreliable projections to the virtual view. Besides, conventional view blending strategy decides the weight for each view mainly based on the baseline distance between the original and the virtual views. With the reliability check, depth quality can be inferred from the wrong projections, and pixels from the more reliable view are assigned with a higher weight in view blending [12].

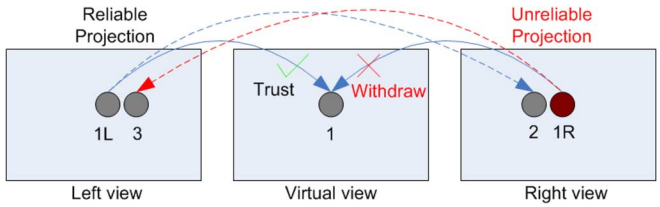

Fig. 2. Schematic of the *i*nter-*v*iew *c*ross *c*heck (IVCC) approach proposed by Yang *et al.* [11]. (Dashed lines) The cross-check processing to determine pixels with unreliable depth values. Then, projections of the unreliable pixels to the virtual view are withdrawn.

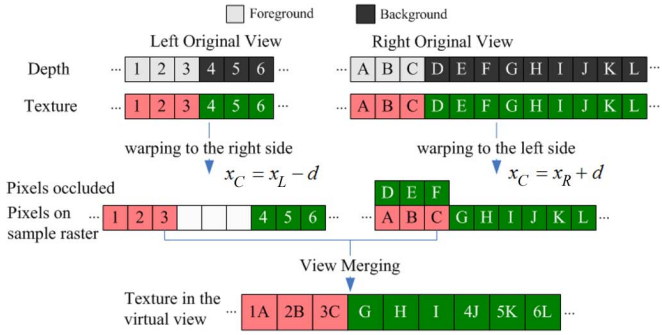

Fig. 3. View synthesis of a line along a FG-BG boundary when the texture and depth edges are well aligned, assuming that the virtual view is in the middle of the two original views and foreground disparity is equal to background disparity plus 3. "1A" means the color value after merging pixel 1 and A.

Generally, this more advanced method intelligently excludes unreliable pixels in view synthesis (not only for those around object boundaries). However, the color difference threshold in the cross check would be large enough to accommodate either illumination difference between the original views or color distortions due to video coding; otherwise, many pixels will be erroneously classified into the unreliable set owing to a too small threshold, resulting in larger holes. Thus, the IVCC method may fail to detect some weak distortions below a certain threshold. Similar to the BCRR and PMLP, foreground pixels with wrong depth values are simply skipped as well, which yields foreground erosion artifacts. In addition, at least two views are required for the cross check, and thus the method is not applicable in view synthesis with a single view input.

III. Boundary Artifact Analysis in View Synthesis

In this section, we explore and identify the fundamental generation mechanisms of boundary artifacts in view synthesis. To facilitate the analysis, we start with the synthesis of boundary regions with both perfect texture and depth, and then make a thorough investigation into the underlying causes of boundary artifacts with imperfect texture and depth, from a new perspective of texture-depth alignment. In this paper, "texture" refers to color information rather than geometric structure.

A. Boundary Synthesis With Ideal Texture and Depth

There are two kinds of object boundaries: the Foreground-Background (FG-BG) boundary where the foreground

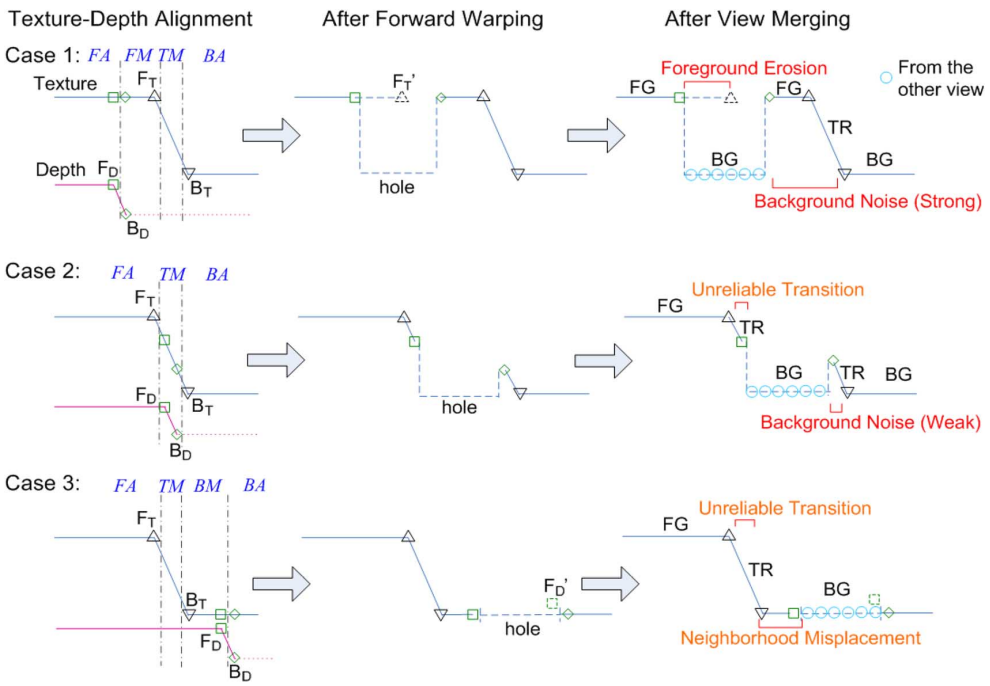

Fig. 4. Three cases of texture-depth alignment at a FG-BG boundary in the left original view and the corresponding synthesis results. (FG: Foreground, TR: Transition, BG: Background.) There are four edge points for texture and depth: Foreground Texture edge point $(\mathrm{F_T})$, Background Texture edge point $(\mathrm{B_T})$, Foreground Depth edge point $(\mathrm{F_D})$, and Background Depth edge point $(\mathrm{B_D})$. $\mathrm{F'_T}$ in Case 1 denotes the correct warped position of $\mathrm{F_T}$ if with a correct foreground depth value, whereas $\mathrm{F'_D}$ in Case 3 denotes the correct warped position of $\mathrm{F_D}$ if with a correct background depth value. Texture-depth alignment is categorized into five types of regions, Foreground Alignment (FA) region, Foreground Misalignment (FM) region, Transition Misalignment (TM) region, Background Misalignment (BM) region, and Background Alignment (BA) region.

is at the left side of the boundary, and likewise the Background-Foreground (BG-FG) boundary. For simplicity without loss of generality, we only discuss with FG-BG boundaries in this paper, and the analysis on BG-FG boundaries is similar. Fig. 3 shows an example of synthesizing a row around a FG-BG boundary when the texture and depth edges are both sharp and well-aligned. In this case, all pixels have correct depth values. Pixel 3 in the left view corresponds to pixel C in the right view, and thus they appear at the same position in the virtual view. After warping, the background in the left view (e.g., pixel 4 to 6 in Fig. 3) is separated from the foreground due to dis-occlusion, leaving a hole in between, while some background pixels in the right view (e.g., pixel D to F) are occluded by the foreground in the virtual view. As a result, the dis-occlusion holes in the warped left view are complementarily filled by the available pixels from the warped right view, and the other pixels are merged with their right-view counterparts. In this case, no background noise or foreground erosion occurs.

However, this ideal condition is rarely satisfied in real applications. A texture boundary captured by camera is usually not so sharp, having a transition area (typically 1 or 2 pixels in width) of mixed foreground and background colors. Besides, stereo matching algorithms often produce inaccurate depth values at object boundaries [21]. Moreover, lossy coding on texture and depth data will further degrade the sharpness of texture and depth boundaries. In short, an ideal object boundary is practically distorted by depth errors or compression artifacts, which compromises the alignment between texture and depth. When texture-depth misalignment takes place around boundary contours, boundary artifacts may be present in synthesized view, which will be discussed in the next subsection.

B. Causes of Boundary Artifacts: A New Perspective

Texture and depth variations at object boundaries exhibit strong coherence, where texture edges correspond to depth edges. Texture (or depth) variations around a FG-BG boundary can be simply modeled as a composition of three parts: the right margin of the foreground in the same foreground color (or depth value), the left margin of the background sharing the same background color (or depth value), and the transition area with mixed color changing from the foreground color to the background color (or with incorrect depth value changing from the foreground depth value to the background depth value), as shown in Fig. 4. For texture, the pixel at the intersection of the foreground and the transition area and that at the intersection of the background and the transition area are defined as the Foreground Texture edge point (denoted as $\mathrm{F_T}$) and the Background Texture edge point $(\mathrm{B_T})$, respectively. Similarly for depth, we have the Foreground Depth edge point $(\mathrm{F_D})$ and the Background Depth edge point $(\mathrm{B_D})$.

For an object boundary, alignment between texture and depth can be classified into three cases based on where a depth discontinuity occurs: depth discontinuity present at the foreground (Case 1), at the transition area (Case 2) or at the background (Case 3) of texture, as shown in Fig. 4. For simplicity, we mainly discuss single depth discontinuity case in this paper, where $\mathrm{B_D}$ is next to $\mathrm{F_D}$ and background depth is the same as the depth of the $\mathrm{B_D}$ pixel (shown as the purple dotted line in Fig. 4). After warping of the left original view to the central virtual view,

pixels around depth discontinuities are separated due to different disparities, leaving holes in between (a wider hole by a stronger depth discontinuity). An arbitrary depth transition can be treated as multiple discontinuities, and each discontinuity results in a hole.

Texture-depth alignment around object boundaries can also be categorized into five aligned or misaligned regions (as shown in Fig. 4): the Foreground Alignment (FA) region and the Background Alignment (BA) region, where the foreground and background texture colors are aligned with foreground and background depth values, respectively; the Foreground Misalignment (FM) region and the Background Misalignment (BM) region in which the foreground and background texture colors are associated with incorrect depth values; and the Transition Misalignment (TM) region where the mixed texture colors do not correspond to any meaningful depth values.

Assuming that the dominance-complement merging method (as discussed in Section II-A) is applied and the warped left view is the dominant view, the holes around FG-BG boundaries in the warped left view will be filled by pixels in the warped right view which probably belong to the background (as to be described in Section III-C), while the other pixels keep untouched. In Case 1 ($\mathrm{F_D} < \mathrm{F_T}$, which indicates $\mathrm{F_D}$ edge point is at the left side of $\mathrm{F_T}$ edge point), a slim slice of foreground texture pixels, located in the FM and TM regions in the original view, are separated from the foreground object and appear inside the background area in the synthesized view, yielding strong background noises. On the other hand, the pixels in the FM region are warped to wrong positions, leaving the original foreground positions as holes. After view merging with the foreground holes filled by background pixels from the other view, the foreground margin appears to be eroded by the background texture, and therefore the foreground erosion artifacts are present. In Case 2 ($\mathrm{F_T} \le \mathrm{F_D} < \mathrm{B_T}$), though the part of texture transition pixels in the TM region is closer to the background in color, perceptually they still appear to be part of the foreground object, especially when the split transition pixels together form a dim curve parallel to the true object boundary. In Case 3, when depth discontinuity occurs at the background ($\mathrm{F_D} \ge \mathrm{B_T}$), all foreground and transition pixels will keep tight with the foreground object, without contaminating the background. For other view merging strategies, the synthesis results are similar, although the artifact intensities may alter.

A non-uniform complex background may cause some other artifacts. Texture transition is introduced by camera aliasing or coding, which can be regarded as a result of low-pass filtering on the color edge between $\mathrm{F_T}$ and $\mathrm{B_T}$. If the $\mathrm{B_T}$ pixels in the original and the virtual views are distinct in color, the color transitions in the two views could be quite different. Thus, in Case 2 and 3, simply copying the transition area in the original view to the virtual view (i.e., warping the pixels in the TM region) may produce an unreliable color transition. Moreover, in Case 3, the pixels between $\mathrm{B_T}$ and $\mathrm{F_D}$ (both inclusive) in the BM region are warped to the neighborhood around the foreground in the virtual view, but in fact they should be warped to the positions next to $\mathrm{B_D}$, e.g., $\mathrm{F_D}$ should be warped to $\mathrm{F'_D}$ if with a correct background depth value, as shown in Fig. 4. In other words, the foreground neighborhood should be occupied by the dis-occluded background pixels instead of the neighboring background pixels in the original view. Thus, the pixels in the BM region, which are actually not adjacent to the foreground in the virtual view, are misplaced in the foreground neighborhood. When pixels around $\mathrm{F_D}$ are different from those around $\mathrm{F'_D}$ in color, the foreground margin in the synthesized view will be wrapped by a stripe of fake background texture which appears rugged or inconsistent with the adjacent background. However, these artifacts of unreliable transition and neighborhood misplacement seldom appear, since the background is usually quite uniform.

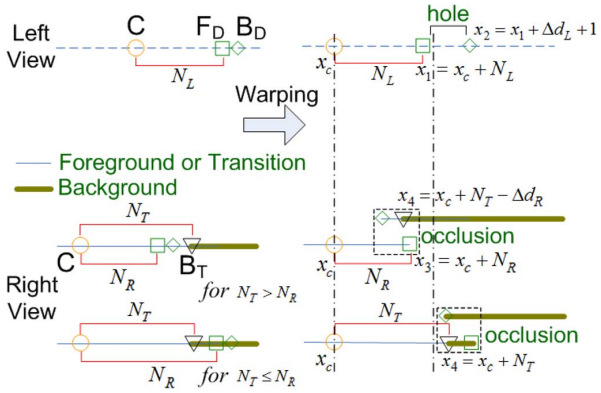

Fig. 5. Positions of the edge points before and after forward warping. For the thick solid lines, the pixels are in background color, and for the thin solid lines, the pixels are in foreground or transition color. Pixel C denotes the foreground object center. Pixel C in the left view and that in the right view are warped to the same position x_c in the virtual view.

In summary, in the FG-BG boundary setting, the pixels on the left of both $\mathrm{F_T}$ and $\mathrm{F_D}$ (inclusive of the leftmost of $\mathrm{F_T}$ and $\mathrm{F_D}$) fall in the FA region, while the pixels on the right from the rightmost of $\mathrm{B_D}$ and $\mathrm{B_T}$ onwards are in the *BA* region. Pixels in both the regions correspond to well aligned texture-depth values, and thus will not cause any artifacts in the synthesized view. However, the texture and depth of the other pixels falling beyond the two regions are considered to be misaligned in three ways, which are the underlying causes of different boundary artifacts generated in view synthesis. The FM region yields foreground erosion and background noises; the TM region produces background noises as well as unreliable transition; and the BM region introduces neighborhood misplacement.

C. Dis-Occlusion Filling in View Merging

In this subsection, we consider the complementary pixels from the other view that fill the dis-occlusion holes. The discussion is still based on the setting of a FG-BG boundary, and the same notations of the edge points $\mathrm{F_D}$, $\mathrm{B_D}$ and $\mathrm{B_T}$ are applied in the following.

Theorem 1: Dis-occlusion holes in the warped left view are mostly likely (under some mostly valid and practically feasible conditions) filled by background pixels from the warped right view.

Proof: First, we discuss the case of single depth discontinuity. We reasonably assume (as illustrated in Fig. 5):

1) The center of the foreground object (denoted as pixel C) in the left view corresponds to the counterpart in the right

view so that they are warped to same position x_c in the virtual view.

2) $\mathrm{F_D}$ in the left (or the right) view is N_L (or N_R) ($N_L > 0, N_R > 0$) pixels away from the object center (i.e., pixel C).
3) $\mathrm{B_T}$ in the right view is N_T ($N_T > 0$) away from pixel C.
4) The difference between the foreground and background disparities in the virtual view from the left view (or the right view) is Δd_L (or Δd_R) ($\Delta d_L > 0, \Delta d_R > 0$).

Then, we denote that $\mathrm{F_D}$ and $\mathrm{B_D}$ in the left view are warped to x_1 and x_2 in the virtual view, while $\mathrm{F_D}$ and $\mathrm{B_T}$ in the right view are warped to x_3 and x_4 in the virtual view, as shown in Fig. 5. As both $\mathrm{F_D}$ and pixel C have the foreground depth value, they are warped with the same disparity (the distance between them remains the same after warping), and we have

$$x_1 = x_c + N_L, \tag{4}$$

$$x_3 = x_c + N_R. \tag{5}$$

Based on the 1D pixel mapping function in (3), we can obtain the warped position of the left-view $\mathrm{B_D}$, that is,

$$x_2 = x_1 + \Delta d_L + 1. \tag{6}$$

For the right view, when $\mathrm{F_D}$ is at the left side of $\mathrm{B_T}$ (i.e., $N_T > N_R$), which represents Case 1 or 2 in Fig. 4, the right-view $\mathrm{B_T}$ corresponds to a background depth value (since it is located on the right from $\mathrm{B_D}$ onwards), and is warped via 1D pixel mapping function in (2) to

$$x_4 = x_c + N_T - \Delta d_R, \quad if\ N_T > N_R. \tag{7}$$

In view of occlusion, the first pixel (denoted as x_B) of background color from left to right in the warped right view is

$$x_B = \max(x_3 + 1, x_4), \quad if\ N_T > N_R, \tag{8}$$

where $\max(A, B)$ takes the larger value of A and B.Similarly, in the case of $N_T \leq N_R$, which represents Case 3 in Fig. 4, the right-view $\mathrm{B_T}$ is associated with a foreground depth value (as it is on the left from $\mathrm{F_D}$ onwards), and is warped to

$$x_4 = x_c + N_T, \quad if\ N_T \leq N_R. \tag{9}$$

Then, the first background pixel in the warped right view appears at

$$x_B = x_4, \quad if\ N_T \leq N_R. \tag{10}$$

By merging the warped left and right views, we can see that the entire dis-occlusion hole (between $\mathrm{F_D}$ and $\mathrm{B_D}$ in the warped left view) is filled by background pixels from the warped right view, if

$$x_1 + 1 \geq x_B. \tag{11}$$

By substituting (4), (5) and (7)–(10) into (11) with further simplifications, we have

$$\begin{cases} N_L \geq N_R \,\&\, \Delta d_R + 1 \geq N_T - N_L, & if\ N_T > N_R \\ N_L + 1 \geq N_T, & if\ N_T \leq N_R. \end{cases} \tag{12}$$

Since depth maps are generated by the same depth estimation algorithm, the depth edge locations in the left and right views should be the same or very close in case of some estimation errors, and then we have $N_L \approx N_R$, which is also treated as a reasonable assumption in [11]. Additionally, depth edge is required to be close to texture edge, e.g., $\mathrm{B_D}$ ($= \mathrm{F_D} + 1$) should be aligned with $\mathrm{B_T}$ as in the ideal case in Fig. 3. Then, we have $N_T \approx N_L + 1 \approx N_R + 1$. When $N_T = N_L + 1 = N_R + 1$, all the three constraints in (12) are satisfied. If N_L differs from N_R in case of depth errors, $N_L \geq N_R$ still has a high probability (at least 50%) to be valid. Similar conclusion can be drawn for $N_L + 1 \geq N_T$. Furthermore, an object boundary usually corresponds to a relatively large foreground-background disparity difference (otherwise, the area is a smooth depth region), and thus $\Delta d_R + 1 \geq N_T - N_L$ is most probably valid for good quality of depth maps. Therefore, we can see that (12) is more likely to be true, which means the entire dis-occlusion hole is filled by background pixels from the other view.

On the other hand, at least one background pixel appears in the dis-occlusion hole, if

$$x_2 > x_B. \tag{13}$$

After substituting (4) to (10) into (13) and further simplifying the inequality, we have

$$\begin{cases} \Delta d_L > N_R - N_L \\ \quad \& \, \Delta d_L + \Delta d_R + 1 > N_T - N_L, & if\ N_T > N_R \\ \Delta d_L + 1 > N_T - N_L, & if\ N_T \leq N_R. \end{cases} \tag{14}$$

According to the above analysis, all the constraints in (14), $\Delta d_L > N_R - N_L$, $\Delta d_L + \Delta d_R + 1 > N_T - N_L$, and $\Delta d_L + 1 > N_T - N_L$, are valid with a high probability (close to 1). Thus, the inequality (14) usually holds except for some rare extreme conditions that the two views have substantially different depth edges or the texture edges are located far away from the depth edges, which means the depth maps have very poor boundary quality.

In the case of multiple depth discontinuities, if at least one background pixel appears in the first hole introduced by the first depth discontinuity, all the other holes will be filled with background pixels. Therefore, it is highly probable that dis-occlusion holes in the warped left view are filled with background pixels from the right view. *Proof completed.*

IV. SMART: Proposed Method for Reducing Boundary Artifacts

A. Suppression of Misalignment & Alignment Enforcement (SMART) between Texture and Depth

Based on the analysis in Section III-B with misalignment cases shown in Fig. 4, we understand and summarize four types of boundary artifacts caused by three forms of texture-depth misalignment. Therefore, we can naturally consider either suppressing misalignment or enforcing alignment for boundary artifact reduction. To minimize the foreground erosion and background noises as well as some other distortions, we consider: 1) to enforce the foreground texture-depth alignment in the FM region by modifying the depth values of the foreground pixels in the region as the foreground depth value, which is to prevent the

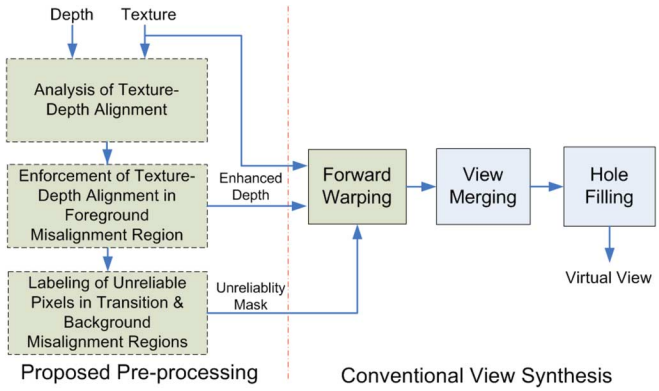

Fig. 6. Framework of the proposed pre-processing method.

foreground erosion as well as the scattered background noises; and 2) to refrain the usage of unreliable pixels in the TM and BM regions to reduce background noises as well as the unreliable transition and neighborhood misplacement.

Accordingly, such a scheme coined as ***s***uppression of ***m***isalignment and ***a***lignment enfo***r***cemen***t*** (SMART) is incorporated as a pre-processing into view synthesis to reduce boundary artifacts, which specifically consists of three major steps, as shown in Fig. 6.

Step 1: Analyze alignment between texture and depth at object boundaries, and locate the four edge points ($\mathrm{F_T}$, $\mathrm{B_T}$, $\mathrm{F_D}$ and $\mathrm{B_D}$) for each object boundary. This is the basis of the following two steps, and a simple yet effective edge point location algorithm is presented in Section IV-B.

Step 2: Enforce the alignment between foreground texture and depth in each view, i.e., if $\mathrm{F_D}$ is at left-side of $\mathrm{F_T}$ (i.e., Case 1), re-align $\mathrm{F_D}$ to $\mathrm{F_T}$ by modifying the depth values of pixels in the foreground misalignment region to the depth value of $\mathrm{F_D}$. This step aims to minimize foreground erosion by using as many foreground pixels as possible based on the location of foreground texture edge point $\mathrm{F_T}$.

Step 3: Suppress forward warping of unreliable pixels in transition and background misalignment regions. For the transition misalignment region between $\mathrm{F_T}$ and $\mathrm{B_T}$ (both exclusive), the suppression is to prevent weak background noises as well as the unreliable transition (e.g., in Case 2 & 3) due to a possible background change in the synthesized view, while the prevention of the pixels in the background misalignment region (from $\mathrm{B_T}$ to $\mathrm{B_D}$, if $\mathrm{B_D}$>$\mathrm{B_T}$) from forward warping is to eliminate the neighborhood misplacement (e.g., in Case 3). An unreliable pixel mask is generated for assisting the process of forward warping to avoid the projections of those labeled unreliable pixels.

It can be seen that the three-step SMART technique attempts to keep all unreliable pixels from contaminating the warped views (suppressing misalignment) as well as to correct some depth errors on the foreground (enforcing alignment). Fig. 7 shows an example of how the SMART scheme works to remove boundary artifacts in Case 1 in Fig. 4. We can see that the depth values in the foreground misalignment region $(\mathrm{F_D}, \mathrm{F_T}]$ are corrected as a consistent foreground depth value to "rescue" the misaligned foreground pixels, thus eliminating or minimizing foreground erosion as well as avoiding background noises. The unreliable pixels in the transition misalignment region between $\mathrm{F_T}$ and $\mathrm{B_T}$ are forbidden in warping, leaving additional holes which are expected to be filled with background pixels from the other view. In this way, the proposed method produces a sharp edge without the above-mentioned artifacts. Similarly, this method is also effective to reduce the boundary artifacts in Case 2 and 3 indicated in Fig. 4. For view synthesis with single view input, directional hole filling can be employed to make up the holes with background texture, as shown in Fig. 7.

As a summary, Table I is presented to compare the effectiveness in reducing different boundary artifacts of the proposed SMART scheme against the three representative approaches discussed in Section II. BCRR [8] and PMLP [9] may neither solve background noises in Case 1 if the misalignment region is wider than the pre-defined unreliable regions (e.g., the background contour [8] or the background boundary layer [9]), nor deal with the unreliable transition or neighborhood misplacement in Case 2 and 3 as both schemes always trust warped pixels at the foreground side. IVCC [12] can detect and remove most boundary artifacts in the three cases but still fail to identify some weak ones that are below the color difference threshold in the cross check, which may likely occur especially for the weak background noises in Case 2. Moreover, none of the three can tackle the foreground erosion in Case 1. In contrast, the proposed SMART is developed to deal with all the boundary artifacts more thoroughly based on our analysis of boundary artifact generation from a new perspective of texture-depth alignment.

B. Edge Point Location

In the proposed method, step 2 and 3 are based on the results of step 1 which locates the edge points for both texture and depth. Here we consider a simple yet effective approach to locate the four types of edge points ($\mathrm{F_D}$, $\mathrm{B_D}$, $\mathrm{F_T}$ and $\mathrm{B_T}$), and more sophisticated schemes could be further incorporated in the location process.

First, a Canny edge detector (the high and low thresholds are empirically determined as 0.2 and 0.08 of the highest gradient magnitude of the image, respectively) is employed to determine the major depth and texture edges, producing continuous and smooth 1-pixel-wide edge curves near the center of the depth and texture transition regions. Object boundaries are located at depth edges with disparity jumps of over T_D pixels. Here, we reasonably assume that in a viewing test viewers are not sensitive to the background noise that is split less than $\theta = 0.1^\circ$ visual angle away from the object, in view that visual contrast sensitivity drops quickly for patterns of spatial frequencies f_S over 10 cycle per degree (cpd). Thus, the threshold T_D is determined by [as illustrated in Fig. 8(a)]:

$$T_D = 2 \cdot \tan(\theta/2) \cdot VD, \tag{15}$$

where the viewing distance VD is assumed to be three times of image height (which is used in subjective viewing tests in MPEG-3DV). For instance, $T_D \approx 4$ for a 1024×768 image.

Then, the four types of edge points are detected by line-based gradient checking in an M × N search window centered at the depth edge, as shown in Fig. 8(b). For simplicity, only the setting

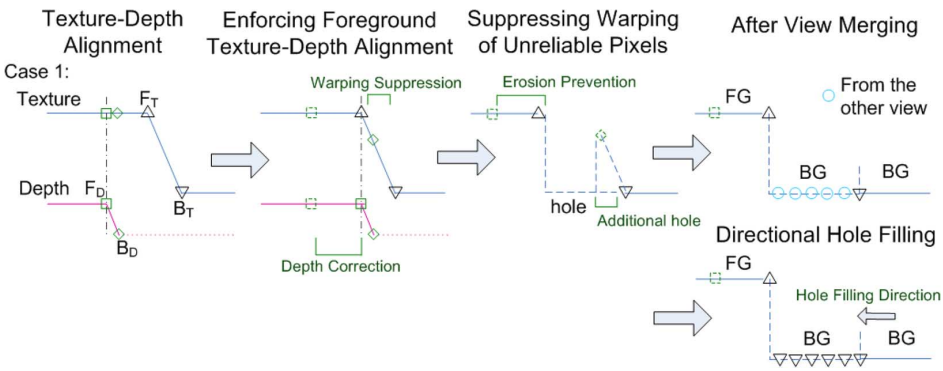

Fig. 7. Example of reducing boundary artifacts (in Case 1) by the proposed method (FG: Foreground and BG: Background).

TABLE I
PERFORMANCE COMPARISONS AMONG DIFFERENT METHODS IN TERMS OF REDUCING THE FOUR TYPES OF BOUNDARY ARTIFACTS

Boundary Artifacts	Methods			
	BCRR [8]	PMLP [9]	IVCC [12]	SMART
Foreground Erosion	invalid	invalid	invalid	valid
Background Noises	partly	partly	partly	valid
Unreliable Transition	invalid	invalid	valid	valid
Neighborhood Misplacement	invalid	invalid	valid	valid

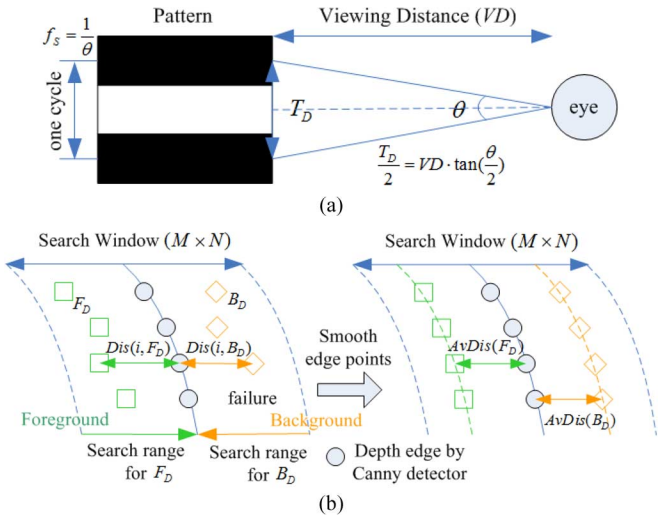

Fig. 8. (a) Illustration of deriving depth discontinuity threshold T_D from visual angle θ. (b) Line-based detection of depth edge points (F_D *and* B_D) in the search window centered at the depth edge by Canny detector and smoothing of the edge points. The edge points unable to be detected by the gradient-based scheme are estimated after the smoothing processing.

of FG-BG boundary is discussed. Here, we define left derivative ($f'_-(x)$) and right derivative ($f'_+(x)$) of a function $f(x)$:

$$f'_-(x) = (f(x) - f(x-2))/2, \tag{16}$$

$$f'_+(x) = (f(x+2) - f(x))/2. \tag{17}$$

To locate depth edge points in each line within the defined search window, the first pixel from *left* (foreground-side) in the line to the Canny depth edge satisfying $|d'_+(x)| \geq Th_d$ and $|d'_-(x)| < Th_d$ is considered as $\mathrm{F_D}$, and the first pixel from *right* (background-side) of the line to the depth edge with $|d'_-(x)| \geq Th_d$ and $|d'_+(x)| < Th_d$ is $\mathrm{B_D}$, where $d(x)$ refers to the disparity value of pixel x ($x-2$ denoting 2-pixel to the left of x) and Th_d is $T_D/2$. Likewise, for locating texture edge points in each line within the search window, the first pixel from *left* in the line to the Canny texture edge (the closest to the Canny depth edge in case of multiple texture edges within the search window) satisfying $|t'_+(x)| \geq Th_{ft}$ and $|t'_-(x)| < Th_{ft}$ is labeled as $\mathrm{F_T}$, while the first pixel from *right* of the line to the texture edge with $|t'_-(x)| \geq Th_{bt}$ and $|t'_+(x)| < Th_{bt}$ is $\mathrm{B_T}$, where $t(x)$ refers to the luminance of pixel x. Empirically, Th_{ft} is determined as 0.25 of edge height, while Th_{bt} is a small value of 6, below which the color difference (possibly resulting in background noises as shown in Fig. 4) is almost invisible. The line-based edge point detection for a BG-FG boundary can be performed similarly.

However, the independent detection of edge points in each line may produce jagged edge points, or some edge points in a few lines may be missed, as illustrated in Fig. 8(b). Thus, positions of the detected edge points are further rectified to produce a smooth curve parallel to the Canny depth or texture edge. Taking $\mathrm{F_D}$ for example, we average the distances between the $\mathrm{F_D}$ edge points and the Canny depth edge points in all lines in the window (denoted as $Dis(i, F_D)$, $i = 1, \ldots, M$), i.e.,

$$AvDis(F_D) = \frac{\sum_{i=1}^{M} Dis(i, F_D) \cdot a(i)}{\sum_{i=1}^{M} a(i)},$$

$$a(i) = \begin{cases} 1, & if\ F_D \text{ detected} \\ 0, & otherwise. \end{cases} \tag{18}$$

All rectified $\mathrm{F_D}$ points are of the distance of $AvDis(F_D)$ to the Canny depth edge, as shown in Fig. 8(b).

V. EXPERIMENTAL RESULTS

We apply the SMART technique in the 1D mode of MPEG view synthesis reference software (VSRS) [22] which follows the conventional view synthesis. The existing representatives of BCRR [8] and IVCC [12] are also tested for comparison. Intermediate virtual views are synthesized with two original views, and the warped left and right views are merged with the blending algorithm as discussed in Section II-A. The test data includes: 1) Middlebury data set [23]: "Cones" (450 × 375), "Teddy" (450 × 375), and "Art" (695 × 555); and 2) MPEG-3DV test sequences: "Mobile" (720 × 540) [24] and "Book_arrival" [25]

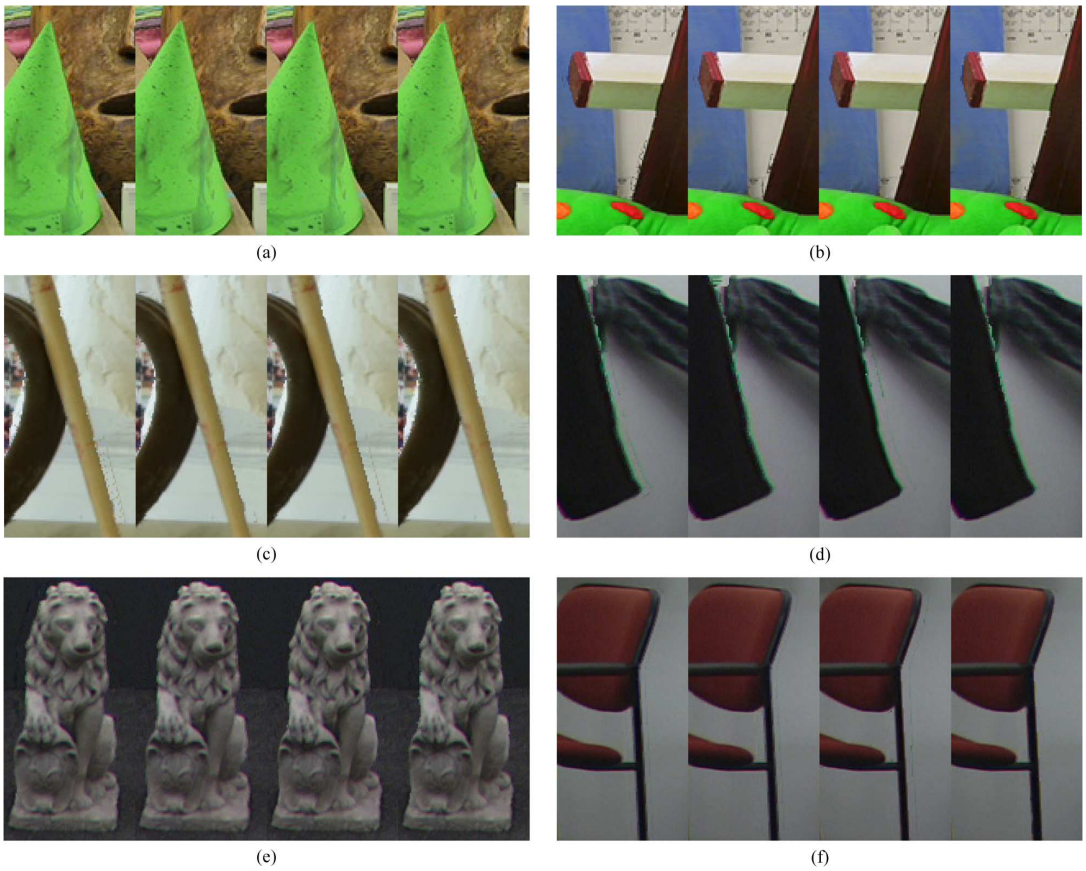

Fig. 9. Samples of synthesized images, (a) "Cones" (view 4 synthesized from views 3 to 5), (b) "Teddy" (view 4 synthesized from views 3 to 5), (c) "Art" (view 3 synthesized from views 2 to 4), (d)–(f) the 1st frame of "Book_arrival" (view 9 synthesized from views 8 to 10). From left to right: 1) VSRS 1D mode, 2) BCRR by Lee and Ho [8], 3) IVCC by Yang *et al.* [12], and 4) the proposed SMART.

(1024×768). The depth maps of the test images are generated with the MPEG depth estimation reference software (DERS) [26] based on graph cuts [27] except for "Mobile" which contains both synthetic texture and depth sequences for each view. The depth maps of "Book_arrival" are generated semi-automatically with some inputs from MPEG experts. Samples of the synthesized images with the three boundary artifact reduction schemes are shown in Figs. 9 and 10.

For the conventional view synthesis with imperfect depth maps, some boundary artifacts are notable. BCRR reduces thin background noises on the background as expected, e.g., in (c)–(e). IVCC works well in cleaning strong background noises, e.g., in Fig. 9(a) and (b), but fails to handle artifacts that are close to the background color, e.g., in Fig. 9(c)–(f).

The proposed method can effectively suppress or eliminate both strong and weak background noises, as can be seen from the figures, which is mainly attributed to the fact that the SMART technique excludes each unreliable pixel region properly from the perspective of local texture-depth alignment. Compared with BCRR [8] and PMLP [9] in which the unreliable regions are uniformly identified with a pre-defined and fixed width, the proposed method can be regarded as an adaptive and enhanced variant. In addition, the proposed method also removes the foreground erosion artifacts, e.g., along the right part of the painting pen shown in Fig. 9(c). Moreover, with the suppression of the color-mixed transition areas in view synthesis, the rendered boundaries present higher contrast and sharpness, which helps to improve subjective visual quality, as shown in Fig. 9(d) and (f).

Since lossy coding on texture and depth data may lead to complex texture-depth misalignment, we further test the three methods in view synthesis with compressed "Mobile" sequences coded by JMVC 6.0 (with default settings). The texture and depth data are coded with the same QP, where three QP values (26, 34 and 42) are selected. The samples of the synthesized 1st frame are shown in Fig. 10(b) and (c). Coding on texture and depth data aggravates boundary artifacts, yielding many irregular fractions of noises in the background and rough edges along the foreground objects. For the "Mobile" sequence, both the proposed SMART and IVCC outperform BCRR, and SMART renders the subjectively best quality boundary regions with the compressed data.

Objective quality assessment for synthesized views remains an open question. The synthesized images visually resemble the original images, whereas they probably misalign or mismatch with the original images in geometry or color [28], which substantially interferes the performance of classical full-reference metric like Peak Signal to Noise Ratio (PSNR) [29], [30]. Here, we use the Structural Similarity (SSIM) [31] metric, a state-of-the-art image quality indicator (usually exhibiting higher correlation to subjective evaluation than PSNR), to measure the overall visual quality of the synthesized images.

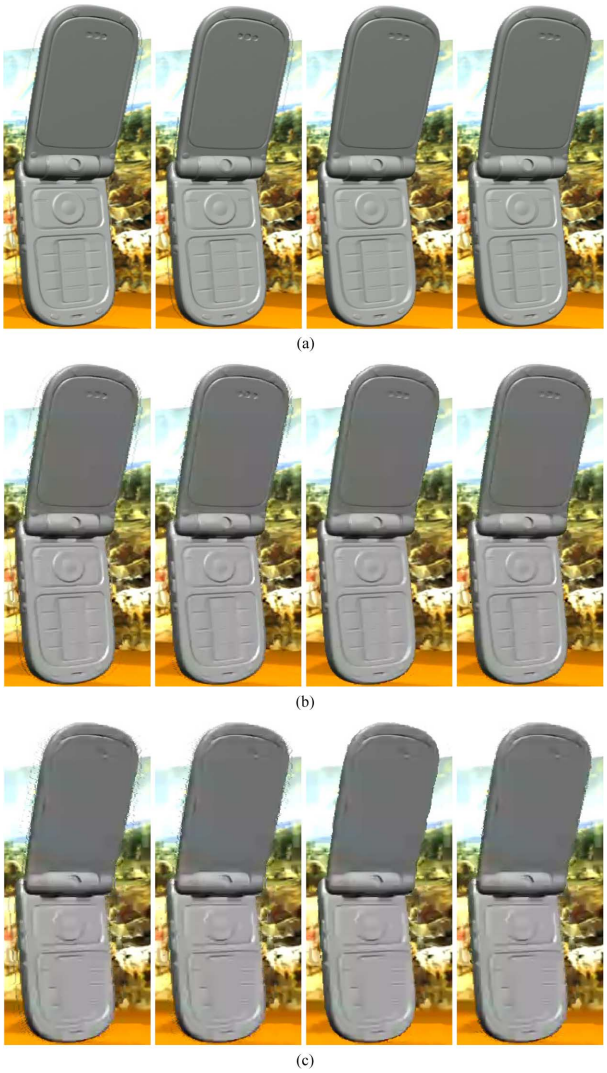

Fig. 10. Samples of the first frame of synthesized "Mobile" (view 5 synthesized from views 4 to 6) with (a) original (uncompressed), (b) compressed ($\mathrm{QP} = 34$), and (c) compressed ($\mathrm{QP} = 42$) texture and depth data. From left to right: 1) VSRS 1D mode, 2) BCRR by Lee and Ho [8], 3) IVCC by Yang *et al.* [12], and 4) the proposed SMART.

TABLE II
SSIM SCORES OF THE SYNTHESIZED IMAGES. "N" DENOTES THE WIDTH OF THE M × N SEARCH WINDOW FOR EDGE POINTS. THE HEIGHT OF THE WINDOW IS UNIFORMLY SET TO BE 3 (I.E., $\mathrm{M} = 3$)

Images	N	QP	Methods			
			VSRS	BCRR[8]	IVCC[12]	SMART
Art	5	/	0.9797	0.9802	0.9796	0.9803
Cones	5	/	0.9648	0.9637	0.9647	0.9642
Teddy	5	/	0.9795	0.9796	0.9805	0.9797
Book_arrival	7	/	0.9232	0.9200	0.9229	0.9230
Mobile	7	/	0.9871	0.9885	0.9852	0.9891
		26	0.9702	0.9718	0.9694	0.9728
		34	0.9351	0.9371	0.9349	0.9385
		42	0.8572	0.8591	0.8598	0.8618

SSIM scores between the synthesized images and the original images (captured by real cameras placed at the same position as the virtual viewpoints) are listed in Table II. Due to the boundary artifact reduction with the proposed method, the synthesized images produced by SMART generally shows higher SSIM scores than the benchmark generated by VSRS 1D mode, especially for the "Mobile" sequences with more visual distortions. For "Cones" and "Book_arrival", the proposed method drops the SSIM score slightly (the same applies for BCRR and IVCC). However, compared with the benchmark, the proposed method still improves some boundary regions while maintaining the same visual quality at the other areas (e.g., the synthesized "Cones" is shown in Fig. 11). The slight loss may be because the proposed method produces more holes in the merged image with the exclusion of unreliable pixels. When the holes correspond to complex texture patterns, the interpolation-based hole filling method cannot well recover the missing color information numerically from neighboring pixels, although the interpolated areas are harmonious with the surrounding pixels visually. Nevertheless, most full-reference quality metrics (including the SSIM metric) treat any waveform difference as quality degradation, and thus decrease the rating in this case.

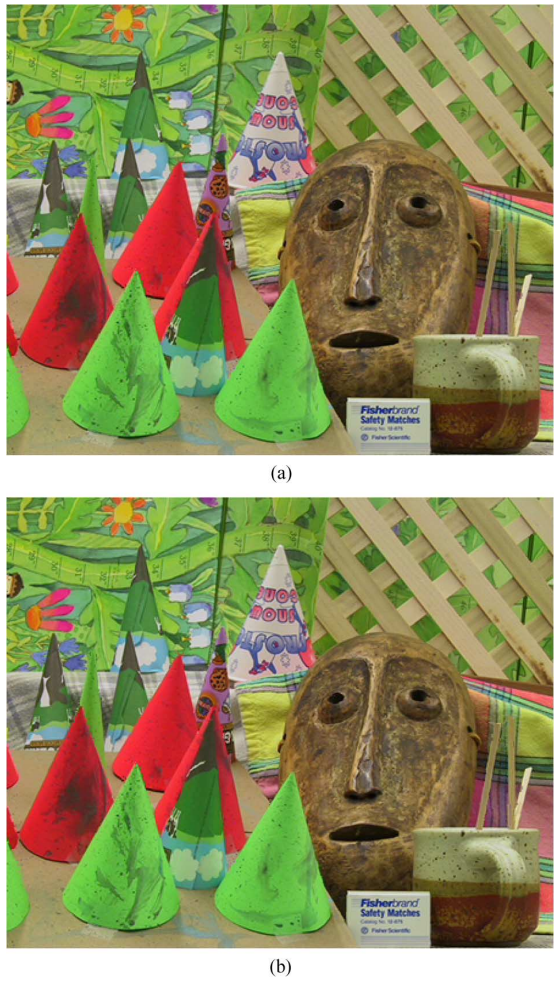

Fig. 11. Synthesized "Cones" (view 4 synthesized from views 3 to 5). (a) VSRS 1D mode, and (b) the proposed SMART.

VI. CONCLUSIONS AND DISCUSSIONS

In this paper, we discuss boundary artifacts in view synthesis of 3D video, and look into three representative methods of

reducing boundary artifacts. We then make an in-depth investigation into the underlying causes of the boundary artifacts from a new perspective of texture and depth alignment around object boundaries, in which we consider four types of boundary artifacts: foreground erosion and background noises which are dominant and occur frequently, as well as unreliable transition and neighborhood misplacement that seldom appear. Three forms of texture-depth misalignment in foreground, transition and background regions have been identified as the causes for the four types of boundary artifacts, respectively.

Accordingly, we propose a novel and effective method to remove boundary artifacts by two means: 1) reduce foreground erosion artifacts by enforcing foreground texture and depth alignment, and 2) mitigate the other three types of artifacts by suppressing misalignments in transition and background regions. Basically, the proposed SMART method mainly exploits the inherent coherence between texture and depth variations along object boundaries to predict and reduce boundary artifacts. Experimental results on view synthesis with original and compressed texture and depth data validate the effectiveness of SMART in visual quality improvement along boundary regions.

It can be seen that SMART demands a robust algorithm to locate the edge points, which is not a major focus in this paper. We therefore adopt a simple yet effective edge location algorithm as described in Section IV-B. However, some jagged cuts along edges are still present in a few cases. More advanced edge point detection together with smoothing of the synthesized object boundaries can be expected to further improve the visual quality of the rendered images. In addition, it may also be worthy of investigating how to more effectively handle the unreliable pixels instead of simply discarding them to enhance the boundary quality in view synthesis.

Visual quality of a synthesized view is critical in 3DV systems. However, current 3DV techniques like depth estimation and data compression inevitably introduce complex texture-depth misalignment, thus yielding annoying boundary artifacts in turn. In order to produce more pleasant synthesized views, we expect the concept of texture-depth alignment presented in this paper to be further incorporated into other processing techniques in 3DV systems.

ACKNOWLEDGMENT

The authors would like to thank Middlebury College, Fraunhofer Institute for Telecommunications Heinrich Hertz Institute (HHI) and Philips for kindly providing the multi-view images, "Book_arrival" and "Mobile" sequences. They are grateful to the three anonymous reviewers for their valuable comments and suggestions, which improved quality of the paper.

REFERENCES

[1] P. Benzie, J. Watson, P. Surman, I. Rakkolainen, K. Hopf, H. Urey, V. Sainov, and C. von Kopylow, "A survey of 3DTV displays: techniques and technologies," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1647–1658, Nov. 2007.

[2] P. Seuntiens, L. Meesters, and W. Ijsselsteijn, "Perceived quality of compressed stereoscopic images: effects of symmetric and asymmetric JPEG coding and camera separation," *ACM Trans. Appl. Perception*, vol. 3, no. 2, pp. 95–109, 2006.

[3] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multiview imaging and 3DTV," *IEEE Signal Process. Mag.*, vol. 24, no. 6, pp. 10–21, Nov. 2007.

[4] C. Fehn, "A 3D-TV approach using depth-image-based rendering (DIBR)," in *Proc. Visu., Imaging Image Process. (VIIP)*, 2003, pp. 482–487.

[5] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. H. N. de With, and T. Wiegand, "The effect of depth compression on multiview rendering quality," *Signal Process.: Image Commun.*, vol. 24, no. 1/2, pp. 73–88, Jan. 2009.

[6] Y. Zhao and L. Yu, "A perceptual metric for evaluating quality of synthesized sequences in 3DV system," in *Proc. Vis. Commun. Image Process. (VCIP)*, Huangshan, China, Jul. 2010.

[7] D. Fu, Y. Zhao, and L. Yu, "Temporal consistency enhancement on depth sequences," in *Picture Coding Symp. (PCS)*, Nagoya, Japan, Dec. 2010, pp. 342–345.

[8] C. Lee and Y. S. Ho, "Boundary filtering on synthesized views of 3D video," in *Int. Conf. Future Gen. Commun. Netw. Symp.*, Sanya, China, 2008, pp. 15–18.

[9] K. Müller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View synthesis for advanced 3D video systems," *EURASIP J. Image Video Process.*, vol. 2008, Article ID 438148.

[10] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. ACM SIGGRAPH*, 2004, pp. 600–608.

[11] L. Yang, T. Yendoa, M. P. Tehrania, T. Fujii, and M. Tanimoto, "Artifact reduction using reliability reasoning for image generation of FTV," *J. Vis. Commun. Image Represent.*, vol. 21, pp. 542–560, Jul./Aug. 2010.

[12] L. Yang, T. Yendo, M. P. Tehrani, T. Fujii, and M. Tanimoto, "Error suppression in view synthesis using reliability reasoning for FTV," in *3DTV Conf. (3DTV-CON)*, Tampere, Finland, 2010.

[13] MPEG Video Group, "Call for contributions on 3D video test material (update)," ISO/IEC JTC1/SC29/WG11 Doc. N9595, Jan. 2008.

[14] L. Zhang and W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV," *IEEE Trans. Broadcast.*, vol. 51, no. 2, pp. 191–199, 2005.

[15] D. Tian, P. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," in *Proc. SPIE 7443*, 2009.

[16] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Process.: Image Comm.*, vol. 24, no. 1/2, pp. 65–72, Jan. 2009.

[17] M. Domañski, M. Gotfryd, and K. Wegner, "View synthesis for multiview video transmission," in *Int. Conf. Image Process., Comput. Vis., Pattern Recog.*, Las Vegas, USA, Jul. 2009, pp. 13–16.

[18] K. Oh, S. Yea, and Y. Ho, "Hole-filling method using depth based in-painting for view synthesis in free viewpoint television (FTV) and 3D video," in *Picture Coding Symp. (PCS)*, Chicago, IL, 2009, pp. 233–236.

[19] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[20] Y. Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, "A Bayesian Approach to Digital Matting," in *Proc. IEEE Conf. Comput. Visi. Pattern Recog.(CVPR)*, 2001, vol. 2, pp. 264–271.

[21] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1–3, pp. 7–42, Apr. 2002.

[22] M. Tanimoto, T. Fujii, and K. Suzuki, 'View synthesis algorithm in view synthesis reference software 2.0 (VSRS2.0)," Lausanne, Switzerland, ISO/IEC JTC1/SC29/WG11 Doc. M16090, Feb. 2009.

[23] "Middlebury Stereo Vision page," 2007 [Online]. Available: http://vision.middlebury.edu/stereo/

[24] F. Bruls, R. K. Gunnewiek, and P. van de Walle, "Philips response to new call for 3DV test material: Arrive book & mobile," ISO/IEC JTC1/SC29/WG11 Doc. M16420, Apr. 2009.

[25] I. Feldmann *et al.*, "HI test material for 3D video," ISO/IEC JTC1/SC29/WG11 Doc. M15413, Apr. 2008.

[26] M. Tanimoto, T. Fujii, M. P. Tehrani, K. Suzuki, and M. Wildeboer, "Depth estimation reference software (DERS) 3.0," Maui, HI, ISO/IEC JTC1/SC29/WG11 Doc. M16390, Apr. 2009.

[27] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, 2004.

[28] J. Lu, Q. Yang, and G. Lafruit, "Interpolation error as a quality metric for stereo: robust or not?," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2009, pp. 977–980.

[29] "Final report from the video quality experts group on the validation of objective quality metrics for video quality assessment," 2000 [Online]. Available: http://www.its.bldrdoc.gov/vqeg/projects/frtv_phaseI

[30] S. Winkler and P. Mohandas, "The evolution of video quality measurement: from PSNR to hybrid metrics," *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 660–668, Sep. 2008.
[31] 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.

Yin Zhao received the B.Eng. degree in information engineering from Zhejiang University, Hangzhou, China in 2008. Since then he has been working towards the Ph.D. degree in Zhejiang University. In 2010, he was a visiting student at Nanyang Technological University, Singapore. His research interests include 3D video processing, video quality assessment, and video coding.

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.

He is currently an Associate Professor at the School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore. His research interests include image/video coding, streaming and processing, 3D video, multimedia systems and applications. He has authored or co-authored over 90 publications and edited a book. He also holds two granted patents.

Dr. Zhu serves as an Associate Editor of IEEE TRANS. ON BROADCASTING and IEEE SIGNAL PROCESSING LETTERS. He has served on technical/program committees, organizing committees and as track/session chairs for over 40 international conferences.

Zhenzhong Chen (S'02–M'07) received the B.Eng. degree from Huazhong University of Science and Technology (HUST) and the Ph.D. degree from Chinese University of Hong Kong (CUHK), both in electrical engineering.

He is currently a Lee Kuan Yew research fellow and Principal Investigator at Nanyang Technological University (NTU), Singapore. Before joining NTU, he was an ERCIM fellow at National Institute for Research in Computer Science and Control (INRIA), France. He held visiting positions at Universite Catholique de Louvain (UCL), Belgium, and Microsoft Research Asia, Beijing. His current research interests include visual perception, visual signal processing, and multimedia communications.

Dr. Chen is a voting member of IEEE Multimedia Communications Technical Committee (MMTC), an invited member of IEEE MMTC Interest Group of Quality of Experience for Multimedia Communications (QoEIG) (2010–2012). He has served as a guest editor of IEEE MMTC E-letter Special Issue on "Human-centric Multimedia Communications" and Journal of Visual Communication and Image Representation Special Issue on "Recent Advances on Analysis and Processing for Distributed Video Systems". He has co-organized several special sessions at international conferences and has served as a technical program committee member of IEEE ICC, GLOBECOM, CCNC, ICME, etc. He received CUHK Faculty Outstanding Ph.D. Thesis Award, Microsoft Fellowship, and ERCIM Alain Bensoussan Fellowship. He is a member of IEEE and SPIE.

Dong Tian received the Ph.D. degree at Beijing University of Technology in 2001. He received the M.Eng. and B.Eng. degrees in automation from the University of Science and Technology of China (USTC) in 1998 and 1995, respectively.

He is currently a principal member research staff in the Multimedia Group of Mitsubishi Electric Research Laboratories (MERL). Prior to joining MERL, he worked with Thomson Corporate Research at Princeton, NJ for over 4 years and became a Senior Scientist, where he devoted the years to an AVC encoder optimization and 3D video projects, especially to the standards of Multiview Video Coding (MVC) and later on 3D Video (3DV) within MPEG. Before that, he spent 4 years (Jan 2002∼Dec 2005) at Tampere University of Technology in Finland as a Post-Doc researcher, when he worked closely with Nokia Research Center to make contributions on the standardization of MPEG-4 AVC /H.264 and its application for mobile applications. He mainly conducts researches on video coding and processing, especially for 3D video signal.

Lu Yu received the B.Eng. degree in radio engineering and the Ph.D. degree in communication and electronic systems from Zhejiang University, China in 1991 and 1996, respectively.

She is currently a Professor in the Institute of Information and Communication Engineering, Zhejiang University. She was a senior visiting scholar in University Hannover in 2002 supported by China Scholarship Council and German Research Foundation (DFG). She was a senior visiting scholar in The Chinese University of Hong Kong supported by the United College Resident Fellow Scheme in 2004. Her research area is video coding, multimedia communication and relative ASIC design, in which she is principal investigator of national R&D projects such as Natural Science Foundation of China, 863 Hi-tech Program and inventor or co-inventor of 31 granted and 21 pending patents. She published more than 100 technical papers and contributed 200 proposals to international and national standards in the recent years.

Dr. Yu now acts as the Chair of the Video-Subgroup of Audio Video coding Standard (AVS) of China and she also was the Co-Chair of Implementation-Subgroup of AVS. She organized the 15th International Workshop on Packet Video as a General Chair in 2006, and host the 78th ISO/IEC JTC1 SC29 WG11 (MPEG) and 21st JVT meeting. She is a member of Technical Committee of Visual Signal Processing and Communication of IEEE Circuits and Systems Society, an area editor of EURASIP Journal Signal Processing: Image Communication.