Dynamic Point Cloud Geometry Compression via Patch-wise Polynomial Fitting

Yingzhan Xu, Wenjie Zhu, Yiling Xu
Cooperative MediaNet Innovation Center
Shanghai Jiao Tong University, China
Email: {xuyingzhan,rebecca28,yl.xu}@sjtu.edu.cn

Zhu Li
Dept of CSEE
University of Missouri, Kansas City, USA
Email: {zhuli}@umkc.edu

***Abstract*—With the boosting requirements of realistic 3D modeling for immersive applications, advent of the newly-developed 3D point cloud has attracted great attention. Frankly, immersive experience using high data volume affirms the importance of efficient compression. Inspired by the video-based point cloud compression (V-PCC), we propose a novel point cloud compression algorithm based on polynomial fitting of proper patches. Moreover, the original point cloud is segmented into various patches. We generated corresponding depth maps via projection of all the patches by focusing on geometry information. Instead of directly compressing the absolute values, we utilized proper polynomial functions to fit in each patch to obtain the differences. Finally, it is satisfying to note that the fitting function effectively represents the patch-wise geometry information. Moreover, new depth maps are obtained with extremely small and stable values, which are more suitable for video-based compression. Different patch-wise fitting parameters are preserved and coded using lossless compression through the open source PAQ project. The proposed approach achieves a noticeable improvement in the compression efficiency while maintaining point cloud quality.**

***Index Terms*— point cloud compression, V-PCC, geometry, polynomial fitting, patch generation**

I. INTRODUCTION

The advancement of scanning technology as well as the growing demand for immersive media have led to an increase in the importance of 3D data format in various applications such as machine vision [1], auto-navigation [2], and medical [3,4]. Realistically, the point cloud has succeeded in recording and describing three-dimensional objects and scenes, based on its newly-developed 3D media format that assists in recording the point geometry information related to attribute information.

A high precision of the point cloud offers a high data volume, and when point cloud information is transmitted and processed, the data efficiency usually does not perform well. Moreover, the irregular and scattered point cloud data increases the complexity of the processing algorithms and takes up a lot of computing space. Therefore, a proper compression algorithm is quite essential in the point cloud applications to overcome these problems.

Numerous studies have explored point cloud compression. Geometry and attribute information are the main properties of the point cloud, which usually coded separately. For attribute compression, the genetic algorithm based intra prediction is introduced [5]. Moreover, a global projection algorithm that maintains a correlation of the color attribute associated with the nearby points in the 3D space is also introduced [6] for the accuracy of the attribute. However, it is obvious that the geometry information is the basis for attribute rendering. The octree structure can be utilized to separate the point clouds and then entropy encode the corresponding leaf nodes to compress the geometry information [7-9]. Moreover, a binary tree structure can also be used to segregate the unorganized points into block structure and eliminate the redundancy of geometry information through residual coding [10-12]. Besides, we can cluster the points into a series of hierarchical point clusters and traverse each cluster from top to bottom. The residual between the traversed point and the top point will be encoded to perform the geometry compression [13,14]. Moreover, a global projection algorithm can convert the 3D data into 2D data and compress geometry information using 2D codecs [6]. This dimensionality data reduction can effectively improve the processing speed and efficiency.

In the last few years, the Moving Picture Experts Group (MPEG) organization has proposed a standardized compression test model for the dynamic point cloud. The main principle is to convert 3D point clouds to 2D video sequences and employ existing video coding algorithms, such as HEVC, for further compression [15]. In this test model, a point cloud is first divided into different patches, and they are projected onto 2D grid based on their surface normal. Then, certain 2D frame is formed via proper packing algorithm, and the process is performed for each point cloud in this dynamic sequence. Therefore, this video-based compression method shows a state-of-the-art outstanding performance.

In the present study, we propose a polynomial surface fitting algorithm to improve the performance of a V-PCC algorithm and achieve better compression results while maintaining point cloud quality. This paper is organized as follows. In Section II, the V-PCC algorithm is introduced. In Section III, we present details of polynomial fitting of depth maps and compression. In section IV, we demonstrate the experimental conditions and the results. Section V concludes the entire paper.

II. V-PCC

The principle behind the V-PCC method is to utilize existing video codecs to compress the geometry and the texture information of dynamic point cloud sequences. At first, the 3D object is divided into patches of different sizes mainly based on the normal vectors of the points that belong to these patches. Then these patches are projected in different directions according to the main normal vector of each patch to generate a padded depth map. After the projecting and padding processes, two video sequences that record the geometry and texture information from point cloud are generated and compressed using existing video codecs. Figure.1 provides an overview of the process used in converting the 3D object into 2D images.

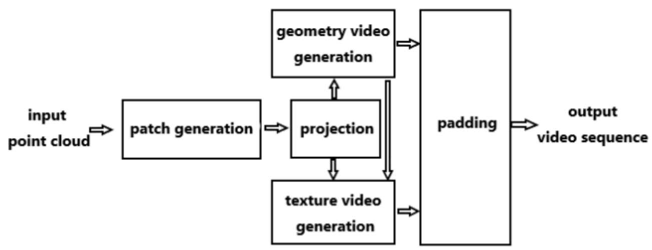

Fig.1 Convert Point Cloud into Video

In the next section we describe the patch generation module and the projection module in detail, and they are used in our proposed compression method.

A. *Patch Generation*

The main effect of patch generation is to reduce the amount of data involved in every process and to improve the efficiency of compression obtained by dividing the complete point cloud objects into smaller scale point clusters. This process mainly includes two steps: initial division and fine division

In initial division, the normal vector is calculated for each point [16]. A rough division is obtained, associating each point with one of the following six oriented planes. This is done by maximizing the dot products of the point normal vectors and the plane normal vectors. The aim is to ensure that each point is associated with the closest normal. More precisely, the planes are defined using the following normal vectors:

(1.0, 0.0, 0.0), (0.0, 1.0, 0.0), (0.0, 0.0, 1.0),

(-1.0, 0.0, 0.0), (0.0, -1.0, 0.0), (0.0, 0.0, -1.0)

In fine division, clustering of adjacent points and extraction of connection components are carried out, and we achieve this process using the proximity search of the kd-tree structure. Moreover, the connection component extraction is mainly used to process the segmentation of the points at the boundary.

B. *Projection*

The projection process aims at mapping the extracted patches onto a 2D grid for video coding. A simple packaging strategy is applied that repeatedly tries to insert the segmented segments in a grid of size W $\times$ H, where W and H are user-defined parameters that correspond to the length and width of the geometry/texture image to be encoded, respectively. The direction of projection is determined using the principal normal vector of the plane with which the segment is associated during the partitioning process. the effect of projection has been presented in figure. 2.

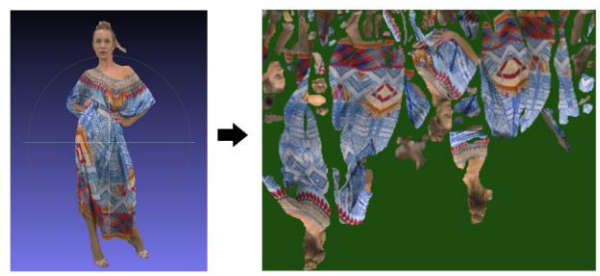

Fig.2 Projection Effect

Via segmentation and projection, the point cloud object is transformed into depth maps to achieve video generation, padding, and other processes. However, the depth values are directly obtained through a spatial coordinate projection of the point cloud. As the depth values are high, these are not convenient for video compression. In the next section, we introduce the method of applying polynomial fitting to reduce the data volume and optimize the depth map.

III. POLYNOMIAL FITTING

Surface fitting can generate the equation to describe the surface based on the position of point information. Moreover, surface fitting can replace the seemingly random distribution with a regular expression to optimize the information storage process. Besides, surface fitting can also be employed to obtain smooth samples in remote sensing scanning [18] and machine learning [19] where we decide to use 3D modeling. The depth map generated by the V-PCC is projected from the point cloud segment, and the depth values maintain good surface continuity and regular surface distribution. Based on these benefits, we can utilize surface fitting expression to accomplish the prediction and reduce the data volume by encoding residuals for the predicted and original values [17]. Moreover, we choose polynomial as the fitting expression because of its simple operation and effective fitting effect.

A. Polynomial Fitting

Polynomial fitting is a surface fitting method to represent each fitting polynomial using several parameters. Through least squares, we can fit these parameters by constantly inserting real point values in our assumed expressions so as to minimize the sum of the squares of the differences between the real depth values and the estimated depth values, For example, we assumed that variable Z has the following relationship with variables x and y:

$$Z = a^x + b^y$$

Using least squares, we can fit the values of a and b if we have enough values for x, y, and Z. In this way, we consider that the three-dimensional coordinates approximately satisfy the geometry relationship, and these are described by the polynomial that contains a and b parameters. Figure 3 shows the effect of the polynomial approach to estimate the spatial relationship among a series of points. The colored surface is the prediction result that assumes these points to be on the fitting surface.

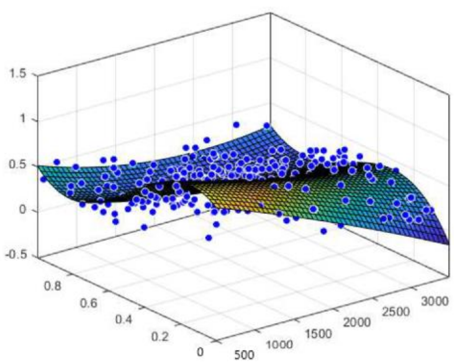

Fig.3 Polynomial Fitting Effect

B. Polynomial Fitting in V-PCC

As described in section II, patches of different sizes are generated after patch generation. Generally, these point cloud segments are quite regular and we can use polynomial fitting to predict the point coordinate information [20].

The number of polynomial terms depends on the complexity of the point cloud segment. Generally, a segment is small and a polynomial with nine terms is enough for prediction. In this model, we predict that the relationship of variables (x, y, z) approximately satisfies the following expression:

$$Z(x,y) = p_0 + p_1 \cdot x + p_2 \cdot y + p_3 \cdot x^2 + p_4 \cdot xy + p_5 \cdot y^2 + p_6 \cdot x^2 y + p_7 \cdot xy^2 + p_8 \cdot y^3$$

Where, Z(x, y) is the value of depth that was predicted at position (x, y). And {p0, p1, p2, p3, p4, p5, p6, p7, p8} are the parameters we need to fit.

After polynomial fitting, we can use these parameters to predict the depth value of each position. Next, we calculate the residuals between the predicted value and the original value. Further, we project residuals to generate residual video sequence, and the new video sequence is compressed instead of the original depth values to conspicuously reduce the data volume of depth map and improve the compression ratio. In addition, we compress the fitting coefficients by employing the open source PAQ project to encode these parameters owing to its lossless coding. Figure 4 shows the method used to generate the video sequence.

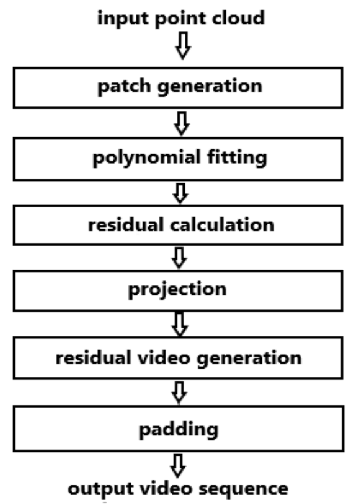

Fig.4 Video Generation Based on Polynomial

The residual values are decoded through the video decoder on the decoding side, and the predicted depth values are calculated by satisfying the expression described using the fitting parameters. Moreover, we reconstructed the original depth value by calculating the difference between the predicted depth value and the corresponding residual value. Through this process, we reconstruct the point cloud object using the calculated original depth values.

IV. EXPERIMENTS

As shown in Table 1, we perform geometry experiments on four dynamic point cloud sequences. Moreover, we test the compression performance on 32 frames of dynamic point clouds in each point cloud sequence.

Table1 Test Database

Sequence	Total number of points
8ivfb_loot_vox10 ('Loot')	25402281
8ivfb_redandblack_vox10('Red')	23266266
8ivfb_soldier_vox10('Soldier')	34409568
8ivfb_longdress_vox10('Longdress')	26698089

We used HM16.16 decoder as a video compression tool and set the code model to All-intra. Our operating environment was a Windows 10 64-bit operating system with a memory RAM of 8.0 GB and an Intel (R) Core (TM) i5-4590 CPU processor. The program was implemented using C++ and Matlab2015a programming languages.

Our proposed polynomial fitting algorithm is based on the V-PCC, so we compare the compression effect of the V-PCC test model and the proposed method on the same data. The rate-distortion curves are shown in figures 5-8, and we use the bit per input point (bpip) after compression to represent the bit rate cost in compression. The peak signal-to-noise ratio (PSNR) is calculated based on the widely-applied point-to-point distortion in MPEG. The geometry PSNR is computed as:

$$PSNR = 10\log_{10}\frac{(3p^2)}{MSE}$$

Where p is the peak constant value defined for each reference point cloud, and MSE is the mean squared point-to-point error.

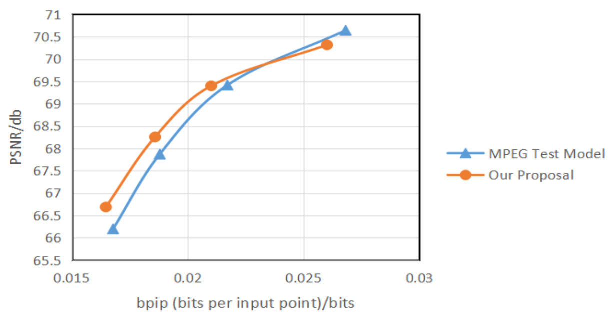

Fig.5 Geometry R-D Curve for ‘Longdress’

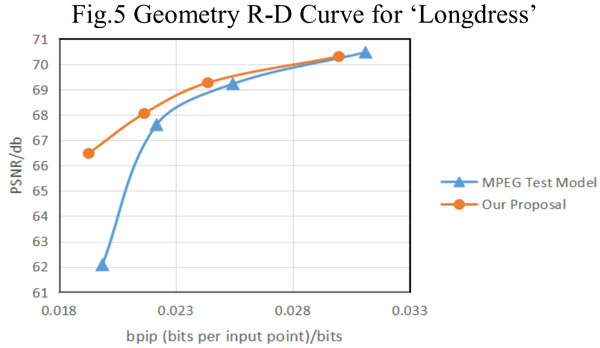

Fig.6 Geometry R-D Curve for ‘Soldier’

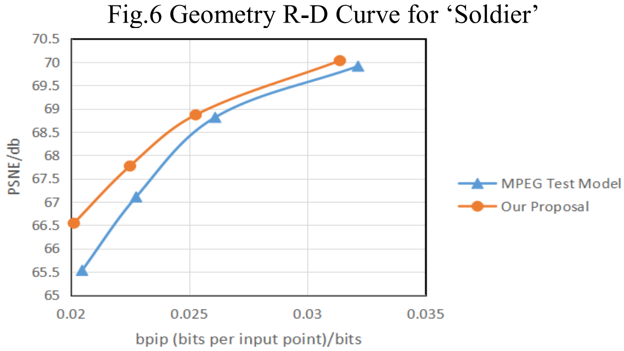

Fig.7 Geometry R-D Curve for ‘Red’

As shown in the rate-distortion curves (R-D curves), the proposed algorithm improves the performance of a V-PCC, reduces the bit cost, and maintains the quality. When the bpip is low, the bpip of our proposal performs better than

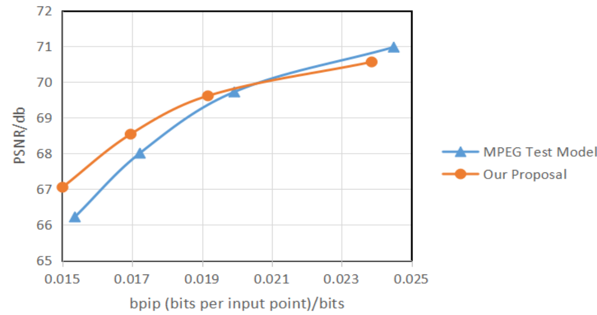

Fig.8 Geometry R-D Curve for ‘Loot’

that of MPEG Test Model with the same PSNR. An improvement is achieved by decreasing the depth values by polynomial fitting. Moreover, our method’s PSNR is little lower when the test is on high bpip.

When the compression is close to distortion-free, the PSNR will be more sensitive to the errors introduced by the fit. As a result, the effect of PSNR is more pronounced than the reducing of bpip. This is shown by the R-D curve. With the test of the database “reaandblack_vox10”, the optimization effect can also be maintained when bpip is high.

We chose a frame of the database “loot_vox10” as a test sample. In Fig. 6, we demonstrated the visual effects of our proposal as compared with the test model, which was a newly introduced V-PCC at the 123rd MPEG meeting.

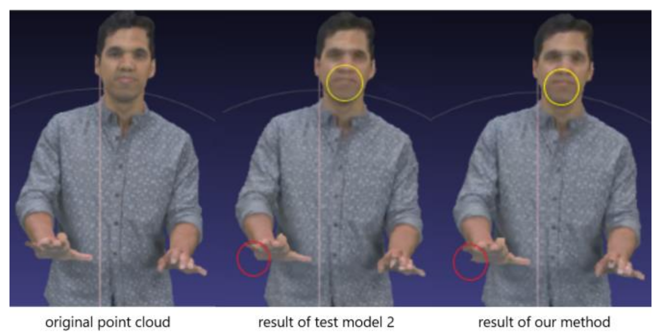

Fig.9 Result of TMC2 and our proposal

From the results of our proposal, we obtained a better performance that maintains the geometry and texture information in the yellow circle. And for some positions, some distortion arose in the red circle. The result shows that we can render and code attributes even better in some locations.

V. Conclusions

In this paper, we propose a dynamic point cloud compression method that combines V-PCC and patch-wise polynomial fitting to carry out geometry compression. Since the surface characteristics of the point cloud segment are obvious, the surface fitting can be used to improve compression efficiency and maintain information quality. Simultaneously, it provides a potential application value in point cloud attribute compression.

REFERENCES

[1] Zhao W , Zhao C , Wen Y , et al. An Adaptive Corner Extraction Method of Point Cloud for Machine Vision Measuring System[C]. International Conference on Machine Vision & Human-machine Interface. IEEE Computer Society, 2010.

[2] Birendra Kathariya, Li Li, Zhu Li, Jose R. Alvarez. Lossless dynamic point cloud geometry compression with inter compensation and traveling salesman prediction[C]. 2018 Data Compression Conference. IEEE, 2018: 2375-0359.

[3] Placitelli A P, Gallo L. 3D point cloud sensors for low-cost medical in-situ visualization[C]. IEEE International Conference on Bioinformatics & Biomedicine Workshops. IEEE, 2011.

[4] Leonardo Ishida Abe, Yuma Iwao, Toshiyuki Gotoh, Seiichiro Kagei, Rogerio Yugo Takimoto, Marcos de Sales Guerra Tsuzuki, Tae Iwasawa. High-speed point cloud matching algorithm for medical volume images using 3D Voronoi diagram[C]. 2014 7th International Conference on Biomedical Engineering and Informatics. IEEE, 2014.

[5] Ke Zhang, Wenjie Zhu, Yiling Xu, Ning Liu. Point Cloud Attribute Compression via Clustering and Intra Prediction[C]. 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB). IEEE, 2018: 2155-5052.

[6] Lanyi He, Wenjie Zhu, Ke Zhang, Yiling Xu. View-Dependent Streaming of Dynamic Point Cloud over Hybrid Networks[C]. Pacific Rim Conference on Multimedia 2018.

[7] Schnabel R, Klein R. Octree-based Point-Cloud Compression[C]. Eurographics. Eurographics Association, 2006.

[8] Hornung A, Wurm K M, Bennewitz M, et al. OctoMap: an efficient probabilistic 3D mapping framework based on octrees[J]. Autonomous Robots, 2013, 34(3):189-206.

[9] Diogo C. Garcia, Ricardo L. de Queiroz. Context-based octree coding for point-cloud video[C]. 2017 IEEE International Conference on Image Processing (ICIP). IEEE. 2017: 2381-8549.

[10] Wenjie Zhu, Yiling Xu, Li Li, Zhu Li. Lossless Point Cloud Geometry Compression via Binary Tree Partition and Intra Prediction[C]. 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP). IEEE, 2017: 2473-3628.

[11] Birendra Kathariya, Ainala Karthik, Zhu Li, Rajan Joshi. Embedded binary tree for dynamic point cloud geometry compression with graph signal resampling and prediction[C]. 2017 IEEE Visual Communications and Image Processing (VCIP). IEEE, 2017.

[12] Kathariya B, Li L, Li Z, et al. Scalable Point Cloud Geometry Coding with Binary Tree Embedded Quadtree[C]. 2018 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2018.

[13] Fan Y, Huang Y, Peng J. Point cloud compression based on hierarchical point clustering[C]. 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. IEEE, 2014.

[14] Ke Zhang, Wenjie Zhu, Yiling Xu. Hierarchical Segmentation Based Point Cloud Attribute Compression[C]. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018). IEEE, 2018: 2379-190X.

[15] Sebastian Schwarz, Marius Preda, Vittorio Baroncini,. Emerging MPEG Standards for Point Cloud Compression[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, DOI 10.1109/JETCAS, 2018.

[16] Hoppe H, Derose T, Duchamp T, et al. Surface reconstruction from unorganized points[J]. ACM SIGGRAPH Computer Graphics, 1992, 26(2):71-78.

[17] Fusaomi Nagata1, Norifumi Horie1, Keigo Watanabe, Maki K. Habib. Curved Surface Fitting Using a Raster-Scanning Window for Smoothing PCD[C]. 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE). IEEE, 2017.

[18] Li X, Xie W, Wang L, et al. Ship detection based on surface fitting modeling for large range background of ocean images[C]. IEEE International Conference on Signal Processing. IEEE, 2017.

[19] Pagnutti G, Minto L, Zanuttigh P. Segmentation and Semantic Labeling of RGBD Data with Convolutional Neural Networks and Surface Fitting[J]. IET Computer Vision, 2017, 11(8):633-642.

[20] Peternell M. Developable surface fitting to point clouds[J]. Computer Aided Geometric Design, 2004, 21(8):785-803.