Inter-camera Coding of Multi-view Video Using Layered Depth Image Representation

Seung-Uk Yoon[1], Eun-Kyung Lee[1], Sung-Yeol Kim[1], Yo-Sung Ho[1], Kugjin Yun[2], Sukhee Cho[2], and Namho Hur[2]

[1] Department of Information and Communications
Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Republic of Korea
{suyoon, eklee78, sykim75, hoyo}@gist.ac.kr
[2] Broadcasting System Research Group
Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Republic of Korea
{kjyun, shee, namho}@etri.re.kr

Abstract. The multi-view video is a collection of multiple videos, capturing the same scene at different viewpoints. If we acquire multi-view videos from multiple cameras, it is possible to generate scenes at arbitrary view positions. It means that users can change their viewpoints freely and can feel visible depth with view interaction. Therefore, the multi-view video can be used in a variety of applications including three-dimensional TV (3DTV), free viewpoint TV, and immersive broadcasting. However, since the data size of the multi-view video linearly increases as the number of cameras, it is necessary to develop an effective framework to represent, process, and display multi-view video data. In this paper, we propose inter-camera coding methods of multi-view video using layered depth image (LDI) representation. The proposed methods represents various information included in multi-view video hierarchically based on LDI. In addition, we reduce a large amount of multi-view video data to a manageable size by exploiting spatial redundancies among multiple videos and reconstruct the original multiple viewpoints successfully from the constructed LDI.

Keywords: multi-view video coding, layered depth image, MPEG.

1 Introduction

The multi-view video is a collection of multiple videos capturing the same scene at different camera locations. If we acquire multi-view videos from multiple cameras, it is possible to generate video scenes from any viewpoints, which means that users can change their views within the range of captured videos and can feel the visible depth with view interaction. The multi-view video can be used in a variety of applications including free viewpoint video (FVV), free viewpoint TV (FTV), three-dimensional TV (3DTV), surveillance, and home entertainment.

Although the multi-view video has much potential for a variety of applications, one big problem is a huge amount of data. In principle, the multi-view video data

Y. Zhuang et al. (Eds.): PCM 2006, LNCS 4261, pp. 432–441, 2006.

are increasing linearly as the number of cameras; therefore, we need to encode the multi-view video data for efficient storage and transmission. Hence, it has been perceived that multi-view video coding (MVC) is a key technology to realize those applications.

ISO/IEC JTC1/SC29/WG11 Moving Picture Experts Group (MPEG) has been recognized the importance of MVC technologies, and an ad hoc group (AHG) on 3-D audio and visual (3DAV) has been established since December 2001. Four main exploration experiments (EE) on 3DAV were performed from 2002 to 2004: EE1 on omni-directional video, EE2 on FTV, EE3 on coding of stereoscopic video using multiple auxiliary components (MAC), and EE4 on depth/disparity coding for 3DTV and intermediate view interpolation. In response to the Call for Comments issued in October 2003, a number of companies have expressed their interests for a standard that enables FTV and 3DTV. After MPEG called interested parties to bring evidences on MVC technologies in October 2004 [1], some evidences were recognized in January 2005 [2] and a Call for Proposals (CfP) on MVC has been issued in July 2005 [3]. Then, the responses to the CfP has been evaluated in January 2006 [4].

In this paper, we propose representation and inter-camera coding methods of multi-view video using the concept of layered depth image (LDI) [5], which is an efficient image-based rendering (IBR) technique. Based on the proposed framework [6], we generate LDI frames from the natural multi-view video, which is different from the previous LDI generation methods mainly using 3-D synthetic objects. We also describe coding methods for the number of layer (NOL) and residual data of the constructed LDI.

The paper is organized as follows. In Section 2, we review our framework [6] for representing multi-view video and explain the generation procedure of LDI from the natural multi-view video. Then, we describe encoding methods of NOL and residual data in Section 3. After experimental results and analysis are presented in Section 4, we draw conclusions in Section 5.

2 Representation of Multi-view Video Based on LDI

An important aim of the multi-view video is to provide view-dependant scenes from the pre-captured multiple videos. This goal is similar to the functionality of image-based rendering (IBR) techniques; the novel view generation using 2-D input images.

Traditionally, IBR has been mainly applied to static objects, architectures, and sceneries. However, there have been several approaches to extend it to the dynamic scenes [7], which are called video-based rendering. Kanade et al. [8] extract a global surface representation at each time frame using 51 cameras (512 x 512) in a geodesic dome. They tried to construct 3-D objects from captured images and render them at interactive rate. Matusik et al. [9] use the images from four calibrated cameras (256 x 256) to compute and shade visual hulls. They could render 8000 pixels of the visual hull at about 8 fps. Carranza et al. [10] used seven inward looking synchronized cameras (320 x 240) distributed

around a room to capture 3D human motion. They used a 3-D human model as a prior to compute 3D shape at each time frame. Yang et al. [11] designed an 8 x 8 camera array (320 x 240) for capturing dynamic scenes. Instead of storing and rendering the data, they transmit only the rays necessary to compose the desired virtual view. In their system, the camera capture rate is 15 fps, and the interactive viewing rate is 18 fps. In 2004, Zitnick et al. [7] proposed efficient view interpolation and rendering methods using multiple videos acquired from eight cameras. However, these approaches are mainly focusing on the real-time rendering rather than the representation and encoding of a huge amount of input video data.

Inspired by these ideas, we have proposed a framework for representation and encoding of multi-view video using the concept of LDI [6]. In our framework, we have obtained LDI frames from natural multi-view video test sequences by 3-D warping using the given depth images. As the concept of LDI, it is possible to generate LDI by storing intersecting points with color and depth. However, this method can only be applied to 3-D computer graphics (CG) models because rays cannot go through the real object. Therefore, we have exploited multiple color and depth images to construct LDI for natural scenes [5][6] and have used the modified LDI data structure [12].

In the previous work [6], the following incremental 3-D warping equation [5] has been used in the warping stage. When $C_1 = V_1 \cdot P_1 \cdot A_1$, $C_2 = V_2 \cdot P_2 \cdot A_2$, the transform matrix $T_{1,2} = C_2 \cdot C_1^{-1}$. C is a camera matrix, V is the viewport matrix, P is the projection matrix, and A is the affine matrix.

$$T_{1,2} \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_2 \cdot w_2 \\ y_2 \cdot w_2 \\ z_2 \cdot w_2 \\ w_2 \end{bmatrix} = T_{1,2} \cdot \begin{bmatrix} x_1 \\ y_1 \\ 0 \\ 1 \end{bmatrix} + z_1 \cdot T_{1,2} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{1}$$

where (x_1, y_1) is the pixel location in C_1, z_1 is the depth at (x_1, y_1). (x_2, y_2) is the warped pixel location in C_2.

However, the problem is that these matrices are designed for 3-D graphics models; there is no clear definition of them for real scenes or objects. Since the camera matrix C was only appropriate for 3-D synthetic scenes, we have calculated a new camera matrix from the given camera parameters contained in multi-view video test sequences. In this paper, we have modified the previous camera matrix C because it does not properly consider the intrinsic characteristics of multiple cameras. It has only considered affine transformations of each camera. The modified camera matrices and the 3-D warping equation are as follows.

$$\dot{C_1} = \dot{A_1} \cdot \dot{E_1}, \dot{C_2} = \dot{A_2} \cdot \dot{E_2}, \dot{T_{1,2}} = \dot{C_2} \cdot \dot{C_1^{-1}} \tag{2}$$

$$\dot{A} = \begin{bmatrix} -f_{s_x} & \theta & t_x \\ 0 & -f_{s_y} & t_y \\ 0 & 0 & 1 \end{bmatrix}, \dot{E} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{bmatrix} \tag{3}$$

$$\dot{T_{1,2}} \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_2 \cdot w_2 \\ y_2 \cdot w_2 \\ z_2 \cdot w_2 \\ w_2 \end{bmatrix} = \dot{T_{1,2}} \cdot \begin{bmatrix} x_1 \\ y_1 \\ 0 \\ 1 \end{bmatrix} + z_1 \cdot \dot{T_{1,2}} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{4}$$

where f_{s_x}, f_{s_y} are focal length, s_x, s_y, are scaling factors, t_x, t_y, are positions of the focal center, and θ is the skew angle. $\dot{A}$ defines the intrinsic camera parameters and $\dot{E}$ is an affine transform matrix expressing a rotation and a translation. Finally, we add an additional row to make a homogeneous 4 x 4 camera matrix $\dot{C}$, because $\dot{A} \cdot \dot{E}$ becomes a 3 x 4 matrix.

3 Inter-camera Coding of Multi-view Video Using LDI

3.1 Color and Depth Components

After generating LDI frames from the natural multi-view video with depth, we separate each LDI frame into three components: color, depth, and the number of layers (NOL). Specifically, color and depth component consists of layer images, respectively. The maximum number of layer images is the same as the total number of views. In addition, residual data should be sent to the decoder in order to reconstruct multi-view images. Color and depth components are processed by data aggregation/layer filling to apply H.264/AVC. NOL could be considered as an image containing the number of layers at each pixel location. Since the NOL information is very important to restore or reconstruct multi-view images from the decoded LDI, it is encoded by using the H.264/AVC intra mode. Finally, the residual data, differences between the input multi-view video and reconstructed ones, are encoded by using the H.264/AVC intra mode.

The data aggregation or layer filling is used in the preprocessing stage [12]. Although H.264/AVC is powerful to encode rectangular images, it does not support shape-adaptive encoding modes. Therefore, we need to aggregate each layer image and then fill the empty locations with the last pixel value of the aggregated image [13]. One problem of the data aggregation is that the resultant images have severely different color distributions. It leads to poor coding efficiency because the prediction among aggregated images is difficult.

The second method is called the layer filling. In order to solve the above problem, we can fill the empty pixel locations of all layer images using pixels in the first layer. Since the first layer has no empty pixels, we can use same pixels in the first layer to fill the other layers. This increases the prediction accuracy of H.264/AVC, therefore data size could be reduced further. We can eliminate the newly filled pixels in the decoding process because the information of NOL is sent to the decoder. It is an eight bit gray scale image that each pixel contains an unsigned integer number representing how many layers there are.

3.2 Coding of Number of Layers (NOL)

For color and depth components, we have applied two kinds of preprocessing algorithms. Still remaining important data to encode are the NOL and residual

data. Therefore, we describe a coding method of NOL and an algorithm to reduce the residual information in this section.

NOL could be considered as an image containing the number of layers at each pixel location. Figure 1 illustrates an example of the NOL image. Usually, the maximum number of layers is the same as the number of cameras used to capture the scene. If we use eight cameras to acquire eight-view video, then the maximum number of layers is eight. The minimum number of layers is one because there always exists more than one layer. In other words, there are no empty pixels in the first layer of LDI.

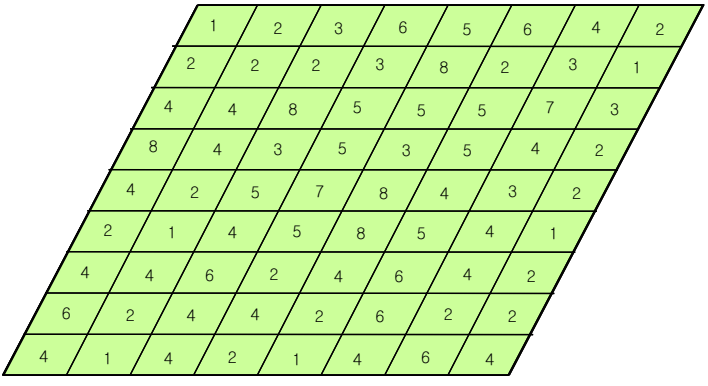

Fig. 1. An example of the NOL image

The physical meaning of the NOL is that it represents the hierarchical structure of the constructed LDI in the spatial domain. Assuming the NOL is known, we can efficiently use empty pixel locations to increase the coherency between pixels. We can freely change the pixel orders, add dummy pixels in the empty locations, and remove them after the decoding because we know where those pixels are.

Since the NOL information is very important to restore or reconstruct multi-view images from the decoded LDI, it is encoded by using the H.264/AVC intra mode. However, if we treat the NOL image as a direct input to the codec, we cannot assure the restoration accuracy. Since the dynamic range of the values of NOL is small, quantization noises can contaminate the reconstructed values. Consequently, it is difficult to restore the original NOL image.

In order to solve this problem, we change the dynamic range of the pixel values of the NOL image by considering both the encoding bits required for the changed dynamic range and the accuracy of restored NOL value.

$$\alpha \cdot nMinLayer \leq \alpha \cdot V_{NOL} \leq \alpha \cdot nMaxLayer, \alpha(\in N) \leq 255/V_{NOL} \quad (5)$$

where *nMinLayer* is the minimum number of layers, *nMaxLayer* is the maximum number of layers, and α is the scaling factor.

3.3 Reduction of Residual Information Using Pixel Interpolation

Theoretically, one way of reducing residual data is to reconstruct multi-views without using the information from the original images. It means that we should maximally exploit depth pixels (DPs) in back layers of LDI, neighboring pixels within a layer image, and spatial relationships between multiple images for the same scene.

In our reconstruction algorithms, there are three steps such as, inverse 3-D warping, reconstruction without residual information, and reconstruction with residual information [12]. In order to reduce residual information, we exploit the neighboring reconstructed images in the second reconstruction stage. We can get intermediate reconstruction results after applying inverse 3-D warping and depth ordering of the back layer pixels. As shown in Fig. 2, we can get intermediate reconstruction results after applying the inverse 3-D warping and depth ordering of the back layer pixels.

Fig. 2. Reconstruction using back layers: (a) view 0, (b) view 1, (c) view 4, (d) view 7

Our approach is to use the neighboring pixels and reconstructed images for interpolating empty pixels of the current reconstructed image. There are mainly two factors influencing the interpolation result: one is spatially located neighboring pixels within the current reconstructed image and the other is temporally located pixels in neighboring reconstructed images. We define the following equation to perform the pixel interpolation.

$$I_S(x,y) = \frac{1}{k} \cdot \sum_{i=0}^{W} \sum_{j=0}^{W} I(R_{(i,j)}) \tag{6}$$

$$I_V(x,y) = \sum_{n=0}^{N-1} a_n \cdot I(R_n), \sum_{n=0}^{N-1} a_n = 1 \tag{7}$$

$$I_E(x,y) = \alpha \cdot I_S(x,y) + (1-\alpha) \cdot I_V(x,y), 0 \leq \alpha \leq 1 \tag{8}$$

where $I_S(x,y)$, $I_V(x,y)$ is the intensity value of the interpolated pixel at the (x, y) position of the current image, respectively, $I_E(x,y)$ is the final interpolated pixel value, k is the valid number of pixels within a W x W window, a_n and α are the weighting factors, N is the number of cameras, and R means the reconstructed image. These equations are only applied to interpolate the empty

pixels of the current image. The weighting factors have been determined by experiments.

Figure 3 shows the reconstruction results after performing the interpolation using the above equations. We can observe that most holes except left-most and right-most sides are recovered with much less visual artifacts compared to the results in Fig. 2.

Fig. 3. Reconstruction results using the pixel interpolation: (a) view 0, (b) view 1, (c) view 4, (d) view 7

4 Experimental Results and Analysis

In our experiments, we have used the "Breakdancers" sequence provided by Microsoft Research. It includes a sequence of 100 images captured from eight cameras; the camera arrangement is 1-D arc with about 20cm horizontal spacing. Depth maps computed by stereo matching algorithms are provided for each camera together with the camera parameters: intrinsic parameters, barrel distortion, and rotation matrix. The exact depth range is also given [7][14].

4.1 Generation of LDIs from Natural Multi-view Video

The main part of generating LDI frames from the natural multi-view video is the incremental 3-D warping. Figure 4 shows the results of 3-D warping using the modified camera matrices. We can observe that actors are slightly rotating as the camera number changes. In order to identify the warping results clearly, we did not interpolate holes. White pixels in each image represent the holes, which are generated by the 3-D warping. Among eight cameras, the fifth camera is the reference LDI view and the warping has been performed from other camera locations to the reference LDI view. When the warping is carried out from the left cameras (view 0, 1, 2, and 3) to the reference camera (view 4), major holes are created along the right side of the actors. On the other hand, holes are mainly distributed in the left side of the actors as the warping is done from the right cameras (view 5, 6, and 7) to the LDI view.

The generated LDI has several layers and the maximum number of layer is the same as the camera number. For the test sequence used in our experiments, each LDI frame can therefore have eight layers in maximum. The layer images (color components) of the constructed LDI frame with depth threshold 3.0 is presented in Fig. 5.

Fig. 4. Results of the incremental 3-D warping: (a) view 0 to view 4, (b) view 1 to view 4, (c) view 7 to view 4

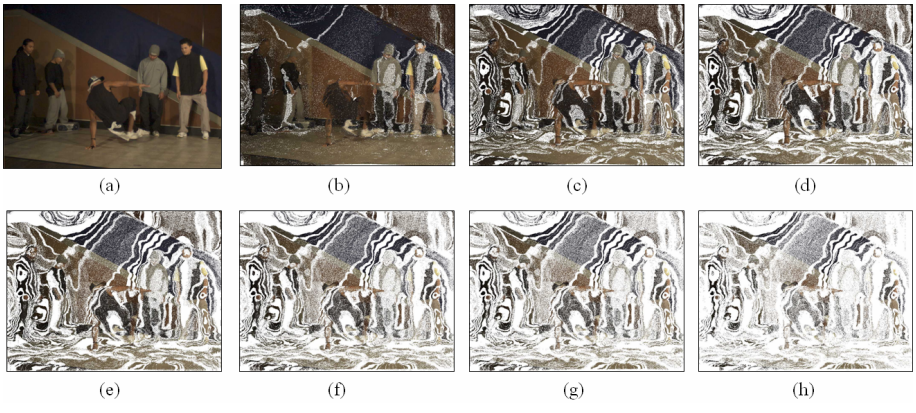

Fig. 5. Layer images of the first LDI frame: (a) 1st layer; (b) 2nd layer, (c) 3rd layer, (d) 4th layer, (e) 5th layer, (f) 6th layer, (g) 7th layer, (h) 8th layer

4.2 Inter-camera Coding of Multi-view Video Using LDI

In Table 1, we have compared the data size between the sum of frames of the test sequence and the generated LDI frame. In Table 1, the sum of frames means the summation of eight color and depth images of the test sequence without encoding. Simulcast using H.264/AVC (color + depth) means the summation of data size calculated by the independent coding of color and depth images.

Table 1 shows the data size by changing the depth threshold value from 0.0 to 5.0, but the data size has not been decreased much as the threshold value is over 3.0 from the experiments. The depth threshold means the difference among actual depth values. The given depth range was from 44.0 to 120.0. The size of NOL data is varying to the encoding condition, mainly by the dynamic range of NOL and quantization parameters. In addition, the depth threshold could affect to the size of them. In Table 1, the size of NOL is computed by using the fixed alpha value of one. From our experiments, about 60 to 70% of the total bitrates

Table 1. Data size for the "Breakdnacers" sequence [kbytes]

	1st 8 Frames	2nd 8 Frames
Sum of frames	25,166	25,166
Simulcast (color+depth)	137.7	132.5
LDI frame (threshold=0.0)	24,520	24,644
Encoded LDI (Layer filling)	71.4	72.9
Number of Layers (NOL)	6.3	6.4
LDI frame (threshold=3.0)	13,924	13,803
Encoded LDI (Layer filling)	48.4	48.2
Number of Layers (NOL)	5.1	5.0
LDI frame (threshold=5.0)	13,808	13,723
Encoded LDI (Layer filling)	46.3	47.0
Number of Layers (NOL)	4.2	4.4

are consumed to encode NOL data as near-lossless fashion and 10% are used for residual data coding.

Still remaining issues of the LDI-based approach are how to select the proper back layer pixels to fill out the current pixel location and how to dynamically adjust bitrates per each component, e.g., color, depth, NOL, and residual data.

5 Conclusions

In this paper, we have described a procedure to generate layered depth images (LDIs) from the natural multi-view video and encoding methods for the number of layers (NOL) and residual data. Incremental 3-D warping has been modified to consider intrinsic characteristics of multiple cameras. For the inter-camera coding of multi-view video, we have applied two kinds of preprocessing algorithms to encode color and depth components of the constructed LDI based on our framework. The number of layers (NOL) and residual data are coded by changing the dynamic range and exploiting pixel interpolation techniques. We have reduced a large amount of multi-view video data to a manageable size by combining the proposed encoding techniques and reconstructed the original multiple viewpoints successfully. Finally, we will investigate temporal prediction structures of the constructed LDI frames in the future.

Acknowledgements. This work was supported by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST).

References

1. ISO/IEC JTC1/SC29/WG11 N6720: Call for Evidence on Multi-view Video Coding. October (2004)
2. ISO/IEC JTC1/SC29/WG11 N6999: Report of the Subjective Quality Evaluation for Multi-view Coding CfE. January (2005)
3. ISO/IEC JTC1/SC29/WG11 N7327: Call for Proposals on Multi-view Video Coding. July (2005)
4. ISO/IEC JTC1/SC29/WG11 N7779: Subjective Test Results for the CfP on Multi-view Video Coding. January (2006)
5. Shade, J., Gotler, S., Szeliski, R.: Layered Depth Images. Proc. of ACM SIGGRAPH, July (1998) 291–298
6. Yoon, S.U., Kim, S.Y., Lee, E.K., and Ho, Y.S.: A Framework for Multi-view Video Coding using Layered Depth Images. Lecture Notes in Computer Science (LNCS), 3767 (2005) 431-442
7. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., and Szeliski, R.: High-quality Video View Interpolation using a Layered Representation. Proc. of ACM SIGGRAPH, August (2004) 600–608
8. Kanade, T., Rander, P.W., and Narayanan, P.J: Virtualized Reality: Constructing Virtual Worlds from Real Scenes. IEEE Multimedia Magazine, Vol. 1, No. 1, (1997) 34-47
9. Matusik, W., Buehler, C., McMillan, L., and Gortler, S.J.: Image-based Visual Hulls. Proc. of ACM SIGGRAPH, (2000) 369-374
10. Carranza, C., Theobalt, C., Magnor, M.A., and Seidel, H.-P.: Free-viewpoint Video of Human Actors. ACM Trans. on Graphics, Vol. 22, No. 3, (2003) 569-577
11. Yang, J.C., Everett, M., Buehler, C., and McMillan, L.: A Real-time Distributed Light Field Camera. Eurographics Workshop on Graphics, (2002) 77-85
12. Yoon, S.U., Lee, E.K., Kim, S.Y., Ho, Y.S., Yun, K., Cho, S., and Hur, N.: Coding of Layered Depth Images Representing Multiple Viewpoint Video. Proc. of Picture Coding Symposium (PCS) SS3-2, April (2006) 1–6
13. Duan, J. and Li, J.: Compression of the LDI. IEEE Trans. on Image Processing, Vol. 12, No. 3, (2003) 365-372
14. Interactive Visual Media Group at Microsoft Research, http://research.microsoft.com/vision/InteractiveVisualMediaGroup/3DVideoDownload/