

POINT CLOUD COMPRESSION AND LOW
LATENCY STREAMING

A Thesis

in

Electrical Engineering

Presented to the Faculty of the University
of Missouri-Kansas City in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

by

KARTHIK AINALA

Bachelor of Technology (B.Tech) in Electronics and Communication Engineering,
Amrita University, India 2012

Kansas City, Missouri

2017

© 2017

KARTHIK AINALA

ALL RIGHTS RESERVED

POINT CLOUD COMPRESSION AND LOW LATENCY STREAMING

Karthik Ainala, Candidate for the Master of Science Degree

University of Missouri–Kansas City, 2017

ABSTRACT

With the commoditization of the 3D depth sensors, we can now very easily model real objects and scenes into digital domain which then can be used for variety of application in gaming, animation, virtual reality, immersive communication etc. Modern sensors are capable of capturing objects with very high detail and scene of large area and thus might include millions of points. These point data usually occupy large storage space or require high bandwidth in case of real-time transmission. Thus, an efficient compression of these huge point cloud data points becomes necessary. 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. In this work we present an alternative way to partition the point cloud data. The point cloud is divided based on the data partition using kd tree binary division instead of Octree's space partition method and forming a base layer. In base layer leaf nodes, the distribution of points is considered and projected to a 2D plane based on the flatness of the node points. Octree and Quadtree based partition is used to further convert the data to bitstreams. These are scalable point cloud bitstreams as we need only specific number of kd nodes in each time for a specific point of view. The use case is navigation in autonomous vehicles where it requires point cloud information up to a specific distance at different speeds. These scalable bitstreams of kd nodes can be used in real time transmission with low latency. Results show that compression performance is improved for geometry compression in point clouds and a scalable low latency streaming model is shown for navigation use case.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled “Point Cloud Compression and Low Latency Streaming,” presented by Karthik Ainala, candidate for the Master of Science degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Zhu Li, Ph.D., Committee Chair

Department of Computer Science & Electrical Engineering

Yugyung Lee, Ph.D.

Department of Computer Science & Electrical Engineering

Cory Beard, Ph.D.

Department of Computer Science & Electrical Engineering

Rufael Mekuria, Ph.D.

External Examiner

CONTENTS

ABSTRACT.....	iii
ILLUSTRATIONS	vi
TABLES	vii
ACKNOWLEDGEMENTS.....	viii
Chapter	
1 INTRODUCTION	1
2 BACKGROUND	5
2.1 MPEG point cloud compression (PCC).....	5
2.1.1 Bounding Box normalization and filtering.....	6
2.1.2 Octree Composition.....	7
2.1.3 Occupancy code entropy coding.....	7
2.2 Principal Component Analysis (PCA).....	7
2.3 Plane Projection Approximation(PPA) based geometry compression.....	9
2.4 Proposed Kd tree and PPA logic	11
2.4.1 Kd tree	11
2.4.2 PPA logic.....	12
3 EXPERIMENTAL AND COMPUTATIONAL DETAILS	14
3.1 Lossy geometry compression with PPA	14
3.2 Lossless geometry compression.....	17
3.3 Scalable point cloud bitstream	19
4 CONCLUSION.....	23
APPENDIX A.....	24

REFERENCE LIST	25
VITA.....	27

ILLUSTRATIONS

Figure	Page
1 Schematic of point cloud compression codec [5]	6
2 Flatness criteria distribution.....	10
3 Kd tree decomposition at level 8	13
4 Octree division up to enhancement layer.....	15
5 Octree division with PPA logic.....	15
6 Flow of the Lossless compression method	17
7 Standard deviation of xyz and uvw.....	18
8 Theta distribution for 4096 leaf nodes	18
9 Scalable point cloud bitstream model	20
10 LDPC staircase code structure	21
11 Decoding time vs loss probability percentage	22

TABLES

Table	Page
1 Compression results for predictive coding geometry	16
2 bpp comparison between PPA, PCC.....	19

ACKNOWLEDGEMENTS

I would like to thank my academic adviser and professor Dr. Zhu Li for providing me an opportunity and helping along the way to complete my thesis work.

I would like to thank University of Missouri - Kansas City for providing me an opportunity to work on this thesis report.

CHAPTER 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]. In this paper the focus is on intra geometry compression techniques.

The compression methods have been compared to the standard reference software for parameters like compression rate, distortion and have shown good results. The octree based codec is one of the MPEG reference software's for the point cloud compression. 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>. 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.

The Test material datasets and the test conditions for the point cloud compression are produced by the MPEG 3DG group. The datasets and test conditions are available in the Call for Proposals(CfP) for Point cloud compression document. Three categories of point cloud datasets are addressed by the CfP and they are as follows.

Category1: Static Objects and Scenes

Category2: Dynamic Objects

Category3: Dynamic Acquisition

Point cloud data sets from these three categories are used for the test material datasets. The test datasets are divided based on the above 3 categories and each category is further divided to 3 Test classes (A, B, C). Test class represents the encoding complexity of a point cloud, Whereas A represents lowest complexity and C is highest complexity. If a new encoding technique is proposed the original order of points in the file need not be maintained in the decoded version. The data type, attributes type and the format information is available in the header of the .PLY and .PCD files. The Category 3 contains dataset from Mitsubishi and Ford Campus Vision dataset and the Mitsubishi dataset is used for the scalable point cloud streaming in this work in chapter 3.3. The point cloud test materials will be tested under different conditions to compare with target bitrates. The 3 categories are tested with the following 3 conditions

- 1) Lossless Geometry & No Attributes
- 2) Lossless Geometry & Lossy Attributes
- 3) Lossy Geometry & Lossy Attributes

Dynamic objects dataset which belong to Category 2 are not required to be tested for 1 and 2 conditions as shown above. In this work the encoding techniques are designed only for static objects.

To summarize, in this work we present the following three codecs and its results:

1) A coding method is presented which introduces 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. The plane projection approximation method is introduced, and the results are compared with the reference software.

2) A coding method is implemented for lossless compression of the geometry attribute of the point cloud data using plane projection approximation (PPA) logic on points of kd leaf-nodes to minimize the variance of the data. The new transformed points of the kd nodes are encoded by PAQ compression [3].

3) A scalable point cloud bitstream is presented which can be used in real time transmission for vehicle navigation use case. The compression method is implemented using an adaptive kd tree binary partition based on the flatness of the kd leaf nodes and the diagonal lengths. These nodes are converted to bitstreams using Octree and Quadtree after PPA on kd leaf nodes.

The rest of the report is organized as follows, in chapter 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 is used to integrate the proposed method. Different partition methods like the Octree, kd tree and Quadtree are also discussed in this chapter. In chapter 3 the three proposed codecs are discussed, while in chapter 4 the results are shown, comparing with existing coding methods and in chapter 5 conclusion and future work is explained.

CHAPTER 2

BACKGROUND

2.1 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 [5], this reference explains the full codec architecture including inter and intra coding. The intra coding part implemented in this codec is used to implement the proposed methods. The intra coding of the input point cloud frame is done by modules 1, 2 and 3 in Figure 1. In the proposed methods, 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. The models are implemented based on these three steps for the Plane projection approximation and learning based compression of point cloud data geometry. These three modules are explained as follows:

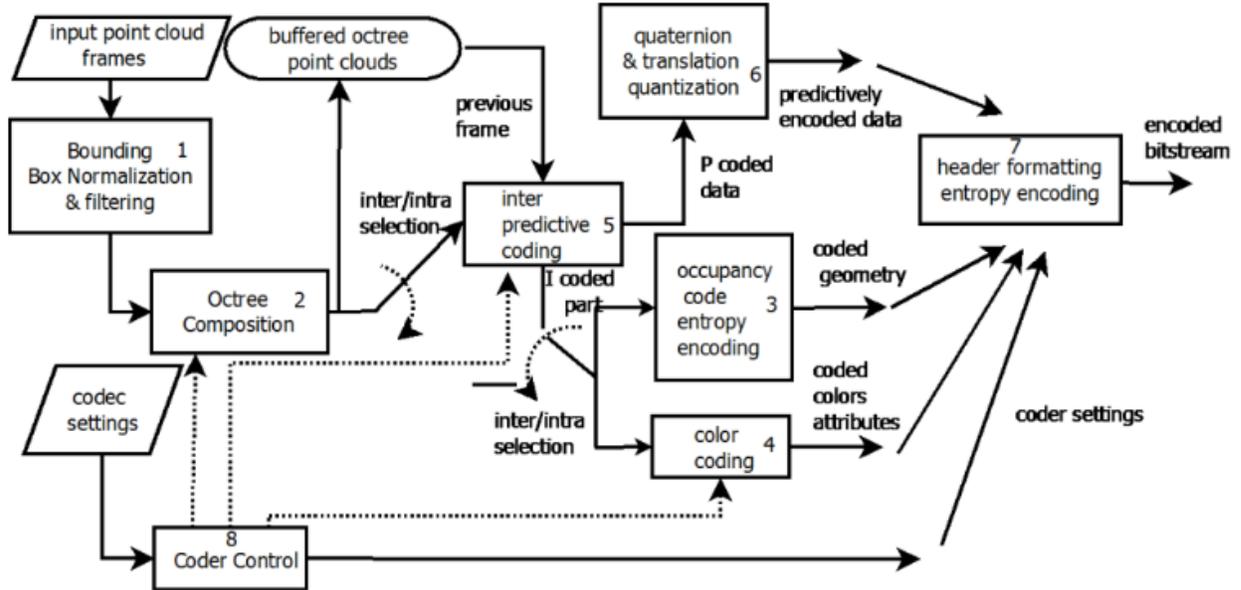


Figure 1: Schematic of point cloud compression codec [5]

2.1.1 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.

2.1.2 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 into 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.

2.1.3 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.

2.2 Principal Component Analysis (PCA)

Principle component analysis (PCA) is a statistical procedure that uses orthogonal transformation to convert a set of observation of possibly correlated variables into a set of values of linearly uncorrelated variables called principle components. The main use of PCA for this work is optimal projection of node-points to PCA subspace such that variance of the projected data is lesser than the original point cloud data.

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 $n = 1, 2, \dots, N$. So, $X = (X_1, X_2, \dots, X_N)^T$. Now, covariance of matrix X is given as:

$$C = X * X^T \quad (1)$$

Consider the eigen value decomposition of C:

$$C = \Phi \Lambda \Phi^{-1} \quad (2)$$

Where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$ is a diagonal matrix containing C's eigen values λ_n such that $\lambda_1 > \lambda_2 > \dots > \lambda_N$. 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 all three eigen vectors and the points of the leaf node are then projected into PCA subspace for optimal rotation as follows:

$$Y = \Phi^T * X \quad (3)$$

Y has the new projected matrix u,v,w instead of original x,y,z. The variance of [u,v,w] is lesser than the [x,y,z] such that the ratio given below is greater than 1 and this ratio represents the compression gain.

$$\text{Gain} = \text{var}([x,y,z]) / \text{var}([u,v,w]) \quad (4)$$

If this gain is greater than 1 then, entropy coding could achieve better compression on [u,v,w] rather than [x,y,z].

2.3 Plane Projection Approximation(PPA) based geometry compression

Recent progress in Graph Signal Processing (GSP) has found several successful applications of GSP to signal compression on non-uniformly sampled signals, examples are [6][2] for compression of SIFT features, color attributes. One common 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 the same principle is applied to achieve local data clustering and adaptive transform by finding flat voxel nodes and then local coordinates transform is applied to have a compact 2D representation for better compression.

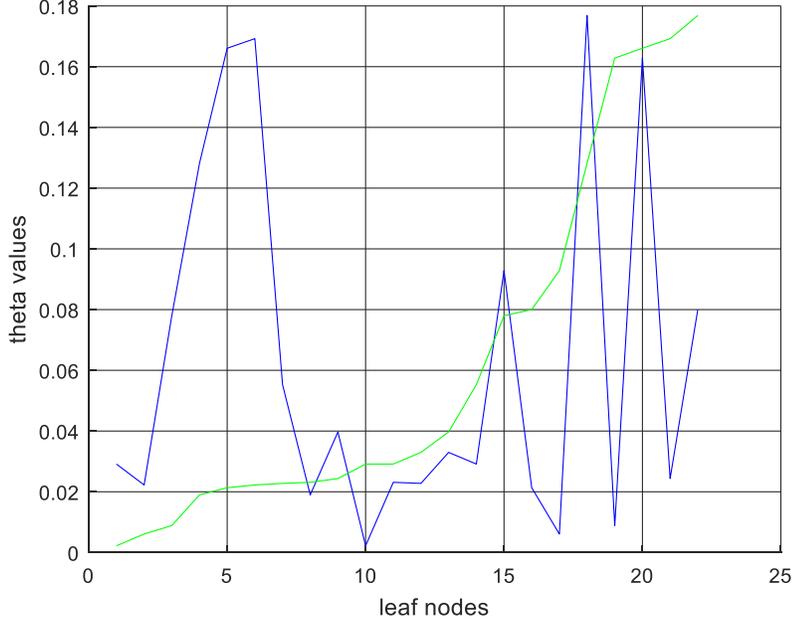
Due to the nature of the point cloud capture process, many local patches are more or less exhibiting flat characteristics indicating that an alternative coding scheme can be more efficient than Octree decomposition. In this work a Plane Projection Approximation (PPA) coding mode is introduced, that project the voxels onto a 2-D plane and coding the geometry points. 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,

$$S = E\{(P_k - \bar{P})^T(P_k - \bar{P})\} \quad (5)$$

Where the mean of the geometry is obtained from the centroids of the voxels, \bar{P} . The Eigen values are computed from 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,

$$\theta = \frac{\min\{\lambda_1, \lambda_2, \lambda_3\}}{\lambda_1 + \lambda_2 + \lambda_3} \quad (6)$$

The distribution of the flatness criteria in equation 6 is illustrated in the Figure 2 below, for the “ski” sequence [7] Octree nodes decomposition upto depth of 6. The sorted θ



is illustrated in green.

Figure 2. Flatness criteria distribution.

As shown above there are quite a number of Octree nodes with voxels more or less lying on a surface and the PPA coding mode is introduced 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,

$$indx = \text{sort}(u * \max_u + v) \quad (7)$$

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 a proper quantization scheme is applied to match the Octree range coding PSNR quality, and then bitstream is generated with a self adaptive entropy coding scheme called PAQ[3].

PAQ is a shallow neural network like context modeling coupled with Arithmetic

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 coding 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.

2.4 Proposed Kd tree and PPA logic

2.4.1 kd tree

Kd tree is a binary data partition scheme where a 3D space is partitioned to equal sizes. The direction of the kd tree partition is set based on the maximum covariance of the data among the three directions (x,y,z). The number of divisions made is said to be the depth or level of the tree. The undivided original bounding box is the root node of the tree and with each division in the level the kd nodes are split in to two nodes with equal points. The nodes at the final level of the tree are called leaf nodes. So, all the points of the original point cloud are equally divided in to the leaf nodes.

The stopping criteria for kd tree division can either be the maximum depth or maximum no. of points in a node. For our work, we ensure that the leaf node size is around 200 points and typically the kd level of 12 gives us that number. The ‘flatness’ of the surface in each node is given as:

$$\theta = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (8)$$

Where $\lambda_1 > \lambda_2 > \lambda_3$ are the eigen values of the covariance matrix of X which is set of

all the points in a node. The θ is the measure of the flatness (of the surface made by) the points in each node as explained in chapter 2.3. The flatness of the leaf nodes leads to more entropy gain. So a θ threshold of 0.03 is considered as the optimum value for entropy gain. The threshold value is decided based on the percentage of the leaf nodes θ fall in to less than 0.03 threshold value. If we further divide to higher kd levels the percentage of flat leaf nodes increase but the size of the leaf node gets smaller than 100 and thereby increasing the kd signalling bits. Like the Octree division the kd nodes are divided up to a certain level to create the enhancement layer. The PPA logic (2.4.2) is then applied on the kd nodes of the enhancement layer to further divide the kd nodes to obtain flat surfaces. An example of kd division up to level 8 on longdress sequence is shown in Figure 3.

2.4.2 PPA logic

Plane projection approximation (PPA) in this work is procedure of applying Principle Component Analysis (PCA) to determine the flatness of the surface in each of the octree or kd node. PCA is used to get the direction of the maximum variance. For point cloud data, we get three directions. If surface is flat enough then all its variances lie in the first two principal components and the variance in the 3rd component is minimum. Since these variances is given by their corresponding eigen values, for flat surface, ratio given in (8) should be the minimum. We define a threshold value such that if $\theta < \text{threshold}$, we consider

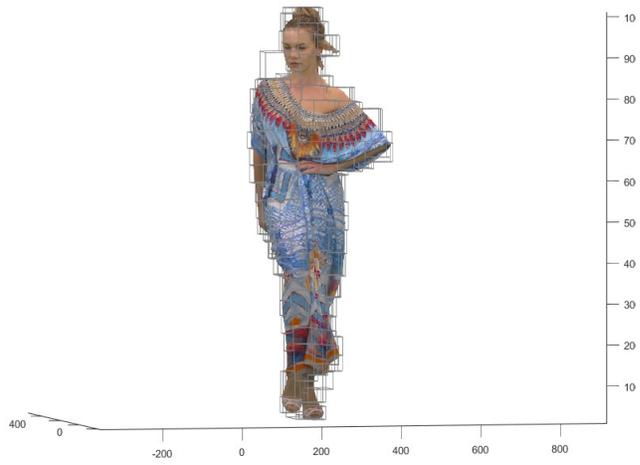


Figure 3. Kd tree decomposition at level 8

the surface as flat. A range of threshold values are used to analyse the flat kd nodes distribution and based on the results a threshold value is selected which gives maximum flat nodes.

CHAPTER 3

EXPERIMENTAL AND COMPUTATIONAL DETAILS

3.1 Lossy geometry compression with PPA

In the MPEG PCC codec the Octree levels (Levels of Detail) are predefined irrespective of the surface geometry of the point cloud data. In this method a fixed Level of Detail is used which produces the 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.

The proposed method is implemented 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 the module Bounding box normalization and filtering, from the pcc library. A radius value is mentioned in the configuration file and the filtering module removes the noisy points from the original pcd file. The proposed encoding model is applied on a pcd data which has 45617 points. The point cloud data is acquired from (<http://vcl.itl.gr/reconstruction/>) which is taken as the reference source for the point cloud codec. The θ value is set to 0.05 as explained in 2.4.2 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.

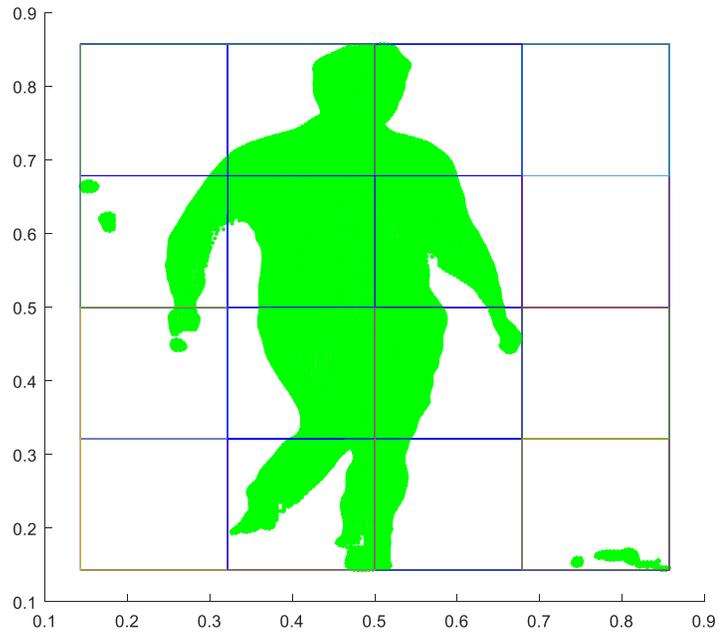


Figure 4 Octree division up to enhancement layer

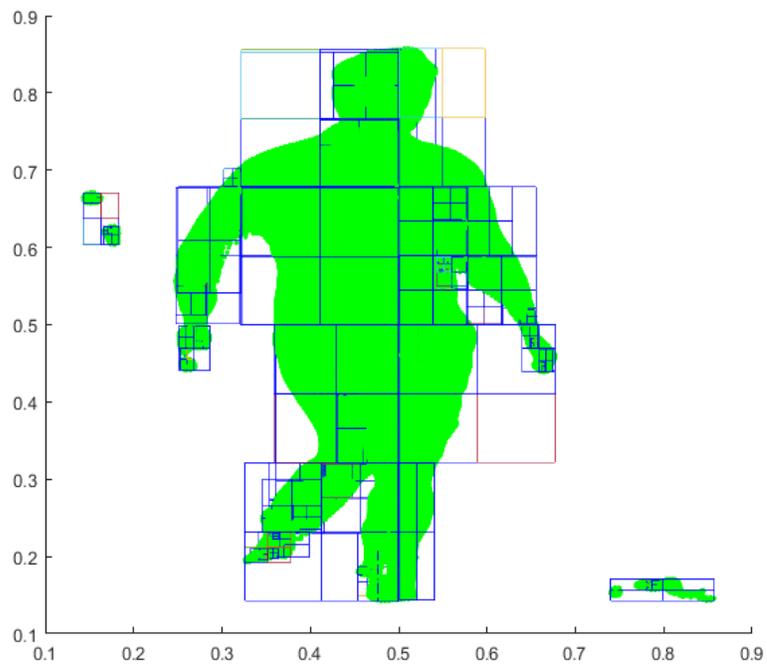


Figure 5 Octree division with PPA logic

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 after PPA is shown in Figure 5. Table 1 shows the results comparison of the proposed plane projection approximation method and the reference software's point cloud codec. It is observed that PPA has significant compression gain over PCC with little loss in PSNR.

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

3.2 Lossless geometry compression

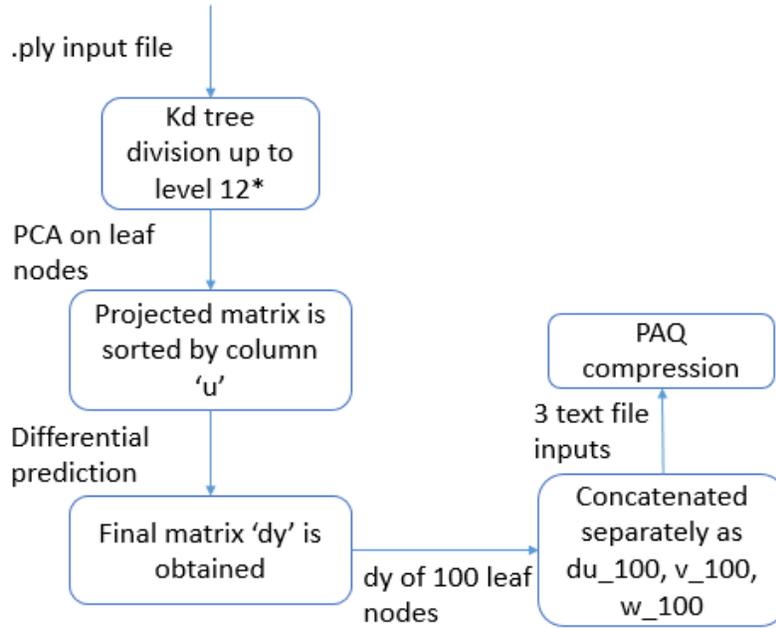


Figure 6: Flow of the Lossless compression method

We used 8i's longdress sequence dataset for our experiment which has around 765821 points. The geometry points of the dataset are considered and color attributes are ignored for this compression method.

As explained in chapter 2.2 the process of transforming the original leaf node xyz to dy is to reduce the entropy of the original leaf nodes. It is evident from Figure 7 that the entropy is reduced as the standard deviation of u,v,w is low compared to xyz. It also justifies the reason for the selection of higher level 12 for the kd tree division of original point cloud.

The theta of all the leaf nodes is found from Equation 8 and its distribution is shown in Figure 8. The flatness of each leaf node is implied by its theta value. By considering the theta value 0.03 as the threshold to measure flatness. The kd level of 12 brings around 85%

of the leaf nodes to flat as seen from distribution in Figure 8 around 3500 out of 4096 leaf nodes are below the theta value of 0.03.

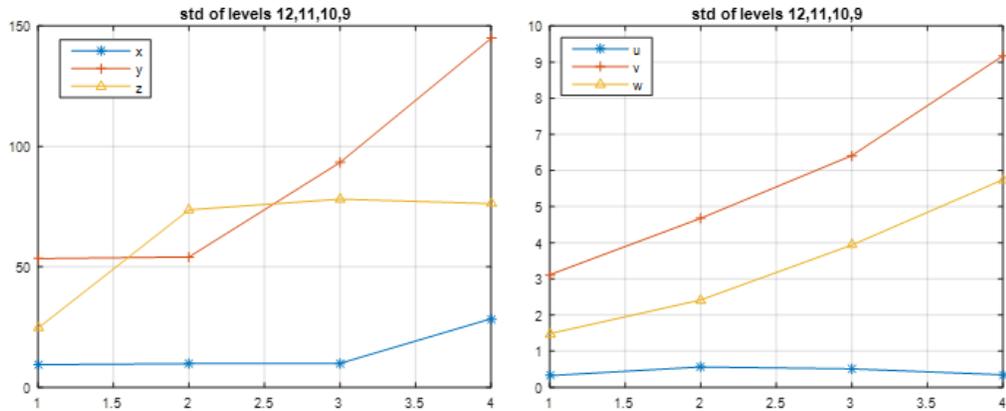


Figure 7: Standard deviation of xyz and uvw.

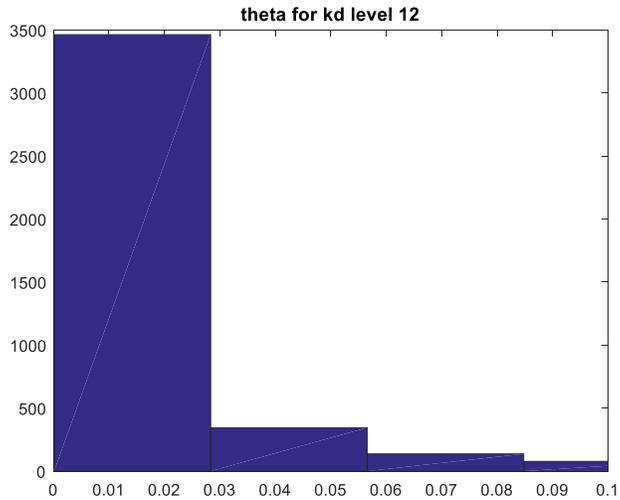


Figure 8: Theta distribution for 4096 leaf nodes

The results of Plane projection approximation (PPA) method is compared with state of the art pcc-mp3dg software. The pcc has both lossy and lossless compression modes. It works by using the octree division of higher levels and the leaf nodes are distributed with one or two geometry points. These points along with octree bits are binarized and later encoded using the range coder. The results from the lossless mode of the pcc-mp3dg are obtained for

longdress point cloud sequence and compared with the lossless ppa in the Table 2.

Table 2. bpp comparison between PPA, PCC.

Method (lossless)	Bits per point (bpp)
ppa	4.39
pcc-mp3dg	20.16

3.3 Scalable point cloud bitstream

A framework model for scalable point cloud bitstream which is used for the vehicle navigation use case is shown. Three data sets provided by MERL are available to test the model. The MERL data set has 3 point cloud data models City tunnel, Overpass and tollbooth with 5 million to 20 million points. Overpass has 5 million points which is huge and we use kd tree binary partition which is a data based partition as shown in Chapter 2.4.1. So if we implement a kd tree division of 10 on the root node ie the original point cloud with 5 million. We get 1024 kd leaf nodes with almost equal number of points. But If we observe the diagonal length of each kd leaf nodes it varies widely based on the sparsity of the points. So if a moving vehicle need the point cloud data in real time we need to stream that particular kd leaf nodes information which is necessary at that point of view(pov). So, If we want to stream the necessary information needed for the vehicle at that speed we should stream the leaf nodes based on their size in the real world.

So, If we observe the leaf nodes at kd levels of 8,9 or 10 the leaf nodes are of same size in number of points but their size in the real world is different and it can be estimated by their diagonal lengths. So we use an Adaptive kd tree division so the kd leaf node levels are not constant and are driven by the kd nodes diagonal lengths.

These kd leaf nodes are converted to bitstreams using Octree division and the depth

of the octree is decided based on the diagonal length of the leaf nodes. So, the selected kd leaf nodes can be sent as a bitstream sequences to view that particular kd node area. To obtain maximum advantage of using the Octree bitstreams we have a condition to limit the number of points in a kd leaf node and it is set to 300 points (approx). This model is shown as blocks in Figure 9 as individual modules.

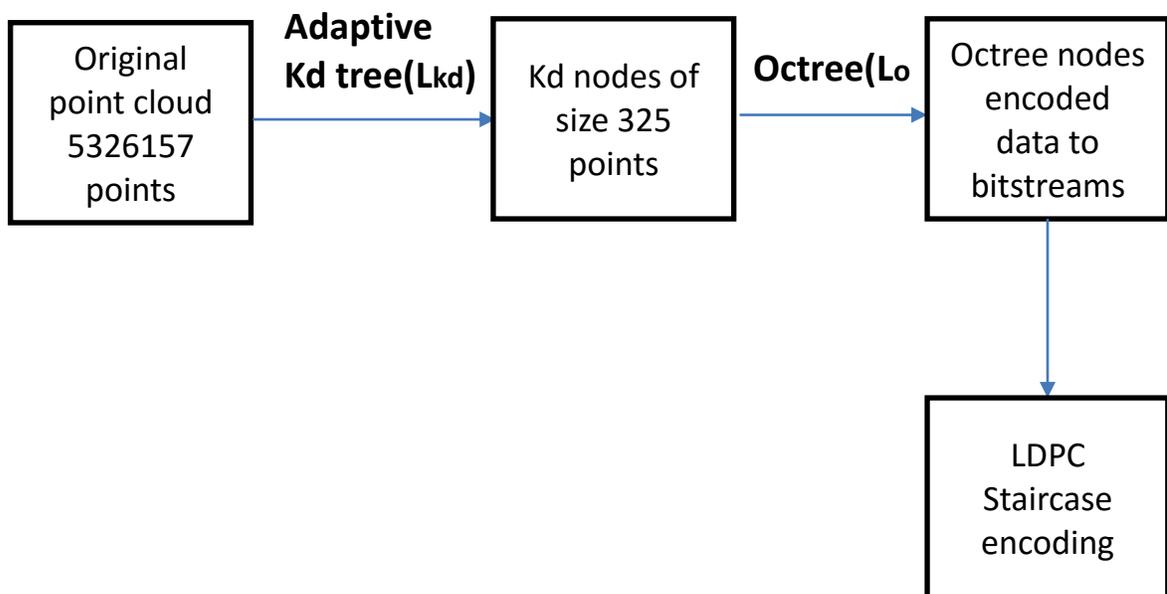


Figure 9: Scalable point cloud bitstream model.

The Octree bitstreams are streamed over the network and the bitstreams are encoded and decoded using Forward Error Correction based codecs. An AL-FEC based codec named LDPC stair case encoding technique [8] is used to test the reliability of the framework in real word. This codec is available as an opensource code from openfec.org. The LDPC staircase code structure is shown in Figure 10, k represents the source symbols, source symbols size is represented by E and usually it is high for LDPC [8]. So, the object transfer length is obtained from L which is k times E .

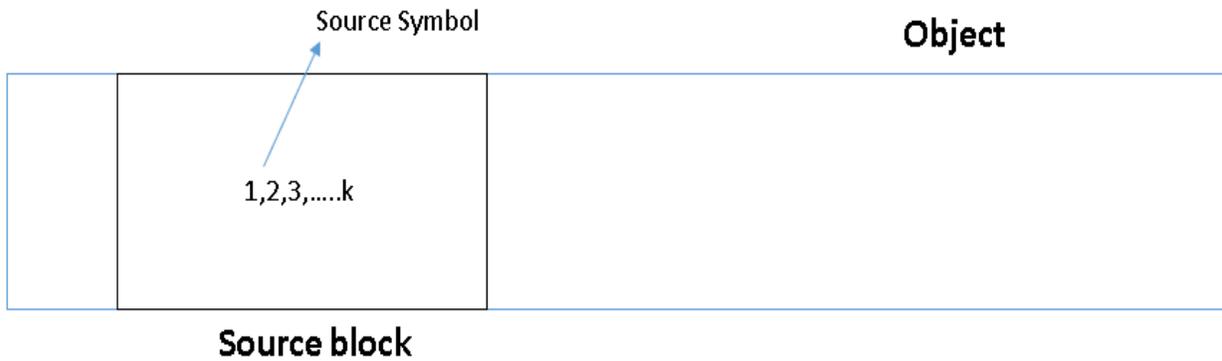


Figure 10: LDPC staircase code structure

If we consider a typical kd nodes bit stream size for the MERL dataset point clouds it will be around 1 million bytes by considering symbol size E as 1024 and number of symbols k as 1024. we can estimate the reliability of encoding and decoding these bitstream with LDPC by using the opensource software and the results are shown in Figure 11 and the MATLAB code is available in Appendix A. The decoding time increases as the loss probability increases and moves above 25 percent. The decoding time which is around 1.8 to 2.8 milliseconds is acceptable in the real time streaming use case even after considering the propagation and transmission delay of the network.

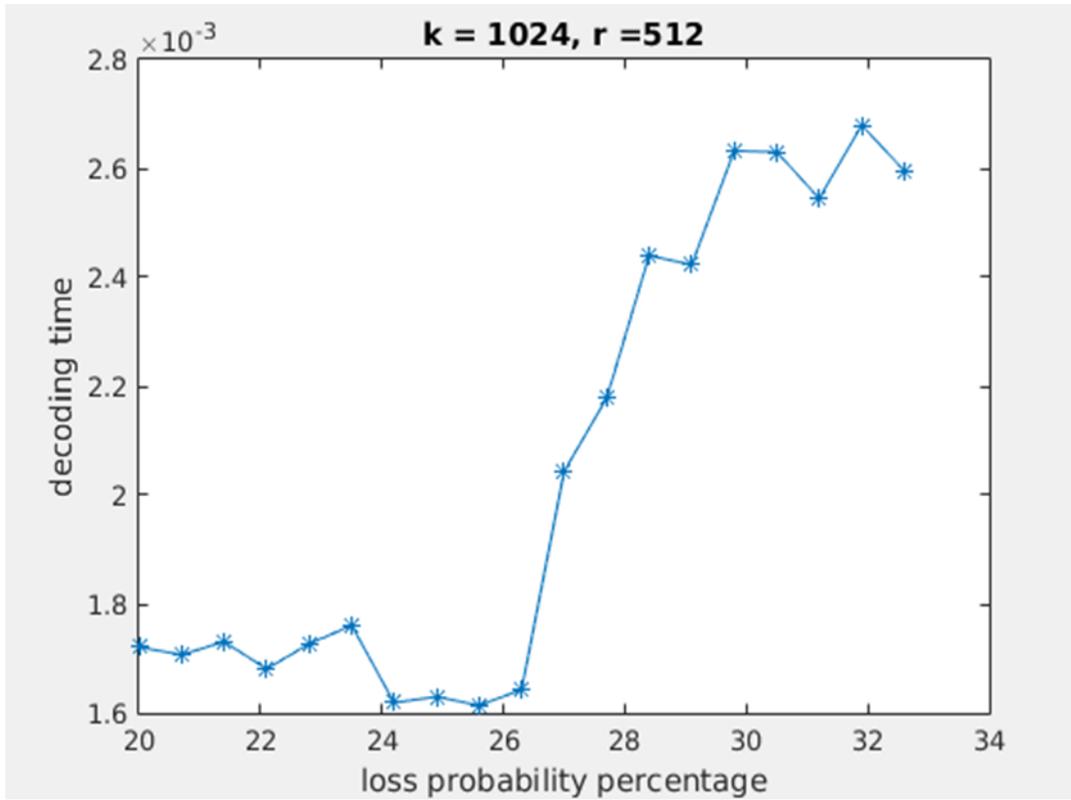


Figure 11: Decoding time vs loss probability percentage

CHAPTER 4

CONCLUSION

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 geometry of the more fine grained details of the point cloud. 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 comparisons can be made more effectively to other proposed methods.

The lossless compression uses the similar technique as above by using PPA approach with kd tree to compress the geometry of the point cloud data and proved itself to be more efficient than the lossless mode of PCC. The assumption that the PCA and differential prediction minimizes the variance of the original xyz, which results to better compression turned out to be true.

The kd tree approach of the ppa mode is also proven to be effective, by using adaptive kd tree and encoding the bitstreams obtained from octree. The bitrate can be changed by varying the voxelization during the octree division. The reliability of decoding the bitstreams using LDPC stair case coding is shown which can be used in the real world navigation use case.

APPENDIX A

CODE FOR DECODING TIME ESTIMATION OF LDPC

```
kval = 1024;
rval = 512;
count = 0;
for lossper = 20:0.7:45
    lossval = lossper; %lossper value assigned HERE
    comp_file = sprintf('%s%d%s%d%s%f', './eperftool -codec=3 -k=', kval, ' -r=', rval, ' -seed=2 -loss=2:', lossval);
    [status,cmdout] = system(fullfile(comp_file));
    count = count + 1;
    lossprobper(count) = lossval;
    Z = textscan(cmdout, '%s', 'Delimiter', ' ');
    dec_status = Z{1}(length(Z{1}));
    dec_status = char(dec_status);
    if find(dec_status == '0')
        dec_timestring = Z{1}(43);
        dec_timestring = char(dec_timestring);
        dec_timecell = textscan(dec_timestring, '%s', 'Delimiter', '=');
        dec_time(count) = dec_timecell{1}(2);
    else
        dec_time(count) = num2cell(0.00);
    end
end

end

figure(44); plot(lossprobper, str2double(dec_time), '-*');
xlabel('loss probability percentage'); ylabel('decoding time');
```

REFERENCE LIST

- [1] R. Schnabel and R. Klein, "Octree-based Point-Cloud Compression," in *Proceedings of the 3rd Eurographics Symposium on Point-Based Graphics, Zurich, Switzerland, 2006*, pp.111-120.
- [2] C. Zhang, D. Florêncio, and C. Loop, "Point Cloud Attribute Compression with Graph Transform," *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, 2014, pp. 2066-2070.
- [3] M. V. Mahoney, "Adaptive Weighing of Context Models for Lossless Data Compression," Technical Report, CS-28005-16, Florida Institute of Technology CS Department, 2005.
<https://cs.fit.edu/~mmahoney/compression/cs200516.pdf>
- [4] Y. Huang, J. Peng, C. Kuo, and 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] R. Mekuria, K. 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. 27, no. 4, pp. 828-842, April 2017.
- [6] X. Xin, Z. Li, and A.K. Katsaggelos "Laplacian Embedding and Key Points Topology Verification for Large Scale Mobile Visual Identification," *Signal Processing.: Image Communication*, vol. 28, no.4, pp. 323-333, 2013.
- [7] D. Alexiadis, D. Zarpalas, and 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/>

- [8] V. Roca, M. Cunche, C. Thienot, J. Detchart, and J. Lacan, "RS + LDPC-Staircase Codes for the Erasure Channel: Standards, Usage and Performance," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Lyon, 2013, pp. 638-644.

VITA

Karthik Ainala received Bachelors of technology in Electronics and Communication Engineering from Amrita University, Kerala, India in 2012. He received “Dean’s international scholarship award” for Master’s program at UMKC. He is graduate research assistant at Multimedia and communications lab, UMKC. He is an MS Research graduate in Electrical and Computer Engineering at UMKC under supervision of Dr. Zhu Li.

His Research interests focus on 3D point cloud data compression using image compression techniques and machine learning algorithms, multimedia communication. Has Experience in Embedded System design.