QUALITY IMPROVING TECHNIQUES IN DIBR FOR FREE-VIEWPOINT VIDEO

Luat Do[1], Svitlana Zinger[1], Yannick Morvan[1] and Peter H. N. de With[1,2]

[1] Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands

[2] Cyclomedia Technology B.V., P.O. Box 68, 4180 BB Waardenburg, The Netherlands

`luat.do@gmail.com, {y.morvan, S.Zinger, P.H.N.de.With}@tue.nl`

ABSTRACT

This paper evaluates our 3D view interpolation rendering algorithm and proposes a few performance improving techniques. We aim at developing a rendering method for free-viewpoint 3DTV, based on depth image warping from surrounding cameras. The key feature of our approach is warping texture and depth in the first stage simultaneously and postpone blending the new view to a later stage, thereby avoiding errors in the virtual depth map. We evaluate the rendering quality in two ways. Firstly, it is measured by varying the distance between the two nearest cameras. We have obtained a PSNR gain of 3 dB and 4.5 dB for the 'Breakdancers' and 'Ballet' sequences, respectively, compared to the performance of a recent algorithm. A second series of tests in measuring the rendering quality were performed using compressed video or images from surrounding cameras. The overall quality of the system is dominated by rendering quality and not by coding.

Index Terms— 3D view interpolation, image based rendering, 3D rendering.

1. INTRODUCTION

Single viewpoint 3DTV and movie is about to break through in the market, due to emerging 3D movies and the availability of low-cost and high-definition display technology. Free-viewpoint video will be an important and innovative feature of 3DTV and an interesting extension [1]. It will allow the user to watch a film or a sport event from his own desired interactively chosen viewpoint. Such an application requires a high-quality 3D video rendering algorithm. Besides consumer applications, 3D display and rendering technology are also being introduced for medical data visualization [2].

For multi-view applications, the scene is typically captured by several cameras at different positions. The intermediate views are then synthesized by interpolation of the two nearest views. Pulli *et al.* [3] show that the highest rendering quality is obtained by using depth maps with individual pixel accuracy. In this paper, we will present a concept of a novel Depth Image Based Rendering (DIBR) algorithm and focus on the performance and possible quality improvements. The

quality tests are replicated from [4] and [5] and the results are compared to that recent work.

Previous work on Depth Image Based Rendering (DIBR) and warping involves warping from one reference image [6] or from two surrounding images [5, 7]. The drawback of the first method is that the rendering quality depends on the distance to the reference camera, as the disocclusions cannot be compensated by other camera views. The two principal problems of DIBR are disocclusions and depth discontinuities that naturally occur between foreground and background objects. Although the algorithm from [5] handles these problems well and produces good results, we show that with a new approach it is possible to clearly outperform those results. The novelty of our approach is that we do not aim at creating a full depth map for the rendered image, because this leads to inherent errors in warping that are difficult to remove. Instead, we process the projected depth maps separately and use them for blending the texture images.

The remainder of this paper is organized as follows. Section 2 outlines our proposed algorithm. We adopt two quality tests from [5] and [4] and apply those to our new approach to create a valid comparison, which is discussed in Section 3. The paper is concluded in Section 4.

2. DEPTH IMAGE BASED RENDERING

In this section, we commence with the fundamental steps of DIBR algorithms and afterwards, we discuss where our new approach differs from the existing proposals. The DIBR algorithms are based on warping [8] a camera view to another view. In multi-view video, the information for warping is taken from the two surrounding camera views to render a new synthetic view. Typically, two warped images are blended to create a synthetic view at the new position. The key to
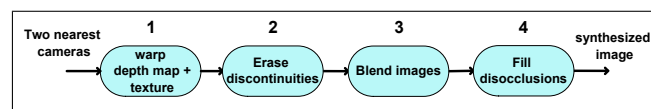

Fig. 1. DIBR algorithm pipeline

our new approach is that the depth image is not blended at an early stage, but the warping results are kept separated and also based on texture warping. In this warping stage, discontinuities (ghosting) aspects are considered prior to blending, which is performed at a later stage. After blending, disocclusions are processed with intelligent foreground and background interpolation. The processing pipeline of the algorithm is shown in Fig. 1. A more detailed reporting of our approach is under development.

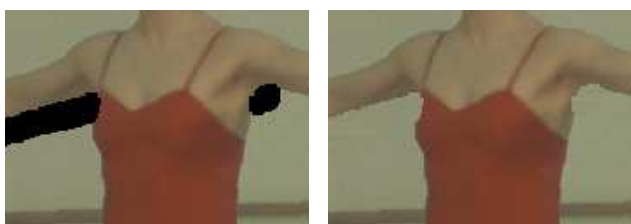(a) Warped depth map before median   (b) Warped depth map after median

Fig. 2. Median filter fills in empty pixels and smoothes the depth image while preserving edges
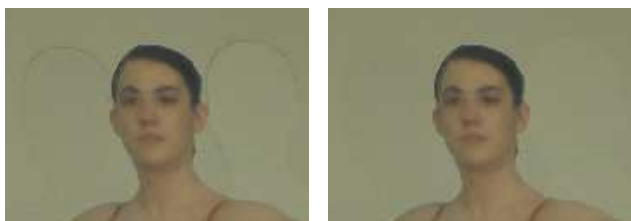
(a) With ghosting errors   (b) Ghosting erased

Fig. 3. Contouring as a result of depth discontinuities

(a) disocclusions after blending   (b) disocclusions filled in

Fig. 4. Disocclusions are filled in with background textures

The first processing step is 3D warping of the two nearest camera views. Unlike the method of Morvan *et al.* [1] and Mori *et al.* [5], where a virtual depth map is first created, both texture and depth are warped simultaneously but kept separated. We have found that first creating a virtual depth map and then doing an inverse 3D warping, results in embedding more inherent errors in the synthetic view. In addition, warping both the depth and texture maps results in considerably less warping operations. A known artifact of 3D warping

is the creation of blank spots within the synthetic image plane due to rounding errors. We have employed a median filter to fill in those blank spots. As a bonus, the median filter will also smooth the depth maps while preserving the edges of objects. Another property of the median filter is that disoccluded regions are not filtered. Fig. 2 illustrates the effectiveness of the median filter.

In the sequel, we discuss how to combat two typical artifacts in DIBR. One of the main problems that occurs in rendering is ghosting errors. This happens when areas at depth discontinuities are warped to the interpolated image. At the discontinuities, the textures of the pixels are a mixture of the background and foreground due to ill-defined borders. When warping, textures of the foreground may be warped to the background. A contour silhouette is then blended into the background (see Fig. 3). Our pipeline has specific processing in the second stage to remove this silhouette. The removal is achieved by 3D warping the coordinates of the discontinuities and then removing their destination coordinates.

In the next processing step, blending of two warped images is performed with a weighted average of the two nearest cameras. The weight is dependent on the relative distance of each camera position to the new position. After the blending process there still might be a few empty areas, resulting from disocclusions, specifically from areas that can neither be viewed from the left nor the right surrounding camera. Disoccluded areas are interpolated with the nearest background textures, as can be seen in Fig. 4.

3. QUALITY MEASUREMENTS

In this section, we will discuss two ways of quality measurements: PSNR evaluation involving the camera configuration of the 3D scene and distortion (PSNR) of a rendered view dependent on the applied compression technique for the surrounding camera views. Since the number of cameras is limited, the camera setup is of primary importance for obtaining a good quality of free-viewpoint rendering. The first series of measurements evaluates the quality of the rendering while varying the distance between the two nearest cameras. This measurement technique has been described in [5]. The RGB images are first transformed to the YUV color space. Then the Peak Signal-to-Noise Ratio (PSNR) of the Y values is calculated. The results are depicted in Fig. 5.

In Fig. 5, the combined distance is defined as $||P-C_1||+||P-C_2||$, where P is the new position and C_1 and C_2 are the positions of the two nearest cameras in 3D point coordinates. In our case, P has the same position as the center camera. It can be seen that our novel rendering algorithm increases the average PSNR with 3 dB and 4.5 dB, for the 'Breakdancers' and 'Ballet' scenes, respectively, as compared to the results presented in [5]. The large difference in PSNR is caused by the larger areas with pixel color differences as rendering is at an earlier stage. The subjective quality difference is smaller,
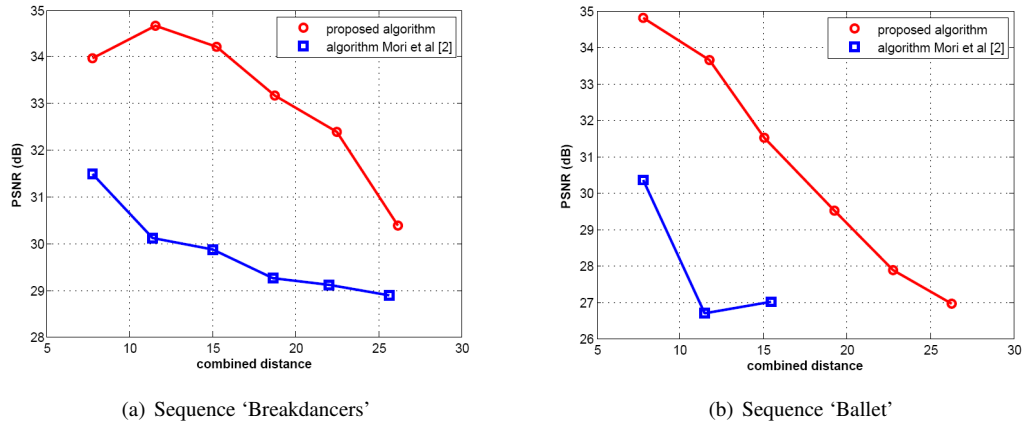
(a) Sequence 'Breakdancers'   (b) Sequence 'Ballet'

Fig. 5. Rendering quality as a function of the combined distance

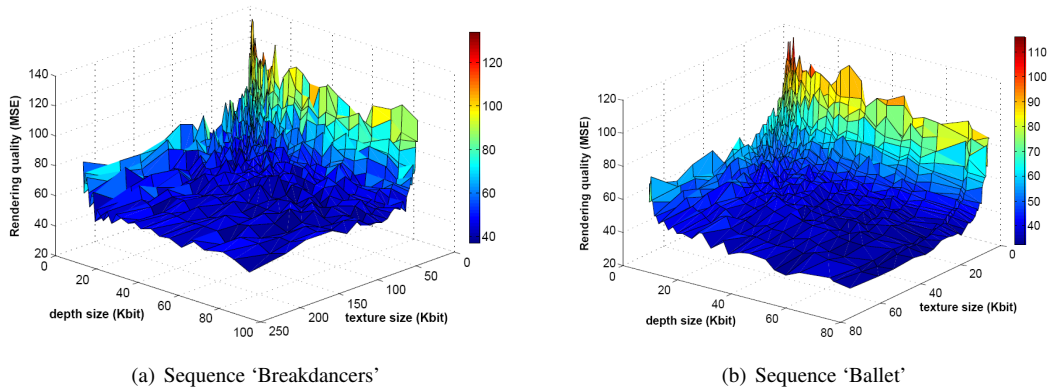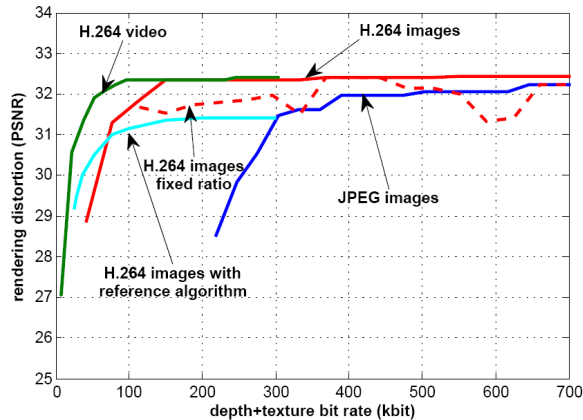(a) Sequence 'Breakdancers'   (b) Sequence 'Ballet'

Fig. 6. R-D surface for the sequences 'Breakdancers' and 'Ballet' with H.264 video compression

we have only noticed some differences on the edges of objects and the smoothness of the pictures.

Let us now investigate the influence of the coding on the rendering quality. Morvan *et al.* [4] have developed a method for calculating the optimal joint texture/depth quantization settings for the encoder. We have performed experiments in two ways. First, frame-based coding using either intra-coding with H.264 [9] or JPEG compression for images, allowing a fair comparison to [4]. Second, we have coded the surrounding camera streams with the regular settings of H.264 to compare the compression rate between video and images. In order to find the optimal joint quantization settings, the joint depth/texture Rate-Distortion (R-D) surface must first be created. Similar to [4], we have performed a full search to find the minimal distortion because we are only interested in the optimal settings and not in the complexity of the search. The rendering quality is expressed as a maximal PSNR for every joint bitrate. The R-D surface for H.264 video is illustrated in Fig. 6 and the rendering quality for JPEG and H.264 image and video encoding are depicted in Fig. 7. For comparison,

the performance of the reference rendering algorithm used in [4], based on [6], is also plotted. The quantization settings for the H.264 encoder start from $q_{min} = 23$ till $q_{max} = 51$. For JPEG encoding, we increment the quantizer setting q from 10 to 80 with steps of 5. The applied data sets are the 'Breakdancers' and 'Ballet' sequences. Each video contains 100 frames of texture and its associated depth maps with a resolution of 1024 by 768 pixels. The scene is recorded with 8 cameras, positioned along an arc spanning about 30° from one end to the other. The depth maps are created off-line and give an indication of the depth of each pixel in the image. For high-quality rendering the depth maps must be very accurate. The data sets satisfy this requirement.

From Fig. 7, it can be observed that for the same joint depth+texture bit rate, our algorithm achieves a higher PSNR. The large performance difference in Fig.7(b) occurs because the reference algorithm uses only one reference camera view to generate the interpolated image. Evidently, applying video coding instead of images achieves a higher compression factor for the same rendering quality. Also, for depth images,
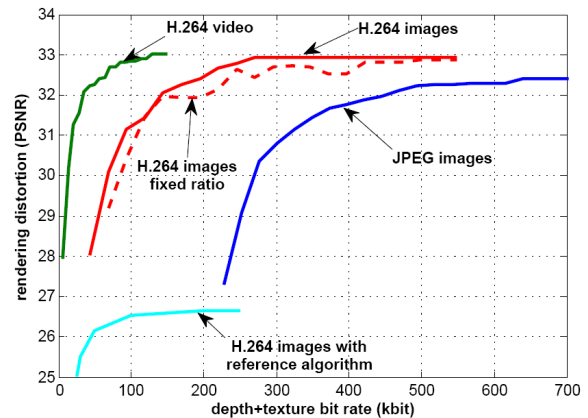
(a) Sequence 'Breakdancers'　　　　　　　　(b) Sequence 'Ballet'

Fig. 7. Rendering quality of various encoders with optimal settings compared to fixed ratio

H.264 is far superior to JPEG encoding. We have also explored the dependence between compression and rendering in the PSNR results. It was found that the maximal PSNR without any compression for the 'Breakdancers' and 'Ballet' scene are 32.3 and 33.0 dB, respectively. From Fig. 7, it can be seen that the rendering qualities are very close to those values, when using the optimal joint depth/texture quantization settings with a data rate higher than 200 kbit. This means that at those bit rates, the quality of the rendering algorithm is highly dominating the obtained PSNR and compression plays a far less relevant role.

Considering the mean ratio between texture and depth bit rates (with optimal settings), we have found that this ratio is 4.0 and 2.2 for H.264 intra-coding using the 'Breakdancers' and 'Ballet' scene, respectively. For H.264 video the mean ratio becomes 1.9 and 0.9. Although the mean ratio is highly scene dependent, this estimated ratios can be a starting point for quickly finding the optimal quantization settings.

4. CONCLUSIONS

We have presented a novel free-viewpoint rendering algorithm using DIBR, which clearly outperforms the existing proposals. The key to our proposal is that the depth and texture map are both warped in the first step and blending of the surrounding views are performed at a later stage. Consequently, errors in creating a virtual depth map are minimized and less warping is needed. After that, discontinuities and disocclusions are each processed in a separate pass. We have shown that our algorithm can handle the major problems of DIBR quite well. From an objective perspective it can be observed that our algorithm outperforms an earlier DIBR method when the relative distance to the reference cameras varies. Furthermore, we have demonstrated that using encoding, it is possible to compress the data considerably without much sacrificing the rendering quality. For future work, we are planning to im-

prove our disocclusions filling method, since we think that this will further enhance the perceptive rendering quality. In addition, we are interested in evaluating the rendering quality on synthetic data to obtain results independent of real-world image and depth maps acquisition problems.

5. REFERENCES

[1] Y. Morvan, *Acquisition, Compression and Rendering of Depth and Texture for Multi-View Video*, Ph.D. thesis, Eindhoven University of Technology, April 2009, to appear.

[2] S. Zinger, D. Ruijters, and P. H. N. de With, "iGLANCE project: free-viewpoint 3d video," in *International Conference on Computer Graphics, Visualization and Computer Vision 2009*, 2009.

[3] T. Duchamp K. Pulli, M. Cohen and W. Stuetzle, "View-based rendering: Visualizing real objects from scanned range and color data," in *In Eurographics Rendering Workshop*, 1997, pp. 23–34.

[4] Y. Morvan, D. Farin, and P. H. N. de With, "Joint depth/texture bit-allocation for multi-view video compression," in *Picture Coding Symposium (PCS)*, 2007.

[5] Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto, "View generation with 3d warping using depth information for ftv," in *3DTV08*, 2008, pp. 229–232.

[6] M. M. Oliveira, *Relief Texture Mapping*, Ph.D. thesis, University of North Carolina, mar 2000.

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

[8] L. McMillan and R. S. Pizer, "An image-based approach to three-dimensional computer graphics," Tech. Rep., 1997.

[9] x264 encoder, "Webpage title: x264 a free h264/avc encoder," 2007, http://developers.videolan.org/x264.html.