Object-based Layered Depth Images for improved virtual view synthesis in rate-constrained context

Vincent Jantet, Christine Guillemot, Luce Morin

OBJECT-BASED LAYERED DEPTH IMAGES FOR IMPROVED VIRTUAL VIEW SYNTHESIS IN RATE-CONSTRAINED CONTEXT

Vincent Jantet [1] *Christine Guillemot* [2] *Luce Morin* [3]

[1] ENS Cachan, Antenne de Bretagne – Campus de Ker Lann – 35170 Bruz, France
[2] INRIA Rennes, Bretagne Atlantique – Campus de Beaulieu – 35042 Rennes, France
[3] IETR - INSA Rennes – 20 avenue des Buttes de Coësmes – 35043 Rennes, France

ABSTRACT

Layered Depth Image (LDI) representations are attractive compact representations for multi-view videos. Any virtual viewpoint can be rendered from LDI by using view synthesis technique. However, rendering from classical LDI leads to annoying visual artifacts, such as cracks and disocclusions. Visual quality gets even worse after a DCT-based compression of the LDI, because of blurring effects on depth discontinuities. In this paper, we propose a novel object-based LDI representation, improving synthesized virtual views quality, in a rate-constrained context. Pixels from each LDI layer are reorganised to enhance depth continuity.

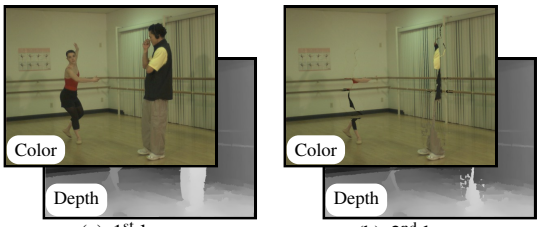
Index Terms— Video Coding, Multi-view Video, Layered Depth Video, Segmentation

1. INTRODUCTION

A multi-view video is a collection of video sequences for the same scene, synchronously captured by many cameras at different locations. Associated with a view synthesis method, a multi-view video allows the generation of virtual views of the scene from any viewpoint [1, 2]. This property can be used in a large diversity of applications [3], including Three-Dimensional TV (3DTV), Free Viewpoint Video (FTV), security monitoring, tracking and 3D reconstruction. However, multi-view videos generate very large amounts of data. This motivates the design of efficient compression algorithms [4].

The chosen compression algorithm is strongly dependent on the data representation and the view synthesis method. View synthesis techniques can be classified into two classes: Geometry-Based Rendering (GBR) techniques and Image-Based Rendering (IBR) techniques. GBR methods require detailed 3D models of the scene, which are difficult to estimate from real multi-view videos. These methods are thus more suitable for rendering synthetic data. IBR methods require some low-detailed geometric information associated with multi-view videos. These methods allow the generation of photo-realistic virtual views at the expense of the size of the acceptable navigation range for the virtual camera.

THIS WORK IS SUPPORTED BY THE FRENCH NATIONAL RESEARCH AGENCY AS PART OF PERSEE PROJECT (ANR-09-BLAN-0170)



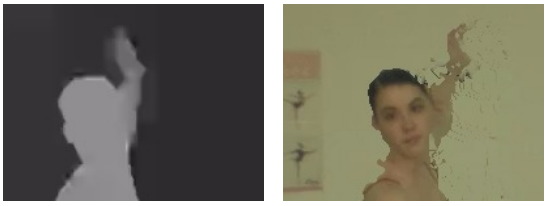
(a) 1st layer. (b) 2nd layer.

Fig. 1. Two first layers (color + depth map) of a classical LDI. Synthesized from "Ballet" [2], views 4–3–5 at $t = 0$, with incremental method [7].

The Layer Depth Image (LDI) representation [5, 6] is one of these IBR approaches. It extends the 2D+Z representation, but instead of representing the scene with an array of depth pixels (pixel color with associated depth values), each position in the array may store several depth pixels, organised into layers. This representation is shown in Figure 1. It efficiently reduces the multi-view video bitrate, and it offers photo-realistic rendering, even with complex scene geometry.

Various approaches to LDI construction have been proposed [6, 7, 8]. All of them organize layers by visibility. The first layer contains all pixels visible from the viewpoint, it is the classical 2D image. The other layers contain pixels in the camera scope, but hidden by objects in previous layers. With this organisation, each layer may contain pixels from the background and pixels from objects in a same neighbourhood, creating texture and depth discontinuities within the same layer. These discontinuities are blurred during layers compression with a classical DCT-based scheme. This blurring of depth discontinuities, shown in Figure 2(a), significantly reduces the rendering quality obtained by classical rendering methods. For example, Figure 2(b) shows artifacts on objects boundaries, rendered by the MPEG-VSRS rendering method [9].

In this paper, we present a novel object-based LDI representation to address both compression and rendering issues. This object-based LDI is more tolerant to compression



(a) Compressed depth map. "Ballet", view 4, MVC (QP=48).

(b) Synthesized virtual view. MPEG-VSRS, camera 3.

Fig. 2. Impact of depth map compression on edge rendering.

artifacts, and compatible with fast mesh-based rendering. Section 2 presents a method for pixels classification into object-based layers, using a region growing algorithm. Section 3 explains how to use inpainting methods to fill holes in the background layer. Section 4 describes how to compress LDI layers, using the MPEG/MVC software. Section 5 briefly presents two rendering methods which have been implemented. Section 6 exposes compression results for both the classical and object-based LDI representations.

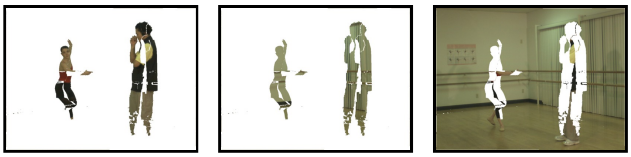
2. OBJECT-BASED LDI

In order to overcome artifacts which result from depth discontinuities, in particular after depth map compression, a novel object-based LDI representation is proposed. This representation organises LDI pixels into two separate layers (foreground and background) to enhance depth continuity. If depth pixels from a real 3D object belong to the same layer, then compression is more efficient thanks to higher spatial correlation which improves effective spatial prediction of texture and depth map. Moreover, these continuous layers can be rendered efficiently (in terms of both speed and reduced artifacts) by using mesh-based rendering techniques.

The number of layers inside a LDI is not the same for each pixel position. Some positions may contain only one layer, whereas some other positions may contain many layers (or depth pixels). If several depth pixels are located at the same position, the closest belongs to the foreground, visible from the reference viewpoint, whereas the farthest is assumed to belong to the background. If there is only one pixel at a position, it is a visible background pixel, or a foreground pixel in front of an unknown background.

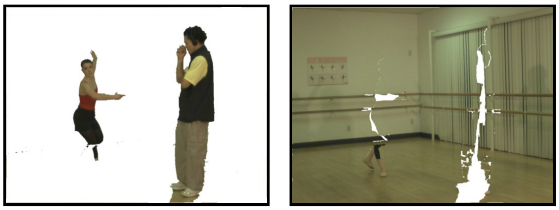
This section presents a background-foreground segmentation method based on a region growing algorithm, which allows organising LDI's pixels into two object-based layers.

First, all positions p containing several layers are selected from the input LDI. They define a region R, shown in Figure 3, where foreground and background pixels are easily identified. Z_p^{FG} denotes foreground depth, and Z_p^{BG} denotes background depth at position p. For each position q outside the region R, the pixel P_q has to be classified as a foreground or background pixel.



(a) Foreground. (b) Background. (c) Unclassified.

Fig. 3. Initialising state of the region growing algorithm.



(a) Foreground. (b) Background.

Fig. 4. Final layer organisation with the region growing classification method.

The classified region grows pixel by pixel, until the whole image is classified, as shown in Figure 4. For each couple of adjacent positions (p, q) around the border of region R such that p is inside R and q is outside R, the region R is expanded to q by classifying the pixel P_q according to its depth Z_q. For classification, Z_q is compared to background and foreground depths at position p. An extra depth value is then given to position q, so that q is associated with both a foreground and a background depth value.

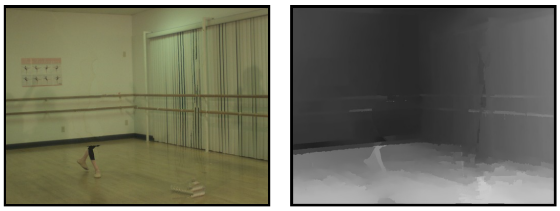
$$P_q \in \begin{cases} \text{foreground} \quad \text{if } (Z_p^{BG} - Z_q) > (Z_q - Z_p^{FG}) \\ \qquad \text{so } Z_q^{FG} = Z_q \text{ and } Z_q^{BG} = Z_p^{BG} \\ \text{background} \quad \text{if } (Z_p^{BG} - Z_q) < (Z_q - Z_p^{FG}) \\ \qquad \text{so } Z_q^{FG} = Z_p^{FG} \text{ and } Z_q^{BG} = Z_q \end{cases}$$

3. BACKGROUND FILLING BY INPAINTING

Once the foreground/background classification is done, the background layer is most of the time not complete (see Figure 4(b)). Some areas of the background may not be visible from any input view. To reconstruct the corresponding missing background texture, one has to use inpainting algorithms on both texture and depth map images. The costly inpainting algorithm is processed once, during the LDI classification, and not during each view synthesis. Figure 5 shows the inpainted background with the Criminisi's method [10].

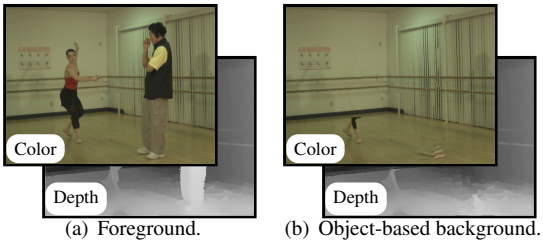
4. COMPRESSION

Both classical LDI and object-based LDI are compressed using the Multi-view Video Codec (MVC) [9], both for texture



(a) Texture. (b) Depth map.

Fig. 5. Background layer obtained after texture and depth map inpainting with the Criminisi's method [10].



(a) Foreground. (b) Object-based background.

Fig. 6. Finals layers of an object-based LDI.

layers, and for depth layers. The MVC codec, an amendment to H.264/MPEG-4 AVC video compression standard, is DCT-based and exploits temporal, spatial and inter-layer correlations. However, MVC does not deal with undefined regions on LDI layers. To produce complete layers, each layer is filled in with pixels from the other layer, at the same position, as shown in Figure 6. This duplicated information is detected by the MVC algorithm, so that it is not encoded into the output data flow and it can be easily removed during the decoding stage.

5. RENDERING

There exists a number of algorithms to perform view rendering from a LDI. This section briefly presents the two methods which have been implemented, focusing respectively on efficiency and quality.

The fastest method transforms each continuous layer into a mesh, which is rendered with a 3D engine, as shown in Figure 7. The foreground mesh is transparent on background region in order to avoid stretching around objects boundaries. Our first experiments, with this method, have shown the feasibility of real time rendering for an eight-views autostereoscopic display.

The second method improves the visual quality of synthesized views by using a point-based projection. It combines the painter's algorithm proposed by McMillan [11], and diffusion-based inpainting constrained by epipolar geometry. Remaining disocclusions areas are filled in with background texture. Figure 8 presents rendering results for both classical and object-based LDI.

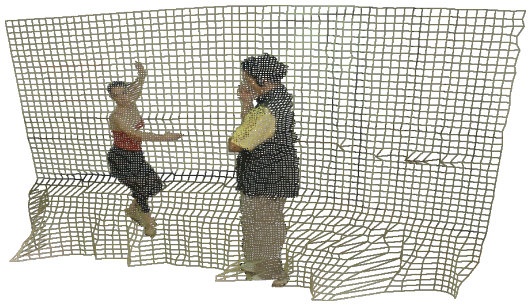


Fig. 7. Fast 3D rendering of a high detailed foreground mesh, onto a low detailed background mesh.

(a) Classical LDI. (b) Object-based LDI.

Fig. 8. Rendering comparison between classical and object-based LDI.

6. RESULTS

The rendered quality of object-based LDI is compared with classical LDI on one side, and state-of-the-art MPEG compression techniques on the other side. Images are taken from "Ballet" data sets, provided by MSR [2]. Only frames for time $t = 0$ are considered.

In the first place, a LDI restricted to two layers, is constructed from three input views: the reference view 4 and side views 3 and 5 alternatively. To deal with unrectified camera sets and reduce correlation between layers, we use the Incremental LDI construction algorithm described in [7]. The corresponding object-based LDI is obtained by applying our region growing classification method on the classical LDI.

Classical LDI and object-based LDI are compressed using the MVC algorithm, as explained in section 4. Several quantization parameters were used, from QP=18 to QP=54, producing compressed output data flows with bit-rates going from 1 Mbit/s to 25 Mbit/s. These compressed data flows are used to synthesize virtual views onto viewpoint 6, using the pixel-based projection method.

In the second place, the state-of-the-art method for multiview video coding is used with the same input data. Views 1, 3, 5 and 7 are coded with the MVC algorithm with various quantization parameters, then the compressed views 5 and 7 are used to synthesize virtual views onto viewpoint 6, using the MPEG/VSRS software [9].

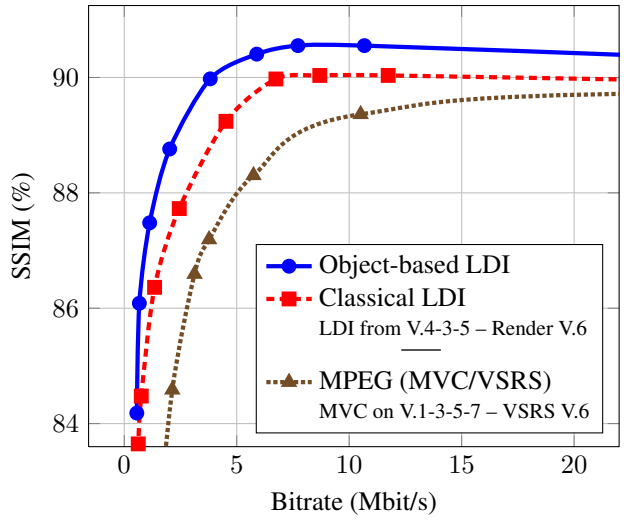


Fig. 9. Rate distortion curves firstly for LDI (object-based or not) compressed by MVC and rendered by our point-based projection, and secondly for multi-view video compressed by MVC and rendered with VSRS algorithm.

Finally, all synthesized views are compared to the original view 6, using the SSIM comparison metrics. Figure 9 presents all the results as three rate distortion curves. For each quantization parameter, object-based LDI can be better compressed than classical LDI, resulting in a smaller bitrate. The rendering quality is also better, resulting in a higher SSIM for the same quantization parameter. Combining these two advantages, the rate distortion curve for the object-based LDI is higher than the one for classical LDI, for every bitrate.

7. CONCLUSION

This paper presents a novel object-based LDI and its benefits for 3D video compression and virtual view rendering. The proposed method to construct these object-based LDI is a foreground and background classification, based on a region growing algorithm which ensures depth continuity of the layers.

These object-based LDI have some attractive features. The reduced number of depth discontinuities in each layer improves compression efficiency and minimizes compression artifacts for a given bitrate. The rendering stage can just be performed with two meshes, moving computations to the GPU, but some small texture-stretching may appear. These artifacts can be avoided by performing ordered projection, which removes cracks and fills disocclusions with background texture.

8. REFERENCES

[1] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Computer graphics and interactive techniques, SIGGRAPH*, New York, NY, USA, 2001, pp. 425–432, ACM.

[2] C.-L. Zitnick, S.-B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, 2004.

[3] A. Smolic, K. Müller, N. Stefanoski, J. Ostermann, A. Gotchev, G.B. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3dtv - a survey," *Circuits and Systems for Video Technology, IEEE Trans. on*, vol. 17, no. 11, pp. 1606–1621, Nov. 2007.

[4] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *Circuits and Systems for Video Technology, IEEE Trans. on*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

[5] J. Shade, S. Gortler, L. He, and R. Szeliski, "Layered depth images," in *Computer graphics and interactive techniques, SIGGRAPH*, New York, NY, USA, 1998, pp. 231–242, ACM.

[6] S.-U. Yoon, E.-K. Lee, S.-Y. Kim, and Y.-S. Ho, "A framework for representation and processing of multi-view video using the concept of layered depth image," *Signal Processing Systems for Signal Image and Video Technology, VLSI Journal of*, vol. 46, pp. 87–102, 2007.

[7] V. Jantet, L. Morin, and C. Guillemot, "Incremental-ldi for multi-view coding," in *The True Vision, 3DTV Conf.*, Potsdam, Germany, May 2009, pp. 1–4.

[8] X. Cheng, L. Sun, and S. Yang, "Generation of layered depth images from multi-view video," *Image Processing ICIP, IEEE Inter. Conf. on*, vol. 5, pp. 225–228, Oct. 2007.

[9] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, "Reference softwares for depth estimation and view synthesis," Apr. 2008.

[10] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Computer Vision and Pattern Recognition CVPR, IEEE Computer Society Conf. on*, June 2003, vol. 2, pp. 721–728.

[11] L. McMillan, "A list-priority rendering algorithm for redisplaying projected surfaces," Tech. Rep., Chapel Hill, NC, USA, 1995.