CLUSTER-BASED POINT CLOUD CODING WITH NORMAL WEIGHTED GRAPH FOURIER TRANSFORM

Yiqun Xu[#], Wei Hu[⋆†], Shanshe Wang[⊥†], Xinfeng Zhang[§], Shiqi Wang[∗], Siwei Ma[⊥†], Wen Gao[⊥†]

[#]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[⋆]Institute of Computer Science and Technology, Peking University, Beijing, China
[⊥]Institute of Digital Media, Peking University, Beijing, China
[§]Viterbi School of Engineering, University of Southern California, USA
[∗]Department of Computer Science, City University of Hong Kong, Hong Kong, China
[#]yiqun.xu@vipl.ict.ac.cn, [†]{forhuwei,sswang,swma,wgao}@pku.edu.cn,
[§]zhangxinf07@gmail.com, [∗]shiqwang@cityu.edu.hk

ABSTRACT

Point cloud has attracted more and more attention in 3D object representation, especially in free-view rendering. However, it is challenging to efficiently deploy the point cloud due to its huge data amount with multiple attributes including coordinates, normal and color. In order to represent point clouds more compactly, we propose a novel point cloud compression method for attributes, based on geometric clustering and Normal Weighted Graph Fourier Transform (NWGFT). Firstly, we divide the entire point cloud into different sub-clouds via K-means based on the geometry to acquire sub-clouds with more uniform structures, which enables efficient representation with less cost. Secondly, for the purpose of reducing the redundancy further, we apply NWGFT to each sub-cloud, in which graph edge weights are derived from the similarity in normal. Finally, extensive experimental results show that, compared with traditional transform based point cloud compression, the proposed approach achieves about 34.34% bit rate reduction on average for Y components of color.

Index Terms— 3D Point cloud compression, color attribute, clustering, Graph Fourier Transform

1. INTRODUCTION

With the development of 3D graphic technology and computation hardware, 3D point clouds have attracted intensive attention in the representation for real scenes or 3D objects, due to its simplicity and accuracy compared with traditional free-view rendering methods using multi-view plus depth maps. 3D point clouds consist of a set of points, with 3D coordinates to represent the geometric information. Attribute data can be added to each point to enrich its functions, *e.g.,* color attribute for visualization and visual feature attribute for localization. Point clouds have been widely applied in many fields, such as 3D immersive tele-presence, 3D broadcasting, cultural and heritage reconstruction [1]. Since 3D points are acquired by directly sampling the surfaces of real objects, the point distribution is not as regular as images or videos, which makes point clouds more difficult to represent compactly. Moreover, multiple attributes on each point significantly increase the burden for transmission and storage of 3D point clouds.

Many approaches have been proposed to support efficient storage and transmission of point clouds. In [2], Schnabel and Klein first introduced the octree structure to compress the geometric structure by splitting the entire point cloud into small voxels, which has been proved to be effective in organizing the 3D point cloud data. The most popular toolbox for 3D point cloud processing, the Point Cloud Library (PCL) [3], is composed of a complete set of processing methods for point clouds. It has been applied as the basic infrastructure for point cloud compression, which also adopted the octree structure to organize 3D point clouds representing the object geometric information.

Besides the geometric information, Huang *et al.* [4] explored the color attribute compression and verified that there is considerable redundancy in color representation. In [5], Rufeal *et al.* presented a hybrid framework in MP3DG-PCC based on PCL. To further improve the compression performance for color signals, Xu [6] proposed an adaptive scanning method based on rate-distortion cost to make color signals in each macro-block contain higher correlation.

In [7, 8], a voxelized point cloud (VPC) is utilized to represent the unstructured 3D point cloud. It quantizes the 3D point cloud into regular grids with dimensions $2^L \times 2^L \times 2^L$, where L is the partition parameter. A voxel is regarded occupied, if it contains at least one point, or else it is unoccupied. Then, the geometry information of voxels is represented by a set of triples. The other attribute value, such as color, for each voxel is denoted as the average value of all the points within the corresponding voxel. In [8], Ricardo $et\ al.$ compressed the color of point clouds via a hierarchical sub-band trans-

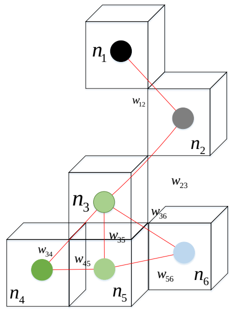

Fig. 1. Graph example for point cloud with 6 points.

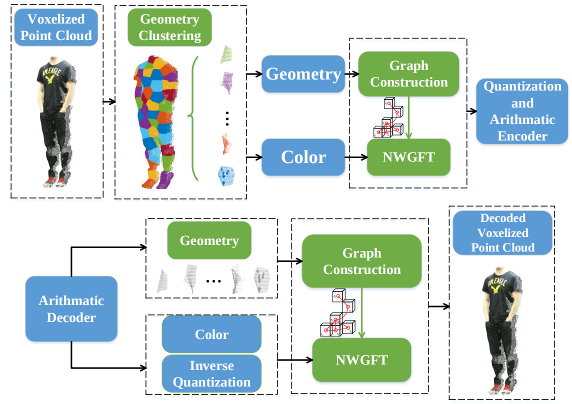

Fig. 2. The framework of the proposal.

form. Based on Graph Fourier Transform (GFT), an adaptive transform computed from a graph to reduce signal correlations, Zhang *et al.* [7] proposed an efficient point cloud compression approach, which compacts the energy of color to achieve higher compression ratio. In order to reduce the computation complexity in matrix decomposition, the original point cloud is usually partitioned into multiple sub-clouds, in each of which a graph is constructed based on point to point distance so as to design the GFT.

Although the GFT has achieved better performance compared with traditional transforms, *e.g.,* DCT, there remains two prominent problems. First, the current edge weight allocation for graphs is inefficient since it ignores the characteristics of neighboring points and cannot reflect the correlation among points efficiently. Second, the point cloud is divided into sub-clouds based on the regular space partition strategy, which may lead to too many isolated sub-clouds and may not exploit regional smoothness efficiently.

In order to address the aforementioned two problems, we propose adaptive geometric partition and NWGFT to further improve the color attribute compression. Compared with the uniform partition of space in [7], we propose to cluster the point cloud according to the point coordinate distribution, which makes each sub-cloud more correlated within. Then, we propose a novel edge weight allocation method for GFT by exploiting the similarity of two normal vectors in relative local space to further remove the correlation of each sub-cloud. Experimental results show that the proposed method significantly outperforms state-of-the-art transform based point cloud compression algorithms.

The remainder of the paper is organized as follows. Section 2 introduces the GFT. In Section 3, we elaborate on the proposed adaptive 3D space partition and NWGFT. Experimental results and conclusion are presented in Section 4 and Section 5, respectively.

2. GRAPH FOURIER TRANSFORM

Graph is able to efficiently represent relationships in the data, which is particularly suitable for irregular and high-dimensional data structures. A graph is defined as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{W})$, where $\boldsymbol{V} = \{v_0, v_1, \cdots, v_{N-1}\}$ denotes the set of nodes with cardinality $N \in N^*$ and $\boldsymbol{W}$ is the weighted adjacency matrix of the graph, in which $w_{i,j}$ refers to the weight allocated for the edge connecting nodes i and j. Accordingly, a graph signal is a function defined on the graph: $\boldsymbol{G} \rightarrow \boldsymbol{f}$, where $\boldsymbol{f} = [f(1), f(2), \cdots, f(N)]^T \in R^N$. For example, we define points in the point cloud as nodes in the graph, and the color attribute of each point as the corresponding graph signal. Fig. 1 shows a simple example for a graph composed of six nodes built within a point cloud. Then, the degree matrix is defined as a diagonal matrix $\boldsymbol{D}$, whose diagonal elements $d_i = \sum_j w_{i,j}$. The combinatorial Laplacian matrix is then defined as,

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}. \tag{1}$$

The Laplacian matrix is symmetric and positive semi-definite, which means it admits a complete set of orthonormal eigenvectors. The GFT basis Φ is then the eigenvector set of the Laplacian matrix. The GFT defined on a graph signal is thus defined as

$$\hat{\boldsymbol{f}} = \boldsymbol{\Phi^T} \boldsymbol{f}. \tag{2}$$

The inverse GFT follows as

$$\boldsymbol{f'} = \boldsymbol{\Phi} \hat{\boldsymbol{f}}. \tag{3}$$

GFT is a content-adaptive linear transform and has been shown to be very useful in compressing certain types of signals, e.g. mesh geometry [9], depth maps [10][11], and other images/videos [12, 13, 14].

3. THE PROPOSED ALGORITHM

We propose to compress attributes on 3D point clouds, such as color and normal, based on geometric clustering and N-WGFT. Without loss of generality, we focus on the widely utilized color attribute to show the efficiency of the proposed method. Specifically, as shown in Fig. 2, for a given point cloud, firstly, we organize it into a voxelized structure with

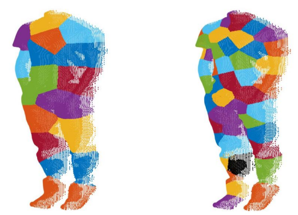

Fig. 3. Boy is partition into 16 clusters (left) and 48 clusters (right).

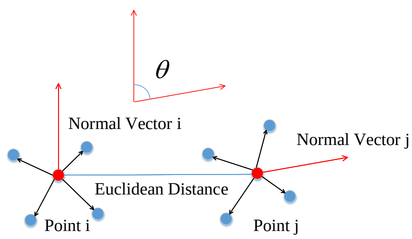

Fig. 4. Edge weight calculation.

a certain level L [15]. A voxel is occupied if it contains at least a point. Occupied voxels closely approximate points in the original cloud [16]. Secondly, The voxelized point cloud (VPC) is divided into K sub-clouds based on the geometry information via K-means. Thirdly, NWGFT is employed to transform the color information in each sub-cloud. Finally, the resulting transform coefficients are processed via uniform quantization and Arithmetic Encoder.

3.1. Geometry Based Clustering

It will cost huge computation resource and time for the decomposition of the Laplacian matrix. Therefore in [7], in order to decrease the computation complexity, sub-clouds are further divided into voxels. However, uniform spatial partition ignores the characteristics of the point cloud, so it would create too many isolated sub-clouds if the point cloud is not very dense, as pointed out in [7]. Also, discontinuous sub-clouds would be yielded.

We address the problem via K-means clustering [17]. The geometry information is selected as the feature for clustering. The VPC is divided into K clusters and each cluster contains n points of VPC on average, so we have

$$K = \frac{N}{n}. \tag{4}$$

In our experiments, n is set to 500, empirically. Moreover, as illustrated in Fig. 3, more clusters contribute to better classification, such as in the leg part of *Boy*. Note that, K-means clustering keeps the continuity of sub-clouds in each cluster.

3.2. Normal Weighted Graph Fourier Transform

After generating K clusters of sub-clouds, we construct graphs independently within each sub-cloud. While previous works usually define edge weights based on the similarity in distances between coordinates, we propose to compute edge weights from normals of points, which describes the geometric similarity more accurately. Specifically, we first set d as a radius for a point i. If the distance between point i and j is less than d, they are regarded as neighbors. Secondly, as shown in Fig. 4, in terms of neighboring points i and j, a local space is constructed via k nearest neighbors method for each of them. We then compute the normal vector of each local space by decomposing the dimension covariance matrix, which serves as a local feature. Finally, the edge weight $w_{i,j}$ between point i and j is defined as

$$w_{i,j} = e^{-\left(\frac{\sin \theta_{i,j}}{\sigma}\right)^2}, \tag{5}$$

where $\theta_{i,j}$ is the angle between two normal vectors on point i and j respectively, and σ is a weighting parameter. In our experiments, d^2, k and σ^2 are set to 300, 15 and 0.4, respectively. Note that, the edge weight defined in (5) is more robust, by combining the features of points and their neighborhoods via normals.

Next, we compute the graph Laplacian $\mathbf{L}$ from the edge weights as in (1), and perform the eigen-decomposition of $\mathbf{L}$ to acquire the eigenvector matrix Φ, which serves as the basis of the proposed NWGFT.

For each sub-cloud, We take the $\boldsymbol{Y}$ component as an example. First, we stack the color attribute $\boldsymbol{Y}$ of VPC into a $n \times 1$ column vector. Then $\boldsymbol{Y}$ is projected into the N-WGFT domain as $\boldsymbol{T} = \boldsymbol{\Phi^T Y}$. Next, $\boldsymbol{T}$ is quantized by $\boldsymbol{T_q} = round(\boldsymbol{T}/Q)$, in which Q denotes the quantization step. Finally, the quantization coefficients are encoded using Arithmetic Encoder and transmitted to the decoder. $\boldsymbol{U}$ and $\boldsymbol{V}$ components are encoded in the same way, separately.

4. EXPERIMENTAL RESULTS

We conducted experiments based on frames extracted from dynamic point clouds, including `Andrew`, `David`, `Phil`, `Ricardo` and `Sarah` representing half of human bodies used in MPEG standard [18], and Dimitris [19] and Boy[1] representing full human bodies. We further partition each point cloud into a $4096 \times 4096 \times 4096$ grid space [15], thus making the number of VPC almost the same as that of the original point cloud.

To verify the performance of the proposed algorithm, we compare with 3 state-of-the-art point cloud compression methods for color, *i.e.*, RAHT [8], DCT based compression and MP3DG-PCC[2]. Herein, RAHT is a wavelet-based

[1] http://www.kscan3d.com/gallery/

[2] http://wg11.sc29.org/svn/repos/MPEG-04/Part16-Animation_Framework_eXtension_(AFX)/trunk/3Dgraphics/

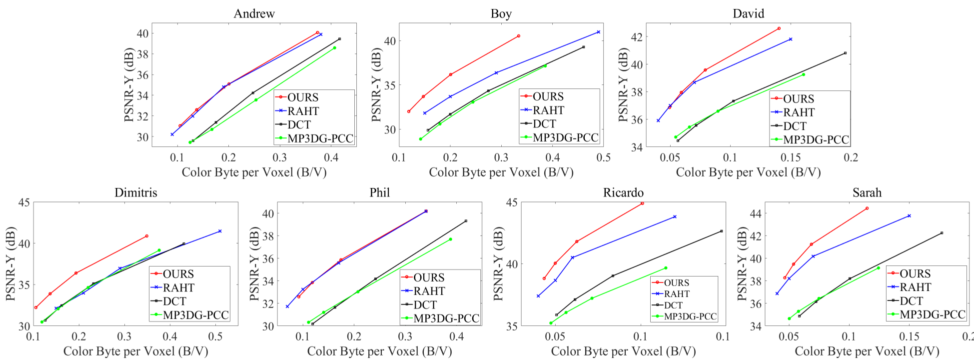

Fig. 5. RD curves for the proposed method, RAHT, DCT and MP3DG-PCC.

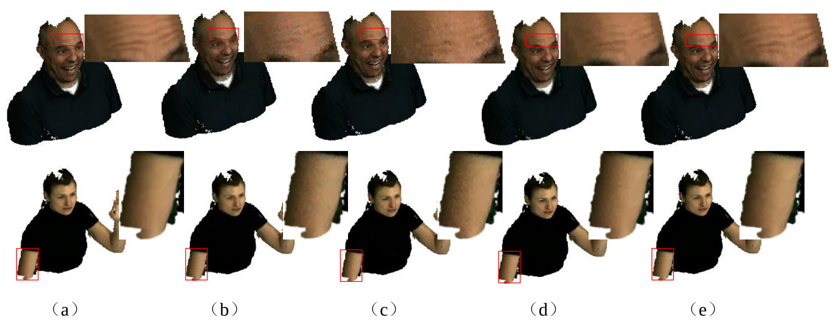

Fig. 6. Rendering results for point clouds compressed with the similar rate. (a) Original point cloud. (b) MP3DG-PCC. (c) DCT. (d) RAHT. (e) proposal.

method. We take the 1D DCT method as a baseline, since 1D DCT is a special case of GFT, as analyzed in [12]. The MP3DG-PCC is a widely adopted open-source point cloud compression API introduced by 3D Graphics (3DG) group of MPEG.

Table 1. Performance of the proposed scheme (compared with RAHT) at quantization step $\{10, 20, 30, 40\}$.

Point Cloud	**BD-BR (Y)**	**BD-BR (U)**	**BD-BR (V)**
Andrew	-1.72%	-1.09%	0.18%
Boy	-27.78%	-41.12%	-52.53%
David	-6.72%	-4.54%	-0.78%
Dimitris	-29.96%	-41.67%	-52.43%
Phil	-1.36%	0.99%	0.21%
Richado	-20.05%	-18.94%	-16.66%
Sarah	-20.50%	-22.17%	-17.04%
Average	-15.44%	-18.36%	-19.86%

Fig. 5 shows the RD curves for different point cloud compression methods. Specifically, we reduce bit rate by 15.44% compared with RAHT, 41.81% compared with DC-T, and 45.76% compared with MP3DG-PCC. The numbers are calculated using the BD-BR [20], which quantifies the difference between two RD curves. As RAHT has the best performance in the competing methods, we further list the bit rate saving over RAHT in Table 1.

Besides, we demonstrate the subjective results of the reconstructed point clouds under similar rates in Fig. 6. We can see that the proposed algorithm preserves more details in the data, and avoids artifacts in smooth regions.

5. CONCLUSION

We proposed adaptive geometric partition and NWGFT to compress the attributes of 3D point clouds. Firstly, we adopt clustering by the K-means method to reduce the computation complexity while making sub-clouds more uniform and compact. Then, we designed an adaptive edge weight allocation strategy by combining each point and its local normal features to make full use of the spatial correlation. Experimental results verified that our proposal reduced the correlation among points more efficiently, showing significant improvement over state-of-the-art point cloud compression methods.

6. REFERENCES

[1] C Tulvan, R Mekuria, and Z Li, "Use cases for point cloud compression (pcc)," in *ISO/IEC JTC1/SC29/WG11 (MPEG) output document N16331*, June 2016.

[2] Ruwen Schnabel and Reinhard Klein, "Octree-based point-cloud compression.," in *Eurographics Symposium on Point-Based Graphics*, 2006, pp. 111–120.

[3] Radu Bogdan Rusu and Steve Cousins, "3d is here: Point cloud library (pcl)," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1–4.

[4] Yan Huang, Jingliang Peng, C-C Jay Kuo, and M Gopi, "A generic scheme for progressive point cloud coding," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 440–453, 2008.

[5] Rufael Mekuria, Kees Blom, and P Cesar, "Design, implementation and evaluation of a point cloud codec for tele-immersive video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2016.

[6] Yiqun Xu, Shanshe Wang, Xinfeng Zhang, Shiqi Wang, Nan Zhang, Siwei Ma, and Wen Gao, "Rate-distortion optimized scan for point cloud color compression," in *IEEE Visual Communications and Image Processing (V-CIP)*, 2017.

[7] Cha Zhang, Dinei Florencio, and Charles Loop, "Point cloud attribute compression with graph transform," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 2066–2070.

[8] Ricardo L de Queiroz and Philip A Chou, "Compression of 3d point clouds using a region-adaptive hierarchical transform," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, 2016.

[9] Zachi Karni and Craig Gotsman, "Spectral compression of mesh geometry," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 279–286.

[10] Godwin Shen, W-S Kim, Sunil K Narang, Antonio Ortega, Jaejoon Lee, and Hocheon Wey, "Edge-adaptive transforms for efficient depth map coding," in *IEEE Picture Coding Symposium (PCS)*, 2010, pp. 566–569.

[11] Gene Cheung, Woo-Shik Kim, Antonio Ortega, Junichi Ishida, and Akira Kubota, "Depth map coding using graph based transform and transform domain sparsification," in *Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*. IEEE, 2011, pp. 1–6.

[12] Cha Zhang and Dinei Florêncio, "Analyzing the optimality of predictive transform coding using graph-based models," *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 106–109, 2013.

[13] Wei Hu, Gene Cheung, Antonio Ortega, and Oscar C Au, "Multiresolution graph fourier transform for compression of piecewise smooth images," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 419–433, 2015.

[14] Wei Hu, Gene Cheung, and Antonio Ortega, "Intra-prediction and generalized graph fourier transform for image coding," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1913–1917, 2015.

[15] Charles Loop, Cha Zhang, and Zhengyou Zhang, "Real-time high-resolution sparse voxelization with application to image-based modeling," in *Proceedings of the 5th High-Performance Graphics Conference*. ACM, 2013, pp. 73–79.

[16] Ricardo L de Queiroz and Philip A Chou, "Transform coding for point clouds using a gaussian process model," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3507–3517, 2017.

[17] M Everingham and A Zisserman, "Automated detection and identification of persons in video using a coarse 3-d head model and multiple texture maps," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 152, no. 6, pp. 902–910, 2005.

[18] Sergio Orts Escolano Charles Loop, Qin Cai and Philip A. Chou, "Microsoft voxelized upper bodies a voxelized point cloud dataset," in *ISO/IEC JTC1/SC29/WG11 ISO/IEC JTC1/SC29/WG1 input document m38673/M72012*, May. 2016.

[19] Alexandros Doumanoglou, Dimitrios S Alexiadis, Dimitrios Zarpalas, and Petros Daras, "Toward real-time and efficient compression of human time-varying meshes," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 24, no. 12, pp. 2099–2116, 2014.

[20] Gisle Bjøntegaard, "Improvement of BD-PSNR model," VCEG-AI11, July 2008.