Multiple Color and Depth Video Coding Using a Hierarchical Representation

Seung-Uk Yoon, *Student Member, IEEE*, and Yo-Sung Ho, *Senior Member, IEEE*

(Invited Paper)

***Abstract*—This paper presents coding schemes for multiple color and depth video using a hierarchical representation. We use the concept of layered depth image (LDI) to represent and process multiview video with depth. After converting those data to the proposed representation, we encode color, depth, and auxiliary data representing the hierarchical structure, respectively. Two kinds of preprocessing approaches are proposed for multiple color and depth components. In order to compress auxiliary data, we have employed a near lossless coding method. Finally, we have reconstructed the original viewpoints successfully from the decoded LDI frames. From our experiments, we realize that the proposed approach is useful for dealing with multiple color and depth data simultaneously.**

***Index Terms*—Layered depth image (LDI), multiple depth images, multiview video, scene representation.**

I. Introduction

THE MULTIVIEW video is a collection of multiple videos capturing a scene at different camera locations. If we acquire it from multiple cameras, we can generate video scenes from any viewpoints. In addition, depth information that can be extracted from multiple videos plays an important role to provide 3-D information of the scene. Therefore, multiple color and depth data can be used in a variety of applications including free viewpoint video (FVV), free viewpoint TV (FTV), 3DTV, surveillance, sports matches, games, and virtual reality (VR).

Although the multiview video with depth has much potential for various applications, one big obstacle is a huge amount of data. In principle, the amount of information in multiview video data increases linearly as the number of cameras. If each video is recorded at a higher resolution, the size of data becomes more tremendous. Because of these reasons, it has been perceived that multiview video coding (MVC) is a key technology to realize those applications.

Recently, MVC is getting a lot of attention; however, most MVC algorithms are based on the H.264/AVC video coding standard. Those MVC techniques are mainly extensions of predictive video coding schemes of a single video. They do not utilize rich 3-D information contained in multiview video.

On the other hand, there have been active researches on image-based rendering (IBR) using multiple images and video-based rendering (VBR) for multiple viewpoint video. Their main target is the seamless rendering of new viewpoints rather than reducing the data size of multiple images or videos. These diverse researches treat similar kinds of data, multiple images or multiview video; however, they usually address coding and rendering of those large amounts of data separately. Moreover, encoding methods of multiple depth video have rarely covered before even though there are some works related to single depth video compression. Therefore, there have been only a few researches trying to solve both problems together.

In this paper, we propose encoding schemes for multiple color and depth video using a hierarchical representation. Unlike most MVC and IBR/VBR techniques, we try to address problems of representation and encoding based on the unified data structure of multiple color and depth video. We use the concept of layered depth image (LDI) [1], one of the image-based rendering techniques, to represent and process multiview video with depth data simultaneously. We describe the overall framework and how to generate LDI frames from the natural multiview video with depth [2], [3]. For color and depth components, two preprocessing algorithms are proposed and examined [3]. For the auxiliary data, an image for the number of layers (NOL), representing the hierarchical structure, a near lossless coding method is applied. Finally, we reconstruct the original viewpoints through the inverse warping and hole-filling techniques using interpolated pixels and residual data.

The paper is organized as follows. In Section II, we review related works for image and video-based representations of multiview data and MVC activities in MPEG. We then describe the proposed approach to generate LDI frames from the natural multiview video in Section III. In Section IV, we explain the framework, encoding, and reconstruction methods for multiview color and depth video. After experimental results are presented in Section V, we draw a conclusion in Section VI.

II. Related Work

A. Image-Based Representations for Multiview Images

Since there have been researches on geometry-based rendering methods, lots of useful modeling and rendering

Manuscript received January 22, 2007; revised May 28, 2007. This work was supported in part by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center (RBRC), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project. This paper was presented in part at the Picture Coding Symposium, Beijing, China, April 2006. This paper was recommended by Guest Editor M. Tanimoto.

The authors are with the Department of Information and communications, Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, Korea (e-mail: suyoon@gist.ac.kr; hoyo@gist.ac.kr).

Digital Object Identifier 10.1109/TCSVT.2007.905363

techniques have developed. Geometry-based rendering, however, requires laborious modeling and long processing time to reconstruct and render complicated real world scenes. As an attractive alternative to avoid these problems, IBR techniques have received much attention. They use 2-D images as primitives to generate an arbitrary view of a 3-D scene. Especially, they do not depend on the complexity of 3-D objects in the scene, but rely on image resolutions [4].

On the other hand, one of the important aims of using the multiview video is to provide view-dependant scenes as users change their viewpoints. This goal is similar to the functionality of IBR: view generation using 2-D input images. We can classify various IBR techniques into three categories based on how much geometry information is used [5], [6]: rendering with no geometry, rendering with implicit geometry, and rendering with explicit geometry. Among a variety of methods, LDI [1] is one of the efficient rendering methods for 3-D scenes of complex geometries. LDI belongs to the category of rendering with explicit geometry. It represents the current scene using an array of pixels viewed from a reference camera location. In order to arrange pixels along the view axis, it uses the depth information for each pixel. This concept is useful to deal with splitted video simultaneously, especially in video surveillance, sports matches, etc.

There are several researches conducted on LDI: LDI tree [7] and adaptive sampling [8] for overcoming a sampling problem, tiling LDIs [9] for modeling and rendering of synthetic static terrains, LDI using pixel grouping [10] to enhance the quality of new viewpoints, and LDI compression [11] to reduce the size of LDI data for synthetic static scenes. Recently, LDI was used to generate soft shadows from opaque objects [12].

Most of them are dealing with a synthetic static scene and targeting for reconstruction and rendering of new viewpoints. Only one paper focuses on compression of LDI data for a static scene. However, we extend the concept of LDI to a natural multiview video for representing, processing, and coding.

B. Video-Based Representations for Multiview Video

IBR techniques have been mainly applied to static objects, architectures, and sceneries. However, there have been other approaches to extend them to dynamic scenes. Kanade *et al.* [13] generate a mesh-based surface representation at every time instant using 51 cameras (512×512) in a 3-D dorm. Their results show artifacts caused by illumination mismatches among multiple views and inconsistent motion when there exists severe changes of the viewing angle. Matusik *et al.* [14] use silhouette images from four cameras (256×256) to compute and shade visual hull, an approximate geometric representation. They can render 8000 pixels of the visual hull at about 8 frames per second (fps). Naemura *et al.* [15] use a view-dependant layer representation and synthesize scenes at 10 fps. They capture 16 videos of small objects and the horizontal resolution of each video is 180 pixels. Carranza *et al.* [16] use convergent seven cameras to capture 3-D human motion. Each video is recorded at a resolution of 320×240 pixels/frame at 15 fps. They use *a priori* human body model that is fit to a 3-D shape of the observed person. Their approach is not designed for a 3-D scene, but for a single 3-D human object. Zitnick *et al.* [17] propose an efficient view interpolation and rendering system using multiple videos (1024×768) captured by eight cameras. Their major target is to help video synthesis and editing in a cost effective way. More VBR references can be found in the literature [18].

However, these approaches are mainly focusing on capturing, modeling, and rendering of 3-D objects. Efficient representation for coding the large input data is less emphasized.

C. MPEG Multiview Video Coding Activity

ISO/IEC JTC1/SC29/WG11 Moving Picture Experts Group (MPEG) has recognized the importance of MVC technologies, and an ad hoc group (AHG) on 3-D audio and visual (3DAV) has established in December 2001 [19]. Four main exploration experiments (EE) on 3DAV were performed from 2002 to 2004: EE1 on omnidirectional video, EE2 on FTV, EE3 on coding of stereoscopic video using multiple auxiliary components (MAC) in MPEG-4, and EE4 on depth/disparity coding for 3DTV and intermediate view interpolation [20]. 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 [21], some evidences were recognized in January 2005 [22] and a Call for Proposals (CfP) on MVC has been issued in July 2005 [23]. Then, the responses to the Call have been evaluated in January 2006 [24]. From the evaluation results, the standardization of MVC is under way.

Most MVC algorithms in MPEG focus only on reduction of data size of multiple color videos. They are based on the H.264/AVC video coding standard, a predictive coding scheme for a single video. In fact, MPEG MVC is considered as a special case of the MPEG scalable video coding (SVC). From these reasons, they rarely use 3-D information contained in multiview video. In addition, they do not consider problems of efficient representation for multiview data and coding of multiple depth sequences. Unlike these approaches, in this paper, we try to represent and process huge amount of multiple color and depth data simultaneously.

III. Representation of Multiview Color and Depth Video Using a Hierarchical Representation

A. Concept of LDI

Among a variety of IBR techniques, LDI is one of the efficient rendering methods for 3-D objects with complex geometries. It represents the current scene using an array of pixels viewed from a camera position. However, each LDI pixel consists of color, depth between the camera and the pixel, and other data that support rendering of LDI. Three key characteristics of LDI are: 1) it contains multiple layers at each pixel location; 2) the distribution of pixels in the back layer is sparse; and 3) each pixel has multiple attribute values. Because of these special features, LDI enables us to render arbitrary views of the scene at new camera positions. Moreover, rendering can be performed quickly with the list-priority algorithm proposed by McMillan [25].

When the rays are emanating from a reference viewpoint (an LDI camera), it is possible to store intersecting points between rays and an object. Each intersecting point contains color and depth information. Fig. 1 represents the conceptual diagram of

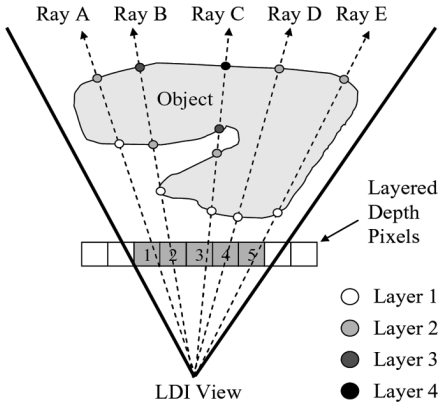


Fig. 1. Conceptual diagram of LDI.

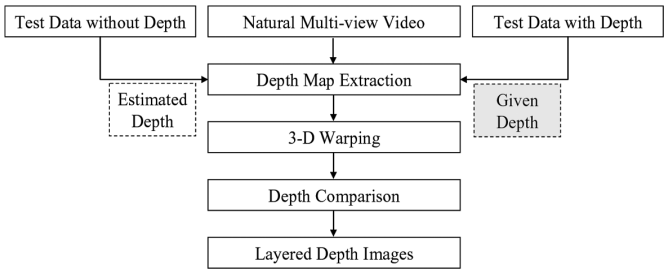


Fig. 2. Generation of LDIs from the natural multiview video.

LDI [1]. As shown in Fig. 1, the first intersecting points construct the first layer of LDI, the second ones build up the second layer, and so on. Consequently, each layered depth pixel (LDP) has different number of depth pixels (DPs), which contain color, depth, and auxiliary information for reconstruction of multiviews. For example, LDP 3 in Fig. 1 has four layers, which contain intersecting points between Ray C and the object.

B. Generation of LDI From Multiview Color and Depth Data

The method described in Fig. 1 is only applicable to 3-D computer graphics (CG) models since rays cannot go through a real object in the physical world. Therefore, we need another approach to generate LDI for real world objects [1].

Fig. 2 shows the overall procedure of the generation of LDIs for real world scenes from natural multiview color and depth video, not from 3-D synthetic models [2], [3].

After obtaining depth information from the multiview video, we perform 3-D warping to generate a single LDI using multiple color and depth images. We start from the following incremental 3-D warping equation [1] because the original McMillan's warping equation [28] is complex and has many parameters to be computed. If we use the incremental warping equation, we can reduce the computational complexity for 3-D warping. When we perform 3-D warping from one camera position i to the other location j, the equation is as follows:

$$C_i = V_i \cdot P_i \cdot A_i, \quad C_j = V_j \cdot P_j \cdot A_j, \quad T_{i,j} = C_j \cdot C_i^{-1} \quad (1)$$

where V is the viewport matrix, P is the projection matrix, and A is the affine matrix. C_i stands for camera matrix at the camera position i. We assume that the warping is performed from the location of camera i to that of camera j. T is the transformation matrix that moves pixels in the image captured from camera i to camera j. Pixels in the image plane of camera i are projected to the world space through the inverse matrix of C_i and those 3-D points are re-projected to the image plane of camera j using the camera matrix C_j. The incremental 3-D warping is then performed by (2)

$$T_{i,j} \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \begin{bmatrix} x_j \cdot w_j \\ y_j \cdot w_j \\ z_j \cdot w_j \\ w_j \end{bmatrix} = T_{i,j} \cdot \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix} + z_i \cdot T_{i,j} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (2)$$

where (x_i, y_i) is the pixel position in an image plane at camera i and z_i is the accompanying per pixel depth value. w is a scaling factor and the warped pixel location is finally obtained after dividing it by w. Usually, the camera matrix C in (1) can be easily calculated for 3-D CG objects. While the viewport matrix is computed from the image resolution, the projection matrix is determined in a graphics library through the orthogonal or perspective view. The affine matrix is calculated from the rotation and translation matrix. However, it is difficult to define those matrices in a natural video.

Because of these reasons, we use a different camera matrix calculated from the given camera parameters of the test sequence, instead of estimating each V, P, and A matrix in the natural video. The modified camera matrices and the 3-D warping equation are as follows [29]:

$$C_i' = A_i \cdot E_i, \quad C_j' = A_j \cdot E_j \quad (3)$$

where C_i' is the camera matrix of camera i, which is calculated from camera parameters, not from CG data. A is a matrix representing intrinsic camera parameters and E is the Euclidean transform expressing rotation and translation of a camera. Definition of A and E are as follows:

$$A = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$E = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{bmatrix} \quad (4)$$

where α_x, α_y are the focal length in pixel unit, s is the skew angle between two axes of the CCD cell, and (x_0, y_0) is the location of the principle point in pixel unit. $R_{3\times3}$ represents the rotational component and $T_{3\times1}$ is the translational factor. In general, $R_{3\times3}$ and $T_{3\times1}$ are given by extrinsic camera parameters that can be calculated from camera calibration techniques. The modified camera matrix $\dot{C}$ is formed by adding an additional row [0 0 0 1] to C'. It makes a homogeneous 4×4 camera matrix because $A \cdot E$ becomes a 3×4 matrix. Finally, the incremental warping in (2) is done using the $\dot{C}$ instead of C.

After performing the 3-D warping, depth comparison and thresholding is conducted for the moved pixels. Since the data size of generated LDI depends on this thresholding, the ex-threshold is chosen by considering the reconstruction quality.

Fig. 3 shows an example of the generation process of LDI for real world scenes using multiple color and depth images. The

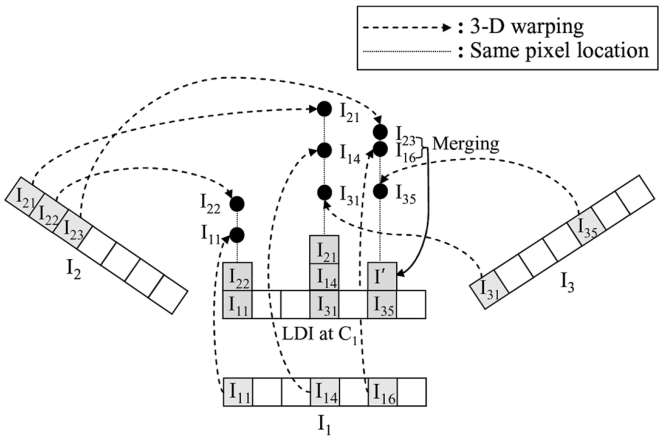


Fig. 3. Generation of LDI from multiple color and depth images.

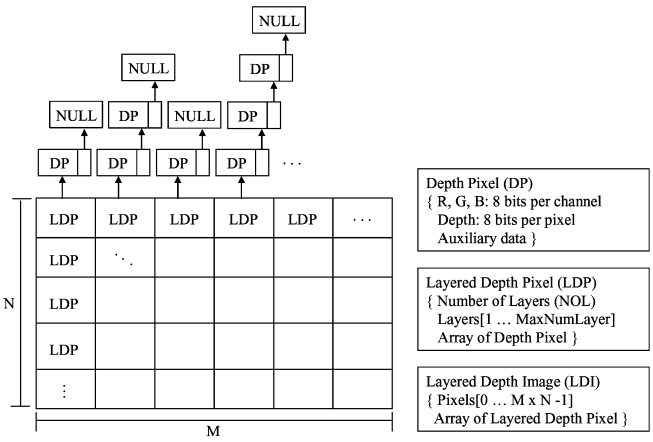


Fig. 4. Data structure of LDI.

LDI scene at camera 1 can be constructed by warping images from all camera locations.

First, we assume that there are three images, $I_i(i = 1, 2, 3)$ where i means the camera index, which are captured at different camera locations such as C_1, C_2, and C_3 in Fig. 3. In addition, each image has per pixel depth value as well as R, G, and B values, and camera parameters are available. I_{11} stands for the first pixel in I_1, I_{21} is that in I_2, and so on. Second, the reference view is selected by the user, e. g., C_1, and 3-D warping is performed to the reference location from all images. Actually, the warping from I_1 to I_1 is the same as I_1 itself. Third, the warped pixels, which still have color and depth values, are sorted according to depth at each pixel location. If the difference of depth values between warped pixels is smaller than the predefined threshold, they are merged into a single point, marked I' in Fig. 3, which has average color and depth values. Otherwise, a new layer is created as depicted in Fig. 3.

LDI contains several attribute data together with multiple layers at each pixel location. A single LDI pixel consists of color, depth, and auxiliary data that support reconstruction of multiview images from the decoded LDI. Therefore, color and depth information contained in multiview video can be properly stored in the data structure of LDI. The overall data structure of LDI is shown in Fig. 4. We use eight bits per color channel and per pixel depth value in DP. The NOL at each LDP location should be stored to recover the hierarchical structure of LDI. We define the maximum NOL is the same as the maximum number of cameras.

C. Analysis of the Multiview Video Test Data

There are various kinds of multiview video test sequences provided by MPEG AHG on MVC [23], [26]. Several proponents provide about 20 sequences with different properties. Recently, MPEG has been issued the CfP on MVC [23] and consequently selected eight sequences as the test sets for the CfP on MVC. They have been chosen by considering diverse features, such as the number of cameras (5, 8, and 100), camera arrangements (1-D parallel, 1-D parallel convergent, 1-D arc, 2-D cross, and 2-D array), frames per second (15, 25, and 30), image resolutions (VGA and XVGA), scene complexity, and camera motions. All the test sequences contain camera parameters for their own camera arrangements and the validation of those camera parameters has been done in MPEG AHG on MVC. The properties of the selected multiview video test sequences are summarized in Table I [3], [23]. Among them, Microsoft Research (MSR) provides the multiview video sequence, "Breakdancers" with camera parameters and depth information [17], [27].

TABLE I
PROPERTIES OF THE SELECTED TEST SEQUENCES FOR THE CFP ON MVC (A: AVAILABLE, N/A: NOT AVAILABLE)

Property	KDDI	MERL	FhG-HHI	Nagoya Univ.	MSR
Sequences	Flamenco2 Race1	Ballroom Exit	Uli	Rena Akko&Kayo	Breakdancers
Resolution	640 x 480	640 x 480	1024 x 768	640 x 480	1024 x 768
Frame Rate (fps)	30	25	25	30	15
Number of Cameras	5/8	8	8	100	8
Camera Arrangement	2-D cross 1-D parallel	1-D arc	1-D parallel convergent	1-D parallel 2-D array	1-D arc
Camera Parameters	A	A	A	A	A
Depth Information	N/A	N/A	N/A	N/A	A

In addition to this test data, we use one more test sequence in our experiments. Although this additional dataset is not the selected for MPEG MVC, it is useful to examine the efficiency of the proposed system for all available dataset at the moment. The sequence is "Ballet" provided by MSR, which also has per pixel depth information as well as camera parameters.

IV. FRAMEWORK FOR REPRESENTATION AND ENCODING OF THE MULTIVIEW COLOR AND DEPTH VIDEO USING LDI

A. System Overview

Fig. 5 shows the overall system diagram for representation and processing of multiview video using the concept of LDI [2], [3]. As shown in Fig. 5, the color and depth frames of the multiview video are gathered and warped to the selected LDI view. Consequently, eight color and eight depth images construct a frame of the LDI sequence. In this paper, the LDI sequence and LDI frames have the same meaning, the sequence of LDI. Once we obtain the LDI frames from the above procedure, we can reconstruct multiview images by applying the inverse warping, the pixel interpolation, and the residual compensation. Since information is lost because of depth thresholding and the limited viewing range caused by camera arrangements, a compensation module is required before the reconstruction of multiviews.

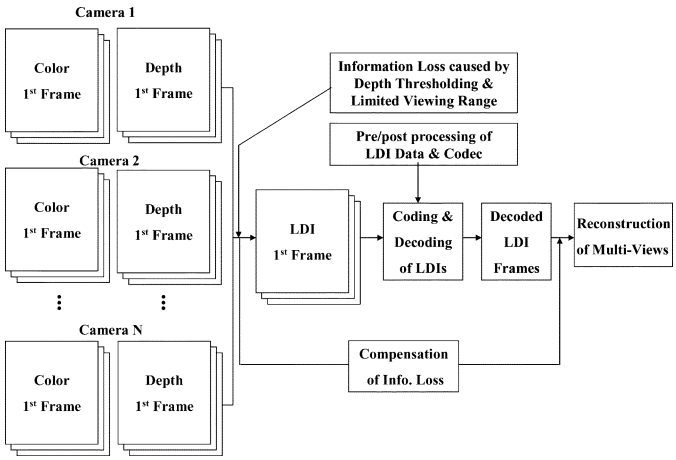


Fig. 5. Framework for representation, encoding, and reconstruction of multiview video using LDI.

TABLE II
PIXEL DISTRIBUTION OF THE FIRST LDI FRAME
(DEPTH THRESHOLD VALUE: 3.0)

Layer	Pixel Occupation [%]	Layer	Pixel Occupation [%]
1	100	5	46.8
2	90.4	6	34.7
3	69.8	7	29.0
4	47.0	8	19.1

In the encoding step, LDI data are preprocessed and H.264/AVC is applied to those processed data adaptively. One important thing to be considered in the encoding process is to employ the specific characteristics of LDI, which are described in the next subsection. Because each layer of LDI contains empty pixels, we should fill or remove those vacant holes before applying encoding techniques.

B. Characteristics of the Generated LDI Frames

Our proposed framework is based on the conversion between multiview color and depth video data and LDI frames. Before encoding the generated LDI frames, we first analyze the properties of the constructed LDI frames [3]. LDI pixel contains color, depth between the camera and the pixel, and auxiliary data that support reconstruction of multiview images from the decoded LDI. Three key characteristics of LDI have been already described in Section III. Because of those specific features, LDI enables us to render arbitrary viewpoints of the scene at new camera positions.

Table II lists the percentage of total number of pixels included in each layer of the generated LDI frame. The LDI frame in Table II is constructed from the first eight color and depth images of the "Breakdnacers" sequence. The pixel occupation is reduced as the layer number increases. It means that the data size of the original multiview video can be reduced by converting them into the data structure based on LDI.

C. Encoder Structure

Fig. 6 illustrates the encoder block diagram [30]. After generating LDI frames from the natural multiview video with depth, we separate each LDI frame into three components: color, depth, and the NOL. The color and depth component consists of layer

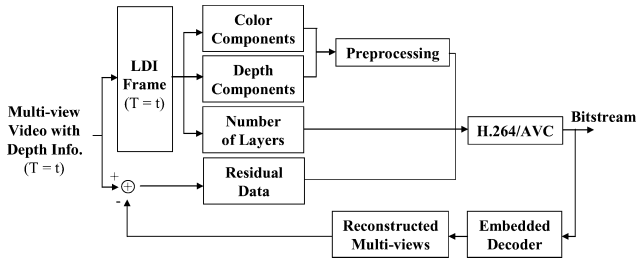


Fig. 6. Encoder block diagram.

images, respectively. The maximum number of layer images is the same as the total number of views. In addition, residual data are sent to the decoder in order to help the reconstruction of multiview images. Color and depth components are preprocessed by data aggregation/layer filling to apply H.264/AVC. NOL could be considered as an image containing different NOL at each pixel location. Since the NOL information is very important to restore or reconstruct multiview images from the decoded LDI, it is encoded by using the H.264/AVC intra-mode with a low quantization parameter from 10 to 15. Finally, the residual data, differences between the input multiview video and reconstructed ones, are encoded by H.264/AVC.

D. Preprocessing of the LDI Frames

We have tested with two kinds of preprocessing algorithms [3]. The first idea is to aggregate scattered pixels into the horizontal or vertical direction [11]. This approach can gather nonempty pixels into one side because each layer of the constructed LDI has different number of pixels. Although H.264/AVC is powerful to encode rectangular images, it does not support shape-adaptive encoding schemes. We, therefore, aggregate each layer image and then fill empty locations with the last pixel value of the aggregated image line by line. First, the scattered pixels in each layer are pushed to the horizontal direction. Second, the aggregated layer images are merged into a single huge image and empty pixels are padded. For example, if each layer image has 1024×768 pixels, the resolution of the huge aggregated image becomes 8192×768. Finally, the generated one is divided into small images with pre-defined resolutions, e.g., 1024×768, to employ H.264/AVC.

The second method is called as layer filling. We can fill the empty pixel locations of all layer images using pixels of the first layer. Since the first layer has no empty pixels, we can use same pixels of the first layer to fill other layers. This process 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 step because information of the NOL is sent to the decoder. It is a gray scale image that each pixel contains an unsigned integer representing how many layers are available.

E. Encoding of the Number of Layers (NOL)

NOL could be considered as an image containing the NOL at each pixel location. Fig. 7 shows an example of the NOL image. Usually, the maximum NOL 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 NOL is eight.

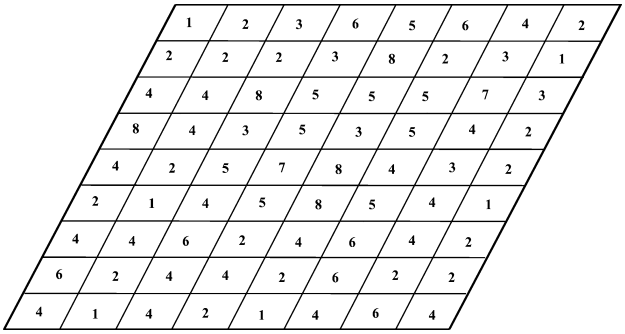


Fig. 7. Example of the NOL image.

The minimum NOL is one since there always exists more than one layer. Namely, there are no empty pixels in the first layer of LDI [29].

The physical meaning of the NOL image is that it represents the hierarchical structure of the constructed LDI in the spatial domain. Assuming NOL is known, we can efficiently use empty pixel locations to increase the correlation between pixels. We can change the pixel orders freely, allocate dummy pixel values in empty locations, and remove them after decoding because we know where those pixels are [29].

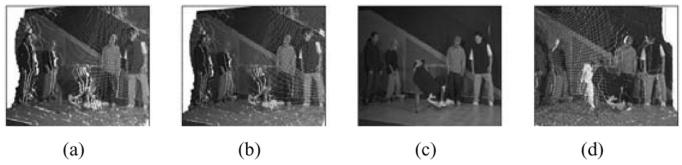
Since the NOL information is very important to restore or reconstruct multiview images from the decoded LDI, it is encoded by using the H.264/AVC intra-mode with a very low QP. From our experiment, QP values from 10 to 15 show similar results when QP 0 is used for NOL. Although the intra-mode is lossy, NOL can be reconstructed in a near lossless fashion using the very low QP. In addition, 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 the restored NOL value.

Finally, the inaccurately reconstructed values of the NOL image could cause different NOL per pixel. They consequently make artifacts in the final reconstructed results. However, the pixel blending at the reconstruction step could correct them within the limited tolerance and the added residual data fill the holes lastly.

F. Reduction of Residual Data Using Pixel Interpolation

Theoretically, we can reduce residual data if we can reconstruct multiviews without using the information of the original images. It means that we should use all available DPs in back layers of LDI, neighboring pixels within a layer image, and spatial relationships between multiple images for the same scene [29].

In our reconstruction algorithms, there are three steps: inverse warping, reconstruction without residual information, and reconstruction with residual information. In order to reduce the residual information, we exploit the neighboring reconstructed images in the second reconstruction step. We can get intermediate reconstruction results after applying inverse 3-D warping and depth ordering of the back layer pixels. As shown in Fig. 8, we can get intermediate reconstruction results after applying the inverse 3-D warping and depth ordering of the back layer pixels.



(a) (b) (c) (d)

Fig. 8. Reconstruction using back layers. (a) View 0. (b) View 1. (c) View 4. (d) View 7.

(a) (b) (c) (d)

Fig. 9. Reconstruction results using the pixel interpolation. (a) View 0. (b) View 1. (c) View 4. (d) View 7.

Our approach is to use neighboring pixels and reconstructed images for interpolating empty pixels of the current reconstructed image [29]. There are mainly two factors influencing the interpolation result: spatially located neighboring pixels within the current reconstructed image and temporally located pixels in neighboring reconstructed images.

The pixel interpolation is performed by

$$I_S(x,y) = \frac{1}{k} \cdot \sum_{i=0}^{W} \sum_{j=0}^{W} I\left(R_{(i,j)}\right) \tag{5}$$

$$I_V(x,y) = \sum_{n=0}^{N-1} a_n \cdot I(R_n), \quad \sum_{n=0}^{N-1} a_n = 1 \tag{6}$$

$$I_E(x,y) = \alpha \cdot I_S(x,y) + (1-\alpha) \cdot I_V(x,y), \quad 0 \le \alpha \le 1 \tag{7}$$

where $I_S(x,y)$ is the interpolated pixel value at the (x,y) position using neighboring pixels of the current image, $I_V(x,y)$ is that using pixels in other views. $I_E(x,y)$ is the final interpolated pixel value, k is the valid number of pixels within a $\mathrm{W} \times \mathrm{W}$ window, a_n and α are the weighting factors, N is the number of cameras, and R means the reconstructed image. We use these equations to interpolate the empty pixels of the current image. The weighting factors have been determined by experiments.

Fig. 9 shows the reconstruction results after we perform the pixel interpolation using the above equations. We can observe that most holes except for left-most and right-most sides are recovered with much less visual artifacts.

G. Reconstruction of Multiple Images From a LDI Frame

Fig. 10 shows the reconstruction procedure of multiview images from the decoded LDI data. We start from the decoded LDI because the reconstruction process can be considered as an inverse procedure of the LDI generation. After we move whole pixels from the decoded LDI to the world coordinate of the reference camera, those 3-D points are re-projected into each camera location. At this time, the target of the inverse 3-D warping is each camera location. Therefore, different numbers of pixels are located at each pixel location of each camera.

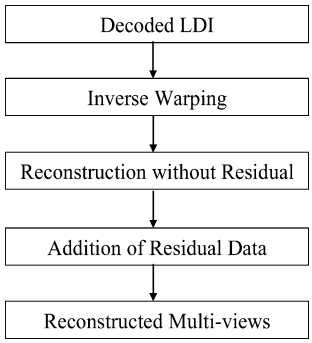


Fig. 10. Reconstruction procedure of multiview images.

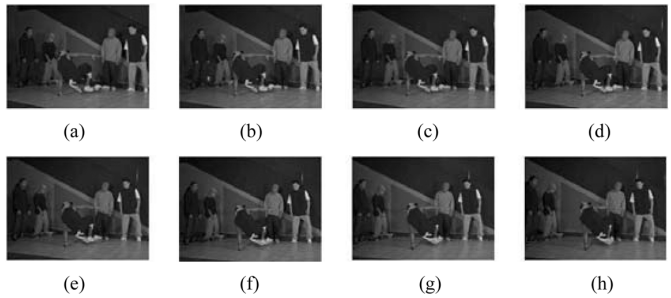


Fig. 11. First frame at each camera location. (a) Camera 0. (b) Camera 1. (c) Camera 2. (d) Camera 3. (e) Camera 4. (f) Camera 5. (g) Camera 6. (h) Camera 7.

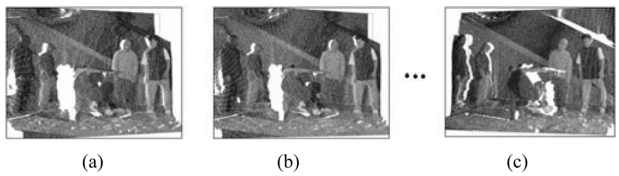


Fig. 12. Results of the incremental 3-D warping. (a) Camera 0 to camera 4. (b) Camera 1 to camera 4. (c) Camera 7 to camera 4.

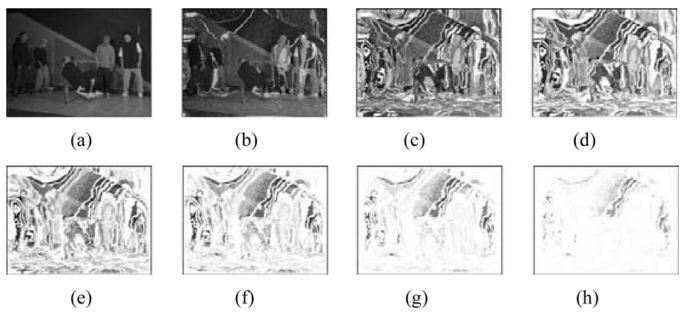


Fig. 13. Layer images of the first LDI frame. (a) First layer. (b) Second layer. (c) Third layer. (d) Fourth layer. (e) Fifth layer. (f) Sixth layer. (g) Seventh layer. (h) Eighth layer.

Now we can use the number of pixels hanging in the back layers to reconstruct the image at each camera location. Those pixels in back layers are called as DPs; each DP contains color, depth, and auxiliary information.

There are many holes in the reconstructed multiview images because information is lost when the pixels are moved to the reference camera location during the inverse 3-D warping. In order to fill the holes, we can use color and depth information contained in DPs of back layers. Each layered depth pixel (LDP) contains different number of DPs and each DP has color and depth information. By using those DPs moved from other camera locations, we can restore some empty pixels of the current viewpoint.

Even though we restore some empty pixels of each view, some empty regions still remain. The main reason is that there are some portions that other cameras cannot capture. For instance, the left-most side of camera 0 and the right-most side of camera 7 of the "Breakdancers" sequence. Actually, the camera arrangement of MSR data is 1-D arc of eight cameras with 20-cm spacing. Therefore, it is natural that we cannot find sufficient information to restore both left-most side and right-most side of the reconstructed images at certain camera locations. In this case, we need additional information to restore those regions. Although we can reduce the empty regions using interpolation methods, inaccurate pixels values can cause artifacts in the reconstructed images.

We, therefore, need to fill the holes using the compensation module, as described in our overall framework in Fig. 5. During the compensation process is performed, we use the residual data extracted from the original multiview images. We can calculate residual information by subtracting empty pixel locations from the original multiview images.

V. Experimental Results and Analysis

A. Generation of LDIs From Natural Multiview Video

The main part of generating LDI frames from the natural multiview video is the incremental 3-D warping. Fig. 11 depicts the first frame at each camera location when the reference camera is camera 4. The results of the incremental 3-D warping are shown in Fig. 12.

We can observe that actors are moving 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 generated by the 3-D warping. In Fig. 12, camera number 4 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 (camera 0, 1, 2, and 3) to the reference camera, 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 (camera 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 "Breakdancer," each LDI frame can, therefore, have eight layers in maximum. The layer images (color components) of the constructed LDI frame with depth threshold 5.0 is presented in Fig. 13. As for depth components, the resultant images are similar except they are represented as 8 bit gray scale images.

B. Preprocessing of the LDI Frames

As we described in Section IV, we have tested with two kinds of preprocessing methods [3], [30]. Fig. 14 represents the results of the data aggregation with the horizontal direction. The first row shows each layer image of the constructed LDI with the

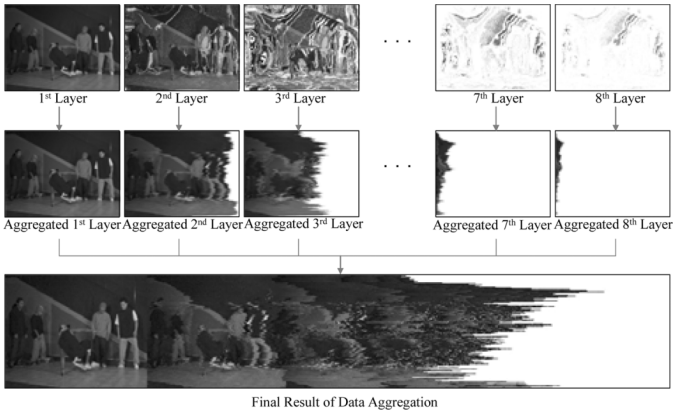


Fig. 14. Results of the data aggregation with the horizontal direction.

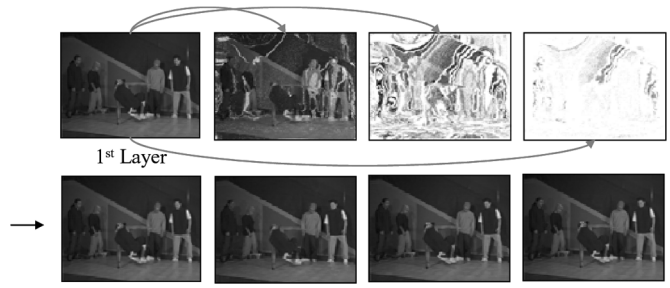


Fig. 15. Results of the layer filling.

depth threshold of 5.0. For each image, data aggregation is performed with the horizontal direction. Finally, each aggregated images are again aggregated into a single one. The third row represents the aggregated result.

The other method is filling empty locations using pixel values in the first layer, which is called as a layer filling technique. The result of layer filling is shown in Fig. 15.

C. Comparison of the Data Size and Analysis

In Tables III and IV, we have compared the data size between the sum of frames of the test sequence and the generated LDI frame [3], [30].

In Tables III and IV, 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.

The threshold value on how much we can allow for the difference among depth values has been selected based on the reconstruction results. Tables III and IV show the data size by changing the depth threshold value from 0.0 to 5.0, but the data size has not decreased drastically as the threshold value is over 3.0 in our experiments. The depth threshold means the difference among actual depth values. For "Breakdancers" and "Ballet" sequences, the given depth range is from 44.0 to 130.0 [27]. Tables III and IV do not list the data size of residual information. The residual information mainly depends on the distance among cameras and the actual viewing range.

In our experiments, we have changed the size of residual data based on the depth threshold value. For the residual data, after we calculate the difference between the original and reconstructed images, we encode the difference image. Moreover, the NOL data size should be added to calculate the final data size; the NOL image is coded by the H.264/AVC intra-mode with the fixed quantization parameter in a near-lossless manner.

TABLE III
DATA SIZE FOR THE "BREAKDANCERS" SEQUENCE

	1st 8 Frames	2nd 8 Frames
Sum of frames (color + depth) [kbytes]	25,166	25,166
Simulcast using H.264/AVC (color + depth) [kbytes]	137.7	132.5
Simulcast using H.264/AVC (color only) [kbytes]	97.4	96.3
LDI frame generated from 16 images [kbytes] (Threshold = 0.0, bit allocation for depth = 8 bits)	24,520	24,644
Encoded LDI frame using the data aggregation (with horizontal direction) [kbytes]	135.4	133.7
Encoded LDI frame using layer filling [kbytes]	71.4	72.9
LDI frame generated from 16 images [kbytes] (Threshold = 3.0, bit allocation for depth = 8 bits)	13,924	13,803
Encoded LDI frame using the data aggregation (with horizontal direction) [kbytes]	131.7	133.8
Encoded LDI frame using layer filling [kbytes]	48.4	48.2
LDI frame generated from 16 images [kbytes] (Threshold = 5.0, bit allocation for depth = 8 bits)	13,808	13,723
Encoded LDI frame using the data aggregation (with horizontal direction) [kbytes]	91.7	93.0
Encoded LDI frame using layer filling [kbytes]	46.3	47.0

TABLE IV
DATA SIZE FOR THE "BALLET" SEQUENCE

	1st 8 Frames	2nd 8 Frames
Sum of frames (color + depth) [kbytes]	25,166	25,166
Simulcast using H.264/AVC (color + depth) [kbytes]	134.4	139.5
Simulcast using H.264/AVC (color only) [kbytes]	92.7	98.2
LDI frame generated from 16 images [kbytes] (Threshold = 0.0, bit allocation for depth = 8 bits)	24,544	24,727
Encoded LDI frame using the data aggregation (with horizontal direction) [kbytes]	133.1	135.6
Encoded LDI frame using layer filling [kbytes]	72.3	76.8
LDI frame generated from 16 images [kbytes] (Threshold = 3.0, bit allocation for depth = 8 bits)	14,153	14,328
Encoded LDI frame using the data aggregation (with horizontal direction) [kbytes]	130.7	135.2
Encoded LDI frame using layer filling [kbytes]	46.7	51.2
LDI frame generated from 16 images [kbytes] (Threshold = 5.0, bit allocation for depth = 8 bits)	13,958	14,022
Encoded LDI frame using the data aggregation (with horizontal direction) [kbytes]	90.4	96.2
Encoded LDI frame using layer filling [kbytes]	46.1	49.7

D. Comparison of PSNR and Analysis

In order to show that the proposed scheme has benefits in terms of coding, we have shown comparison results in terms of peak sinal-to-noise ratio (PSNR) Y versus bit rate for the "Breakdancers" sequence in Fig. 16. The proposed method is compared with others, which have been evaluated as the responses to the CfP of MVC [24].

The important fact in Fig. 16 is that all other methods encode color components only, but the proposed one compresses color, depth, NOL, and residual data. In other words, the PSNR curve in Fig. 16 for the proposed method is the result of the total bit rates for depth, NOL, and residual as well as color data. However, the other curves present the PSNR Y versus bit rate for the color component only. The coding results are displayed for three rate points: low, middle, and high bit rates.

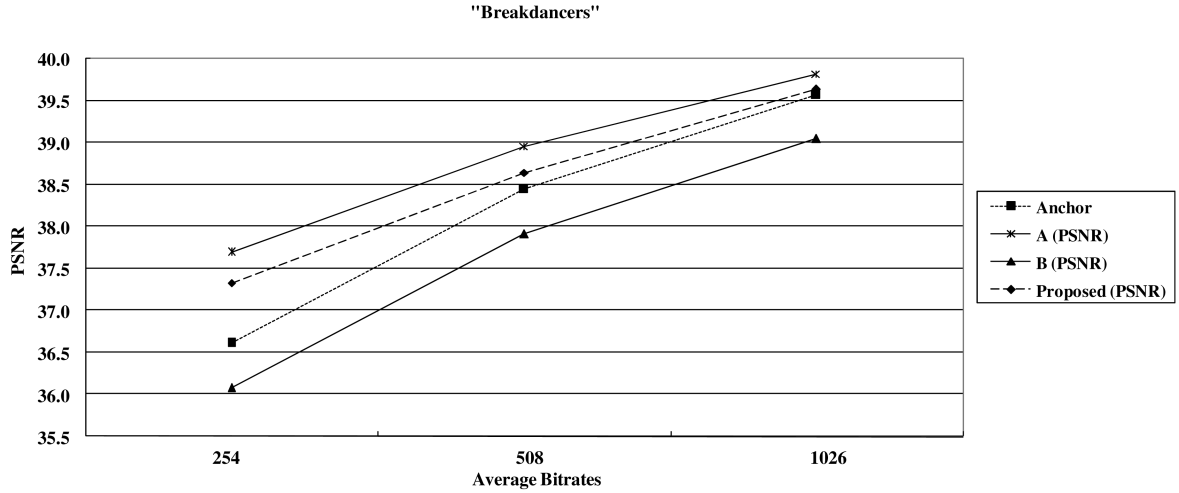


Fig. 16. PSNR Y versus bit rate curve.

In Fig. 16, the anchor means that all views are encoded separately using H.264/AVC with specified parameters [23]. All PSNR curves represent the average values over all views. Except for the anchor coding result, it is not possible to compute the exact bit rate for each view because all data are merged into one bit stream. We have calculated the total bit rate per LDI frame and divided it by the number of views because each LDI frame contains data for all viewpoints hierarchically. Since the rate control mechanism is not implemented for the LDI frames, we have manually allocated the total bit rate to each component of the LDI frame. Its four components have been encoded by the proposed methods. We have assigned approximately 70% of the total bit rate to NOL, 20% to the color and depth components, and 10% to the residual data. From our experimental results, we observe that the proposed method has benefits in terms of coding efficiency because it shows a better PSNR curve than the anchor and a few others dealing with the color component only, even though it contains all the encoded bits for depth, NOL, and residual as well as color data.

There are several problems to be considered in future experiments. First, the relationship between the NOL and the quality of reconstructed views should be analyzed carefully. Second, shape adaptive transforms, such as a shape-adaptive discrete cosine/wavelet transform could be used to encode LDI data because H.264/AVC supports only the 4×4 integer transform. Finally, temporal prediction schemes between constructed LDI frames could be investigated using more test sequences with depth information. Remaining issues of the LDI-based approach are how to select the proper back layer pixels to fill out the current pixel location, how to dynamically allocate total bits to each component, e.g., color, depth, NOL, and residual, and how to compare the performance of depth coding. Since the LDI frame contains the depth information and the PSNR value may not be the best measure for evaluating the depth coding performance, we need to develop proper comparison metrics considering view generation results using the depth information.

E. Reconstruction Results of Multiple Viewpoints

The result of the inverse 3-D warping is depicted in Fig. 17. As we can observe in Fig. 17, there are many holes in the resultant images since information has been lost due to the 3-D warping and long camera distance.

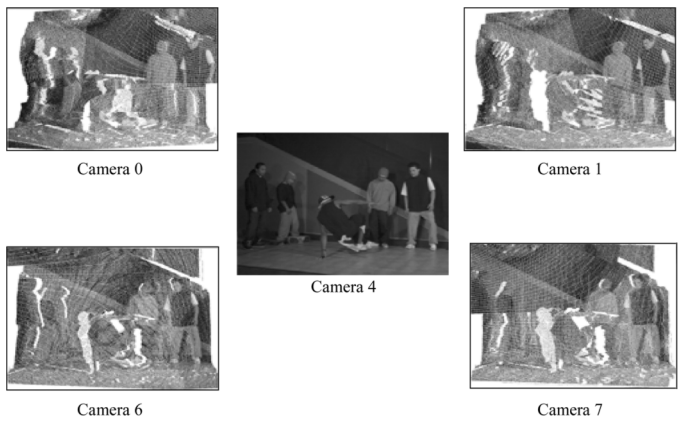


Fig. 17. Results of inverse 3-D warping.

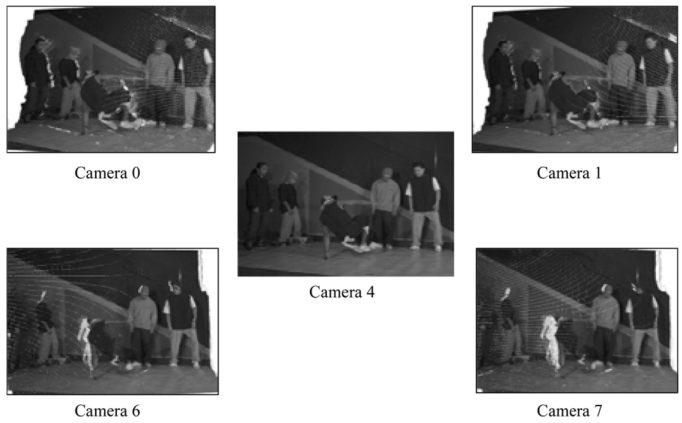


Fig. 18. Reconstruction results without residual information.

In order to fill the holes, we have used color and depth information contained in DPs of back layers. By using those DPs from other camera locations, we can restore some empty pixels. The reconstruction result using the additional DPs hanging in the back layer of LDI is depicted in Fig. 18.

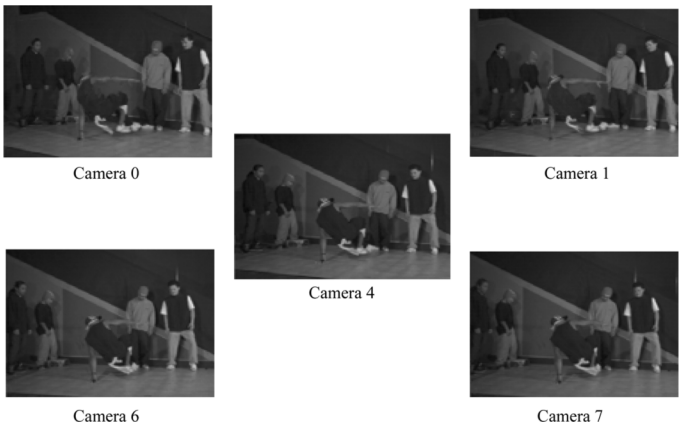


Fig. 19. Recosntruction results after applying residual information.

In Fig. 18, we did not use residual data from the original images, thus the left and right-most side of the reconstructed images have large holes. As we have already mentioned in Section V, the major reason of these results is the long distance between cameras. If we move pixels from camera 0 to camera 4, the distance between these two cameras become about 80 cm. Therefore, it is reasonable that there exist certain regions that several cameras cannot acquire sufficient information for them.

In order to fill remaining regions, we have calculated residual information by subtracting empty pixel locations from the original multiview images. Fig. 19 represents the final reconstruction results using the residual information. Although there are some artifacts in the final reconstruction results, it could be improved by exploiting more accurate method of selecting proper DPs to fill the holes.

VI. Conclusion

In this paper, we have described a hierarchical representation of multiple color and depth video, preprocessing methods, encoding schemes of each component, and the reconstruction procedure for multiview video data with depth information. Experimental results show that the size of the original multiple color and depth video data, which are several tens of megabytes, are reduced to about a few hundred kilobytes. For example, the size of the test multiview video data with 100 frames is about 2.5 Gbytes, but it has been reduced more than a thousand times. In addition, even though the proposed representation contains color, depth, the NOL, and residual data, its PNSR performance is comparable with others dealing with color information only. The reconstruction results present that the original viewpoints are properly restored through the inverse warping, the pixel interpolation, and the compensation with residual data. The proposed approach can treat multiple color and depth videos simultaneously using a hierarchical representation and reduce the overall data to a manageable size. We, therefore, believe that our approach could be useful to represent, process, and encode multiview color and depth video data effectively.

References

[1] J. Shade, S. J. Gortler, L. W. He, and R. Szeliski, "Layered depth images," in *Proc. SIGGRAPH'98*, Jul. 1998, pp. 231–242.

[2] S. U. Yoon, S. Y. Kim, E. K. Lee, and Y. S. Ho, "A framework for multiview video coding using layered depth images," *Lecture Notes Comput. Sci. (LNCS)*, vol. 3767, pp. 431–442, Nov. 2005.

[3] S. U. Yoon, S. Y. Kim, E. K. Lee, and Y. S. Ho, "A framework for representation and processing of multiview video using the concept of layered depth image," *J. VLSI Signal Process. Syst.*, vol. 46, no. 2–3, pp. 432–441, Mar. 2007.

[4] S. U. Yoon, S. Y. Kim, and Y. S. Ho, "Preprocessing of depth and color information for layered depth image coding," *Lecture Notes Comput. Sci. (LNCS)*, vol. 3333, pp. 622–699, Nov. 2004.

[5] H. Y. Shum and S. B. Kang, "A review of image-based rendering techniques," *Proc. IEEE/SPIE Visual Commun. Image Process.*, pp. 2–13, Jun. 2000.

[6] H. Y. Shum, S. B. Kang, and S. C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.

[7] C. F. Chang, G. Bishop, and A. Lastra, "LDI tree: A hierarchical representation for image-based rendering," in *Proc. SIGGRAPH'99*, Aug. 1999, pp. 291–298.

[8] R. Namboori, H. C. Teh, and Z. Huang, "An adaptive sampling method for layered depth image," in *Proc. Comput. Graph. Int.*, Jun. 2004, pp. 206–213.

[9] J. Shade, M. F. Cohen, and D. P. Mitchell, "Tiling layered depth images," Univ. Washington, Seattle, Tech. Rep. #02-12-07, 2000.

[10] H. Kim, S. Kim, B. Koo, and B. Choi, "Layered depth image using pixel grouping," in *Proc. Vis. Syst. Multimedia*, Oct. 2001, pp. 121–127.

[11] J. Duan and J. Li, "Compression of the LDI," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 365–372, Mar. 2003.

[12] Y. H. Im, C. Y. Han, and L. S. Kim, "A method to generate soft shadows using layered depth image and warping," *IEEE Trans. Visual. Comput. Graph.*, vol. 11, no. 3, pp. 265–272, May/Jun. 2005.

[13] T. Kanade, P. W. Rander, and P. J. Narayanan, "Virtualized reality: Constructing virtual worlds from real scenes," *IEEE Multimedia Mag.*, vol. 1, no. 1, pp. 34–47, Jan.–Mar. 1997.

[14] W. Matusik, C. Buehler, L. McMillan, and S. J. Gortler, "Image-based visual hulls," in *Proc. SIGGRAPH'00*, Jul. 2000, pp. 369–374.

[15] T. Naemura, J. Tago, and H. Harashima, "Real-time video-based modeling and rendering of 3-D scenes," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 66–73, Mar./Apr. 2002.

[16] C. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel, "Free-viewpoint video of human actors," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 569–577, Jul. 2003.

[17] 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, Aug. 2004.

[18] Video-Based Rendering. [Online]. Available: http://www.video-based-rendering.org/

[19] *List of Ad hoc Groups Established at the 58th Meeting in Pattaya*, ISO/IEC JTC1/SC29/WG11 N371, 2001.

[20] A. Smolic and D. McCutchen, "3DAV exploration of video-based rendering technology in MPEG," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 348–356, Mar. 2004.

[21] *Call for Evidence on Multi-view Video Coding*, ISO/IEC JTC1/SC29/WG11 N6720, 2004.

[22] *Report of the Subjective Quality Evaluation for Multi-view Coding CfE*, ISO/IEC JTC1/SC29/WG11 N6999, 2005.

[23] *Call for Proposals on Multi-view Video Coding*, ISO/IEC JTC1/SC29/WG11 N7327, 2005.

[24] *Subjective Test Results for the CfP on Multi-view Video Coding*, ISO/IEC JTC1/SC29/WG11 N7779, 2006.

[25] L. McMillan, "A list-priority rendering algorithm for redisplaying projected surfaces," Univ. North Carolina, Charlotte, UNC Tech. Rep. TR95-005, 1995.

[26] *Updated Call for Proposals on Multiview Video Coding*, ISO/IEC JTC1/SC29/WG11 N7567, 2005.

[27] *Interactive visual media group at Microsoft Research* [Online]. Available: http://research.microsoft.com/vision/InteractiveVisualMediaGroup/3DVideoDownload/

[28] L. McMillan, "An image-based approach to three-dimensional computer graphics," Ph.D. dissertation, Dept. Comp. Sci., Univ. North Carolina, Chapel Hill, 1997.

[29] S. U. Yoon, E. K. Lee, S. Y. Kim, Y. S. Ho, K. Yun, S. Cho, and N. Hur, "Inter-camera coding of multiview video using layered depth image representation," *Lecture Notes Comput. Sci. (LNCS)*, vol. 4261, pp. 432–441, 2006.

[30] 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," in *Proc. Picture Coding Symp.*, Apr. 2006, vol. SS3-2, pp. 1–6.

Seung-Uk Yoon (S'06) received the B.S. degree in electronic engineering from Sogang University, Seoul, Korea, in 2000, and the M.S. degree in information and communications engineering from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 2002, where he is currently working toward the Ph.D. degree in the Information and Communications Department.

His research interests include representation and coding of multiview color and depth data, layered depth image compression, image-based rendering, immersive media processing, and 3-D scene representation.

Yo-Sung Ho (SM'06) received the B.S. and M.S. degrees in electronic engineering from Seoul National University, Seoul, Korea, in 1981 and 1983, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 1990.

He joined Electronics and Telecommunications Research Institute (ETRI), Daejon, Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, NY, where he was involved in development of the Advanced Digital High-Definition Television (AD-HDTV) system. In 1993, he rejoined the technical staff of ETRI and was involved in development of the Korean DBS digital television and high-definition television systems. Since 1995, he has been with Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, where he is currently Professor of Information and Communications Department. His research interests include digital image and video coding, image analysis and image restoration, advanced video coding techniques, digital video and audio broadcasting, three-dimensional video processing, and content-based signal representation and processing.