International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-3, January 2020

Tensor Tucker Decomposition based Geometry Compression on Three Dimensional LiDAR Point Cloud Image

PL. Chithra, A. Christoper Tamilmathi

Abstract*: *Data Visualization in static images is still dynamically growing and changing with time in recent days. In visualization applications, memory, time and bandwidth are crucial issues when handling the high resolution three dimensional (3D) Light Detection and Ranging (LiDAR) data and they progressively demand efficient data compression strategies. This shortage is strongly motivating us to develop an efficient 3D point cloud image compression methodology. This work introduces an innovative lossless compression algorithm for a 3D point cloud image based on higher-order singular value decomposition (HOSVD) technique. This algorithm starts with the preprocessing method which removes the unreliable 3D points and then it combines the HOSVD together with the normalization, predictive coding followed by Run Length encoding to compress the HOSVD coefficients. This work accomplished lower mean square error (MSE), high (infinitive) Peak signal noise ratio (PSNR) to produce the lossless decompressed 3D point cloud image. The storage size has been reduced to one by fourth of its original 3D LiDAR point cloud image size.

Keywords* : *Higher-order singular value decomposition, Point Cloud Compression, Predictive coding, Run-length coding, Tensor decomposition, Tucker decomposition.

I. INTRODUCTION

3D Point cloud is an unordered massive raw data of 3D objects captured by the sensors. Each point has the position (x, y, z), a color (R, G, B) or (Y, U, V) and other things like transparency, time of the acquisition, angle, etc. The static point cloud consists of several tens of millions of 3D points depending on the application. Each point occupies the 8/10 bits of memory, reduced by the compression technique based on different techniques such as Octree, depth image handling, global and local topology method. Point cloud data are captured for many purposes, including generating 3D CAD models for industry manufactured parts, for metrology and quality review, and for a multitude of visualization, animation, rendering, and mass customization applications.

Revised Manuscript Received on January 30, 2020.
***** Correspondence Author
Dr.PL. Chithra*, Professor, Department of Computer Science,University of Madras, Gundy Campus, Chennai- 600 025,Tamil Nadu, India. Email: chitrasp2001@yahoo.com
A.Christoper Tamilmathi, Research Scholar, Department of Computer Science, University of Madras,Gundy Campus,Chennai- 600 025, Tamil Nadu, India. Email: amathimphil5@gmail.com

Geographic information systems, point clouds are one of the sources used to make a digital elevation model of the terrain.

The 3D point cloud applications are facing computational and data management challenges when working on massive data due to the limitation of the memory, time, transmission bandwidth and quality of the decompressed image. This work prefers lossless compression to avoid the loss of information during the compression process. The higher-order dimensional image always occupies, the more memory spaces compared with the low dimensional data image. In this situation, tensor decompositions and in specific tucker models are a promising mathematical technique for higher-order compression and dimensionality reduction in the field of visualization and graphics.

The remaining section of the article is organized as follows. Section II discusses the related work of the proposed method. In Section III, explains the proposed compression technique. Experimental results are discussed in Section IV. The conclusion provided in Section V.

II. RELATED WORK

A. Point Cloud Compression

Arnaud et al., discussed the main idea is that depth maps implicitly define a global manifold structure of the underlying surface of a point cloud. It is possible to only work on the parameter domain and to modify the point cloud indirectly [1]. Paval et al., presents a new 3ad point cloud scanning method for planes detection, it includes several image processing methods like the connected component labeling and the shape border detection [2]. Dai Cheng-Qiu et al., stated the algorithm of 3D point cloud data based on the octree. First, the coordinates of 3D point cloud data converted to Morton without distortion were discussed. As the Morton is too discrete, an optimization algorithm for the Morton is proposed by IT on this basis finally [3]. Mirjana et al., introduced space-filling curve dictionary-based compression, employs dictionary-based compression in the spatial data management domain and enhances it with indexing capabilities by using space-filling curves[4]. Javier et al., used a fast variant of Gaussian Mixture Models and an expectation-Maximization algorithm to replace the points grouped in the previous step with a set of Gaussian distributions.

These learned models can be used as features to find matches between two consecutive poses and apply 3D pose registration using RANSAC [5]. Hamindreza and Andreas discuss the use of Conventional image-based compression methods for panorama images to encode the range, reflectance, and color value for each point [6]. Jae-Kyun et al., proposed hybrid range image coding algorithm, which predicts the radial distance of each pixel using previously encoded neighbors adaptively in one of the three coordinate domains: range image domain, height image domain, and 3D domain [7]. From the above 3D point cloud compression not based on the tensor decomposition techniques. Tensor decomposition based image compression is suitable for point cloud data since 3D point cloud structure is based on the 3D tensor format.

B. Compression based on Tucker Decomposition

Ekta et al. proposed the OCTEN the compression-based online parallel implementation for the CP decomposition and it is suitable for the bigger tensor data [8]. Jin Li et al., focus on low complexity nonnegative tensor decomposition (NTD) based compression method of hyperspectral images. This method is based on a pairwise multilevel grouping approach for the NTD to overcome its high computational cost [9]. Jianze Li et al., proposed the algorithm for point cloud denoising based on Tucker decomposition to compress this patch tensor to be a core tensor of smaller size then remove the noise using hard thresholding [10]. Maria Ishteva et al., presented the higher-order orthogonal iterations and outline two new algorithms, based on the trust-region and conjugate gradient methods on manifolds [11]. Rafael Ballester et al., proposed TTHRESH: tensor compression for Multidimensional visual data [12]. Haider Hameed al-Mahmood and Zainab Al-Rubaye stated the lossless image compression based on Predictive coding and Bit-plane slicing [13]. Bo Du et al., introduced the hyperspectral image compression based on Patch-based low-rank tensor decomposition [14]. From the above tensor decomposition compression related work shows that the tensor decomposition process only implemented on the 3D object image compression, not on the 3D point cloud image. To the best of our knowledge it is the first tensor decomposition based 3D point cloud geometry compression algorithm compression based on Predictive coding and Bit-plane slicing [13]. Bo Du et al., introduced the hyperspectral image compression based on Patch-based low-rank tensor decomposition [14]. From the above tensor decomposition compression related work shows that the tensor decomposition process only implemented on the 3D object image compression, not on the 3D point cloud image. To the best of our knowledge it is the first tensor decomposition based 3D point cloud geometry compression algorithm.

C. Tucker/HOSVD Decomposition

In this work, tensors refer to a 3-dimensional array of N=3. Vectors (tensor of dimension 1) are written in bold lower case as in $\boldsymbol{x} = (x_1,\ldots,x_N)$, matrices (tensors of dimension 2) in bold capitals i.e. **U**, and general tensors with three-dimension written in bold capitals Monotype Corsiva i.e., $\mathcal{T}$. We generally use the notation and definitions from [multidimensional base paper]; in specific, rows and columns in matrices generalize to tensors as fibers. The n-th mode unfolding technique of a tensor $\mathcal{T}$, places all n-mode fibers next to each other as columns of a matrix and are denoted as $\mathcal{T}_{(n)}$. The tensor-times-matrix product (TTM) contracts a tensor's n-mode fibers along a matrix's rows and is demoted as $\mathcal{T} \times_n \mathbf{U}$[12].

D. Tensor Tucker Decomposition

Tucker decomposition method decomposes the original PCD tensor into a one core tensor and multiple numbers of factor matrices. A Singular value decomposition (SVD) technique applied on third order tensor for three coordinate system to reduce the dimension is called as Higher-Order singular value decomposition (HOSVD) technique. A third-order tensor $\mathcal{T} \in F^{d1\times d2\times d3}$ where $\mathbb{F}$ is either $\mathbb{R}$ or $\mathbb{C}$.

The tucker decomposition can be denoted in (1) as follows

$$\mathcal{T} = \mathcal{T}_1 \times_1 \mathbf{U}^1 \times_2 \mathbf{U}^2 \times_3 \mathbf{U}^3 \quad (1)$$

where $\mathcal{T}_1 \in F^{d1\times d2\times d3}$ is the core tensor, a third-order tensor that contains the 1-mode, 2-mode and 3-mode singular Values of $\mathcal{T}$, which are defined as the Frobenius norm of 1-mode, 2-mode and 3-mode slices of tensor $\mathcal{T}$ respectively. $\mathbf{U}^1$,$\mathbf{U}^2$ and $\mathbf{U}^3$ are unitary matrices in $F^{d1\times d1}$, $F^{d2\times d2}$, $F^{d3\times d3}$ respectively. The j-mode product (j=1, 2, 3) of $\mathcal{T}$ by $\mathbf{U}^j$ is denoted as $\mathcal{T}_1 \times \mathbf{U}^j$ with entries are given in (2), (3) & (4) [19].

$$(\mathcal{T}_1 \times_1 \mathbf{U}^1)(d_1,d_2,d_3) = \sum_{j1=1}^{d1} \mathcal{T}_1(j1,d2,d3)\,\mathbf{U}^1(d1,j1) \quad (2)$$

$$(\mathcal{T}_1 \times_2 \mathbf{U}^2)(d_1,d_2,d_3) = \sum_{j2=1}^{d2} \mathcal{T}_1(d1,j2,,d3)\,\mathbf{U}^2(d2,j2) \quad (3)$$

$$(\mathcal{T}_1 \times_3 \mathbf{U}^3)(d_1,d_2,d_3) = \sum_{j3=1}^{d3} \mathcal{T}_1(d1,d2,j3)\,\mathbf{U}^3(d3,j3) \quad (4)$$

The tucker decomposition decomposes a third order tensor into a core tensor and set of factor matrices.

III. PROPOSED WORK

The architecture of the proposed 3D point cloud compression method is shown in figure 1. The proposed algorithm consists of four main stages. First, the outliers in the input point cloud (PCD) data have been removed by using statistical subspace detection technology. Second, the HOSVD is applied on an input PCD to produce three orthogonal factor matrices and a three-dimensional core tensor. Third, the Quantization process applied independently on both core tensor and the factor matrices to discretize the element values. Fourth, the resultant core tensor and factor matrix have been compressed by two-dimensional (2D) predictive coding followed by Run Length coding.

A. Statistical Subspace Detection

The acquired 3D PCD is usually polluted due to the multi-reflection of an object from the ground. These erratic data lead to flawed values and cause representation failures of objects in point clouds.

This proposed pre-process method resolves the issue of unreliable raw PCD by detecting and removing the unreliable 3D points. This pre-process step consists of three steps. First form the subspace by using the Euclidean distance value of each pair of

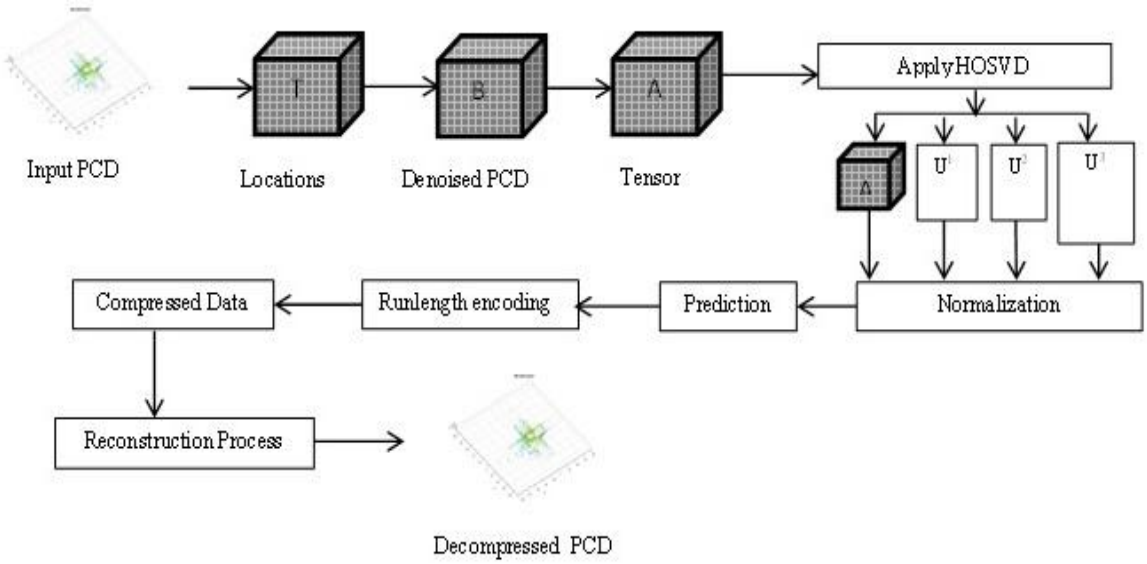

Fig. 1. General architecture of proposed 3D point cloud compression.

3D points in a point cloud. Next, estimate the threshold value based on the mean and standard deviation value of each point's subspace. Then find the unreliable points which are having higher distance value than the threshold value and eliminate it from the 3D PCD. The pre-processing resultant image only contains reliable points in the PCD. These reliable point cloud images considered as an original point cloud image $\mathcal{B}$ for the remaining steps of the proposed algorithm is shown in figure 1.

B. Tucker Transformation

In the 2D image, the SVD process unfolding the data into left-singular vectors (column-space), singular values (eigen values) and right-singular vectors (row-space). In a 3D PCD, the HOSVD has projects on to high variance subspaces in order to reduce dimensionality of an image. HOSVD is applied on the third-order tensor $\mathcal{A} \in \mathbb{R}^{I1 \times I2 \times I3}$ from the previous step of the algorithm to decompose it into core tensor $\mathcal{B} \in \mathbb{R}^{R1 \times R2 \times R3}$ and three Factor matrices $\mathbf{U}^n \in \mathbb{R}^{In \times Rn}$. Figure 2 shows the HOSVD (Tensor Tucker decomposition) subspaces.

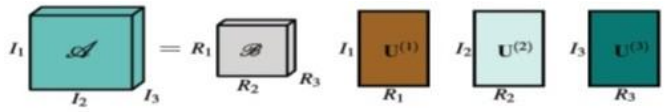

Fig. 2. The HOSVD (Tensor Tucker decomposition) subspaces.

The decomposition of the 3D point cloud tensor structure has been shown in figure 3.

Fig. 3. The architecture of HOSVD (Tensor Tucker decomposition).

In this proposed method, the input tensor $\mathcal{A}$ is unfolded as $\mathcal{A}_n \in \mathbb{R}^{I \times J}$ and the compute the symmetric tensor ($\underline{\mathcal{A}_n}$) for each unfolded $\mathcal{A}_n$ by using the (6)

$$\underline{\mathcal{A}_n} = \mathcal{A}_n - \mathcal{A}_n^T \quad (6)$$

where $\underline{\mathcal{A}_n}$ is a symmetric tensor, $\mathcal{A}_n$ is an unfolded tensor $\mathcal{A}$ and $\mathcal{A}_n^T$ is a transpose of $\mathcal{A}_n$.

Next, the left singular vector $\mathbf{U}^n$ have been obtained from the eigenvalue decomposition using (7).

$$\mathbf{U}^n = \text{eigen } \underline{\mathcal{A}_n} \quad (7)$$

Then the rightmost singular vector V^n has been calculated from (8).

$$V^n = \mathbf{U}^{n'} . \mathcal{A}_n \quad (8)$$

where $\mathbf{U}^{n'}$ is a transpose of $\mathbf{U}^n$.

Finally, the core tensor has been derived from the original tensor $\mathcal{A}$ and the factor matrices $\mathbf{U}^n$, expressed in (9).

$$G = \mathcal{A} \times_1 \mathbf{U}^{1-1} \times_2 U^{2-1} \times_3 U^{3-1} \quad (9)$$

The core tensor is the same structure of the original tensor with reduced size.

The reduced dimensions subspaces occupy fewer memory spaces than the original 3D point cloud data. The HOSVD transformation of the proposed method has been explained in Algorithm 1.

Algorithm 1: HOSVD transformation of 3D PCD of size $I_1 \times I_2 \times I_3$

```
{
  //HOSVD Transformation
For n=1 to 3 do
   {
```

$\mathcal{A}_n$ =Unfold ($\mathcal{A}$, n)
//Symmetric matrix of size $I_n \times I_n$
$\underline{\mathcal{A}_n} = \mathcal{A}_n - \mathcal{A}_n^T$
//full decomposition: eigen values $\varepsilon^{(n)}$ in non increasing //order
$\mathbf{U}^n$=eigen ($\underline{\mathcal{A}_n}$)
$V^n = \mathbf{U}^{n'} . \mathcal{A}_n$
//Back to the original size
$\mathcal{A}$ = fold ($\mathcal{A}_n$)
}
}

// $\mathcal{A}$ is the combination of core and $\mathbf{U}^1$ $\mathbf{U}^2$ $\mathbf{U}^3$

Algorithm 2: Prediction

```
{
//Prediction Error
For i=2 to size [A1,1]–1
  {
  For j=2 to size [A1,2]
     {
  fx = A1[i, j-1]+A1[i-1,j]
  e[i-1, j] = e[i-1,j]–fx
     }
  }
}
```

Algorithm 3: Run-length Encoding
Input: Stream of bits
Output: Elements and number of occurrences

```
Function [d, c] = RLE(x)
 {
   ind=1
   d[ind]=x[1]
   c[ind]=1
   for i=2 to length (x)
     {
    if x[i-1] = = x[i]
      {
           C[ind]=c[ind]+1
    else
         ind=ind+1
        d[ind]=x[i]
        c[ind]=1
      }
     }
 }
```

C. Normalization

Normalization is playing an important role in order to scale the data of an attribute so that it falls in a smaller range. Normalization is most essential when we are dealing with attributes in a different range, [15]. So they are normalized to bring all the attributes on the same range. In this proposed work normalize the high deviated values by using the following (10).

$$N = (i - mnv) \times (Mxr - Mnr) ./ (mxv - mnv) \quad (10)$$

where N is a normalized value of an i^{th} element, i is an element location, mnv is a minimum value in the point cloud data set, mxv is a maximum value in the data set, Mxr and Mnr is a maximum and minimum range value of the normalization function respectively.

D. 2D Predictive lossless Coding

The Predictive coding eliminates the redundancy of an eigenvalue of the pixels by extracting the new information between the pixels. This new information or prediction error defined as the difference between the actual and the predicted value of the adjacent pixels. In the 2D prediction coding, the function applied on a pixel in a left-to-right, top-to-bottom scanning position of an image. The predictive value formally calculated from the linear combination of n previous pixel [16].

The prediction error has been calculated by subtracting the accumulated value top pixels from the current pixel value, This prediction coding system consists of encoding and decoding phases with identical functions. The prediction error has been encoded by the Run-length encoding technique to generate the compressed data stream. It is a suitable algorithm to achieve the best compression result. The algorithm 2 discusses the step by step process of 2D prediction coding.

E. Run Length Encoding

Run-length encoding (RLE) is a very efficient form of lossless image compression that applys on a stream of bits having same value occurring consecutive times to reduce the length of stream. It encodes the sequence and it store only a single value and its count. Some pre-defined code must be substituted for an eliminated bits. Based on this code receiver can reconstruct Original signal [17]. Step by step procedure of Run-length encoding explained in algorithm 3.

IV. EXPERIMENTAL RESULTS

A. Metrics

The following metrics are used to measure the robustness and performance of the proposed algorithm based on tensor decomposition. The metrics of point cloud compression evaluated with Peak Signal to Noise Ratio and Mean Square Error (MSE)[12] expressed in (11).

$$\text{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[|I(i,j) - I'(i,j)|^2] \quad (11)$$

where M and N are the number of rows and number of the columns of a 3d point cloud image. I and I' are the original and reconstructed PCD image. i and j are the locations that are the specified location in the PCD image. The minimum MSE value indicates that the minimum difference between the original and reconstructed point cloud. PSNR value is calculated by the MSE value [12] shown in (12).

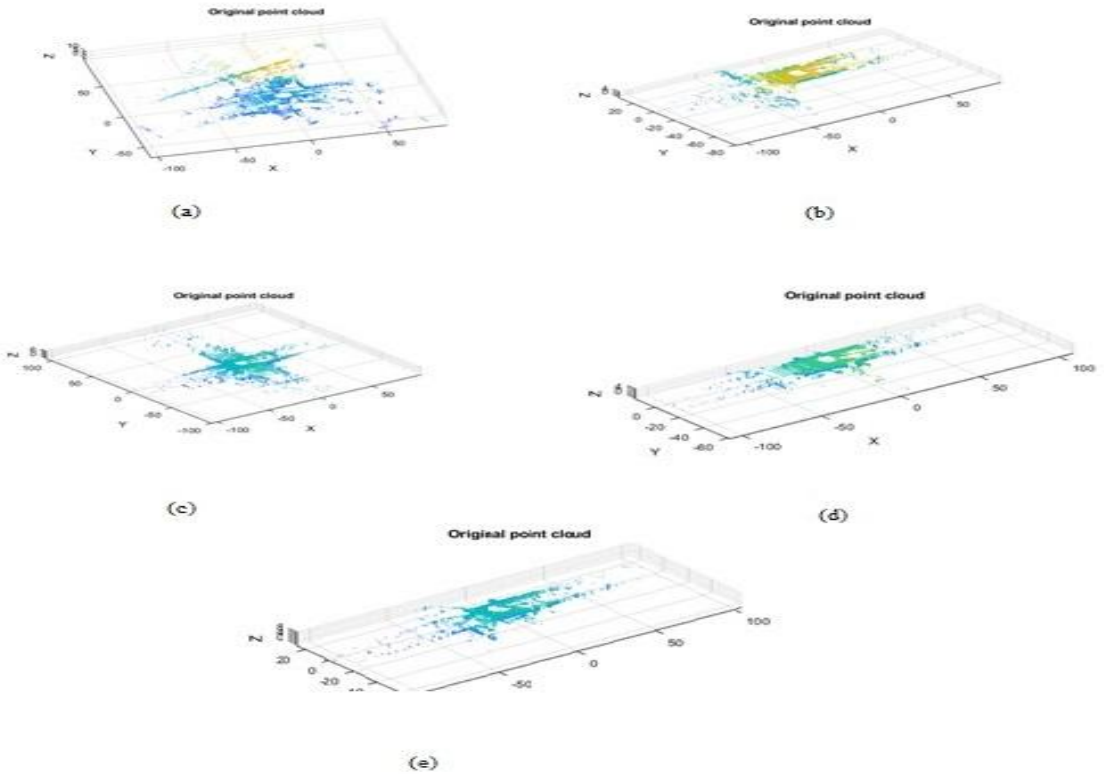

Fig. 4. Input 3D LiDAR point Cloud dataset for the Proposed Compression method. a) Scan b) Scan 3725 c) Scan 4868 d) Scan 5463 e) Scan 5563.

TABLE-1 : Input Point Cloud Image Description

S. No.	Name of the file	Number of Points	Original file Size
1	Scan	56677	1.08 MB
2	Scan 3725	59902	1.14 MB
3	Scan 4868	55503	1.05 MB
4	Scan 5463	58894	1.10 MB
5	Scan 5563	58110	1.12MB

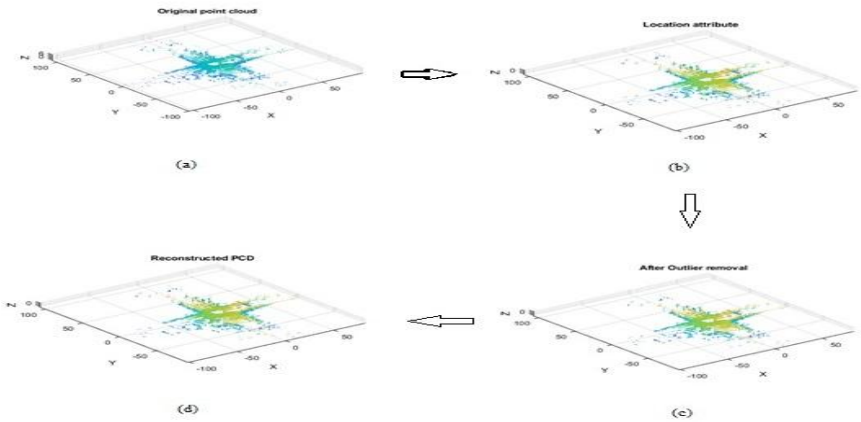

Fig. 5. Processing steps of the proposed algorithm. a) Original 3D Point Cloud image. b) Location attribute c) After outlier removal. d) Reconstructed 3D point cloud image.

TABLE-2: Performance evaluation of the proposed method

Point cloud name	No.of Points	No.of points after outlier Removal	Size of Core Tensor	Size of $\mathbf{U}^1$	Size of $\mathbf{U}^2$	Size of $\mathbf{U}^3$	DM	HD	Compressed File Size	Compression Ratio
Scan	56677	56372	3 x 3 x 9	3 x 3	3 x 3	18791 x 9	0	2.4645e-05	0.24 MB	4.4681
Scan 3725	59902	59611	3 x 3 x 9	3 x 3	3 x 3	19871 x 9	0	4.6677e-05	0.25 MB	4.4697
Scan 4868	55503	55339	3 x 3 x 9	3 x 3	3 x 3	18447 x 9	0	4.9847e-05	0.23 MB	4.4674
Scan 5463	58894	58690	3 x 3 x 9	3 x 3	3 x 3	19564 x 9	0	4.6797e-05	0.24 MB	4.4692
Scan 5563	58110	57994	3 x 3 x 9	3 x 3	3 x 3	19332 x 9	0	4.7634e-05	0.25 MB	4.4689

$$PSNR = 10log_{10}\left(\frac{255^2}{MSE}\right) \quad (12)$$

Sydney urban 3D LiDAR dataset [19] is used to evaluate the effectiveness of the proposed algorithm. Sydney datasets contain the sequences of 3D point cloud images with different angle appearance. In this evaluation, five different PCD images were selected from every 500 sequences of point cloud images that are shown in Figure 4 and the description tabulated in Table 1. These PCD data are tested in the windows environment using Matlab 2019b. The location attribute has been extracted from the point cloud and compressed by the proposed algorithm using different stages of the algorithm. The first unreliable points are identified and removed by the statistical subgroup method. Then inlier points are converted into a tensor format structure. HOSVD algorithm applied to the 3D tensor to split the data into vector blocks. These blocks are same in size of the original tensor, but in reduced space. Normalization and prediction process is the quantization process that has been applied to the vectors and then Run-length entropy encoding has been applied to produce the compressed image.

Figure 5 shows the step by step process of the proposed algorithm. One of the sample Sydney urban area point cloud image Scan 4868 shown in figure 5.a. The proposed compression work performs on an extracted location attribute displayed in figure 5.b. After the preprocessing step, the point cloud only contains the reliable points which are shown in figure 5.c. Figure 5.d displays the reconstructed PCD image on a receiving end. This proposed work supports lossless compression. Hence reconstructed point cloud perfectly matches with reliable PCD images. Table 2 displays the output of the HOSVD algorithm.

From table 2 it observed that the Distance Metric (DM) between original and reconstructed point cloud is zero, it indicates that the reconstructed PCD image is same as the original PCD image due to the lossless compression of the proposed method. The Hausdroff distance (HD) indicates that the maximum distance between the two PCD images. The Scan 3725 sample PCD image achieved better compression ratio than the other sample images.

V. CONCLUSION

This proposed method introduced the lossless 3D point cloud image compression based on the HOSVD method. It consists of four states to achieve the compressed data, such that first one: preprocessing state to remove the unreliable 3D points from PCD, second: Apply HOSVD transformation to reduce the dimension of the original tensor into core and factor matrices. This transformed structure further reduced the number of the bit representation by applying normalization, prediction and run-length encoding. This work applied to a Sydney Urban 3D LiDAR dataset to verify the performance of the efficiency of the proposed compression method. The experimental results show that low MSE and high PSNR value achieved to produce the lossless 3D point cloud compression with good compression ratio and high processing speed. The storage size has been reduced to one by fourth of its original 3D LiDAR point cloud image size.

REFERENCES

1. Arnaud Bletterer, Frederic Payan. M. Antonini, A. Meftah. “Point Cloud Compression using Depth Maps”.Society for Imaging Science and Technology, pp. 3DIPM-397.1 to 3DIPM 3971.5, 2016. DOI: 10.2352/ISSN 2470 -1173.2016.21.
2. P. Chmelar, Lubes Rejfek, L. Beran, Martin Dobrovolny. “ A Point Cloud Decomposition by the 3D level scanning for planes detection”. International Journal of Advanced and Applied Sciences. 4(11),pp. 121-126, 2017.
3. Dai Cheng-Qiu, C. Min, F.Xiao-yong. “ Compression Algorithm of 3D Point Cloud Data Based on Octree. The Open automation and control systems Journal, 7, pp.879 - 883. 2015.
4. M. Pavlovic, H.Gildhoff, K. Bastian, A. Ailamaki. “ Dictionary compression in point cloud data management”. SIGSPATIAL, 2017. Available: http://doi.org/10.1145/3139958.3139969.
5. Javier Navarrete, D. Viejo, M. Cazorla. “Compression and registration of 3D Point Clouds using GMMs”. Pattern Recognition Letters. 110, pp. 8-15, 2018. https:// doi.org/10.1016/j.patrec. 2018.03.017.
6. H. Houshiar, A. Nuchter. “ 3D Point Cloud Compression using Conventional Image Compression for efficient Data transmission”.
7. Jae – Kyun Ahn, Kyu-Yul Lee, Joe - Young Sim, Chang – Su Kim. “ Large scale 3D point Cloud Compression using Adaptive Radial distance Prediction in Hybrid co-ordinate domains”. IEEE Journal of Selected Topics in Signal Processing. V. 9.No. 3, April 2015 .

8. E.Gujral, R. Pasricha, Tianxiong Yang, Evangelos E. Papalexakis. "OCTEN: Online Compression-based Tensor Decomposition", arXive: 1807.01350v1, 3 July 2018.
9. J. Li, Z. Liu. "Compression of Hyper Spectral images using an accelerated nonnegative tensor decomposition", Published by De Gruyter Open,15, pp. 992-996. 2017.
10. J. Li, X. Zhang, T. Tran. "Point Cloud Denoising based on Tensor Tucker Decomposition", arXiv: 1902.07602v1, 20 February 2019.
11. M. Ishteva, L. D. Lathauwer, P. –A.Absil, S. V. Huffel. "Dimensionality Reduction For Higher-Order Tensors: Algorithms and Applications",
12. R. B. Ripoll, P. Lindstrom, R. Pajarola. " TTHRESH: Tensor compression for Multidimensional visual data", arXiv: 1806.05952v2, 6 March 2019.
13. H. H. Al-Mahmood, Z. Al-Rubaye. " Lossless Image Compression based on Predictive Coding and Bit-plane Slicing", International Journal of Computer Applications. V.93, No. 1, May 2014.
14. B. Du, M. Zhang, L. Zhang, R. Hu, D. Tao. "PLTD: Patch Based Low-Rank Tensor Decomposition for Hyperspectral Images", IEEE Transaction on Multimedia, DOI: 10.1109
15. Normalization available at https://www.geeksforgeeks.org/data-normalization-in-data-mining/ accessed on Aug. 2019.
16. Predictive lossless coding available at https://pdfs.semanticscholar.org/7adb/3a9c92af81a22c4d7ba4ad9aa8a41d33d484.pdf accessed Aug. 2019
17. Run-length Encoding, https://www.techiedelight.com/run-length-encoding-rle-data-compression-algorithm/ accessed Aug. 2019
18. Sydney Urban 3D LiDAR data set from http://www-personal.acfr.usyd.edu.au/a.quadros/objects4.html accessed on August 2019.
19. Basics of Tensor Decomposition available at https://en.wikipedia.org/wiki/Tucker_decomposition accessed on August 2019.

AUTHORS PROFILE

Dr. PL. Chithra, is the Professor in the Department of Computer Science at the University of Madras. She received her M.C.A and Ph.D. degrees from Alagappa University and University of Madras, Tamil Nadu, India respectively. Her area of interest includes Digital Image Processing, Pattern Recognition and Artificial Neural Networks. She is currently working on 3D Medical Image segmentation, 3D Point Cloud Compression, Real time Image Processing techniques, Deep Learning and Network Security. She is a Ph.D. and M. Phil research supervisor for guiding recent trend technologies since 2000. She has conducted several refresher courses and published more than 50 papers in national and international journals. She has been serving as Organizing Chair and Program Chair of several International conferences.

A. Christoper Tamilmathi, is the Ph.D. Research scholar at Department of Computer Science, University of Madras. She received M.C.A., degree from Madurai Kamaraj University and M.Phil. from University of Madras, Tamil Nadu, India. She has qualified State Level Eligibility Test (SET) and National Eligibility Test (NET). Her research area includes Digital Image Processing, 3D Point Cloud Compression and Artificial Neural Networks.