Incremental-LDI for Multi-View Coding

Vincent Jantet, Luce Morin, Christine Guillemot

INCREMENTAL-LDI FOR MULTI-VIEW CODING

Vincent Jantet [1], Luce Morin [2], and Christine Guillemot [1]

[1] INRIA Rennes, Bretagne Atlantique - Campus de Beaulieu - 35042 Rennes, France
[2] IETR - INSA Rennes - 20 avenue des Buttes de Coësmes - 35043 Rennes, France

ABSTRACT

This paper describes an Incremental algorithm for Layer Depth Image construction (I-LDI) from multi-view plus depth data sets. A solution to sampling artifacts is proposed, based on pixel interpolation (inpainting) restricted to isolated unknown pixels. A solution to ghosting artifacts is also proposed, based on a depth discontinuity detection, followed by a local foreground / background classification. We propose a formulation of warping equations which reduces time consumption, specifically for LDI warping. Tests on Breakdancers and Ballet MVD data sets show that extra layers in I-LDI contain only 10% of first layer pixels, compared to 50% for LDI. I-LDI Layers are also more compact, with a less spread pixel distribution, and thus easier to compress than LDI Visual rendering is of similar quality with I-LDI and LDI.

***Index Terms*—** Video Coding, Virtual Reality, Multi-view Video, Layered Depth Video, View Interpolation

1. INTRODUCTION

A multi-view video is a collection of video sequences captured for the same scene, synchronously 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. The huge amount of data contained in a multi-view sequence needs an efficient compression [4].

The compression algorithm is strongly linked to the data representation and the view synthesis methods. View synthesis approaches can be classified in two classes. Geometry-Based Rendering (GBR) approaches use a detailed 3D model of the scene. These methods are useful with synthetic video data but they become inadequate with real multi-view videos, where 3D models are difficult to estimate. Image-Based Rendering (IBR) approaches are an attractive alternative to GBR. Using the acquisition videos accompanied by some low-detailed geometric information, they allow the generation of photo-realistic virtual views.

The Layer Depth Image (LDI) representation [5, 6] is one of these IBR approaches. In this representation, pixels are no more composed by a single color and a single depth value, but can contain several colors and associated depth values. This representation reduces efficiently the multi-view video size, and offers a fast photo-realistic rendering, even with complex scene geometry.

Various approaches to LDI compression have been proposed [6, 7, 8], based on classical LDI's layers constructions [6, 9]. The problem is that layers generated are still correlated, and some pixels are redundant between layers. This paper proposes an Incremental LDI construction (I-LDI) to reduce the inter-layer correlation. The number of layers is significantly reduced for an equivalent final rendering quality. Techniques are then proposed to overcome visual artifacts, like sampling holes and ghosting artifacts [2, 9, 10].

2. LAYERS GENERATION

LDI can be generated from real multi-view + depth video sequences by using a warping algorithm. This algorithm, detailed in section 3, uses a view and the associated depth map to generate a new viewpoint of the scene. However, this classical LDI construction (described in section 2.1) usually produces some correlations between layers. An alternative construction algorithm that we call I-LDI (for Incremental LDI) is described in section 2.2.

2.1. Classical LDI construction

Given a set of viewpoints and one depth map per view, the classical algorithm for LDI construction [6, 9] proceeds in three steps, summarized in figure 1. First, an arbitrary viewpoint is chosen as the reference viewpoint (it is usually chosen among input viewpoints). Then, each (or a subset of) input views is warped onto this reference viewpoint, using the warping algorithm described section 3. Finally, all these warped views are merged into a single LDI model, where each pixel position may contains many layered depth pixels.

There are many merging policies depending on the application. Keeping all depth pixels results in unnecessarily high

Acknowledgement: This work has been funded by the brittany region in the context of the french national project Futurimage.

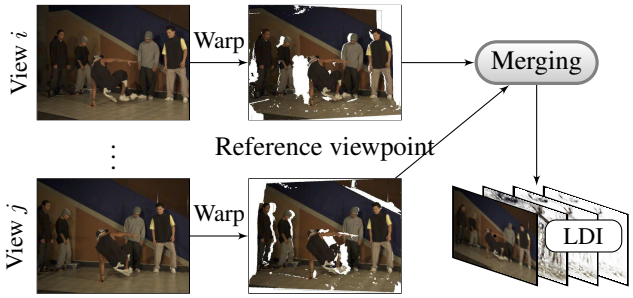

Fig. 1. Classical LDI construction scheme.

redundant layers. It is preferable to keep at each pixel location, only pixels whose depth value significantly differs from that of the others. We use a threshold Δ_d on the depth value to eliminate pixels with very similar depth value.

The first three layers of such a LDI are presented in figure 2. Layered pixels are ordered on their depth value. The first layer is composed of pixels with smallest depth, the second layer contains pixels with second smallest depth, and so on. We observe that, except for the first one, layers are partially empty, but non-empty pixels are sparsely distributed all over the layer. Furthermore, many pixels are redundant between the layers. These characteristics make it difficult to efficiently compress the LDI.

(a) 1st layer (b) 2nd layer (c) 3rd layer

Fig. 2. First layers of an LDI frame. All 8 inputs views are used for the generation. ($\Delta_d = 0.1$)

2.2. I-LDI construction

To reduce correlation between LDI layers, we propose an incremental construction scheme, illustrated in figure 3, based on residual information extraction [10]. First, the reference view is used to create an I-LDI with only one layer (the view itself). Then, this I-LDI is warped iteratively on every other viewpoint (in a fixed order), and we use a logical exclusion difference between the real view and the warped I-LDI to compute the residual information. This information is warped back into the reference viewpoint and inserted in the I-LDI layers. By this method, only required residual information from side views is inserted, and no pixels from already defined areas are added to the L-LDI. On the other side, all the information present in the MVD data is not inserted in the I-LDI.

The first three layers of such an I-LDI are presented in figure 4. Compared to LDI layers, I-LDI layers contain fewer pixels, and these pixels are grouped in connected clusters.

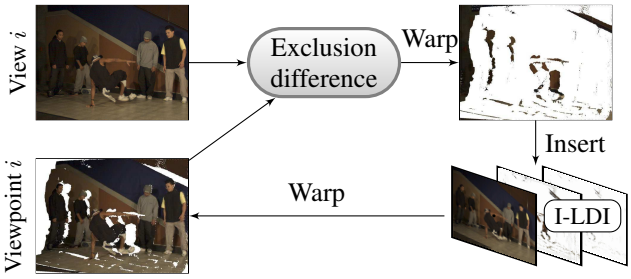

Fig. 3. Step of I-LDI construction for view i, with residual information extraction.

Some other characteristics of these layers are discussed in section 6.

(a) 1st layer (b) 2nd layer (c) 3rd layer

Fig. 4. Firsts layers of an I-LDI frame. All 8 inputs views are used for the generation, in a B-hierarchic order.

3. WARPING

In this section, we explicit the equations used in the warping process. Let $(X, Y, Z, 1)$ be a 3D point in homogeneous coordinates, which is projected onto pixel $p_1 = (x_1, y_1, 1)$ in view V_1 and pixel $p_2 = (x_2, y_2, 1)$ in view V_2. Pixel coordinates p_i in view V_i are derived from the projection equations:

$$\omega_i \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} C^i_{1,1} & C^i_{1,2} & C^i_{1,3} & C^i_{1,4} \\ C^i_{2,1} & C^i_{2,2} & C^i_{2,3} & C^i_{2,4} \\ C^i_{3,1} & C^i_{3,2} & C^i_{3,3} & C^i_{3,4} \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where C^i is the 3×4 projection matrix depending on the viewpoint i, and ω_i is an arbitrary scale factor. Knowing both camera parameters and the depth map Z_{p_1} associated to view V_1, the warping equations provide p_2 coordinates as a function of p_1 and Z_{p_1}. Warping algorithm works in two steps. The first step uses p_1 and Z_{p_1} to estimate the 3D point (X, Y, Z_{p_1}). The second step uses this estimated 3D point to evaluate the pixel position p_2 in the new viewpoint image.

To solve the first step, we need to inverse the projection equation (1). Let (L_1), (L_2) and (L_3) be the three linear equations corresponding to the matrix notation (1) which are combined as follows:

$$\begin{aligned} & (C^1_{2,2} \cdot 1 - C^1_{3,2} \cdot y_1) \cdot (L_1) \\ + \quad & (C^1_{3,2} \cdot x_1 - C^1_{1,2} \cdot 1) \cdot (L_2) \\ + \quad & (C^1_{1,2} \cdot y_1 - C^1_{2,2} \cdot x_1) \cdot (L_3) \end{aligned} \quad (2)$$

Unknown parameters Y and ω_1 can then be eliminated by simplifying equation (2) giving:

$$\begin{array}{rr} & X \cdot \det\left(\left[C^1_{.,1}\middle|C^1_{.,2}\middle|p_1\right]\right) \\ + & Z_{p_1} \cdot \det\left(\left[C^1_{.,3}\middle|C^1_{.,2}\middle|p_1\right]\right) \\ + & \det\left(\left[C^1_{.,4}\middle|C^1_{.,2}\middle|p_1\right]\right) \end{array} = 0 \qquad (3)$$

where $C^1_{.,i}$ is the i^{th} column of the C^1 matrix. A direct form for the point's abscissa X is given by equation (3), and the same kind of equation could be written to estimate Y.

Compared to a classical matrix inversion, some coefficients of the equation (3) only depend on p_1 and do not change during warping of all layered depth pixels at a same pixel location. By implementing this optimization, we reduce by 49% the number of multiplications needed to warp a full LDI, and almost as much for it's time consumption.

Each pixel is warped independently of the others. To avoid the use of a depth buffer, we implemented the McMillan's priority order list algorithm [11]. Warping results are shown in figure 5(a).

4. HOLES FILLING BY INPAINTING

Directly applying warping equations may cause some visual artifacts, due mostly to disocclusion and sampling [2, 9, 10]. This section describes our simple inpainting method to fill sampling holes, visible in figure 5(a).

Let V_p be the pixel color at the p position, and W_p a neighborhood around p. Undefined pixels p can be interpolated as:

$$V'_p = \frac{1}{k} \sum_{q \in W_p} V_q \qquad (4)$$

where k is the number of defined pixels within W_p.

This inpainting solution is used both during the rendering stage, and during the I-LDI construction. During the rendering stage, it improves the visual quality by interpolating all missing pixels. During the I-LDI construction, it is used carefully to fill only the sampling holes, and to leave disocclusion areas unchanged. Results are shown in figure 5(b). If a pixel is undefined due to a sampling effect, it should be surrounded by many defined pixels, which mean a high k value. If the pixel is undefined due to a large disocclusion area, it should be surrounded by many undefined pixels, which mean a low k value. The classification is done by comparing k with a threshold Δ_k.

5. GHOSTING ARTIFACTS REMOVAL

In real pictures, pixels along object boundaries receive the contribution from both foreground and background colors. Using these blended pixels during the rendering stage results in ghosting artifacts (visible in figure 6(a)). We remove these blended pixels from the reference view before I-LDI construction. Their color is thus imported by side cameras during the I-LDI construction.

(a) Basic Warping (b) Sampling holes filled

Fig. 5. View warping and sampling holes filling

Blended pixels in a view can be identified by an edge detector performed on the associated depth map. Let p be a pixel position, we estimate the depth mean $\bar{d}_p$ and depth variance v_p within a neighborhood W_p around p. Pixels near a boundary have a high variance, but among these pixels, only those from the background side of a boundary may cause a visible artifact. We then remove pixels p such $v_p > \Delta_v$ and $d_p > \bar{d}_p$ where Δ_v is a threshold.

The result of our ghosting removal method is visible in figure 6. The silhouette behind the person is erased.

(a) Without boundaries detection. (b) With boundaries detection.

Fig. 6. Ghosting artifacts removal results from rendering view. (W_p is a 11×11 window)

6. EXPERIMENTAL RESULTS

Experiments have been conducted on Breakdancers and Ballet data sets from MSR [2]. Parameters of the 8 acquisition cameras and all associated depth maps are already estimated and provided within the data. The viewpoint number 4 is considered as the reference viewpoint. Only frames for time $t = 0$ are considered.

For the LDI construction, all 8 acquired views are warped into the reference viewpoint. A small merging threshold value $\Delta_d = 0.1$ is used in following comparisons. For the I-LDI construction, views are inserted in a B-hierarchical order (4; 0; 7; 2; 6; 1; 5; 3). Thresholds are set by experiments: $\Delta_v = 20$ for boundary detection and $\Delta_k = 60\% \cdot N$ for inpainting, where N is the number of pixels within the W_p window. Both inpainting and boundary detection are done within W_p a 11×11 window.

All 8 input views are used, but all pixels from each view are not inserted in the LDI. Because of the depth threshold in the LDI construction, and of the exclusion difference in I-LDI construction, some pixels from side views are ignored.

Figure 7 presents the ratio of pixels from each view which is really inserted in the LDI. We can observe that few pixels are inserted from view 5 to 8, means these views become almost useless with the I-LDI construction scheme. Using only a subset of acquired views (the reference and the extreme views) provides almost the same I-LDI layers.

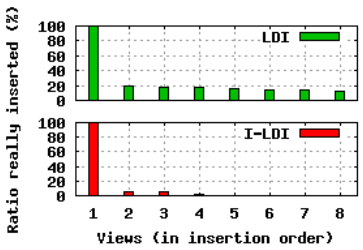

Fig. 7. Utilization rate of acquired views during layers construction

Figure 8 shows the ratio of defined pixels in each layers for both LDI and I-LDI construction schemes. For both constructions, the first layer contains 100% of it's pixels, and differences appear for extra layers. For the LDI, extra layers represent more than 50% of the size (in number of pixels) of the first layer, whereas for the I-LDI, extra layers represent less than 10%. Layers beyond the 3rd one are quite empty and can be ignored. The visual rendering is of similar quality with both LDI and I-LDI construction scheme. Local rendering artifacts may appear, depending on views insertion order.

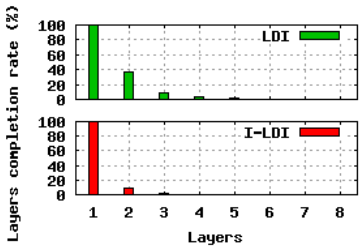

Fig. 8. Layers completion rate for LDI and I-LDI

7. CONCLUSIONS AND FUTURE WORK

This paper presents an incremental procedure to generate LDI from natural multi-view images. The minimum information to fill disocclusion areas is inserted into LDI layers which makes layers easier to compress. They contain 80% less pixels, and with a more compact distribution. To overcome visible artifacts, some simple solutions have been proposed. The sampling holes filling by pixel interpolation provides good results. The ghosting removal by depth discontinuity detection may cause some luminosity discontinuity between textures from different views.

In future work, we will investigate an improved disocclusion detection to insert into the I-LDI all occluded textures. An alpha merging approach will be used to reduce luminosity discontinuity artifacts. Finally, the compression stage will be investigated with a full video sequence.

8. REFERENCES

[1] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 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 Transactions 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 Transactions on*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

[5] J. Shade, S. Gortler, L. He, and R. Szeliski, "Abstract layered depth images," 1998.

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

[7] S.-U. Yoon, E.-K. Lee, S.-Y. Kim, Y.-S. Ho, K. Yun, S. Cho, and N. Hur, "Coding of layered depth images representing multiple viewpoint video," 2006.

[8] J. Duan and J. Li, "Compression of the layered depth image," *Image Processing, IEEE Transactions on*, vol. 12, no. 3, pp. 365–372, Mar. 2003.

[9] X. Cheng, L. Sun, and S. Yang, "Generation of layered depth images from multi-view video," *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 5, pp. V –225–V –228, 16 2007-Oct. 19 2007.

[10] K. Müller, A. Smolic, K. Dix, P. Kauff, and T. Wiegand, "Reliability-based generation and view synthesis in layered depth video," *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, pp. 34–39, Oct. 2008.

[11] L. Mcmillan, "A list-priority rendering algorithm for redisplaying projected surfaces," Tech. Rep., 1995.