MULTIPLE DESCRIPTION CODING OF PLANE-BASED 3-D SURFACES

Sung-Bum Park†, Chang-Su Kim‡, and Sang-Uk Lee§

†Samsung Electronics Co., Ltd., Korea
‡School of Electrical Engineering, Korea University, Seoul, Korea
§School of Electrical Engineering and INMC, Seoul National University, Seoul, Korea
E-mail: sbpark@ieee.org, cskim@ieee.org, sanguk@ipl.snu.ac.kr

ABSTRACT

We present a multiple description coding (MDC) scheme for 3-D plane-based surfaces. First, planes are split into two disjoint subsets, called descriptions, each of which provides an equal contribution in 3-D surface reconstruction. To optimize the quality of the decoded surface, planes in each description are adaptively compressed according to the channel error condition. Then, the two compressed bitstreams are transmitted over distinct channels to the decoder. At the decoder, if both channels are available, the two bitstreams are decoded and merged together to reconstruct a high quality surface. If only one channel is available, we employ a hole filling method to fill visual holes and reconstruct a smooth 3-D surface. Therefore, the proposed algorithm provides an acceptable 3-D surface, even when one channel is totally lost.

***Index Terms*—** Error resilient transmission, multiple description coding, plane-based 3-D surface, graph coloring, and hole filling.

1. INTRODUCTION

Much research effort has been made for robust transmission of 3-D data, since 3-D compressed bitstreams are very sensitive to channel errors. In [1], Yan *et al.* proposed a 3-D data partitioning scheme to alleviate the error propagation problem in 3-D bitstreams. In [2], an unequal error protection (UEP) method was proposed to improve the error robustness of progressively compressed 3-D bitstreams. Recently, Park *et al*. proposed an error concealment scheme for 3-D data, which can recover the visual quality of corrupted mesh surfaces [3]. In severe error conditions, however, these conventional techniques fail to provide an acceptable quality reconstruction of 3-D surfaces and yield visually annoying artifacts.

Multiple description coding (MDC) has been applied to 2-D video data to protect the quality of reconstructed videos [4, 5]. An MDC encoder generates two or more bitstreams, called descriptions, of the same importance. These descriptions are transmitted over distinct channels to the decoder. The decoder is designed to reconstruct the signal from one or both descriptions according to channel conditions. Therefore, an acceptable quality reconstruction can be achieved, even though one channel is totally lost. Similarly, if 3-D data can be split into two subsets of the identical importance, MDC is a promising framework for robust transmission of 3-D data over noisy channels.

‡The work of C.-S. Kim was supported by the MIC, Korea, under the ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0017).

Different from 2-D image and video, 3-D data are often expressed by geometry primitives, which have irregular structures, such as polygonal meshes [6] and point clouds [7]. Due to the irregularity of geometry primitives, it is challenging to extract two descriptions of the identical importance from the conventional 3-D representation. Note that the MDC was applied to the mesh data compression [8, 9]. To facilitate the processing of 3-D geometry, Park *et al*. recently introduced a plane-based representation of 3-D points [10]. They divided the whole 3-D volume into regular grid cubes, and approximated point samples within each cube with a plane patch [10]. The regular cube structure makes it easy to split 3-D geometry into two subsets.

In this paper, we propose a robust transmission system for 3-D point data based on the MDC framework. First, we approximate input point samples by multiscale plane patches [10]. Then, the plane patches are split into two descriptions using a graph coloring scheme [11]. In order to minimize overall reconstruction errors, planes in each description are encoded adaptively according to the channel conditions. Then, the compressed bitstreams are transmitted over distinct channels to the decoder. At the decoder side, the two bitstreams are independently decoded and merged together to form a 3-D surface. If both channels are available, the decoder reconstructs a high quality surface. On the other hand, if only one of the channels is available, the surface is recovered by a novel hole filling algorithm. Therefore, a lower but acceptable quality surface is obtained, even when one channel is totally lost.

This paper is organized as follows. The proposed MDC encoding and decoding schemes are described in Section 2 and Section 3, respectively. Then, Section 4 evaluates the error robustness of the proposed algorithm. Finally, Section 5 concludes this paper.

2. MDC OF PLANE-BASED 3-D SURFACES

The block diagram of the proposed MDC system is shown in Fig. 1. To extract geometry primitives with a regular structure, an input point data is expressed by the plane-based 3-D representation [10]. The resulting plane set is then split into two disjoint subsets of the equal importance. Finally, the two descriptions are compressed to minimize the expected distortion and transmitted over distinct channels to the decoder.

2.1. Coloring of Plane-Based 3-D Surfaces

To generate two descriptions from a plane-based 3-D surface, we split the original plane set D_0 into two disjoint sets $D_{1,0}$ and $D_{1,1}$. Since the plane patch set D_0 is represented over the regular cubes, the simplest way to obtain $D_{1,0}$ and $D_{1,1}$ is to divide the regular

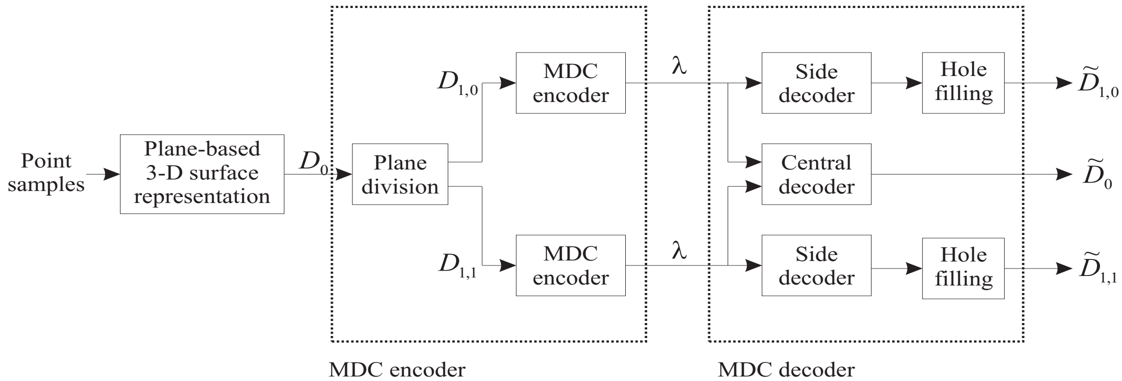


Fig. 1. The block diagram of the proposed MDC algorithm. λ indicates the channel error probability.

cubes using the quincunx subsampling [4]. However, the quincunx sampling may yield an unequal sampling of 3-D surfaces, since it only splits the whole volume equally without the consideration of surface characteristics.

In graph theory [11], a graph coloring method extracts disjoint sets of nodes from a given graph G. When the plane set D_0 is associated with the graph G_{D_0}, the division of D_0 can be seen as a graph coloring problem of G_{D_0} with chromatic number $\chi(G_{D_0}) = 2$, *i.e.*, two disjoint sets. Note that a graph is two-colorable, if it is divided into two disjoint sets and each element in one set is connected to the elements in the other set only. This is desirable since the proposed algorithm attempts to reconstruct the loss of one set using the information in the other set. However, a graph is two-colorable if and only if it is bipartite.

A typical 3-D mesh G_{D_0} is not bipartite and some adjacent nodes should be assigned the same color. These nodes can be seen as a penalty or a cost in the graph coloring problem [12]. In this work, we attempt to minimize the overall cost due to these miscolored nodes.

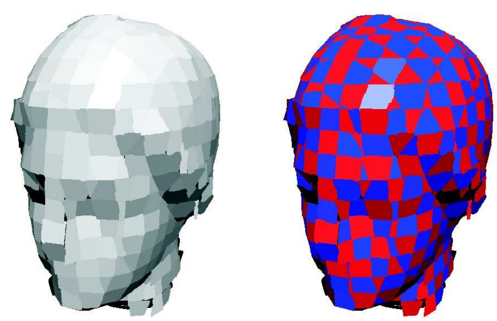
Given the plane set D_0 with the graph G_{D_0}, we get the red plane set $\mathcal{P}_R$ for $D_{1,0}$ and the blue plane set $\mathcal{P}_B$ for $D_{1,1}$ to minimize the coloring cost function, which is defined as

$$(\mathcal{P}_R, \mathcal{P}_B) = \arg \min_{\substack{P_i \in \mathcal{P}_R, \\ P_j \in \mathcal{P}_B}} \left[\sum_{P_i \in \mathcal{P}_R} C_R(P_i) + \sum_{P_j \in \mathcal{P}_B} C_B(P_j) \right], \quad (1)$$

where $C_R(P_i)$ is the number of red planes adjacent to a red plane $P_i \in \mathcal{P}_R$, and $C_B(P_j)$ is the number of blue planes adjacent to a blue plane P_j, respectively. To locally minimize the cost function in (1), we propose the following iterative algorithm. It is noted that a similar algorithm was introduced for the generalized graph coloring problem with $\chi(G) \geq 3$ in [12].

1. Choose initial sets $\mathcal{P}_R$ and $\mathcal{P}_B$ with the equal number of planes.
2. Find the red plane $P_i \in \mathcal{P}_R$ with the maximum $\Delta C_{RB}(P_i)$.
3. Find the blue plane $P_j \in \mathcal{P}_B$ with the maximum $\Delta C_{BR}(P_j)$.
4. If $\Delta C_{RB}(P_i) + \Delta C_{BR}(P_j) \leq 0$, stop. Otherwise, color P_i as blue and P_j as red, and go to step 2.

$\Delta C_{RB}(r_i)$ is the change in the cost function, when $P_i \in \mathcal{P}_R$ turns blue. Similarly, $\Delta C_{BR}(P_j)$ is the differential cost for the color flipping of P_j. Note that the overall cost in (1) decreases maximally in every iteration, though the iterative method cannot guarantee the coloring result that has the globally minimum cost.



(a) (b)

Fig. 2. The division of the 3-D 'Venus' model: (a) the plane-based representation and (b) its division into blue and red subsets.

Fig. 2 shows the coloring of the 3-D 'Venus' plane model. Each node corresponds to a cube region, and the 6-connectivity is assumed between cubes. We observe that red planes and blue planes are equally distributed on the 'Venus' surface and adjacent planes tend to have different colors. In other words, the proposed algorithm splits the 3-D surface into two disjoint sets efficiently.

2.2. Encoding of Plane Parameters

In the MDC decoder, there are three possible surface reconstructions, $\tilde{D}_0$, $\tilde{D}_{1,0}$, and $\tilde{D}_{1,1}$, according to the channel condition. We assume that two channels fail independently with the same probability λ. Then, the expected distortion of the reconstructed 3-D surface is given by

$$(1-\lambda)^2 d(D_0, \tilde{D}_0) + \lambda(1-\lambda) d(D_0, \tilde{D}_{1,0}) + \lambda(1-\lambda) d(D_0, \tilde{D}_{1,1}) + \lambda^2 \phi, \quad (2)$$

where $d(\cdot)$ is the distortion metric between 3-D models, $d(D_0, \tilde{D}_0)$ is the central distortion, $d(D_0, \tilde{D}_{1,0})$ and $d(D_0, \tilde{D}_{1,1})$ are the side distortions, and ϕ is the distortion when no information is received. Our objective is to minimize the expected distortion according to the given channel error rate λ.

To minimize the expected distortion, we encode the parameters of the planes in $\mathcal{P}_{\mathcal{R}} \cup \mathcal{P}_{\mathcal{B}}$. $\tilde{D}_{1,0}$ depends on $\mathcal{P}_{\mathcal{R}}$ and $\tilde{D}_{1,1}$ depends on $\mathcal{P}_{\mathcal{B}}$, whereas $\tilde{D}_0$ is affected by both $\mathcal{P}_{\mathcal{R}}$ and $\mathcal{P}_{\mathcal{B}}$. Therefore, the minimization of the cost function in (2) yields the plane set $\mathcal{P} = \mathcal{P}_{\mathcal{R}} \cup \mathcal{P}_{\mathcal{B}}$, given by

$$\mathcal{P} = \arg\min_{\mathcal{P}_{\mathcal{R}},\mathcal{P}_{\mathcal{B}}} [(1-\lambda)d(D_0, \tilde{D}_0(\mathcal{P}_{\mathcal{R}}, \mathcal{P}_{\mathcal{B}})) + \lambda d(D_0, \tilde{D}_{1,0}(\mathcal{P}_{\mathcal{R}})) + \lambda d(D_0, \tilde{D}_{1,1}(\mathcal{P}_{\mathcal{B}}))]. \quad (3)$$

Note that, when only one description is available, we incorporate the hole filling method to interpolate the missing planes of the other description, as shown in Fig. 1. During the hole filing, the plane in a missing cube is affected by the other planes in the neighboring cubes. Due to this cross dependency among the planes, it is computationally very complex to find the globally optimal solution of (3).

As an alternative, we propose an iterative plane modification algorithm to obtain a locally optimal solution of (3). The proposed algorithm refines each plane P_t to P_t^*, while keeping the plane parameters of the other planes fixed. Let us assume that P_t belongs to $\mathcal{P}_{\mathcal{R}}$ and its neighbor planes in $\mathcal{P}_{\mathcal{B}}$ are P_i, $i = 0, 1, 2, 3$, without the loss of generality. Then, the refinement equation for P_t, which reduces the cost in (3), is given by

$$P_t^* = \arg\min[(1-\lambda)d_p(P_t, P_t^*) + \lambda \sum_{i=0}^{3} d_p(P_i, \tilde{P}_i(P_t^*)) + \lambda d_p(P_t, \tilde{P}_t(P_0, P_1, P_2, P_3))], \quad (4)$$

where $d_p(\cdot)$ denotes the distortion between planes, and $\tilde{P}_i(\cdot)$ is an interpolated plane in the hole filling process. Since the last term in (4) does not depend upon P_t^*, (4) can be reduced to

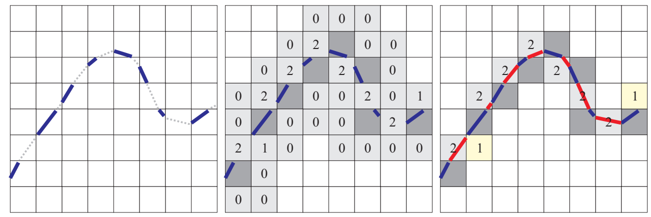
$$P_t^* = \arg\min\left[(1-\lambda)d_p(P_t, P_t^*) + \lambda \sum_{i=0}^{3} d_p(P_i, \tilde{P}_i(P_t^*))\right]. \quad (5)$$

Various metrics can be used to measure the distance $d_p(P_t, P_t^*)$ between P_t and P_t^* in a cube. In this work, we add up the distances from the sampled points on P_t to the plane P_t^*. The points on P_t are sampled using the domain grids of resolution 512×512 on the xy, yz and zx faces, respectively. Similarly, we compute $d_p(P_i, \tilde{P}_i(P_t^*))$. However, since each local plane is closely related to its adjacent planes, the sampling of points on P_i is carried out using only the joint boundary face of P_t^* and P_i.

The refinement of the plane parameters is iteratively performed through all planes in $\mathcal{P}$. The iteration stops, when the difference between the results of the ith iteration and the $(i+1)$th iteration becomes negligible. Then, the final plane parameters are encoded using the plane compression algorithm in [13], and transmitted to the decoder.

3. MDC DECODING ALGORITHM

In the decoder, two decoding scenarios are possible. As shown in shown in Fig. 1, if both channels are available, we decode each bitstream and merge the decoded descriptions together to reconstruct the 3-D plane surface $\tilde{D}_0$. On the other hand, if one channel fails, the decoder uses the other channel only. However, when the half of plane surface information is lost, the reconstructed 3-D model has holes on its surface, which degrade visual quality severely. To overcome this issue, we detect and fill holes by exploiting the high correlation between two plane subsets $D_{1,0}$ and $D_{1,1}$.



(a) (b) (c)

Fig. 3. The proposed hole filling scheme: (a) the reconstructed surface using only one description, (b) the hole candidates, and (c) the hole recovery, where red lines depict the interpolated planes.

Fig. 3 illustrates the proposed hole filling scheme. In Fig. 3 (a), the surfaces from one description are depicted as blue lines, and those from the other missing description as dotted gray lines. First, we collect all empty cubes around the intact cubes as hole candidates. Specifically, all cubes are traversed in the raster scan order and the number of adjacent intact planes K is recorded for each cube, as shown in Fig. 3 (b). In this work, if K is larger than or equal to 2, the cube is declared to be a hole. Then, for each hole, we estimate a new plane $\tilde{P}$. From the K planes, we sample the boundary points $\mathbf{p}_i$'s, $i = 0, 1, \cdots, L-1$, which are located on the faces of the missing cube. For the sampling, we use the global grid of resolution $2^{16} \times 2^{16} \times 2^{16}$. Finally, we estimate the parameters of the plane $\tilde{P} : (\tilde{\mathbf{a}}, \tilde{\mathbf{n}})$ by

$$\tilde{\mathbf{a}} = \frac{1}{L}\sum_{i=0}^{L-1} \mathbf{p}_i,$$

$$\tilde{\mathbf{n}} = \arg\min \sum_{i=0}^{L-1} [\tilde{\mathbf{n}} \cdot (\mathbf{p}_i - \tilde{\mathbf{a}})]^2, \quad (6)$$

subject to $|\tilde{\mathbf{n}}| = 1$, where $\tilde{\mathbf{a}}$ is a point on $\tilde{P}$ and $\tilde{\mathbf{n}}$ is the normal vector. In Fig. 3 (c), the proposed algorithm recovers new planes on hole regions, which are depicted by red lines.

4. SIMULATION RESULTS

The performance of the proposed MDC system is evaluated on the 3-D 'Dancer' model in Fig. 5. The 'Dancer' model has $23K$ planes on $128 \times 128 \times 128$ grid cubes. As the distortion metric, we adopt the Hausdorff distance between the original model D_0 and the reconstructed model $\tilde{D}_0$, $\tilde{D}_{1,0}$, or $\tilde{D}_{1,1}$. Note that the Hausdorff distance is also used in [14] to evaluate the distortion between 3-D mesh models. Then, the obtained distortions are expressed in terms of the peak signal-to-noise ratio (PSNR), where the peak difference is the diagonal length of the bounding cube for the 3-D model.

Fig. 4 shows the trade-off relationship between the central distortion $d(D_0, \tilde{D}_0)$ and the side distortion

$$1/2[d(D_0, \tilde{D}_{1,0}) + d(D_0, \tilde{D}_{1,1})]$$

according to the channel error rate λ. In this test, the bitrate is set to 1.0 bits per plane (bpp). As λ becomes higher, the qualities of the reconstructed 3-D models from the side decoders become better, while the central reconstruction becomes worse.

Next, we assume that only one bitstream is available at the decoder side. The corrupted surface in Fig. 5 (a) is recovered in Fig. 5

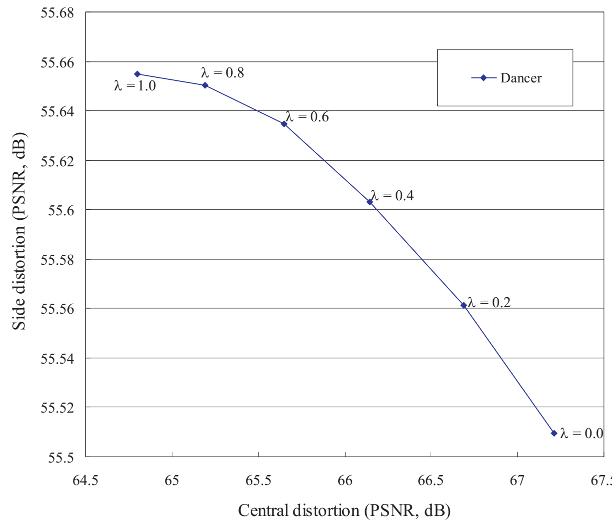
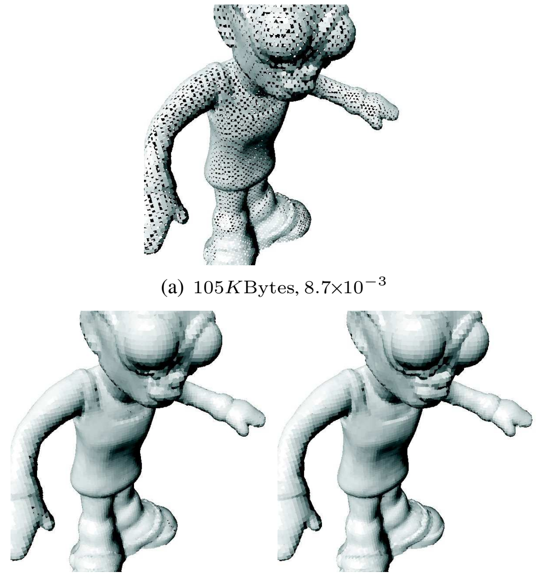


Fig. 4. The relationship between the central distortion and the side distortion according to the channel error probability λ.



(a) $105K\text{Bytes}, 8.7{\times}10^{-3}$

(b) $105K\text{Bytes}, 3.2{\times}10^{-4}$ (c) $146K\text{Bytes}, 1.0{\times}10^{-6}$

Fig. 5. The hole filling of the 'Dancer' model: (a) the reconstructed surface from only one description, (b) the recovered surface using the hole filling algorithm, and (c) the error-free decoding of two bitstreams. The first number represents the bit rate, while the second number is the distortion.

(b), using the hole filling algorithm. The annoying holes are faithfully removed. Moreover, compared with the error-free reconstruction in Fig. 5 (c), the recovered surface in Fig. 5 (b) exhibits only modest degradation. The proposed algorithm consumes about 40% more bits ($210K$ bytes) to encode $D_{1,0}$ and $D_{1,1}$ separately than to encode the whole model into a single description ($146K$ bytes). However, using additional bits, the proposed algorithm provides an acceptable 3-D data reconstruction, even when one channel is totally lost. These results demonstrate that the proposed algorithm guarantees a minimum but acceptable quality reconstruction of 3-D data even in severe channel conditions.

5. CONCLUSION

In this paper, we proposed an algorithm for robust transmission of 3-D surfaces based on the MDC framework. First, we split a plane-based 3-D surface into two descriptions using the graph coloring scheme. In order to minimize the expected distortion, we encoded the parameters of plane patches according to the channel error rate. Finally, we proposed a novel 3-D surface recovery algorithm, which reconstructs an acceptable quality surface even when only one description is available. The simulation result demonstrated that the proposed MDC algorithm is a promising framework for robust transmission of 3-D data over noisy channels.

6. REFERENCES

[1] Z. Yan, S. Kumar, and C.-C. J. Kuo, "Mesh segmentation schemes for error resilient coding of 3-D graphic models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 138–144, Jan. 2005.

[2] G. Al-Regib, Y. Altunbasak, and J. Rossignac, "Error-resilient transmission of 3D models," *ACM Trans. Graph.*, vol. 24, no. 2, pp. 182–208, Apr. 2005.

[3] S.-B. Park, C.-S. Kim, and S.-U. Lee, "Error resilient 3-D mesh compression," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 885–895, Oct. 2006.

[4] C.-S. Kim and S.-U. Lee, "Multiple description coding of motion fields for robust video transmission," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 9, pp. 999–1010, Sept. 2001.

[5] Y. Wang, A. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. IEEE*, vol. 93, no. 1, pp. 57–70, Jan. 2005.

[6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. ACM SIGGRAPH*, July 1992, pp. 71–78.

[7] S. Rusinkiewicz and M. Levoy, "Qsplat: A multiresolution point rendering system for large meshes," in *Proc. ACM SIGGRAPH*, July 2000, pp. 343–352.

[8] P .Jaromersky, X. Wu, Y. Chiang, and N. Memon, "Multiple description geometry compression for networked interactive 3d graphics," in *Proc. ICIG 2004*, Dec. 2004, pp. 468–471.

[9] M. O. Bici and G. B. Akar, "Multiple description scalar quantization based 3d mesh coding," in *Proc. IEEE ICIP 2006*, Oct. 2006, pp. 553–556.

[10] S.-B. Park, H. Choi, and S.-U. Lee, "Multiscale surface representation and rendering for point clouds," in *Proc. IEEE ICIP*, Oct. 2004, pp. 1939–1942.

[11] J. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[12] D. Brélaz, "New methods to color the vertices of a graph," *Commun. ACM*, vol. 22, no. 4, pp. 251–256, Apr. 1979.

[13] S.-B. Park, H. Choi, and S.-U. Lee, "3-D geometry compression using multiscale plane based representation and zerotree based coding," in *Proc. IEEE ICIP*, Oct. 2006, pp. 537–540.

[14] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, June 1998.