A Novel Inpainting-Based Layered Depth Video for 3DTV

Ismaël Daribo and Hideo Saito

***Abstract*—Layered Depth Video (LDV) is recognized as a promising 3D video data representation for supporting advanced 3D video services required in Multiview Video (MVV) systems such as Three-Dimensional Television (3DTV). This representation consists of one full or central view in the form of a video-plus-depth sequence as the main layer, and additional enhancement layers including residual texture and depth data that represent side views. LDV is thus both a derivative of and an alternative to Multiview Video-plus-Depth (MVD) representation by only transmit one full view with associated residual data over the channel. There is a risk, however, of residual data information rapidly increasing as the distance between the center view and side views increases. This occurs when parts of the central view are not visible in the side views, leaving blank spots called disocclusions. These disocclusions may grow larger, which increases the amount of residual data that needs to be sent with the main layer. In this paper, we address the residual layer generation problem. We propose an inpainting-based LDV generation method to reduce the amount of residual data to send by retrieving the missing pixels from the main layer. In the proposed method, we take into account the depth information by distinguishing between foreground and background parts of the scene, at low complexity, during the texture and structure propagation stage of the inpainting process. Experimental results demonstrated the effectiveness of the proposed method.**

***Index Terms*—DIBR, inpainting, layered depth video, 3DTV.**

I. Introduction

INTEREST in 3D data representation for 3D video communication end-to-end services has grown rapidly within the last few years, particularly in Three-Dimensional Television (3DTV) production, which is considered the next generation of multimedia consumer products. 3DTV has a long history. Over the years, a consensus has been reached: 3DTV broadcast services can only be successfully introduced to the public if the perceived image quality and the viewing comfort is at least comparable to conventional Two-Dimensional Television (2DTV). Various improvements to 3D technology have raised more interest in multiview video applications, such as 3DTV that offers depth perception without the need for special glasses. The development of stereoscopic displays [1], [2] has been increasingly promoted by several display manufacturers. Although there is no doubt that High Definition Television (HDTV) has succeeded in significantly increasing the realism of television, it still lacks one very important feature: the representation of natural depth perception. At present, 3DTV can be considered the logical next step for complementing HDTV to incorporate 3D perception into the viewing experience. Multiview Video (MVV) systems have gained significant interest recently, particularly in terms of view synthesis approaches. View synthesis usually falls in two categories: Geometry-Based Rendering (GBR) and Image-Based Rendering (IBR). Typically, GBR exploits the 3D geometric texture knowledge of the scene, which requires 3D models of the objects. However, such models require millions of polygons, complex lighting models, extensive texture mapping, and great computational cost. IBR techniques have received attention as an attractive alternative to GBR for view synthesis. Instead of 3D geometric primitives, a collection of images are used to generate other images. Among a variety of IBR techniques, the Layered Depth Image (LDI) [3] is one of the most efficient synthesizing view methods for complex 3D scenes. LDI generation was first investigated in stereo case [3], and then later in a multiview camera set-up [4].

An extension of LDI representation called Layered Depth Video (LDV) has been proposed as a 3D video data representation. LDV is considered suitable associated 3D video data representation that provides one full or central view as a main layer, and additional enhancement layers that include residual texture and depth data to represent the side views. LDV is then both a derivative of and an alternative to Multiview Video-plus-Depth (MVD) representation: it only transmits one full view (with associated residual data) over the channel, and afterwards, the non-transmitted side views are generated by view synthesis as a view transfer between the central and side views. The central view is then projected onto side views by IBR. The problem, however, in that every pixel does not necessarily exist in every view, which results in the occurrence of holes when the central view is projected. View synthesis then exposes the parts of the scene that are occluded in the central view and make them visible in the side views. This is a process known as "disocclusion".

One way of dealing with these disocclusions would be to rely on pre-processing the depth video to allow the reduction of depth data discontinuities in a way that decreases the disocclusions. However, this would mean introducing filtering-induced distortion to the depth video, which would reduce the user's original depth perception. It is possible to remove disocclusions by considering more complex multi-dimensional data representations, such as LDV data representation, that allow the storage of additional depth and color values for pixels that are occluded in the central view. This extra data provides the necessary information to fill in disoccluded areas in rendered, novel views.

Manuscript received July 15, 2010; revised February 07, 2011; accepted February 14, 2011. Date of publication April 07, 2011; date of current version May 25, 2011. This work is supported by the National Institute of Information and Communications Technology (NICT), Japan.

The authors are with the Department of Information and Computer Science, Keio University, Yokohama 223-8522, Japan (e-mail: daribo@hvrl.ics.keio.ac.jp; saito@hvrl.ics.keio.ac.jp).

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.2011.2125110

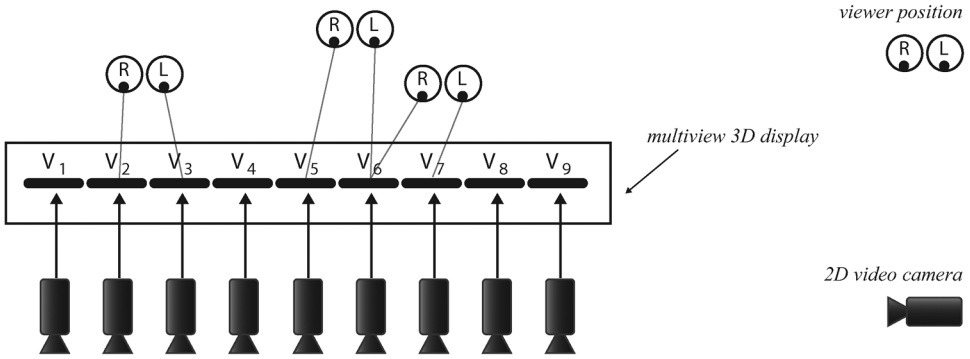

Fig. 1. Efficient support of multiview autostereoscopic displays based on MVV content.

One solution suggested by Tauber *et al.* [5] consists of combining IBR with inpainting techniques to deal with large disocclusions, due to the natural similarity between damaged holes in paintings and disocclusions in view synthesis. Image inpainting, also known as image completion [6], fills in pixels in a large missing region with the information derived from the pixels that surround it. Image and video inpainting has a wide range of applications, such as removing overlaid text and logos, restoring scans of deteriorated images by removing scratches or stains, compressing images, and creating artistic effects. State-of-the-art methods are broadly classified as structural inpainting or as textural inpainting. Structural inpainting reconstructs using prior assumptions about the smoothness of the structures in the missing regions and boundary conditions, while textural inpainting considers only the available data from texture exemplars or other templates in the existing regions.

Initially introduced by Bertalmio *et al.* [7], structural inpainting uses either isotropic diffusion or the more complex anisotropic diffusion to propagate boundary data in the isophote[1] direction, and prior assumptions about the smoothness of structures in the missing regions. Textural inpainting uses either statistics or template knowledge of patterns inside the missing regions, commonly modeled by Markov random fields (MRF). Levin *et al.* suggest to extract relevant statistics about the known parts of the image and then combining them in an MRF framework [8].

In this study, we used the work done by Criminisi *et al.* [9] as a starting point. They attempted to combine the advantages of both structural and textural inpainting approaches by using a very insightful principle: the texture is inpainted in the isophote direction according to its strength. However, this remains a purely 2D approach. We propose here to extend this idea by adding depth information to distinguish pixels belonging to either the foreground or the background. Clearly indicating which contour of the holes is close to the object of interest and which one is in the background neighborhood significantly improves the inpainting algorithm in this context. We also propose taking advantage of the depth information with a low additional computational complexity and no loss of optimality.

The rest of the paper is organized as follows. We introduce some related background information on 3D video formats in Section II, and we describe residual layer generation in Section III. Section IV briefly reviews Criminisi's inpainting algorithm and Section V addresses the problem of reducing the amount of residual data to be transmitted. Our final conclusions are drawn in Section VII.

[1]Isophotes are level lines of equal gray-levels. Mathematically, the direction of the isophotes can be interpreted as $\nabla^{\perp} I$, where $\nabla^{\perp} = (-\partial_y, \partial_x)$ is the direction of the smallest change.

II. Background

To support 3DTV system requirements, many 3D video data representation have been investigated in terms of their complexity, efficiency, and functionality according to the following general requirements:

- can utilize as many existing delivery infrastructure and media as possible,
- require minimal change to device components,
- backwards compatibility - it is unacceptable for 3D services to impair existing devices,
- can support a wide range of display devices and allow for future extension,
- are high quality.

Stereoscopic systems are the most well-known and simple acquisition techniques for 3D video data representation. Stereoscopic video can provide a 3D impression by using left and right videos as a pair, thereby creating a stereo camera system, while a monoscopic 2D video cannot. A pair of 2D videos is acquired: one for the left eye, and the other for the right. As a generalization of stereo video, MVV can be considered an extension of the stereo video data representation to a higher number of views.

Multiview autostereoscopic displays project multiple views into the viewing zone at the same time essentially, the consecutive views act like stereo pairs (Fig. 1). As a result, head motion parallax viewing can be supported within practical limits, but the amount of data to be processed and transmitted increases significantly compared to using conventional stereo data or 2D video. The development of a wide range of multiview autostereoscopic displays and MVV applications increases the number of output video needs. Users can therefore choose their own viewpoint (*e.g.* super bowl XXXV sport event, bullet time effect, *etc*). Advanced 3D video applications like this require a 3D video format that can render a continuum of output views or a very large number of different output views at the decoder side. MVV formats are still not sufficient to support such requirements without extensively increasing the number of input views and consequently the bandwidth.

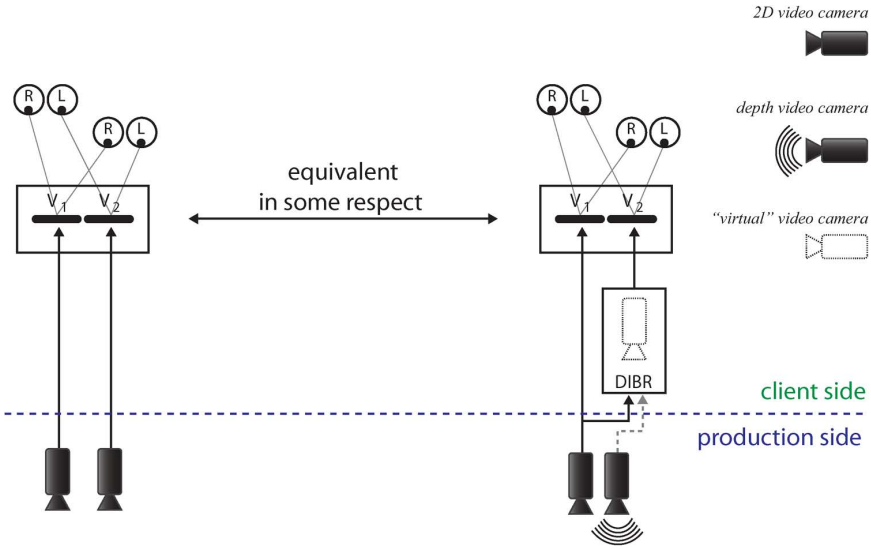

Fig. 2. Efficient support of stereo autostereoscopic displays based on video-plus-depth content.

texture image *associated depth map*

Fig. 3. Texture picture and its associated depth map.

Video-plus-depth data representation has been introduced to overcome this issue. It can respond to the stereoscopic vision needs at the receiver side as shown in Fig. 2, and at the same time decrease dramatically the transmission bandwidth compared to the conventional stereo video data representation. Initially studied in the computer vision field, the video-plus-depth format provides a regular 2D video enriched with its associated depth video (see Fig. 3).

The 2D video provides the texture information, the color intensity, and the structure of the scene, while the depth video represents the Z-distance per-pixel between the optical center of the camera and a 3D point in the visual scene. Hereafter, the 2D video may be denoted as texture video in opposition to the depth video.

Great effort has been made to estimate depth information from multiple 2D video inputs. Thanks to recent advances in semiconductor processes, it is possible to directly capture depth video using a time-of-flight (TOF) camera [10], also known as a depth camera. The TOF camera is based on TOF technology that measures the distance between the camera and the scene in real time. This camera emits infrared light that is reflected by the environment and then comes back to the camera's sensor. The traveling time of the light is then measured for each pixel of the sensor and used to compute the depth of the scene. The depth video can be regarded as a monochromatic texture-less video signal. Generally, the depth data is quantized with 8 bits, i.e., the closest point is associated with the value 255 and the most distant point is associated the value 0. With that, the depth video is specified as a smoothed gray level representation.

At the client side, the second color video corresponding to the second view is reconstructed from the transmitted video-plus-depth data by means of depth image based rendering (DIBR) techniques [11]–[13]. The ability to generate a stereoscopic video from video-plus-depth data at the receiver side is an extended functionality compared to conventional stereo video data representation. Consequently, the 3D impression can be adjusted and customized after transmission. However, because view-synthesis-induced artifacts increase dramatically with the distance of the rendered viewpoint, video-plus-depth can support only a very limited continuum around the available original view. To overcome this issue, MPEG started an activity developing a 3D video standard that would support these requirements [14]. This standard is based on a video-plus-depth (MVD) format as shown in Fig. 4. Video-plus-depth data is combined with multiview video (MVV) data to form the MVD format, which consists of multiple 2D videos, each of which has an associated depth video.

The final step in this process is rendering multiple intermediate views from the received data by DIBR. At this point, the central and side views are fully processed and transmitted. As an alternative to fully transmitting the side views in addition to the central view, LDV can decrease redundancies between the views by only considering the central view as the main layer and some residual data as enhancement layers. This new representation can deliver targets high-quality, high-resolution images with lower bitrates than those deliver by MVD. In the next section, we discuss the importance of residual layer extraction generation.

III. Residual Layer Generation

We will describe residual layer generation within a three video camera system composed of one central and two side views (left and right), as illustrated in Fig. 4. The generation

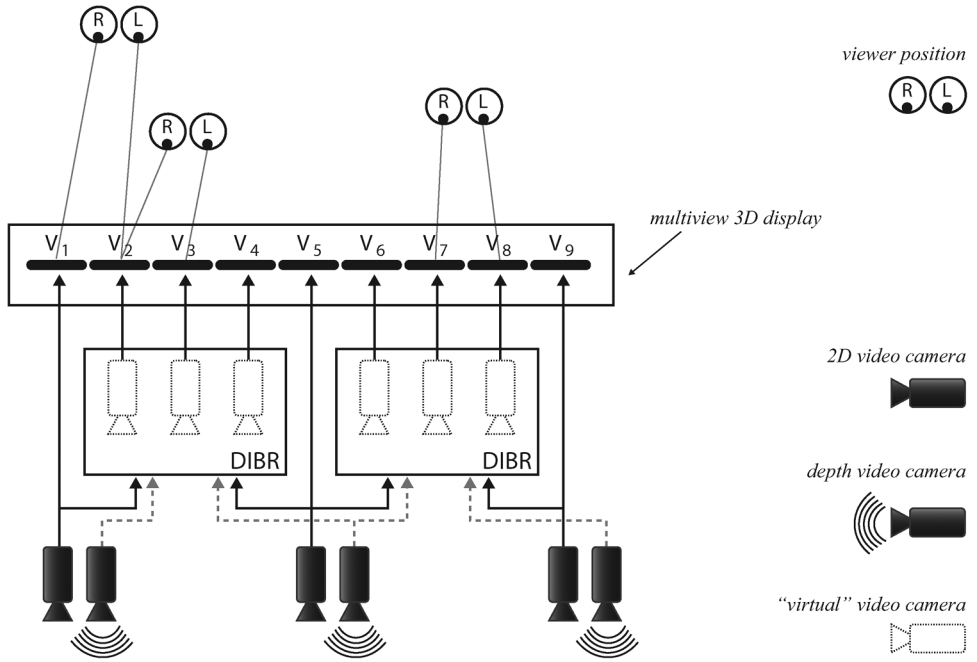

Fig. 4. Efficient support of multiview autostereoscopic displays based on MVD content.

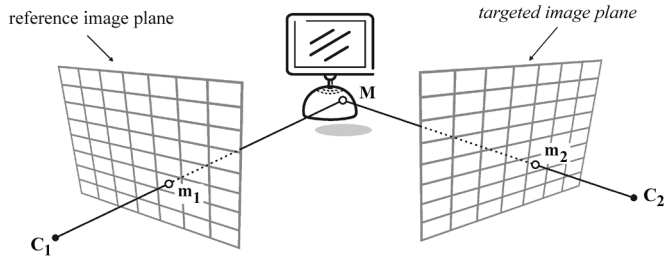

Fig. 5. 3D image warping: Projection of a 3D point on two image planes in homogeneous coordinates.

process can be separated into two main parts. First, the central view is transferred to each side view by DIBR using the given depth video. This process is called as 3D warping. Next, by subtraction, it is possible to determine which parts of the side views are covered in the central view. These are then assigned as residual data for texture and depth and transmitted while the rest is omitted. This process includes a function for mapping points from the central view (the reference image plane) to the side views (the targeted image plane) as illustrated in Fig. 5 and described in the next section.

A. 3D Warping

First, we introduce some notations. The intensity of the reference view image I_1 at pixel coordinates (u_1, v_1) is denoted by $I_1(u_1, v_1)$. The pinhole camera model is used to project I_1 into the second view $I_2(u_2, v_2)$ with the given depth data $Z(u_1, v_1)$. Conceptually, the 3D image warping process can be separated into two steps: a back-projection of the reference image into the 3D-world, followed by a projection of the back-projected 3D scene into the targeted image plane [11]. If we look at the pixel location (u_1, v_1), first, a back-projection per-pixel is performed from the 2D reference camera image plane I_1 to the 3D-world coordinates. Next, a second projection is performed from the 3D-world to the image plane I_2 of the target camera at pixel location (u_2, v_2), and so on for each pixel location. To perform these operations, three quantities are needed: $\boldsymbol{K}_1$, $\boldsymbol{R}_1$, and $\mathbf{t}_1$, which denote the 3×3 intrinsic matrix, the 3×3 orthogonal rotation matrix, and the 3×1 translation vector of the reference view I_1, respectively. The 3D-world back-projected point $\mathbf{M} = (x, y, z)^\top$ is expressed in non-homogeneous coordinates as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{R}_1^{-1} K_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \lambda_1 - \boldsymbol{R}_1^{-1} \mathbf{t}_1 \tag{1}$$

where λ_1 is a positive scaling factor.

Looking at the target camera quantities, $\boldsymbol{K}_2$, $\boldsymbol{R}_2$ and $\mathbf{t}_2$, the back-projected 3D-world point $\mathbf{M} = (x, y, z, 1)^\top$ is then mapped into the targeted 2D-image coordinates $(u', v', 1)^T$ in homogeneous coordinates as:

$$\begin{pmatrix} u_2' \\ v_2' \\ w_2' \end{pmatrix} = \boldsymbol{K}_2 \boldsymbol{R}_2 \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \boldsymbol{K}_2 \mathbf{t}_2. \tag{2}$$

We can therefore express the targeted coordinates function of the reference coordinates by

$$\begin{pmatrix} u_2' \\ v_2' \\ w_2' \end{pmatrix} = \boldsymbol{K}_2 \boldsymbol{R}_2 \boldsymbol{R}_1^{-1} \boldsymbol{K}_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \lambda_1 - \boldsymbol{K}_2 \boldsymbol{R}_2 \boldsymbol{R}_1^{-1} \mathbf{t}_1 + \boldsymbol{K}_2 \mathbf{t}_2 \tag{3}$$

It is common to attach the world coordinates system to the first camera system, so that $\boldsymbol{R}_1 = \boldsymbol{I}_3$ and $\mathbf{t}_1 = \mathbf{0}_3$, which simplifies (3) into

$$\begin{pmatrix} u_2' \\ v_2' \\ w_2' \end{pmatrix} = \boldsymbol{K}_2 \boldsymbol{R}_2 \boldsymbol{K}_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \lambda_1 + \boldsymbol{K}_2 \mathbf{t}_2 \tag{4}$$

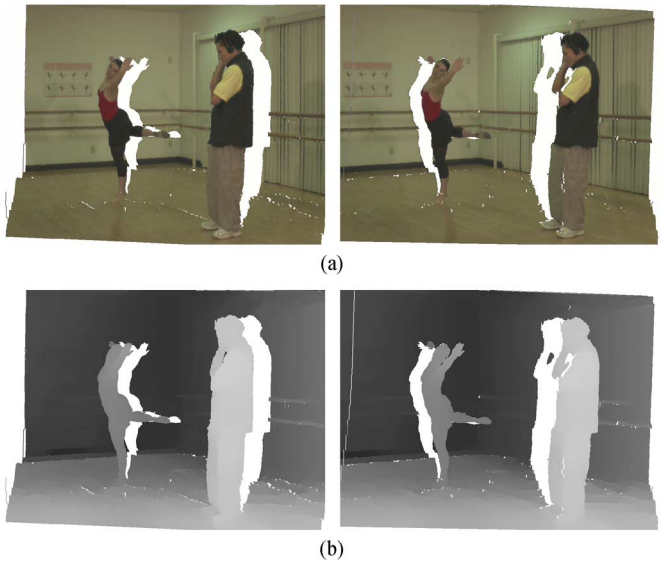

Fig. 6. 3D warping of the central view into both side views. (a) Projected texture image. (b) Projected depth map.

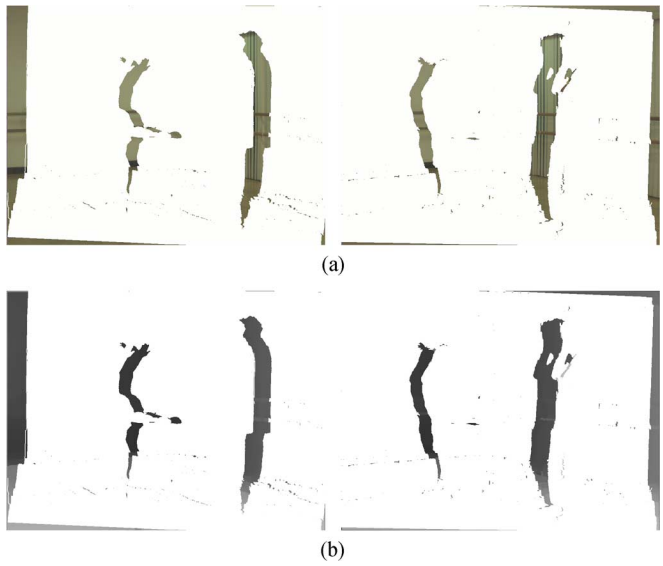

Fig. 7. Residual data in both side views. (a) Residual texture image. (b) Residual depth map.

where $(u_2', v_2', w_2')^\top$ is the homogeneous coordinates of the 2D-image point $\mathbf{m}_2$, and the positive scaling factor λ_1 is equal to

$$\lambda_1 = \frac{z}{c} \quad \text{where} \quad \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \boldsymbol{K}_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \tag{5}$$

In the final step, the homogeneous result is converted into pixel location as $(u_2, v_2) = (u_2'/w_2', v_2'/w_2')$.

Note that z is the third component of the 3D-world point $\mathbf{M}$, indicating the depth information at pixel location (u_1, v_1) of image I_1. This data is considered key side information for retrieving the corresponding pixel location on the other image I_2.

B. Residual Layer

As mentioned earlier, 3D warping of the central view into both side views reveals the covered parts (as shown in Fig. 6) which then need to be transmitted in addition to the central view. These disoccluded regions are mainly concentrated along the depth discontinuities of foreground objects. The side views are reduced to residuals, as shown in Fig. 7, by subtracting the projected central view from a given side view. This results in a significantly reduced data rate.

At the user side, the central view and residual data are extracted to reconstruct original side views (see Fig. 8), leading to a new viewing experience and a high degree of user interactivity.

IV. Criminisi's Inpainting Algorithm

Criminisi *et al.* [9] first reported that exemplar-based texture synthesis contains the process necessary to replicate both texture and structure. They used the sampling concept from Efros and Leung's approach [15], and demonstrated that the quality of the output image synthesis is greatly influenced by the order in which the inpainting is processed.

With input image I and missing region Ω, the source region Φ is defined as $\Phi = I - \Omega$ (see Fig. 9). The algorithm performs the synthesis task by using a best-first filling strategy that entirely depends on the priority values that are assigned to each patch on boundary $\delta\Omega$. Given patch Ψ_p centered at point p for some $p \in \delta\Omega$ (see Fig. 9), they define its priority $P(p)$ as the product of two terms:

$$P(p) = C(p) \cdot D(p), \tag{6}$$

where $C(p)$ is the *confidence* term that indicates the reliability of the current patch and $D(p)$ is the *data* term that gives special priority to the isophote direction. These terms are defined as

$$C(p) = \frac{1}{|\Psi_p|} \sum_{q \in \Psi_p \cap \Phi} C(q) \quad \text{and} \quad D(p) = \frac{\langle \nabla^\perp I_p, \mathbf{n}_p \rangle}{\alpha} \tag{7}$$

where $|\Psi_p|$ is the area of Ψ_p (in terms of number of pixels within patch Ψ_p), α is a normalization factor (*e.g.* $\alpha = 255$ for a typical gray-level image), $\mathbf{n}_p$ is a unit vector orthogonal to $\delta\Omega$ at point p, and $\nabla^\perp = (-\partial_y, \partial_x)$ is the direction of the isophote. $C(p)$ represents the percentage of non-missing pixels in patch Ψ_p and is set at initialization to $C(q) = 0$ for missing pixels in Ω, and $C(q) = 1$ everywhere else. Once all the priorities on $\delta\Omega$ are computed, a block matching algorithm derives the best exemplar $\Psi_{\widehat{q}}$ to fill in the missing pixels under the highest priority patch $\Psi_{\widehat{p}}$, previously selected, as follows

$$\Psi_{\widehat{q}} = \arg\min_{\Psi_q \in \Phi} \left\{ d(\Psi_{\widehat{p}}, \Psi_q) \right\} \tag{8}$$

where $d(.,.)$ is the distance between two patches, defined as the Sum of Squared Differences (SSD). Having found the source exemplar $\Psi_{\widehat{q}}$, the value of each pixel-to-be-filled $p' \in \Psi_{\widehat{p}} \cap \Omega$ is copied from its corresponding pixel in $\Psi_{\widehat{q}}$. After patch $\Psi_{\widehat{p}}$ has been filled, the confidence term C(p) is updated as follows

$$C(p) = C(\widehat{p}), \quad \forall p \in \Psi_{\widehat{p}} \cap \Omega. \tag{9}$$

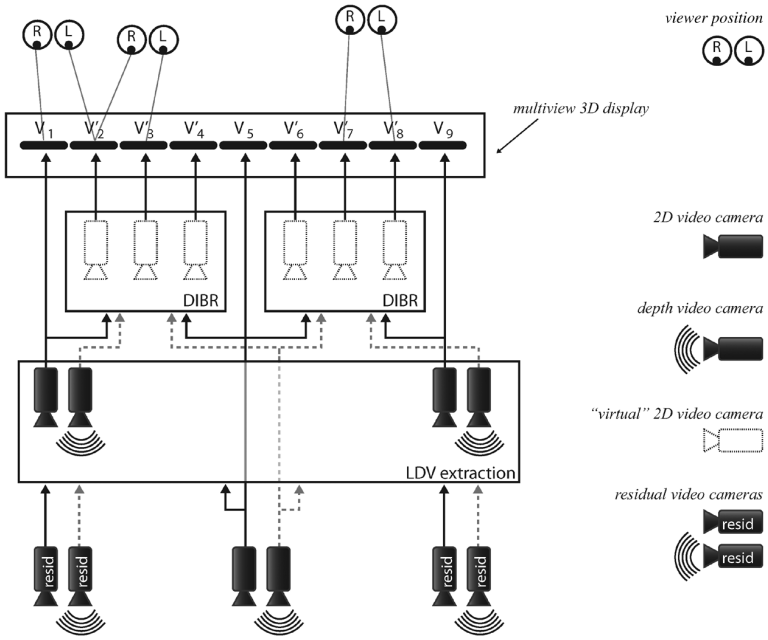

Fig. 8. Efficient support of multiview autostereoscopic displays based on LDV content.

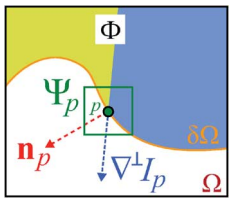

Fig. 9. Notation diagram [9].

V. Depth-Aided Texture and Structure Propagation

Although missing areas in paintings and disocclusions from 3D image based rendering present conceptual similarities, we have some specific *a priori* knowledge about disocclusions. Disocclusions are the result of displaced foreground object that reveals some background areas. Filling in the disoccluded regions using background pixels therefore makes more sense than foreground ones. To implement this, Cheng *et al.* developed a view synthesis framework [16], in which the depth information constrains the search range for the texture matching and then a trilateral filter utilizes the spatial and depth information to filter the texture image, thus enhancing the view synthesis quality. In a similar study, Oh *et al.* proposed replacing the foreground boundaries with background ones located on the opposite side [17]. They intentionally manipulated the disocclusion boundaries so that they only contained pixels from the background and then applied one of the existing inpainting techniques.

Based on these works, we propose a depth-aided texture inpainting method using Criminisi's algorithm principles that gives background pixels higher priority than foreground ones.

A. Priority computation

Given the associated depth patch Z in the targeted image plane, in our definition of priority computation, we propose weighting the previous priority computation in (6) by adding a third multiplicative term:

$$P(p) = C(p) \cdot D(p) \cdot L(p), \tag{10}$$

where $L(p)$ is the level regularity term, defined as the inverse variance of the depth patch Z_p:

$$L(p) = \frac{|Z_p|}{|Z_p| + \sum\limits_{q \in \Psi_p \cap \Phi} (Z_p - \overline{Z_p})^2} \tag{11}$$

where $|Z_p|$ is the area of Z_p (in terms of number of pixels), and $\overline{Z_p}$ the mean value. We give more priority to patch overlaying at the same depth level, which naturally favors background pixels over foreground ones.

B. Patch Matching

Considering the depth information, we update (8) as follows:

$$\Psi_{\hat{q}} = \arg\min_{\Psi_q \in \Phi} \left\{ d(\Psi_{\hat{p}}, \Psi_q) + \beta \cdot d(Z_{\hat{p}}, Z_q) \right\} \tag{12}$$

where the block matching algorithm is processed in the texture and depth domains through parameter β, which allows us to control the importance given to the depth distance minimization. By updating the distance measure, we favor the search of patches with the same depth level.

Based on the proposed depth-based inpainting method, we can to some extent retrieve the missing pixels by taking advantage of the available texture inside the central view. This leads to reduced residual data. The last step consists of deciding which parts of the projected central view still need to be transmitted. We propose considering the inpainting-induced artifacts as the new residual data because there are significantly fewer of them than the original disocclusions as shown in Fig. 12. Moreover, by significantly reducing the residual areas, we overcome the overlap issue mentioned by Muller *et al.* [18]. The merging of

(a) (b) (c)

Fig. 10. Example of disoccluded regions to fill in ("Ballet" sequence, frame 21) and the resulting residual to send. (a) Full resolution. (b) Zoom. (c) Residual.

(a) Criminisi's inpainted (b) Oh's + Criminisi's inpainted (c) proposed inpainted

Fig. 11. Example of hole-filling using (a) the original Criminisi's algorithm, (b) the Oh's + Criminisi's, and (c) the one enhancement by warped depth information ("Ballet" sequence, frame 21). (a) Criminisi's inpainted. (b) Oh's + Criminisi's inpainted. (c) Proposed inpainted.

residual data of both side views into one buffer can further reduced the data rate.

VI. Experimental Results

The Multiview Video-plus-Depth (MVD) sequence "Ballet" provided by Microsoft [19] was used to test the proposed method. Calibration parameters are supplied with the sequences. The depth video provided for each camera was estimated via a color-based segmentation algorithm [20]. The "Ballet" sequence represents two ballet dancers at two different depth levels. Due to the large baseline between the central camera and the side cameras, more disocclusions appear during the 3D warping process.

As we can see in Fig. 10, the disocclusion boundaries belong both to the foreground and background part of the scene. This makes conventional inpainting methods less efficient.We will now compare our method with the work described by Oh *et al.* [17]. Oh *et al.* proposed a pre-processing step [17] that addresses the issue of disocclusion boundaries belonging to both the foreground and background. To provide a fair comparison, we applied Criminisi's inpainting algorithm after the foreground/background boundaries copy-and-paste [17]. We will refer to this combination as "Oh's + Criminisi's inpainting". Most of the related work focused on frameworks where at least two reference views were warped on a virtual viewpoint. As a result, fewer disocclusions were revealed, and the disoccluded regions were smaller. It is important to note that in our problem, only one reference view is available (i.e., the central view), leading to large disocclusions, in which conventional inpainting methods tend to be ineffective.

Results using our region-filling method, the original Criminisi's algorithm, and Criminisi's algorithm combined with Oh *et al.*'s method are shown in Fig. 11. Comparing the three methods clearly demonstrated that our algorithm better preserves the contours of foreground objects and can enhance the visual quality of the inpainted images. This is achieved by propagating the texture and structure from the background regions, while Criminisi's algorithm makes no distinction between the two. Despite the fact that the Oh's + Criminisi's scheme combination can distinguish the foreground from the background along the disocclusion boundaries, this method is not well suited for large disocclusions and depends mainly on the well-computed segmentation of disocclusion boundaries.

We can observe on the Fig. 13 the quality improvement obtained with our method, and in Fig. 12 the significant residual data reduction. The residual data reduction, which is significant, is shown in Fig. 12, and the quality improvement obtained by our method is clearly visible in Fig. 13. We managed to contain

(a) Criminisi's residual (b) Oh's + Criminisi's residual (c) proposed residual

Fig. 12. Example of residual data using (a) original Criminisi's algorithm, (b) Oh's + Criminisi's, and (c) one enhancement by warped depth information ("Ballet" sequence, frame 21). (a) Criminisi's residual. (b) Oh's + Criminisi's residual. (c) Proposed residual.

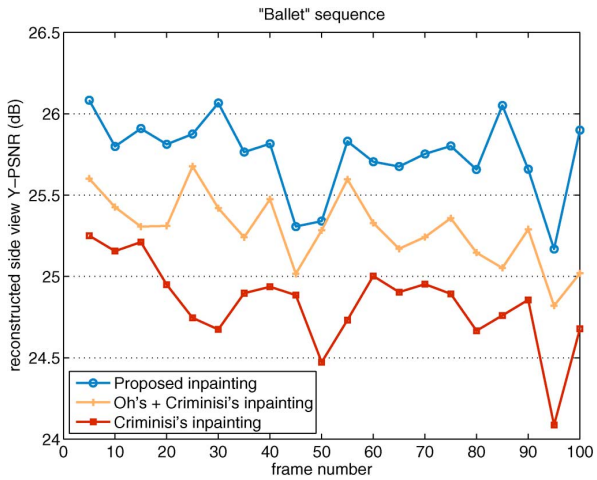

Fig. 13. Objective PSNR results.

any computational increase, or pre-processing requirement by adding a new depth-based term to the patch priority computation. This is significant because many previous methods require pre-processing processes [17] or inevitably lead to expensive additional computations [16].

VII. Conclusion

In this paper, we addressed the problem of residual layer generation in LDV data representation by proposing the use of a post-process on the disoccluded areas based on inpainting techniques, which are well-known for their ability to propagate texture and structure along the contours of "holes". The proposed method uses Criminisi's algorithm, and depth information has been added to the priority computation and the patch matching. The proposed method relies on texture and structure propagation while also taking into account the depth information by distinguishing between the foreground and background parts of an image.

Improvements can be made to visual quality without any increase in complexity, and no pre-processing is required. We took advantage of the available texture inside the central view to fill in the disocclusions. Experimental results demonstrated that the visual quality of the inpainted image can be improved in particular, by preserving the foreground contours, thus, reducing the amount of residual data to be sent in addition to the full transmission of the central view.

Several issues remain that warrant further research. In future studies, we intend to concentrate on temporal coherence to ensure spatial and temporal stability and robustness against errors in the foreground depth map.

References

[1] N. A. Dodgson, "Autostereoscopic 3D displays," *Comput.*, vol. 38, pp. 31–36, Aug. 2005.

[2] Y. Zhu and T. Zhen, "3D multi-view autostereoscopic display and its key technologies," in *Asia-Pacific Conf. Inf. Process. (ACIP)*, 2009, vol. 2, pp. 31–35.

[3] J. W. Shade, S. J. Gortler, L.-W. He, and R. Szeliski, "Layered depth images," in *Computer Graphics*, Jul. 1998, vol. 32, Annual Conference Series, pp. 231–242 [Online]. Available: http://grail.cs.washington.edu/projects/ldi/

[4] X. Cheng, L. Sun, and S. Yang, "Generation of layered depth images from multi-view video," in *Proc. of the IEEE Int. Conf. Image Process. (ICIP)*, San Antonio, TX, 2007, vol. 5.

[5] Z. Tauber, Z.-N. Li, and M. Drew, "Review and preview: Disocclusion by inpainting for image-based rendering," *IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev.*, vol. 37, no. 4, pp. 527–540, Jul. 2007.

[6] B. Furht, *Encyclopedia of Multimedia*, 2nd ed. New York: Springer, 2008.

[7] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. of the annual Conference on Computer graphics and interactive techniques (SIGGRAPH)*, New Orleans, USA, Jul. 2000, pp. 417–424.

[8] A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nice, France, Oct. 2003, vol. 1, pp. 305–312.

[9] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, 2004.

[10] T. Oggier, M. Lehmann, R. M. Rolf Kaufmann, M. S. , P. Metzler, G. Lang, F. Lustenberger, and N. Blanc, "An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (swissranger)," in *Proc. SPIE Conf. Opt. Syst. Design*, 2003, vol. 5249, pp. 634–545.

[11] L. McMillan, Jr., "An image-based approach to three-dimensional computer graphics," Ph.D. dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC, 1997.

[12] W. Mark, "Post-rendering 3D image warping: Visibility, reconstruction, and performance for depth-image warping," Ph.D. dissertation, University of North Carolina at Chapel Hill, , NC, Apr. 1999.

[13] M. M. Oliveira, "Relief texture mapping," Ph.D. dissertation, University of North Carolina at Chapel Hill, , NC, 2000.

[14] A. Smolic, K. Mueller, P. Merkle, N. Atzpadin, C. Fehn, M. Mueller, O. Schreer, R. Tanger, P. Kauff, and T. Wiegand, "Multi-view video plus depth (MVD) format for advanced 3D video systems," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-W100 doc.*, San Jose, CA, Apr. 2007.

[15] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Kerkyra, Greece, Sep. 1999, vol. 2, pp. 1033–1038.

[16] C.-M. Cheng, S.-J. Lin, S.-H. Lai, and J.-C. Yang, "Improved novel view synthesis from depth image with large baseline," in *Proc. Int. Conf. Pattern Recog.*, Tampa, Finland, Dec. 2008, pp. 1–4.

[17] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video," in *Proc. Picture Coding Symp. (PCS)*, Chicago, IL, USA, May 2009, pp. 1–4.

[18] K. Muller, A. Smolic, K. Dix, P. Kauff, and T. Wiegand, "Reliability-based generation and view synthesis in layered depth video," in *Proc. IEEE Workshop Multimedia Signal Process. (MMSP)*, Cairns, Queensland, Australia, Oct. 2008, pp. 34–39.

[19] "Sequence microsoft ballet and breakdancers," 2004 [Online]. Available: http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload/

[20] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. Annu. Conf. Comput. Graph. Interactive Tech. (SIGGRAPH)*, Aug. 2004, vol. 23, pp. 600–608, no. 3.

Ismaël Daribo received the Engineering degree from the IMAC Graduate Engineering School, France and the Master's degree from University of Paris-Est, France, in 2005. He received the Ph.D. degree from Telecom ParisTech in 2009.

During the summer of 2008, he was a visiting scholar researcher under the JSPS program at Keio University, Japan, and he returned to the university in 2010, where he currently holds a research associate position. His research interests include all technical aspects of 3D video communication end-to-end services, including multiple camera acquisition, 3D video data representation, and 3D video coding and depth image-based rendering on 3D displays.

Hideo Saito received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Japan, in 1987, 1989, and 1992, respectively.

He has been on the faculty of the Department of Electrical Engineering, Keio University, since 1992. He stayed in the Robotics Institute, Carnegie Mellon University as a visiting researcher from 1997 to 1999. Since 2006, he has been a professor in the Department of Information and Computer Science, Keio University. He is currently the leader of the research project "Technology to Display 3D Contents into Free Space" supported by CREST, JST, Japan. His research interests include computer vision, mixed reality, virtual reality, and 3D video analysis and synthesis. He is a senior member of IEEE, and IEICE, Japan.