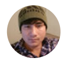
An improved enhancement layer for octree based point cloud compression with plane projection approximation

Conference Paper · September 2016

DOI: 10.1117/12.2237753

6 authors, including:

Rufael Mekuria
Centrum Wiskunde & Informatica



Birendra Kathariya
University of Missouri - Kansas City

An improved enhancement layer for octree based point cloud compression with plane projection approximation

Karthik Ainala, Univ. of Missouri-Kansas City (United States); Rufael N. Mekuria, Unified Streaming B.V. (Netherlands); Birendra Khathariya, Zhu Li, Univ. of Missouri-Kansas City (United States); Ye-Kui Wang, Rajan Joshi, Qualcomm Inc. (United States)

ABSTRACT

Recent advances in point cloud capture and applications in VR/AR sparked new interests in the point cloud data compression. Point Clouds are often organized and compressed with octree based structures. The octree subdivision sequence is often serialized in a sequence of bytes that are subsequently entropy encoded using range coding, arithmetic coding or other methods. Such octree based algorithms are efficient only up to a certain level of detail as they have an exponential run-time in the number of subdivision levels. In addition, the compression efficiency diminishes when the number of subdivision levels increases. Therefore, in this work we present an alternative enhancement layer to the coarse octree coded point cloud. In this case, the base layer of the point cloud is coded in known octree based fashion, but the higher level of details are coded in a different way in an enhancement layer bit-stream. The enhancement layer coding method takes the distribution of the points into account and projects points to geometric primitives, i.e. planes. It then stores residuals and applies entropy encoding with a learning based technique. The plane projection method is used for both geometry compression and color attribute compression. For color coding the method is used to enable efficient raster scanning of the color attributes on the plane to map them to an image grid. Results show that both improved compression performance and faster run-times are achieved for geometry and color attribute compression in point clouds.

Keywords: principal component analysis, geometry compression, enhancement layer, octree.

1. INTRODUCTION

3D Point clouds are a commonly used representation for 3D object scans and 3D visual data. Scanned point clouds typically contain thousands to millions of points, requiring large storage space and/or transmission bandwidth. For real-time and bandwidth constrained point cloud based applications, compression of this data is crucial. Point Cloud data typically contain geometry and color attribute information, so compression should target both. For geometry, coding methods based on octree composition have typically been proposed such as [1], while for color and attribute coding methods complementary to octree such as graph transform have been proposed [2].

These methods have been compared to different methods for parameters like compression rate, distortion and computational load etc. and have shown good results. However, due to the recent progress in augmented reality, immersive 3D data capture and rendering etc. the need for efficient real-time compression of dense 3D point cloud data is becoming more urgent. In such applications, the point cloud data frames are often captured at high frame rates (such as like video frames) and are much denser. For real-time and efficient compression of such dense 3D Point cloud data octree based is not always a good approach. In octree based methods run-time of the encoding method is related exponentially to the number of subdivisions, resulting in heavy computation for very high detail/dense octrees. Further, the serialization based compression becomes less efficient for the higher levels of detail. To address both the computation and compression efficiency problem of octree based point cloud compression, we propose an alternative enhancement layer for octree based point cloud compression. This enhancement layer works beyond a fixed level of the octree (up to which the compression efficiency and run-time are still good) and provides a method to code the point data beyond this level. It is based on a plane projection approximation and coding of residuals. The plane approximation allows both more efficient coding of geometry coordinates by storing 2D coordinates and residuals, and color attribute data, by raster scanning of the attributes after rotating the coordinates with PCA and then code the geometry and color attributes by raster scan the 2D plane. This approach better preserves affinity of geometry and color attributes for subsequent compression pipeline to be efficient..

Starting from the octree based MPEG reference software for point cloud compression, we have developed this enhancement layer as an improvement to this coding framework. We have selected this codec which operates in real time on commercial hardware as our reference software as it is open source and in use for the development of a point cloud standard in MPEG. The codec is available at https://github.com/RufaelDev/pcc-mp3dg and will soon be updated with code to perform the plane projection approximation as presented in this paper (we are working on integrating this method with this software framework).

To summarize, in this work, we present a coding method that can be used as an enhancement layer in octree based point cloud compression codecs, making octree coding more suitable for real-time encoding with very dense point clouds such as typically acquired using 3D scanning. We present both the use of the method which is based on plane projection and includes techniques for exploiting this for geometry and color attribute coding. The method is well aligned with the current MPEG Point cloud compression software and proposed as an improvement to this framework.

The rest of the paper is organized as follows, in section 2 we present the MPEG Point cloud compression framework based on octree composition and other techniques. This illustrates the typical current approach for coding point clouds and the underlying framework for point cloud compression that we will embed our proposed method into. In section 3 we present the method for coding geometry data using the plane projection approach, while in section 4 we present how the method can be used to code color attribute data. In section 5 we present the results, comparing with existing coding methods and in section 6 we conclude the paper.

2. MPEG POINT CLOUD COMPRESSION (PCC)

The architecture of the MPEG Point Cloud Compression combines features from the octree based 3D point cloud codecs [4] and [1] which are based on occupancy codes and surface approximations respectively. Further it includes techniques known from video coding such as inter frame predictive coding. The complete block diagram of this architecture is shown in figure 1 taken from the reference [3], this reference explains the full codec architecture including inter and intra coding. We are extending the intra coding part implemented in this codec. The intra coding of the input point cloud frame is done by modules 1, 2 and 3 in Fig 1. In our proposed extension, the bounding box computation for starting the octree in (1), the octree composition in (2) stay the same, but an extra module is added in box (3) for coding the enhancement layer based on plane projection triangulation. Alternatively, the proposed enhancement layer can be incorporated in a separate functional block.
The 3 steps involved in encoding the intra frame are Bounding Box normalization and filtering, Octree composition and occupancy code entropy coding. We have implemented our model based on these three steps for our Plane projection approximation and learning based compression of point cloud data geometry. These three modules are explained as follows:

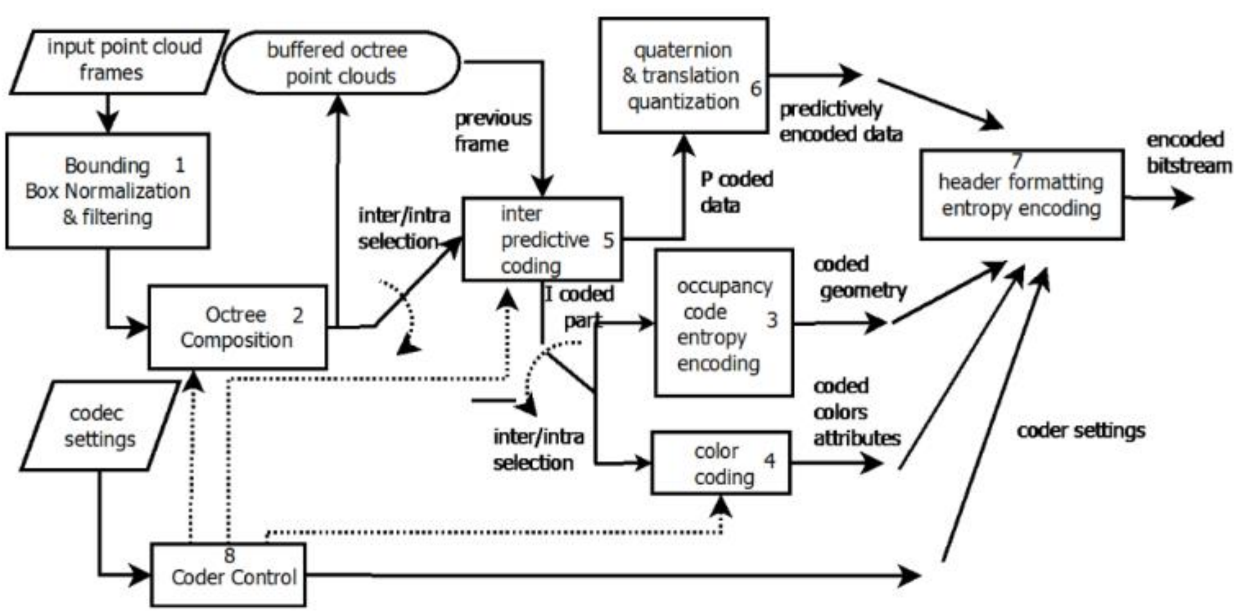


Fig 1 Schematic of point cloud compression codec [3]

i) Bounding Box normalization and filtering: The bounding box of a point cloud frame is computed as box with fixed lower and upper corners. The bounding box changes from frame to frame so the correspondence between frames is lost which is not favorable for inter-prediction. In our case we are working on single point cloud frame so we don't process the bounding box predictive algorithm for inter prediction. We proceed with the normalization and outlier removal procedure implemented to the point cloud data. The outliers are the erroneous points in the point cloud which are introduced during 3D scanning with multiple cameras. These outliers are removed using radius removal filter. The point cloud data is also normalized and now the coordinates are between 0 and 1.

ii) Octree Composition: An octree is a tree data structure suitable for sparse 3D data where each branch node represents a cuboid bounding volume in space. Every branch has up to eight children i.e. one for each sub-octant of the parent cell. These 8 child cells have occupied cells which have at least one point coordinate in its cell, and empty cells which do not have any. Out of these eight cells only the occupied cells are further divided in to sub branches. This happens recursively and number of levels is predefined in the configuration file. For the proposed approach the number of octree levels is limited, and our enhancement layer module will work on the coarse octree and original point cloud to create an enhancement layer bit stream for coding level of details beyond the coarse coded octree.

iii) Occupancy code entropy coding: As mentioned earlier about the approach in [4] and [1] which are computationally expensive the pcc codec follows a modified approach as presented in [9] which is based on carry less byte based range coder. The proposed enhancement layer can also use this entropy coding mode, but alternatively can work with a PAQ learning based entropy encoder which improves the compression performance.

This paper presents the following improvements over the MPEG Point Cloud Compression software: In the MPEG PCC codec the level of octree division (Levels of Detail) are predefined irrespective of the surface geometry of the point cloud data. In our method we too have a fixed Level of Detail with the final layer represented as an enhancement layer. The occupied octree cells of the enhancement layer are divided to further levels based on the flatness threshold of the cell. So if the points in the octree cell are not flat enough to project on a plane they are decomposed to next level. The octree cells which are flat enough to clear the threshold become the leaf nodes. These leaf nodes are projected to different plane by applying PCA. In this way the flat or plane areas can be encoded more effectively after projection further avoiding multiple octree divisions. The residuals of the new projected PCA data of each leaf node is encoded using a learning based entropy encoder. By setting the flatness threshold it is also possible to always code the geometry and attributes in the enhancement layer using plane projection approximation.

3. PLANE PROJECTION APPROXIMATION (PPA) BASED GEOMETRY COMPRESSION

Recent advances in Grapsh Signal Processing (GSP) has found several successful applications of GSP to signal compression on non-uniformly sampled signals, examples are [15][2] for compression of SIFT features, color attributes. One intuition from GSP approach is that an adaptive transform is employed to better represent local data segmentation and have local filtering that achieve better compression efficiency. In this work we apply the same principal by achieve local data clustering and adaptive transform by identifing flat voxel regions and then apply local coordinates transform to have a compact 2D representation for better prediction and compression.

Due to the nature of the point cloud capture process, many local patches are more or less exhibiting flat characteristics that indicates an alternative coding scheme than Octree decomposition can be more efficient. In this work we introduce a Plane Projection Approximation (PPA) coding mode, that project the voxels onto a 2-D plane and coding the geometry and color attributes via raster scann on the plane. This involves first the flatness test. Let $\{P_k\} \in R^3$ be a set of voxels belong to a certain Octree node, we compute the geometry covariance,

$$S = E\{(P_k - \bar{P})^T(P_k - \bar{P})\} \tag{1}$$

Where the mean of the geometry is given by the centroids of the voxels, $\bar{P}$. Then we compute the Eigen values of the covariance matrix S, as $\{\lambda_1, \lambda_2, \lambda_3\}$, sorted by the Eigen value. Then the flatness measurement of the voxels in this node is computed as the Eigen value ratio,

$$\theta = \frac{min\{\lambda_1,\lambda_2,\lambda_3\}}{\lambda_1+\lambda_2+\lambda_3} \quad (2)$$

The distribution of the flatness criteria in Eq. 2) is illustrated in the Fig. 2 below, for the “ski” sequence [12] Octree nodes decompostion upto depth of 6. The sorted θ is illustrated in green.

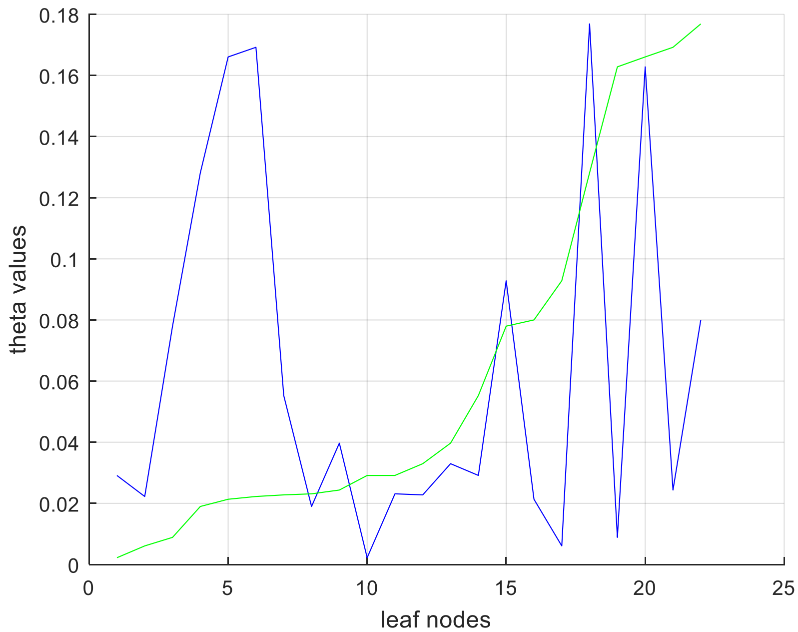


Figure 2. Flatness criteria distribution.

As indicated, there are quite a portion of Octree nodes consists of voxels more or less lying on a surface, therefore we introduce the PPA coding mode as follows, compute a 3x3 rotation matrix R via PCA, and rotate the [x, y, z] coordinates of voxels $\{P_k\}$ to [u, v, w], in descending order of variances, then we compute the raster scan order by sorting the voxels by,

$$indx = \text{sort}(u*max\_u + v) \quad (3)$$

This in effect gives us a raster scan of voxels after projection to a 2-D plane. The coding of the geometry is therefore achieved by differentially coding the [u, v, w] coordinates, and then applying a proper quantization scheme to match the Octree range coding PSNR quality, and then generate bitstream with a self adaptive entropy coding scheme called PAQ[10].

PAQ is a shallow neural network like context modeling coupled with Arithemetic coding to achieve high efficiency lossy compression. It is an improvement over PPM (Prediction by Partial Matching) by having many different models, and many flexible contexts. PAQ uses arithmetic coding similar to PPM but unlike arithmetic coding that uses single prediction at a time, PAQ has different models connected to different contexts that outputs many predictions. Arithmetic needs one prediction at a time, so a context mixer is used to combine all the predictions into one single prediction. Neural network algorithms are used for context mixing.

4. PLANE PROJECTION APPROXIMATION BASED COLOR COMPRESSION

As for efficient point cloud compression, both geometry and color attribute compression are important, therefore we introduce here a method for color coding based on the same flatness criterion as in section 3. The method is based on a principle component analysis (PCA) of the points in the flat voxel, followed by a raster scan and mapping to an image coder.

Principle component analysis (PCA) is a statistical procedure that uses orthogonal transformations to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principle components [6]. The main use of PCA is dimensionality reduction. The idea behind this is that any data point with multi-features or multi-dimensional may have two or more features that are correlated. If we can search a direction in same or another dimension where all the correlated features (or variables) vary similarly then those correlated features can be represented by a single feature in that direction. Thus we can reduce the dimension by rotating the original dimension optimally in that direction where the variance is maximum.

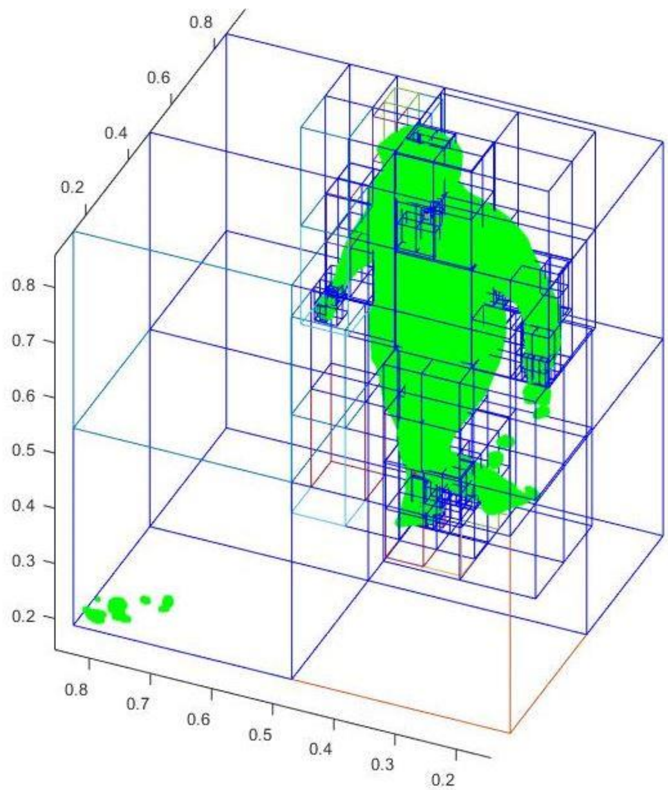


Fig. 2. Octree Decomposition with threshold 0.05

For point cloud data, every point in space is represented by three variables which are x, y, z coordinates. If there are N points in a leaf node then $X_n = \{x_n, y_n, z_n\}$ where n = 1,2,…….,N. So, $X = (X_1, X_2 \ldots X_N)^T$. Now, covariance of matrix X is given as:

$$C = X * X^T \tag{4}$$

Consider the eigenvalue decomposition of C:

$$C = \Phi \Lambda \Phi^{-1} \tag{5}$$

where $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_N)$ is a diagonal matrix containing C's Eigen values λ_n such that $\lambda_1 > \lambda_2 \ldots > \lambda_N$. Φ is the eigen vector matrix with first column as first principle component which is associated with λ_1 and so on.

After Eigen value decomposition of the point data of a leaf node, we get three Eigen values and 3x3 Eigen vector. We choose only two Eigen vectors associated with two largest Eigen values. The points of the leaf node are then projected into PCA subspace (space with reduced dimension) for optimal rotation as follows:

$$Y = \Phi^T * X \tag{6}$$

Y now has two variables u and v instead of x, y, z along with corresponding RGB color information. Figure 3 is an example of 2D projection of 3D points cluster from one of the leaf nodes. If there are M occupied leaf nodes, then PCA will be applied to all occupied M nodes separately.

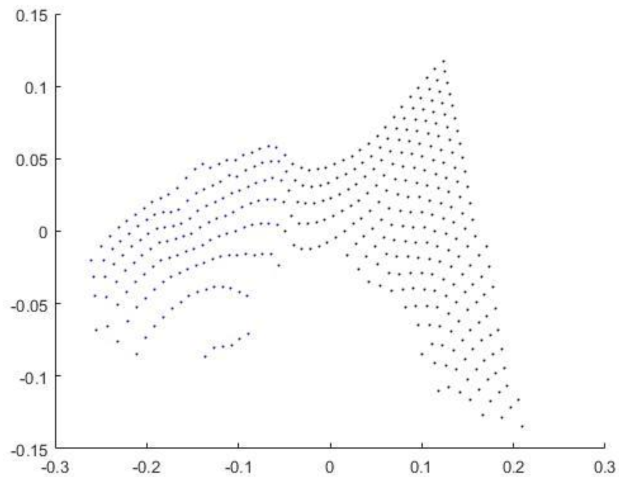


Fig. 3. Optimally projected data points from one of the leaf node

Finally, color scanning is performed. However, from figure 3 we can see that the points are scattered all over the plane with few coordinates without corresponding point. Raster scanning which is performed on 2D image is not going to work in this case. To eliminate ambiguity in the priority for the scanning we first sort the points based on the value of first principle component. This sorting of points also preserves the bond between geometry and color attribute. As, images are formed of points from each leaf-nodes the direct bond between geometry and color is lost. However, there is indirect bond between them as geometry in sorted and color is scanned in the same sorted order. To sort points, we scale 'u' value with scaling factor 'α' and add the second principle component value to it as follows:

$$S = \alpha * u + v \qquad (5)$$

In this paper, we set α to 10 and sort the projected points using values given by eq. (5). This sorted vector of u, v and color attribute is reshaped to form an image. Width is set to a certain value whereas length will vary with the number of points in each leaf node. While reshaping points into images, the last row might not fill to its full width and is padded with zero to its width. After reshaping, M images will get formed corresponding to each occupied leaf node. These images are stacked in the order given by Depth first scan indexing to form one large image. The last step is to encode the final image with some image compression scheme such JPEG compression (which was used in our implementation).

5. RESULTS

We have implemented our method by reusing the code from the point cloud compression library and the MPEG Point cloud compression software [3]. The noise free normalized point cloud data (.pcd file) is acquired by using a module from the pcc library and implemented our logic. We have applied our encoding model on a pcd data which has 45617 points. We acquired the point cloud data file from (http://vcl.iti.gr/reconstruction/) which is taken as the reference source for the point cloud codec. The θ value we assumed is 0.05 and the octree division for the enhancement layer stops at level 6. These 6 levels of octree divisions are shown in the Figure 4 and Figure 5.

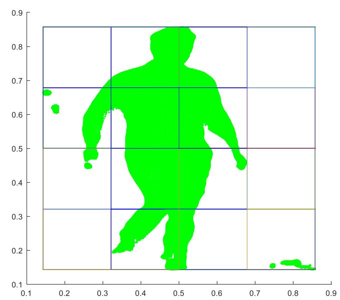


Fig 4

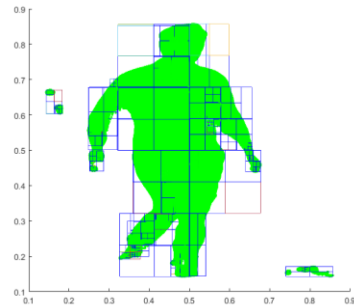


Fig 5

The 6 levels of octree divisions resulted 22 occupied octree cells in the enhancement layer. Our PPA method is applied to the enhancement layer and the octree division(level of detail) is increased by 4. So now the level of detail is 10 and the total number of leaf nodes (occupied octree cells) are 459. The final octree division node distribution is shown in Fig 5.

TABLE 1.COMPRESSION RESULTS FOR PREDICTIVE CODING GEOMETRY

	Compressed output(bytes)	PSNR(dB)
PPA (plane projection approach)	49305	73
PCC (point cloud codec)	86893	77

Next the color compression scheme is tested. Each point of Point Cloud Data has color information in RGB format. So, for each point 3x8 = 24 bits is required to represent color information. Tables below display the no. bits required to represent color information for each point after compression. Experimentation is performed on two different datasets (Human Skiing [12] and Human Motion) and results are tabulated in two tables, one for each dataset. The table shows compression in bits/points for different JPEG Quality level and for different flatness (θ) of voxels. Also, for each flatness, the total number of occupied leaf-nodes is also shown in the table. From Table II and Table III, we can observe that with the decrease in the flatness threshold value, the number of occupied leaf-nodes increases as octree decomposition level goes deeper and deeper. The number of bits required to represent the color information for each point gets lower with the decrease in quality of JPEG compression and with increase in the flatness threshold value which is as expected. While comparing with compression achieved by encoder built in Point Cloud Library (PCL) [11] which has compression around 8.4 bits/voxel for 38 dB luminance reconstruction error, the compression achieved by this method is comparatively high.

TABLE II COMPRESSION RESULT FOR "HUMAN SKIING" DATASET COLOR CODING

θ	Bits/Point Attribute (Human Skiing)					
	JPEG Quality					No. of occupied leaf-node
	10	40	60	80	100	
0.01	2.4493	4.5335	5.6132	7.7658	26.6181	2926
0.05	1.1315	2.5931	3.3415	4.7434	15.6613	301
0.1	1.1739	2.6049	3.3079	4.5824	14.6262	88
0.12	1.128	2.5913	3.2653	4.4887	14.243	48

TABLE III COMPRESSION RESULT FOR "HUMAN MOTION" DATASET COLOR CODING

θ	Bits/Point Attribute (Human Motion)					
	JPEG Quality					No. of occupied leaf-node
	10	40	60	80	100	
0.01	1.0076	1.7334	2.0869	2.6481	5.9677	694
0.05	0.67139	1.2765	1.5548	2.0253	4.8646	123
0.1	0.6333	1.2468	1.5292	2.0138	5.2314	38
0.12	0.58435	1.1798	1.4545	1.9341	5.162	8

IV. CONCLUSION AND FUTURE WORK

This work presented a coding method based on plane projection that can serve as an enhancement layer for octree point cloud compression. The plane projection is used to develop a method for coding both color attributes and geometry of the more fine grained details of the point cloud. The initial results presented in this paper look promising and will be tested more rigorously in future work. Further, there is a need to improve the predictive coding by implementing the enhancement layer for inter-predictive compression. The PAQ Neural network can be retrained to have further gains in the compression levels. The PPA approach can be implemented in the MPEG Point Cloud Compression code so that the comparisions can be made more effectively to other proposed methods. The work is of particular interest for recent work in VR/AR [13][14]

6. REFERENCES

[1] R. Schnabel, and R. Klein , "Octree-based Point-Cloud Compression," in *SPBG*, 2006, pp. 111-120.

[2] D. Florencio, C. Loop C. Zhang, "Point Cloud Attribute Coding with the Graph transform," in *IEEE International Conference on Image Processing*, Paris, October 2014.

[3] "Design, Implementation and Evaluation of a Point Cloud Codec for Tele-Immersive Video" R Mekuria, K Blom, P Cesar IEEE Transactions on Circuits and Systems for Video Technology sept. 2016

[4] Y. Huang, J. Peng, C. Kuo, C. 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] O. Devillers and P. Gandoin, "Geometric compression for interactive transmission," in Proceedings of the International Conference on Information Visualization. London, England, UK: IEEE, 2000, pp. 319–326

[6] J. Peng and C. Kuo, "Octree-based progressive geometry encoder," in In Internet Multimedia Management Systems IV. Edited by Smith, John R.; Panchanathan, Sethuraman; Zhang, Tong. Proceedings of the SPIE. Citeseer, 2003, pp. 301–311.

[7] Y. Huang, J. Peng, C. Kuo, and M. Gopi, "Octree-based progressive geometry coding of point clouds," in Proceeding of the Eurographics Symposium on Point-Based Graphics, Zurich, Switzerland, 2006, pp. 103–110.

[8] S. Fleishman, D. Cohen-Or, M. Alexa, and C. Silva, "Progressive point set surfaces," ACM Transactions on Graphics (TOG), vol. 22, no. 4, pp. 997–1011, 2003

[9] J. Kammerl. , N. Blodow, R.B. Rusu, S. Gedikli, and E. Steinbach M Beetz, "Real-time compression of point cloud streams," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on , pp.778,785, 14-18 May*, 2012.

[10] M. Mahoney, Adaptive Weighing of Context Models for Lossless Data Compression, Florida Tech. Technical Report CS-2005-16, 2005.

[11] R. Rusu 3D is here: Point Cloud Library (PCL) IEEE International Conference on Robotics and Automation (ICRA) Shanghai 2011, pp. 1-4

[12] D. Alexiadis, D. Zarpalas, P. Daras, “Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras“, IEEE Transactions on Multimedia, VOL. 15, NO. 2, pp. 339-358, February 2013 http://vcl.iti.gr/reconstruction/

[13] Microsoft Holoportation, https://www.microsoft.com/en-us/research/project/holoportation-3/

[14] Replay Technology: http://replaytechnology.com

[15] Xin Xin, Zhu Li, Aggelos K. Katsaggelos “Laplacian embedding and key points topology verification for large scale mobile visual identification” *Sig. Proc.: Image Comm*. 28(4): 323-333 (2013)