Hierarchical Decomposition of Depth Map Sequences for Representation of Three-Dimensional Dynamic Scenes

Sung-Yeol KIM¹, Student Member and Yo-Sung HO†(a), Member

SUMMARY In this paper, we propose a new scheme to represent three-dimensional (3-D) dynamic scenes using a hierarchical decomposition of depth maps. In the hierarchical decomposition, we split a depth map into four types of images: regular mesh, boundary, feature point and number-of-layer (NOL) images. A regular mesh image is obtained by down-sampling a depth map. A boundary image is generated by gathering pixels of the depth map on the region of edges. For generating feature point images, we select pixels of the depth map on the region of no edges according to their influence on the shape of a 3-D surface, and convert the selected pixels into images. A NOL image includes structural information to manage the other three images. In order to render a frame of 3-D dynamic scenes, we first generate an initial surface utilizing the information of regular mesh, boundary and NOL images. Then, we enhance the initial surface by adding the depth information of feature point images. With the proposed scheme, we can represent consecutive 3-D scenes successfully within the framework of a multi-layer structure. Furthermore, we can compress the data of 3-D dynamic scenes represented by a mesh structure by a 2-D video coder.

key words: hierarchical decomposition of depth maps, 3-D dynamic scenes, depth image-based representation, 3-D TV

1. Introduction

As high-speed networks are widely used and technologies for three-dimensional (3-D) display systems have been improved rapidly, various multimedia applications using 3-D audio-visual data have appeared, such as 3-D TV and 3-D games. Especially, 3-D TV [1] is considered a main theme for a future broadcasting system supporting high-quality audio-visual services and user-friendly interactions, such as multi-view image generations [2].

In order to realize 3-D TV, it is essential to develop an efficient scheme to represent 3-D dynamic scenes. Recently, a depth image-based representation (DIBR) [3] has been issued as a core technology for 3-D TV. Since DIBR can render complicated 3-D scenes that are difficult to model geometrically, it is considered a suitable technology to represent 3-D dynamic scenes. Color images and synchronized depth maps are used in DIBR.

For acquiring depth maps, stereo matching algorithms with binocular images have been intensively studied for several decades [4]–[6]. In recent, multiple viewpoint images can be used to obtain more accurate depth maps [7]. Furthermore, as sensor technologies for obtaining depth information are being developed, we can also capture more accurate per-pixel depth data from real scenes directly using a depth camera system [8].

In general, depth maps are provided as gray-level images and include depth values of 256 levels. Intensities in depth maps indicate the depth information at each pixel position. Hence, we can regard the position and intensity of each pixel in a depth map as geometry information in the 3-D space. Moreover, the corresponding color image can be regarded as a texture image that covers a geometric surface made by a depth map.

There are two approaches to represent 3-D dynamic scenes with DIBR techniques: mesh-based representation [9], [10] and image-based reconstruction [11]. In a mesh-based representation, we extract feature points from a depth map, and then generate consecutive 3-D surfaces by applying a mesh triangulation technique. The main advantage of the mesh-based representation is a high rendering speed, because we employ a graphic accelerator to render 3-D scenes. However, we sometimes suffer from dealing with its data due to their irregularities. In other words, positions of feature points are not constant frame by frame. Hence, it is hard for us to apply signal processing techniques into the data directly.

On the other hand, an image-based reconstruction uses all pixels in a depth map and generates 3-D scenes utilizing a 3-D warping technique. Naturally, we can easily apply general signal processing techniques to its data. However, it is difficult to render consecutive 3-D surfaces with the image-based reconstruction in real time, because we need time to carry out some complicated algorithms and to apply reasonable hole-filling techniques. Especially, if we do not know the accurate information about camera parameters, it is hard to generate 3-D views successfully using a 3-D warping technique.

In this paper, we focus on the generation of 3-D dynamic scenes using a mesh-based representation in DIBR. We propose a new scheme to represent 3-D dynamic scenes, called as a hierarchical decomposition of depth maps. Although our approach is a kind of a mesh-based representation, it can maintain regularities like an image-based reconstruction. In addition, the proposed scheme can render consecutive 3-D scenes progressively within the framework of a multi-layer representation.

The remainder of this paper is organized as follows. In Sect. 2, we introduce the hierarchical decomposition of depth maps. Then, Sect. 3 explains how to render 3-D
dynamic scenes with the proposed scheme, and Sect. 4 presents a multi-layer representation of 3-D dynamic scenes. After providing experimental results in Sect. 5, we conclude in Sect. 6.

2. Hierarchical Decomposition of Depth Maps

In the hierarchical decomposition of depth maps, we split depth maps into four types of images: regular mesh, boundary, feature point and number-of-layer (NOL) images. These four types of images are used to represent 3-D dynamic scenes instead of depth maps. First, we can define depth maps \( \psi_n \) by Eq. (1).

\[
\psi_n(i, j) = D_n(i, j) \quad 0 \leq i < W, 0 \leq j < H, 0 \leq n < N
\]  

Here, the term of \( i \) and \( j \) indicates the horizontal and vertical position in a depth map, respectively, and \( D(i, j) \) means the intensity at \((i, j)\). The term of \( n \) is the frame number of a depth map, and \( N \) is the total number of depth maps. \( W \) and \( H \) are the horizontal and vertical resolution of depth maps, respectively.

2.1 Regular Mesh Images

In a 3-D mesh model, we define a valence of a vertex as the total number of its neighbor vertices or faces. When valences of all vertices in a 3-D mesh model are equal, we call the model as a regular mesh. We apply the concept to obtain regular mesh image from depth maps. A regular mesh image is obtained by down-sampling a depth map. First of all, we define a grid cell as the unit of the hierarchical decomposition. When the size of a grid cell is \( p \times q \), we down-sampling a depth map with the horizontal sampling rate \( p \) and the vertical sampling rate \( q \). We can obtain a regular mesh image \( B_n \) by following Eq. (2).

\[
B_n(i, j) = \psi_n(x, y) \begin{cases} 
 x = i \cdot (p - 1) \\
 y = j \cdot (q - 1)
\end{cases} 
0 \leq x < W, 0 \leq y < H, 0 \leq n < N
\]  

Here, the term of \( i \) and \( j \) indicates the horizontal and vertical position in a regular mesh image, respectively.

Figure 1 explains how to generate a regular mesh image from a depth map. First, we select 4 depth pixels at the corner of a grid cell, and then the selected pixels are gathered into an image. Naturally, when the resolution of a depth map is \( W \times H \) and the size of a grid cell is \( p \times q \), the resolution of the regular mesh image is \((W/p + 1) \times (H/q + 1)\).

We define the size of a grid cell as \( 2^n \times 2^n \) resolution, such as \( 16 \times 16, 8 \times 8 \), or \( 16 \times 8 \). Once we choose the size of a grid cell, we should maintain it for each depth map during the hierarchical decomposition. When we set the size of grid cell, we should be careful to select the size of a grid cell, because it is inversely proportional to the amount of distortion of generated 3-D scenes. We commonly use the size of a grid cell as \( 16 \times 16 \) or \( 8 \times 8 \).

2.2 Boundary Images

A boundary image is used to refine a 3-D surface generated by a regular mesh image for the region of edges. Since most of the serious distortions are mainly occurred in their areas, we should deal with the region of edges carefully. The boundary image is employed to handle the region of edges in a depth map independently. In order to make the boundary image, we first extract regions of edges by applying a Sobel filter into a depth map vertically and horizontally. The reason we utilize depth maps instead of color images is that they are related to an object-based representation. In other words, depth maps will not be disturbed by lights or surroundings.

After extracting the region of edges, we employ a quad-tree structure to represent 3-D surfaces of grid cells on the region of edges. In a quad-tree structure, we determine whether a grid cell includes the region of edges or not. If the grid cell is on the region of edges, we partition the grid cell into four sub-grid cells. If the sub-grid cell still includes the region of edges, we repartition the sub-grid cell into four sub-grid cells again.

In addition, we employ a full modeling technique to represent 3-D surfaces for a grid cell. When the region of edges occupies more than a half of the grid cell, we represent the grid cell using a full modeling mode. Hence, there are five patterns for 3-D surfaces generated by boundary images, as shown in Fig. 2. The type 1, type 2, type 3, and type 4 are for quad-tree structure modes and the type 5 is for a full modeling mode.

For generating a boundary image, we gather the depth information of quad-tree structures and a full modeling into an image. Figure 3 explains how to obtain a boundary image from a depth map. In a quad-tree structure mode, we need 10 depth pixels to describe a grid cell. The depth pixels are assigned into a boundary image along a raster scanning.
order. On the other hand, we need 21 depth pixels to describe a grid cell in a full modeling mode. We also assign 21 depth pixels into the boundary image as a quad-tree structure mode.

Here, we note that we can regenerate quad-tree structures and a full modeling only if we know the depth pixels and the type information. In order words, the vertical and horizontal positions for depth pixels can be predicted by the size of a grid cell and the resolution of depth maps, and then we can generate the two modes with the type information and a boundary image.

We set the buffer size of a boundary image as the same as one of a regular mesh image to maintain resolution consistence. When the buffer is overflowed by the depth information of the region of edges, we allocate another buffer to generate the next boundary image. The empty space in the buffer is assigned a default value, 255. Therefore, all boundary images have the same resolution with regular mesh images.

2.3 Feature Point Images

After obtaining regular mesh and boundary images, we generate feature point images from depth maps. Feature points mean depth pixels in a grid cell affecting the shape of 3-D surfaces. In this paper, we extract feature points on the region of no edges. As we deal with the region of edges by boundary images in Sect. 2.2, we handle the region of no edges with feature point images. Feature point images are used to enhance the visual quality of the region of no edges in 3-D dynamic scenes.

2.4 NOL Images

In order to generate a feature point image from a depth map, we first select a depth pixel in a grid cell that has a great influence on the shape of a 3-D surface. Then, we select the next one according the influence. The 1st feature points in each grid cell are gathered into an image to generate the 1st feature point image, and the 2nd feature points are also gathered into another image to generate the 2nd feature point image. Naturally, the resolution of feature point images is equal to the one of regular mesh images. Figure 4 shows the generation of feature point images.

When we extract feature points, we employ a maximum distance algorithm [12]. Figure 5 shows the flow chart to extract feature points constantly. Basically, dominant stereo algorithms or depth acquisition techniques using a depth camera utilized smoothness and consistency constraints to obtain geometrically continuous depth information on the region of no edges [7],[8]. Therefore, the algorithm was operated with geometrically continuous depth data on the region of no edges. In general, two or three feature points are enough to represent a grid cell on the region of no edges.
dynamic scenes. A NOL image is composed of structural data, which are the number of feature points in a grid cell and the type information of quad-tree structures or a full modeling mode of the corresponding boundary image. In other words, we can recognize the elements of a 3-D surface, such as the number of feature points and the region of edges from a NOL image. Naturally, there is a NOL image for a depth map. Figure 6 explains what a NOL image plays a role in.

As we mentioned in Sect. 2.2, there are five types in a boundary image. Each type indicates a quad-tree structure mode or a full modeling mode. If the number of feature points in a grid cell is \( n \), the NOL for the types will be from \( n + 1 \) to \( n + 5 \). The NOL information from \( n + 1 \) to \( n + 5 \) has a one-to-one correspondence with the type information from type 1 to type 5. If the NOL in a grid cell is \( n + 1 \), the grid cell is represented by four pixels in a regular mesh image and 10 pixels for the type 1 of a quad-tree structure in a boundary image. In case NOL is \( n + 5 \), a grid cell is represented by four depth pixels in a regular mesh image and 21 pixels for a full modeling mode, type 5. In addition, when NOL in a grid cell is \( n \), the grid cell is represented by four pixels in a regular mesh and \( n \) feature points of each feature point image. Naturally, when NOL in a grid cell is zero, the grid cell is represented by only four pixels in a regular mesh image.

3. Rendering of 3-D Dynamic Scenes

In order to render 3-D dynamic scenes, we define their initial surfaces with regular mesh, boundary, and NOL images. For obtaining the initial surfaces, we make regular mesh surfaces with regular mesh images. Figure 7 (a) shows the rendering result of the wire frame mode for a 3-D surface with a regular mesh image, and Fig. 7 (b) shows the rendering result of the 3-D surface. As shown in Fig. 7 (a), all valances of the 3-D scene are six, i.e. a regular mesh. However, it is not enough for the initial surface of a 3-D scene because there are serious distortions in the region of edges as shown in Fig. 7 (b).

Therefore, we add the boundary surface generated by boundary images into the regular mesh surface to obtain an initial surface for a 3-D scene. Figure 8 show the surface by a boundary image and the result of the initial surface for a 3-D scene. As shown in Fig. 8 (c), the 3-D surface on the region of edges is enhanced comparing with regular mesh surfaces in Fig. 7 (b).

After constructing initial 3-D surfaces with regular mesh, boundary, and NOL images, we will meet serious distortions, holes, close to the region of edges owing to the difference of depth values between in regular mesh images and in quad-tree structures and a full modeling mode. For preventing holes, exceptional processing is required to fill out them. When depth values on boundaries in a grid cell

![Fig. 7](image_url)

(a) Boundary surface  (b) Initial surface  (c) Edge region enhancement of the 3-D surface

Fig. 8 Representation of initial surfaces for 3-D scenes.

![Fig. 9](image_url)

(a) Holes  (b) After filling holes

Fig. 9 Filling holes of boundary images.

![Fig. 10](image_url)

(a) Wire frame for final surface  (b) Result after texture mapping  (c) No edge region enhancement of 3-D surface

Fig. 10 Representation of a 3-D scene with four types of images.
are different with depth values in regular mesh images, we carry out a mesh triangulation to fill the regions. With such an exceptional process, we prevent serious distortions. Figure 9 shows the result of filling holes.

After generating initial surfaces, we improve visual qualities of the 3-D initial surfaces by adding the depth values in feature point images. With depth values in regular mesh and feature point images, we generate more complex surfaces in the region of no edges with Delaunay triangulation as a unit of a grid cell. Figure 10 shows the final rendering result of a 3-D scene. As shown in Fig. 10(c), the 3-D surface for the body of a man is enhanced comparing with the surface in Fig. 8(c).

4. Multi-Layer Representation

In order to serve 3-D dynamic scenes to the satisfaction of consumers, we also need to develop the functionality to support quality scalability [13], [14]. In other words, we should be able to control the amount of required data for 3-D dynamic scenes within the framework of multi-layer representation [15] according to target applications and network conditions. For example, we should provide consumers with minimum data of 3-D contents within reliable visual qualities in mobile applications.

Progressive meshes [16] give us an efficient solution about how to represent hierarchical layers for 3-D mesh models. However, we need more efforts to control the mechanism for consecutive 3-D mesh scenes. In addition, we sometimes suffer from dealing with their data due to irregularities when we try to apply signal processing techniques into consecutive 3-D surfaces.

Basically, multi-layer representation of 3-D dynamic scenes is divided into two layers: a base layer and enhancement layers. A base layer means the minimum required data to generate 3-D dynamic scenes and enhancement layers are additional data to improve the visual quality of the 3-D surfaces generated by the base layer. In the hierarchical decomposition of depth maps, we regard initial surfaces generated by regular mesh, boundary, and NOL images as a base layer, and the depth information in feature point image as enhancement layers. Figure 11 explains the multi-layer representation of 3-D dynamic scenes using the hierarchical decomposition.

5. Experimental Results

We have tested the performance of our proposed scheme with two test sequences, as shown in Fig. 12. Test data were Home-shopping and Breakdance sequences. Home-shopping sequences have 100 frames with 720 × 480 resolutions, while Breakdance sequences have 100 frames with 1024 × 768 resolutions. Depth maps for Home-shopping are captured by a depth camera, ZCam™ [17], [18]. On the other hand, depth maps for Breakdance sequences are obtained by a stereo matching algorithm [7], [19]. The experiment was implemented on an Intel based PC (Dual 3.2 GHz Pentium 4 Xeon, 2 GB DDRRAM) under Microsoft Windows XP.

5.1 Hierarchical Decomposition of Depth Maps

Figure 13 shows the four types of images extracted from depth maps of the test sequences. In this experiment, we set the size of grid cell as 16 × 16. The resolutions of the images for the Home-shopping and Breakdance sequences are 46 × 31 and 65 × 49, respectively. The first line and the second line image sequences indicate the series of regular mesh and boundary images, respectively, from the 1st frame to the 5th frame. The third line is for the series of NOL images, and the fourth and fifth ones are for the series of the first and the second feature point images.

As shown in Fig. 13, we could generate four types of images regularly. The main problem of mesh-based representation in DIBR was the irregularities, which caused us to handle its data difficult. With the hierarchical decomposition of depth maps, we could find a solution to give regularity to the data of the mesh-based representation, and handle the data easier than previous approaches, as well as maintaining high rendering speed.
5.2 Generation of Multi-Layer Representation

We generated a base layer and two enhancement layers for the test sequences. Figure 14 shows the result of multi-layer representation of the 60th frame of Home-shopping sequences and the 70th frame of Breakdance sequences. Figure 14(a) shows the result of a 3-D scene represented by only base layers, and Fig. 14(b) shows the result by base layer and two enhancement layers.

As shown in Fig. 14, we could generate a multi-layer structure by the hierarchical decomposition of depth maps. As a result, we could control the number of enhancement layers in proportion to visual qualities requested by target applications. In the band-limited applications, such as a mobile 3-D TV and Internet broadcasting, we could serve 3-D dynamic scenes within reliable visual qualities. Especially, since we selected a base layer as the combination of regular mesh, boundary, and NOL images, we could define the minimum required data for 3-D dynamic scenes structurally, while previous approaches had problems to choose them.

As shown in Fig. 15, when we compared estimated depth maps from the 3-D surfaces generated by a base layer with original depth maps, we could note that the proposed base layer represented an initial surface of 3-D dynamic scenes successfully with maintaining reliable visual qualities. Figure 16 shows the result of 3-D dynamic scenes applying a texture mapping algorithm.

In addition, we compared our proposed scheme with 3-D full modeling. In general, the 3-D full modeling generates a 3-D scene with all depth pixels in a depth map. In this experiment, we used 25 pixels for each grid cell like type 5 in a boundary image. As shown in Table 1, we could increase rendering speed for 3-D dynamic scenes more than about 20 times comparing with the 3-D full modeling, because our scheme rendered the region of edges and no edges of a 3-D scene adaptively.
In the 3-D full modeling, since there were tremendous amount of triangles to render as shown in Table 1, it needed a more time to generate 3-D surfaces than the proposed scheme. However, the advantage of the 3-D full modeling is much simpler in the aspect of the design of rendering device that our scheme. Especially, when we added feature point images into initial surfaces generated by a base layer, we needed a mesh triangulation technique, such as Delaunay triangulation.

### 5.3 Compression of Depth Maps

We compressed the data of 3-D dynamic scenes for the test sequences using a H.264/AVC video codec. We could use a conventional 2-D video coder directly to code 3-D data unlike previous approaches, because we converted irregular 3-D data into regular 2-D images. We compared the coding results of our approaches with the coding results of original depth maps. In a H.264/AVC coder, we set quantization parameter as 30. Table 2 shows the coding result of original depth maps. In Table 3, ‘Base’ means the summation of coding bits of regular mesh, boundary, and NOL images, and ‘EL1’ and ‘EL2’ mean the coding results after adding feature point image and the second feature point image, respectively.

When we used ‘EL2’, we reduced coding bits of depth maps about 5%–28% in comparison with one of original depth maps without serious visual quality degradation. However, we should code NOL images with a lossless manner, because NOL managed other layer images. If some losses happen during transmission, there will be serious distortion in a 3-D scene. On the other hand, the coding of original depth maps will be more robust for transmission errors. As a result, we need to develop a robust coding scheme for these errors.

### 6. Conclusion

In this paper, we proposed a new scheme to represent 3-D dynamic scene using a hierarchical decomposition of depth maps. With the proposed scheme, we could represent and render consecutive 3-D scenes successfully while supporting the framework of a multi-layer representation, and we also reduced the amount of bits needed to code depth maps. In addition, we increased rendering speed for 3-D dynamic scenes more than about 30 times in comparison with a 3-D full modeling. Finally, we could control visual qualities of 3-D dynamic scenes according to target applications. We expect the proposed scheme to be used to generate 3-D contents for next-generation broadcasting or various 3-D applications.

### Acknowledgments

This work was supported in part by the Institute for Information Technology Advancement (IITA), in part by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

### References


**Sung-Yeol Kim** received his B.S. degree in Information and Telecommunication engineering from Kangwon National University, Korea, in 2001 and M.S. degree in Information and Communication Engineering at the Gwangju Institute of Science and Technology (GIST), Korea, in 2003. He is currently working towards his Ph.D. degree in the Information and Communications Department at GIST, Korea. His research interests include digital signal processing, video coding, 3-D mesh representation, 3-D mesh compression, 3-D television, and realistic broadcasting.

**Yo-Sung Ho** received both B.S. and M.S. degrees in electronic engineering from Seoul National University, Korea, in 1981 and 1983, respectively, and Ph.D. degree in Electrical and Computer Engineering from the University of California, Santa Barbara, in 1990. He joined the Electronics and Telecommunications Research Institute (ETRI), Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, New York, 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 Korea direct broadcast satellite (DBS) digital television and high-definition television systems. Since 1995, he has been with the Gwangju Institute of Science and Technology (GIST), where he is currently a professor in the Information and Communications Department. His research interests include digital image and video coding, image analysis and image restoration, advanced coding techniques, digital video and audio broadcasting, 3-D television, and realistic broadcasting.