

# Palette-based Color Attribute Compression for Point Cloud Data

Li Cui<sup>1</sup> and Euee S. Jang<sup>2\*</sup>

<sup>1</sup> Department of Computer & Software, Hanyang University  
Seoul, Republic of Korea;  
and Department of Computer Science & Technology, Yanbian University  
Yanji, China  
[e-mail: Lcui2000@gmail.com]

<sup>2</sup> Department of Computer & Software, Hanyang University  
Seoul, Republic of Korea  
[e-mail: esjang@hanyang.ac.kr ]

\*Corresponding author: Euee S. Jang

*Received August 14, 2018; revised November 22, 2018; accepted December 30, 2018;  
published June 30, 2019*

---

## Abstract

Point cloud is widely used in 3D applications due to the recent advancement of 3D data acquisition technology. Polygonal mesh-based compression has been dominant since it can replace many points sharing a surface with a set of vertices with mesh structure. Recent point cloud-based applications demand more point-based interactivity, which makes point cloud compression (PCC) becomes more attractive than 3D mesh compression. Interestingly, an exploration activity has been started to explore the feasibility of PCC standard in MPEG. In this paper, a new color attribute compression method is presented for point cloud data. The proposed method utilizes the spatial redundancy among color attribute data to construct a color palette. The color palette is constructed by using K-means clustering method and each color data in point cloud is represented by the index of its similar color in palette. To further improve the compression efficiency, the spatial redundancy between the indices of neighboring colors is also removed by marking them using a flag bit. Experimental results show that the proposed method achieves a better improvement of RD performance compared with that of the MPEG PCC reference software.

---

**Keywords:** Point cloud, palette color, clustering, spatial redundancy, color attribute compression

---

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00067, Development of ICT Core Technologies for Safe Unmanned Vehicles).

## 1. Introduction

Recently, 3D models are widely used in many applications such as virtual reality (VR), animation, game, scientific visualization and immersive communications. With the significant development in 3D data acquisition techniques, 3D point cloud representation has been widely used to represent 3D objects or scenes instead of traditional mesh-based method. A 3D point cloud is often collected in millions of points, which consist of geometry, color, normal, and other attributes. This promotes the development of 3D visualization and data compression technologies for realistic 3D applications.

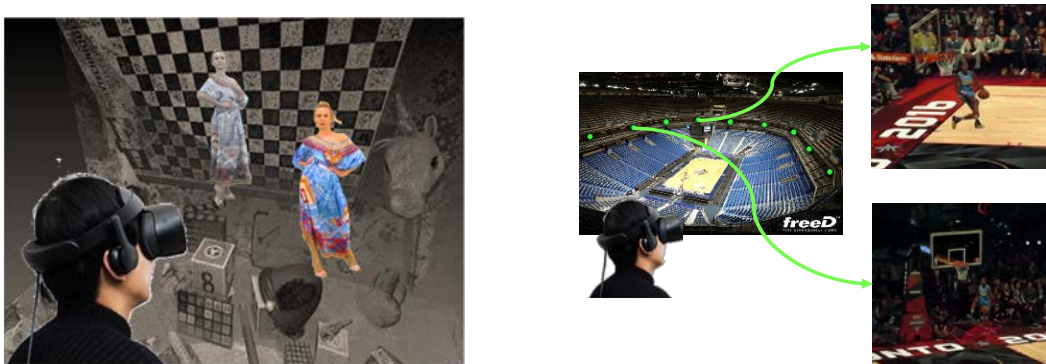
Recent advances of 3D camera technology have driven demand for the technologies related to point cloud. Modern cameras provide the depth capture capability to acquire and process point clouds in real time. And some of them have built-in features to enable 3D scanning using mobile devices such as Sony Xperia XZ1 and Apple iPhoneX. For example, Xperia XZ1 can capture 10k points to support some 3D applications and iPhone X can capture more than 30k points to enable facial recognition [1][2]. Advanced point cloud sensing technologies are being applied to recent point cloud depth cameras. For example, a point cloud depth camera from Huawei is capable of reconstructing 300k points in 10s and another camera from Qualcomm can capture 10k points of depth, with the spacing between points as 0.1mm [3][4]. The ongoing improvement of 3D camera technology results in more points captured to enable the better consumer experience.

Meanwhile, the devices related to virtual technologies such as consumer head-mounted VR displays have been widely used in video game industry [5][6]. Low latency and fast refresh rates are required in these commodity devices to provide consumers with good VR experience. However, the reconstructed point cloud in these applications typically contains millions of points to preserve a high level of detail. And the applications need to process complex computations for interactive rendering. Visual quality observed by consumers is generally constrained by the limited memory storage of the devices and network bandwidth due to the large rendering cost. Thus, providing efficient method to compress 3D point cloud data becomes important for the applications. For example, it being supported in a real-time application as shown in Fig. 1(a) can result in a richer VR experience with less memory. And while it being used for a television application as shown in Fig. 1(b), broadcasting of VR can be allowed over bandlimited channels to consumer devices. 3D Graphics (3DG) group in Motion Picture Experts Group (MPEG) has been working on the standardization of point cloud compression technologies for several years to cope with massive 3D point clouds.

According to mesh-based models, there have been many algorithms for static and dynamic mesh compression proposed over the last decade for storage and transmission purposes [7][8]. The static mesh compression is to compress 3D mesh data through exploiting spatial and topological correlation among geometry, connectivity, and other attributes [9][11]. Dynamic mesh compression exploits temporal and optionally spatial dependency among mesh data being associated with dynamic scene content (e.g., vertices, faces, and texture) [12].

The development of 3D data acquisition technology has resulted in a rapidly increasing volume of 3D objects. Efficient compression becomes very important to handle a massive number of realistic 3D point cloud data in 3D applications. In the conventional mesh-based representation, the connectivity information is used to construct the topological relationship of polygons. To maintain the original topological relationship among meshes, connectivity data

should be compressed lossless. In point-based representation, a 3D object is described as a set of points without any further topological information. The topological relationship among points is no longer necessary for point cloud compression technologies. Thus, the design of point-based compression method must meet different requirements compared with the mesh-based compression method due to the aforementioned features (i.e., a huge amount of data and no need to handle topological information) of point cloud data.



**Fig. 1.** Two example applications of PCC technique employed in a VR device

In recent years, the development of point cloud compression technologies is attracting more and more attention. There have been several compression methods proposed to exploit spatial and temporal correlations among geometry, normal, color, and other attribute information of points [13]-[19]. Some of them represent each point cloud as a graph based on points and their geometric relationship with nearby points; and use the graph-based transforms to encode geometry and attribute data. Cohen et al. handle the color attribute data over the graphs and compress them through block-based prediction and graph transform coding [20]. A Gaussian process transform is used to encode colors and other attributes for static 3D point cloud in [13]. Several methods for compressing dynamic 3D point cloud are based on graph and estimate motions between sequences to remove the temporal redundancy [14][15]. 3D point cloud sequences are represented by high-resolution sub-divisional triangular meshes and motion estimation and graph wavelet transform are applied to encode the meshes [16]. Although graph-based methods are useful to remove spatial redundancy, they are characterized by high complexity due to the computational cost of constructing graph.

Other compression methods have been proposed to assign points according to their spatial distribution such as hierarchical point clustering and those using octree data structure. Most of them are typically corresponding to geometry data compression for 3D point cloud. Coarse approximation and level of details are used to form hierarchical clusters of geometry data for 3D point cloud in [17]. Octree-based compression methods often represent points according to their geometrical information; then, apply different spatial resolution according to different depth of octree to allow progressive transfer of point cloud data [18][19].

In this paper, we focus on color attribute compression for 3D point cloud. There have been several graph-based methods for color attribute compression of 3D point cloud [21]-[24]. The graph transform (GT)-based compression methods for 3D point cloud utilize the spatial correlation among points, and the set of points in 3D point cloud is partitioned into same size of blocks and GT is provided to these blocks [21]. The points in 3D point cloud according to their coordinates were down-sampled to a uniform grid, which in turn was partitioned into

blocks so that GT could be directly applied; and the latter work also includes 3D intra block prediction [20]. While the partitioned blocks can be sparsely populated and few points may be adjacent, Robert et al. use block-based prediction and GT to compress point clouds that contain sparsely-populated blocks [22]. It is proved that Gaussian process transform (GPT) can be used to model the statistics of the signal on points based on their coordinates in [23]. And GPT is in turn used in a transform coding system to encode color attribute data of 3D static point cloud [13]. However, these GT based methods require complex matrix decompositions. A hierarchical transform is used to reduce the complexity of transform computation which is derived according to the colors associated with geometry data in spatial space [24].

Meanwhile, it is validated that the conventional image compression methods such as principal components analysis (PCA), vector quantization (VQ), run length encoding (RLE), and JPEG can be used to compress point cloud data [19][25][26]. PCA is used for the points in the flat voxel and a raster scan is followed to mapping points into blocks. RLE is used to remove the spatial redundancy among adjacent color data. JPEG is used to compress point data based on block level.

The proposed method utilizes the spatial redundancy among colors to construct a color palette using K-means clustering method and quantizes color data based on the palette colors. Additionally, we observe that the color value of a point has high correlation with those of points that are closer in distance. To further compress color data, one flag bit for each color is used to mark whether the current color being similar with previous one.

The rest of the paper is organized as follows. In Section 2, a base reference software of MPEG PCC is described. We describe the proposed method in Section 3. In Section 4, the experimental results are presented in detail. Finally, we conclude this paper in Section 5.

## 2. Related Works

An exploration activity has been started to promote the development of 3D point cloud compression (PCC) technology in MPEG. A base reference software for MPEG PCC (here after called the anchor) has been developed to provide necessary functionality and methodology for efficient representation and compression of 3D point cloud [26]. The architecture of MPEG PCC is based on bounding box normalization and outlier filter, octree composition, surface approximation, and occupancy code entropy coding as shown in Fig. 2.

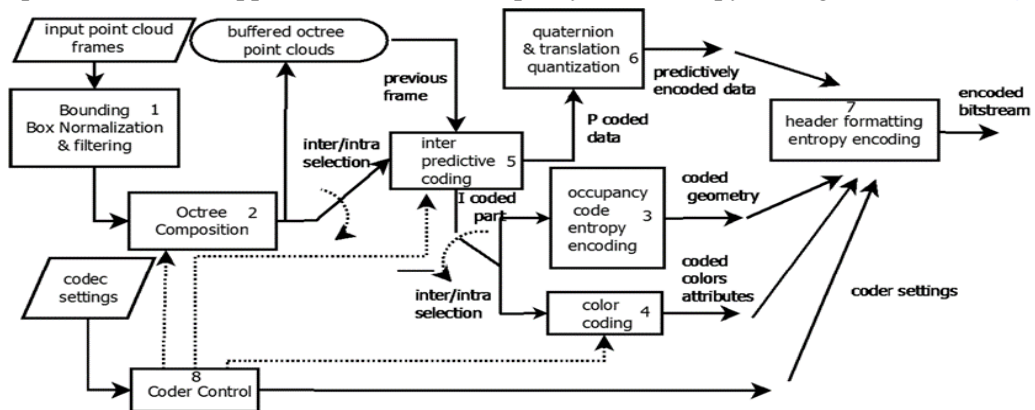


Fig. 2. Schematic of MPEG PCC anchor encoder [26]

Through bounding box normalization and outlier filter, the geometry data of input point cloud are transformed from original high-accuracy data to approximately similar integer data for performing further coding operation easily. Then the anchor codec works with a specified voxel size of an octree. All enclosed points in the octree will be decimated to the voxel centroids. Additionally, voxel centroids can be added to the output bitstream to refine the precision of the octree voxel centers. At decoder, all points are decoded as their voxel centroids in the octree grid and refined to their centroid positions if available.

In the anchor codec, the geometry data as well as attribute data are organized by using octree data structure in the anchor codec, while the attribute data is coded separately by using specialized encoder. JPEG compression method for MPEG PCC is used to compress color attribute data. Color attribute data are compressed by traversing the octree in depth first order and mapping the average color of the points in each leaf voxel to an image grid. Although JPEG can achieve high rates of compression for image compression, it may be not efficient for point cloud compression. The specified number of points being mapped to blocks result in only the local spatial redundancy utilized by the JPEG-based compression method.

A hybrid compression method has been investigated to improve the performance of the anchor codec, which combines VQ-based and RLE-based methods [25]. The hybrid method can achieve better rate-distortion performance than that of the JPEG method, but it reduces little distortion at the low bitrate. The reason is that the hybrid method only utilizes the local special redundancy too. Due to the burden of large amount of point data, the high compression ratio with an acceptable decrease in PSNR is the performance requirement for point cloud-based applications.

### 3. Proposed Method

In this paper, a palette-based compression method for 3D point cloud is presented to utilize the global spatial redundancy among points through clustering color data in color space. A high-level block diagram of the proposed method is shown in Fig. 3, which includes data analysis, geometry and attribute data coding, and entropy coding.

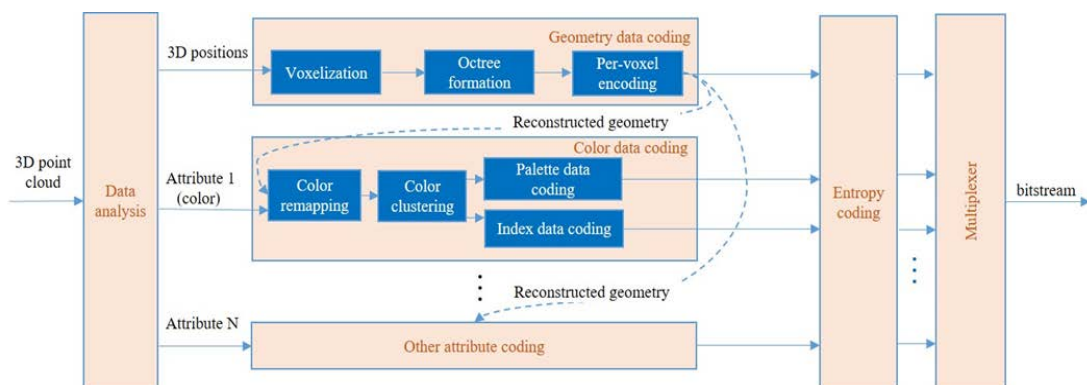


Fig. 3. Block diagram of the proposed method

Through data analysis, the geometry data and attribute data (e.g., color, normal and reflectance) are separated from the original point cloud and organized as easy to be further processed independently. Geometry data coding includes three processes to encode 3D positions. The voxelization is to convert the input 3D position data into voxel representation by using bounding box. The octree formation is to represent voxelized 3D positions as an

octree structure. Through per-voxel encoding process, each voxel in a P-frame is encoded with conventional prediction and quantization methods. This process can be skipped when an I-frame is encoded. Reconstructed geometry information as the output of geometry data coding is used to determine which corresponding attribute data are to be encoded. As the first step of color attribute coding, the color remapping process is to map one or more color values to one color value based on the encoded geometry information. If the encoded geometry is lossless, this process can be skipped. Through color data clustering, the K-means clustering method is employed to cluster color data and the centroids of the clusters are used to construct a color palette. The index of the centroid color in palette is used to represent those colors in corresponding cluster. The index data is compressed by utilizing the spatial redundancy through the index data coding process. After geometry and attribute data have been encoded independently, entropy coding method is used respectively for all encoded data. Finally, the entropy coded bits are merged to organize the encoded bitstream.

### 3.1 Color Data Clustering

K-means clustering is an iterative method to minimize the quantization error. It consists of two separate phases including the  $k$  centroids of cluster to be calculated and the nearest centroid for each color to be found. These two phases are repeated until there are no changes in the centroids of cluster.

Let us assume that a set of color data  $P = \{p_0, p_1, \dots, p_N\}$  has to be cluster into  $k$  number of cluster. Let  $C = \{c_0, c_1, \dots, c_k\}$  be the cluster centroids and  $D = \{d_0, d_1, \dots, d_N\}$  be the indices of  $P$ . The algorithm of k-means clustering can be summarized as follows.

**Step 1:** Initialize  $C = \{c_0, c_1, \dots, c_k\}$ .

**Step 2:** For each color in  $P$ , calculate the mean square error  $e$ , between the centroid in  $C$  and each color in  $P$  using the relation given below.

$$e_j = \|p_j - c_i\|^2, \text{ for } i \in [1, K] \text{ and } j \in [1, N] \quad (1)$$

**Step 3:** Take each color to its nearest centroid based on error  $e$ . Then, each color corresponds an index in  $D$  as follows.

$$d_j = \underset{j \in [1, N]}{\operatorname{argmin}} e_j \quad (2)$$

**Step 4:** Update the centroids  $C$  using the relation given below.

$$c_i = \frac{1}{t_i} \sum p_j, \text{ for all } d_j = i \quad (3)$$

where  $t_i = \operatorname{count}(d_j)$

**Step 5:** Repeat the steps 2-4 until  $C$  is no change.

In this paper, the proposed method is performed to obtain local minimum of quantization error. The color data in  $P$  is randomly down-sampled; then, the initialized centroids in  $C$  are determined according to the square errors between the origin and the sampled colors.

### 3.2 Index Data Encoding

After the color clustering process, the centroids of clusters are used to construct a color palette. The index of the centroid in palette is used to represent the color data. We introduced one-bit-flag-per-point method to compress index data through exploring the similarity of the neighboring index data.

It is assumed that the number of colors ( $N$ ) corresponds to the number of index data ( $N$ ). The set of index data is mapped into 1D array and each index is represented by  $R$  (i.e.,  $R = \operatorname{ceiling}(\dots)$ ) bits. If the current color is similar to its previous one, only one bit flag is enough to indicate it instead of using  $R$  bits. If not, it needs to be represented by the

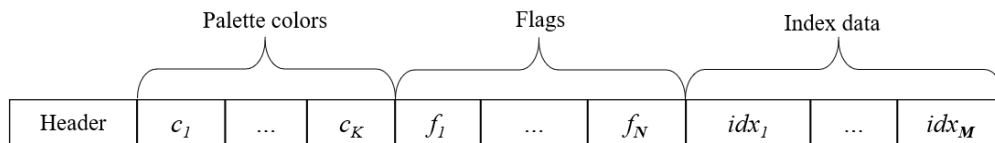
corresponding index data. Accordingly, while there are the number of color data ( $L$ ) which can be represented only with one-bit flag, the whole index data with can be represented by use  $N+(N-L)\times R$  number of bits instead of the original  $N\times R$  number of bits. This implies that the proportion of the similar index data among the whole index data ( $L/N$ ) should be larger than  $1/R$ . And as a larger  $L$  is obtained, a higher compression ratio can be achieved. We perform the clustering process to investigate the number of the similar index data  $L$  among the whole index data  $N$ . The investigation results based on different static point clouds are given in **Table 1**.  $K$  is the number of colors in palette and each index is represented by 8 bits ( $R=8$ ). Every  $L/N$  in the results is bigger than 12.5 percent, which certificates that the one-bit-flag-per-point method can be used to compress the index data in the proposed method.

**Table 1.** Proportion of the similar index data among the whole index data

Point Cloud	$L/N$			
	$K=8$	$K=32$	$K=128$	$K=256$
<i>Egyptian_mask</i>	80.8%	51.9%	25.6%	15.9%
<i>Staeue_Klimt</i>	74.0%	56.1%	42.6%	37.0%
<i>Head(00039)</i>	67.9%	47.7%	33.2%	25.3%
<i>House_without_roof</i>	79.9%	63.0%	46.5%	38.9%
<i>Landscape(00014)</i>	66.9%	44.9%	27.6%	21.1%
<i>Stanford_Area4</i>	82.3%	69.5%	59.3%	52.7%

### 3.3 Bitstream Structure

The encoded result of color attributes in each 3D point cloud includes a header part, a color palette, a list of one-bit flags and encoded index data as shown in **Fig. 4**. The header part includes several parameters such as the number of colors in palette  $K$ , the total number of colors  $N$  and the number of encoded index data  $M$ . The color palette is comprised of the  $K$  number of colors. The flag part is composed of the  $N$  number of one-bit-flag data to indicate whether the current color is similar to its previous one.



**Fig. 4.** Bistream structure of the proposed method

## 4. Experimental Results and Analysis

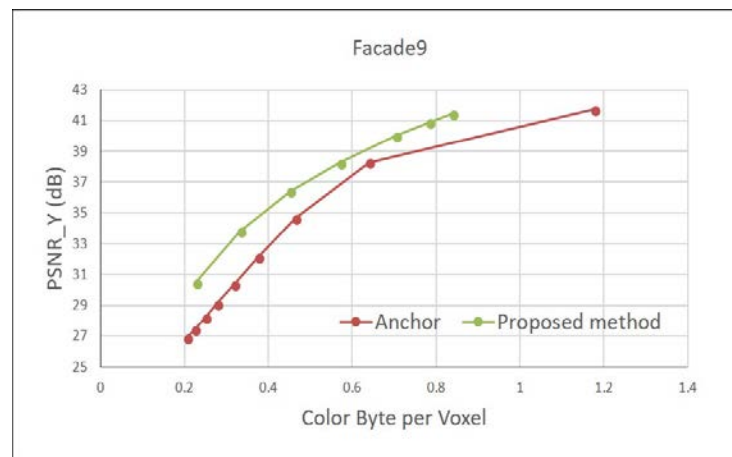
To investigate the performance of the proposed method, we implement our method based on the anchor [26]. We used 19 static point clouds that are test datasets used in MPEG-3DG. Some of them have simple distribution in color space and others are sparse.

The rate-distortion (RD) performance between the proposed method with different sets of palette colors  $\{8, 16, 32, 64, 128, 192, 256\}$  and the anchor at different quality factors  $\{60, 65, 70, 75, 80, 85, 90, 95, 100\}$  has been compared. The results of the RD performance are classified into five types based on the relative position of the two curves (i.e., the curve of the

proposed method and the curve of the anchor) as shown in **Table 2**. **Fig. 5** shows four examples of the RD performance comparisons to describe four types (i.e., BETTER, CROSS, SIMILAR BETTER and SIMILAR WORSE) in detail. The number of test data for the BETTER type is seven and their proportion in test set is 36.8% shown in **Table 2**. And six number of the BETTER-type data are similar, which are accounted for 31.6% of the proportion in our test set. Based on the two examples shown in **Fig. 5 (a)** and **(c)**, we can conclude that the proposed method can achieve better RD performance for 68.4% of test data than that of the anchor. The proposed method used for the CROSS-type data as shown in **Fig. 5(b)** achieves lower RD performance than that of the anchor at the low bitrate, while being better at the high bitrate. The three SIMILAR WORSE-type data include *Facade64* (19,714,629 points), *Landscape14* (72,145,549 points) and *House57* (5,001,077 points). These SIMILAR WORSE-type data have sparse distribution in color space, so the limited number of colors in palette is not enough to represent the whole color data efficiently.

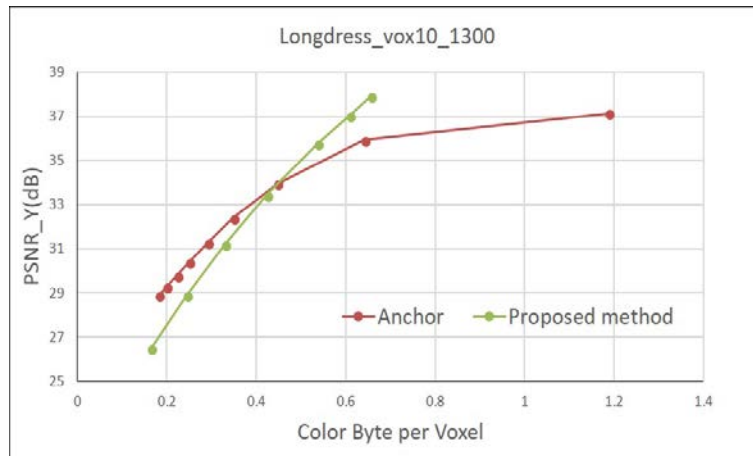
**Table 2.** Proportion comparison among five types of RD curves

Type	Name of test data	Number of test data	Proportion
BETTER	<i>Arco_Valentino_Dense, Façade9, Palazzo_Carignano_Dense, Queen_frame_0200, Soldier_vox10_0690, Stanford_Area2, Stanford_Area4</i>	7	36.8%
CROSS	<i>Longdress_vox10_1300, Redandblack_vox10_1050, Shiva35</i>	3	15.8%
SIMILAR BETTER	<i>Egyptian_mask, Façade15, Frog67, Head39, Loot_vox10_1200, Staue_Klimt</i>	6	31.6%
SIMILAR WORSE	<i>Façade64, House57, Landscape14</i>	3	15.8%
WORSE	-	0	0.0%

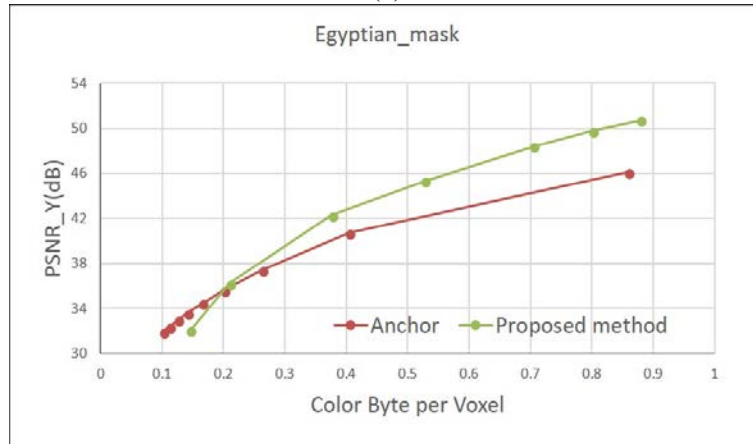


(a)

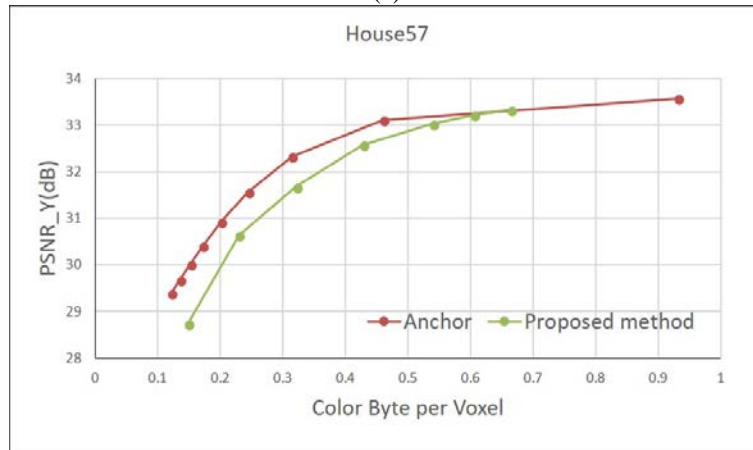




(b)



(c)

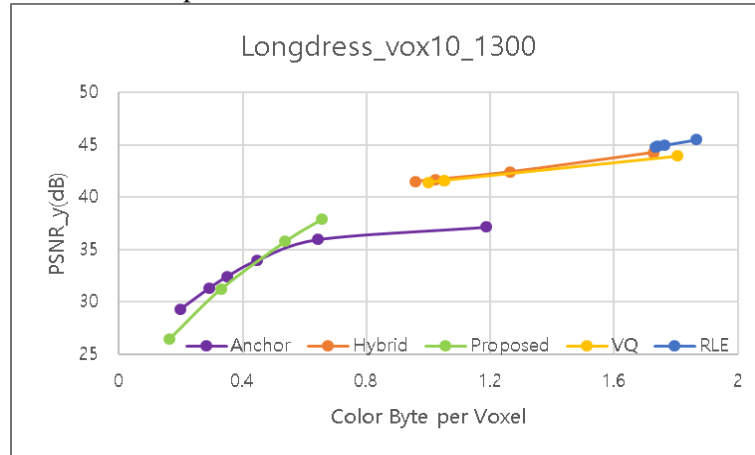


(d)

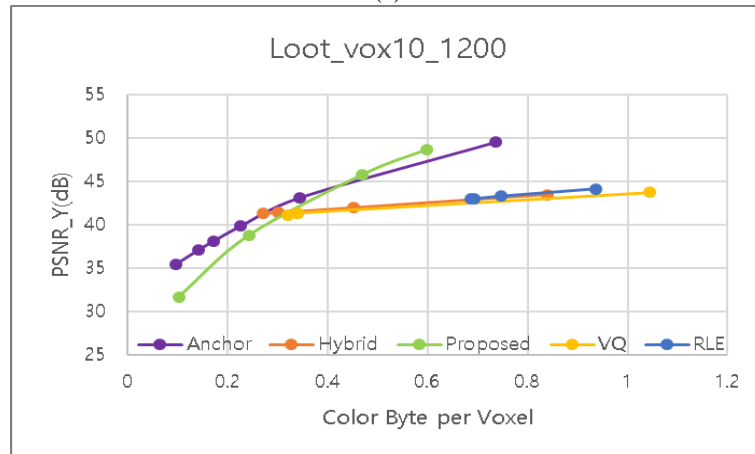
**Fig. 5.** Examples of RD performance comparison for four types (a) BETTER (b) CROSS (c) SIMILAR BETTER (d) SIMILAR WORSE

Additionally, the RD performance among the proposed method with other methods has been compared and the results are shown in Fig. 6. The methods include VQ, RLE and Hybrid are set with different palette colors {8, 32, 128, 256}, and the anchor is set at different quality

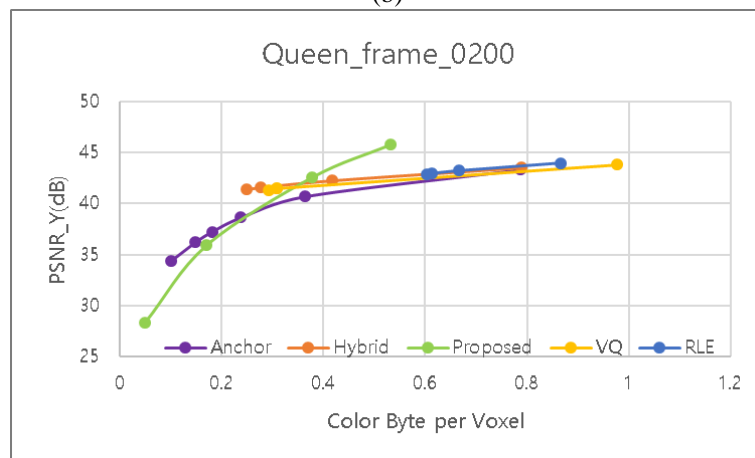
factors {65, 80, 85, 90, 95,100}. Through comparison of the results, we can conclude that the proposed method can achieve best RD performance than the other methods in the byte per voxel (bpv) range from 0.4 to 0.6. It means that the proposed method can achieve high compression ratio with an acceptable decrease in PSNR.



(a)



(b)



(c)

Fig. 6. RD performance comparison among the proposed method with others

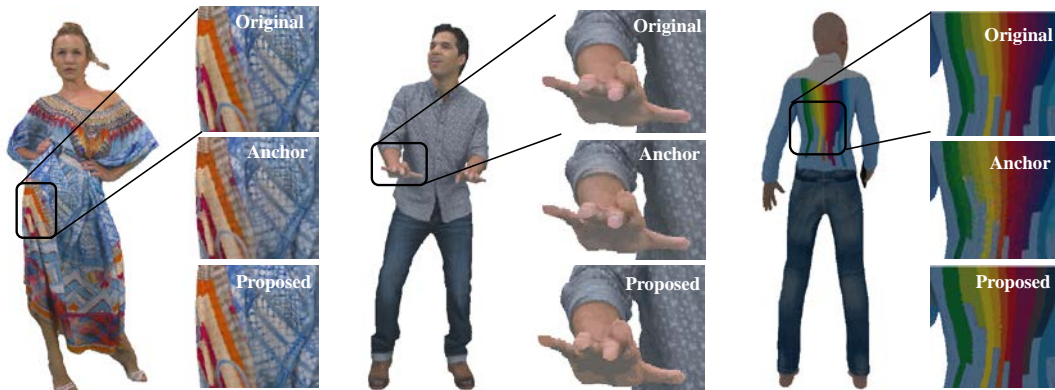
Then, a part of the experimental results are chosen to compare the quality of compressed point clouds based on similar bpv. **Table 3** shows the PSNR comparison results of the anchor and the proposed method for three examples based on specific bitrates. The threshold for the anchor is the quality factor of JPEG (QF), as well as the threshold for the proposed method is the number of color data in the palette (NC). **Fig. 7** shows the quality comparison of rendering results based on the results in **Table 3**. The results show that the proposed method is better than the anchor to be used for the test sets with simple distribution in color space (i.e., *Longdress\_vox10\_1300*, *Queen\_frame\_0200*). While for the test data set with complex distribution in color space (i.e., *Loot\_vox10\_1200*), the proposed method has no obvious advantage due to the limitation of the number of color data in palette. Through the comparison of the results, we can conclude that the proposed method can utilize the global spatial redundancy of the whole color data through clustering. It can achieve better quality for the test data with the simple distribution in color space. Additionally, **Table 3** shows the comparison of the decoding time per point between that of the proposed method ( $T_P$ ) and the anchor ( $T_A$ ). The reduction in decoding time ( $\Delta T$ ) is also measured, which is defined as follows:

$$\Delta T = (T_P - T_A)/T_A \times 100\% \quad (4)$$

The results should not be taken as the accurate comparison of complexity, since none of the tested methods are optimized in any sense. Rather, the results can be taken as an indication that the proposed method is not showing any evidence that the proposed method is more or less complex than the anchor method.

**Table 3.** Performance comparison between the anchor and the proposed method for three examples

Test data	Number of points	Method	Threshold	PSNR	bpv	Decoding time (ms/p)	$\Delta T$
<i>Longdress_vox10_1300</i>	857,966	Anchor	QF=95	35.93	0.64	0.309	5.50%
		Proposed	NC=256	37.90	0.66	0.326	
<i>Loot_vox10_1200</i>	805,285	Anchor	QF=90	38.11	0.23	0.271	7.38%
		Proposed	NC=32	38.78	0.24	0.291	
<i>Queen_frame_0200</i>	1,000,993	Anchor	QF=90	34.76	0.18	0.280	-5.36%
		Proposed	NC=32	35.89	0.17	0.265	



**Fig. 7.** Quality comparison of rendering results based on the proposed method and the anchor for three examples (**Left**) *Longdress\_vox10\_1300* (**Middle**) *Loot\_vox10\_1200* (**Right**) *Queen\_frame\_0200*

## 5. Conclusion

In this paper, we presented a palette-based compression method to compress color attributes of 3D point cloud. A clustering method is used to remove spatial redundancy among adjacent color data and one bit flag per color is used to indicate whether the current color data can be represented by the similar color in palette compared with neighboring one. From the experimental results, it is shown that the proposed method compared with that of the anchor. The results of RD performances showed that the proposed method can achieve quality improvement based on different bitrate range for different test data. The development of rate-control method for color attribute compression will be our future task.

## References

- [1] "Specifications for Xperia™ XZ1," Sonymobile.com. [Online]. [Article \(CrossRef Link\)](#).
- [2] "iPhone-X tech specs," Apple.com. [Online]. [Article \(CrossRef Link\)](#).
- [3] Millen Yanachkov, "Huawei P11 may feature a camera that rivals Apple's TrueDepth system on the iPhoneX," Phonearena.com, 2017. [Online]. Available: [Article \(CrossRef Link\)](#).
- [4] "Qualcomm First to Announce Depth-sensing Camera Technology Designed for Android Ecosystem," Qualcomm.com, 2017. [Online]. Available: [Article \(CrossRef Link\)](#).
- [5] J. B. Kim and H. S. Jun, "Vision-based location positioning using Augmented Reality for Indoor navigation," *IEEE Trans. on Consumer Electronics*, vol. 54, no. 3, pp. 954-962, Aug. 2008. [Article \(CrossRef Link\)](#).
- [6] H. Heo, E. C. Lee, K. R. Park, C. J. Kim, and M. C. Whang, "A realistic game system using multi-modal user interfaces," *IEEE Trans. on Consumer Electronics*, vol. 56, no. 3, pp. 1364-1372, Aug. 2010. [Article \(CrossRef Link\)](#).
- [7] J. Peng, C. S. Kim and C. C. Jay Kuo, "Technologies for 3D mesh compression: A survey," *J. Visual Communication Image Representation*, vol. 16, no. 6, pp. 688-733, Dec. 2005. [Article \(CrossRef Link\)](#).
- [8] A. Maglo, G. Lavoué, F. Dupont, and C. Hudelot, "3D mesh compression: Survey, Comparisons and Emerging Trends," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 44:1-44:41, Feb. 2015. [Article \(CrossRef Link\)](#).
- [9] M. Deering, "Geometry Compression," in *Proc. of SIGGRAPH '95*, pp. 13-20, 1995. [Article \(CrossRef Link\)](#).
- [10] D. Y. Lee, S. B. Sull, and C. S. Kim, "Progressive 3D mesh compression using MOG-based Bayesian entropy coding and gradual prediction," *The Visual Computer*, vol. 30, no. 10, pp. 1077-1091, Mar. 2014. [Article \(CrossRef Link\)](#).
- [11] F. Caillaud, V. Vidal, F. Dupont, and G. Lavoué, "Progressive compression of arbitrary textured meshes," *Computer Graphics Forum*, vol. 35, no. 7, pp. 475-484, Oct. 2016. [Article \(CrossRef Link\)](#).
- [12] L. Váša, S. Marras, K. Hormann, and G. Brunnett, "Compressing dynamic meshes with geometric Laplacians," *Computer Graphics Forum*, vol. 33, no. 2, pp. 145-154, May 2014. [Article \(CrossRef Link\)](#).
- [13] R. L. de Queiroz and P. A. Chou, "Transform coding for point clouds using a gaussian process model," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3507-3517, July 2017. [Article \(CrossRef Link\)](#).
- [14] D. Thanou, P. A. Chou and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1765-1778, April 2016. [Article \(CrossRef Link\)](#).
- [15] R. L. de Queiroz and P. A. Chou, "Motion-compensated compression of dynamic voxelized point clouds," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3886-3895, Aug. 2017. [Article \(CrossRef Link\)](#).

- [16] A. Anis, P. A. Chou and A. Ortega, "Compression of dynamic 3D point clouds using subdivisional meshes and graph wavelet transforms," in *Proc. of IEEE ICASSP*, pp.6360-6364, 2016. [Article \(CrossRef Link\)](#).
- [17] Y. Fan, Y. Huang, and J. Peng, "Point cloud compression based on hierarchical point clustering," in *Proc. of IEEE SIPAASC*, Kaohsiung, pp. 1-7, 2013. [Article \(CrossRef Link\)](#).
- [18] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of Point Cloud Streams," in *Proc. of IEEE ICRA*, Saint Paul, MN, pp. 778-785, 2012. [Article \(CrossRef Link\)](#).
- [19] K. Ainala, R. N. Mekuria, B. Khathariya, Z. Li, Y. K. Wang, and R. Joshi, "An improved enhancement layer for octree based point cloud compression with plane projection approximation," in *Proc. of ADIP*, pp.22-25, 2016. [Article \(CrossRef Link\)](#).
- [20] R. A. Cohen, D. Tian, and A. Vetro, "Point cloud attribute compression using 3D intra prediction and shape-adaptive transforms," in *Proc. of IEEE DCC*, USA, pp. 141-150, 2016. [Article \(CrossRef Link\)](#).
- [21] C. Zhang, D. Florêncio and C. Loop, "Point cloud attribute compression with graph transform," in *Proc. of IEEE ICIP*, Paris, pp. 2066-2070, 2014. [Article \(CrossRef Link\)](#).
- [22] R. A. Cohen, D. Tian and A. Vetro, "Attribute compression for sparse point clouds using graph transforms," in *Proc. of IEEE ICIP*, pp. 1374-1378, 2016. [Article \(CrossRef Link\)](#).
- [23] P. A. Chou, and R. L. de Queiroz, "Gaussian process transforms," in *Proc. of IEEE ICIP*, Phoenix, AZ, pp. 1524-1528, 2016. [Article \(CrossRef Link\)](#).
- [24] R. L. de Queiroz and P. A. Chou, "Compression of 3d point clouds using a region-adaptive hierarchical transform," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3947-3956, Aug. 2016. [Article \(CrossRef Link\)](#).
- [25] L. Cui, H. Y. Xu, and E. S. Jang, "Hybrid color attribute compression for point cloud data," in *Proc. of IEEE ICME*, Hong Kong, pp. 1273-1278, 2017. [Article \(CrossRef Link\)](#).
- [26] R. Mekuria, K. Blom and P. Cesar, "Design, implementation and evaluation of a point cloud codec for tele-immersive video," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828-842, April 2017. [Article \(CrossRef Link\)](#).



**Li Cui** received the B.S. degree from Yanbian University, Yanji, China in 2000 and the M.S. degree from Honam University, Korea in 2007. She is currently working toward a Ph.D. at Hanyang University, Seoul, Korea. And she is also a lecturer in the Department of Computer Science and Technology, College of Science and Engineering, Yanbian University, Yanji, China. Her current research interests include MPEG Reconfigurable Graphics Coding, Computer Graphics and Point Cloud Compression.



**Eusee S. Jang** received a B.S. from Jeonbuk National University, Korea and a Ph. D. from SUNY at Buffalo, NY, USA. He is a Professor in the Dept. of Computer Science & Engineering, Hanyang University, Seoul, Korea. His research interests include image/video coding, reconfigurable video coding, and computer graphics objects. He has authored more than 150 MPEG contribution papers, more than 30 journal or conference papers, 35 pending or accepted patents, and two book chapters. Dr. Jang has received three ISO/IEC Certificates of Appreciation for contributions to MPEG-4 development. He received the Presidential Award from the Korean Government for his contribution to MPEG standardization.