# Projection based dynamic point cloud compression using 3DTK toolkit and H.265/HEVC

Emil Dumic[1], Anamaria Bjelopera[2], Andreas Nüchter[3]
[1]Dept. of Electrical Engineering, University North, 104. brigade 3, 42 000 Varazdin, Croatia, edumic@unin.hr
[2]Dept. of Electrical Engineering and Computing, University of Dubrovnik,
Cira Carica 4, 20 000 Dubrovnik, Croatia, anamaria.bjelopera@unidu.hr
[3]Department of Informatics VII - Robotics and Telematics, Julius-Maximilians-University Würzburg,
Würzburg, Germany, andreas.nuechter@uni-wuerzburg.de

***Abstract*—This paper presents novel compression method for dynamic point clouds based on projections and the H.265/HEVC video coder. We used 3DTK – The 3D Toolkit to create equirectangular projection images and x265 as H.265/HEVC coder, to compress and extract created projection images. Compression was introduced in 3DTK generated projection image size (different pixel size, but compatible with later video compression), as well as lossless/lossy video compression. Visual inspection shows better results for compression only using different projection resolution, with lossless video compression. Lossy video compression adds noise and creates additional points, resulting in lower visual quality.**

***Index Terms*—3DTK, dynamic point clouds, compression**

## I. INTRODUCTION

Point cloud technology is being addressed as one promising solution for different static and dynamic visualization applications. Other purposes include geographic information systems, 3D models of an environment, medical imaging, etc. Recently, the JPEG standardization committee created an initiative called JPEG Pleno, which aims to provide a standard framework for representing new imaging modalities, such as texture-plus-depth, light field, point cloud (PC), and holographic imaging [1].

Due to the data size of a static and dynamic point cloud, which can be very big, it is important to efficiently store its data for transmission and later inspection or visualization. Different compression methodologies have been proposed lately, which could be generally divided into octree-based methods, graph-based methods and projection-based methods. Particularly, projection-based methods are suitable for dynamic point clouds, because existing video coders achieve higher compression ratios due to of temporal redundancies between adjacent point clouds.

Several methods for projection-based static point cloud coding have been proposed recently. In [2] authors propose a best effort projection based compression method for point clouds. To take advantage of the well-developed 2D compression algorithms, the regularized 3D point cloud is projected onto specified planes as different views while position information and related attributes are preserved. Joint depth- and color-dependent block-wise prediction was also utilized to further reduce the inter-view redundancy between the projected 2D images. Point clouds are then successfully reconstructed via a corresponding decoding process. In the paper [3] the authors present novel point cloud reduction methods based on panorama images. Several projection based algorithms have been used: equirectangular projection, cylindrical projection, Mercator projection, rectilinear projection, Pannini projection, stereographic projection and Albers equal-area conic projection. Different compressions are achieved by using distinct resolution for panorama images. It is shown that the reduced point clouds are ideally suited for feature based registration on panorama images. In [4] the same authors present the use of conventional image based compression methods for 3D point clouds. They map the point cloud onto panorama images, using equirectangular projection, to encode the range, reflectance and color value for each 3D point. Results of several lossless compression methods and the lossy JPEG on point cloud compression were presented. Lossless compression methods are designed to retain the original data, while lossy compression methods sacrifice the details for higher compression ratio. Projection-based point cloud coding was calculated using "3DTK – The 3D Toolkit"[5]. Recently, a new compression algorithm for dynamic point clouds based on projections has been tested in [6].

This paper is organized as follows: Section II describes algorithm for projection based dynamic point cloud compression. Section III presents results and discussion and finally section IV gives the conclusions and explains further research activities.

## II. ALGORITHM FOR PROJECTION BASED PC COMPRESSION

In this section we will describe programs and parameters used to generate video streams from dynamic point clouds. To process point cloud to projection, as well as projection to point cloud, we used 3DTK [5]. After every point cloud was transformed to projection, we used FFmpeg [7] and x265 coder for further processing, to combine several projections in one video sequence. The same tool was also used to extract projections from each video sequence. Concisely, 3DTK was

2nd International Colloquium on Smart Grid Metrology (SMAGRIMET)
Split, Croatia, April 9-12, 2019.

firstly used to create projections from PCs. FFmpeg was then used to create video sequence from several projections and extract projections back from video sequence. Finally, 3DTK was used to generate PCs from projections.

We used dynamic point clouds that are given at the URL [8], i.e. first 20 point clouds for ”longdress” (longdress_vox10_1051.ply - longdress_vox10_1070.ply) from 8i Voxelized Full Bodies (8iVFB v2). For later 3DTK processing, they were firstly centered and normalized in the range 0-1 using Meshlab [9]. Also, they were converted from ASCII to binary format for more fair comparison regarding compression ratio.

3DTK – The 3D Toolkit [5] (3DTK) provides algorithms and methods to process 3D point clouds. It includes automatic high-accurate registration (6D simultaneous localization and mapping, 6D SLAM) and other tools, e.g., a fast 3D viewer, plane extraction software, etc. In our experiments, we used the scan_to_panorama application to create projections from PC and panorama_to_scan to create PC from projections. Specific parameters for mentioned applications are given in Tab. I.

The FFmpeg tool [7] is the leading multimedia framework, able to decode, encode, transcode, mux, demux, stream, filter and play many different audio and video formats. We used x265, part of ffmpeg, which is H.265/HEVC video coder. Specific used parameters are as follows: lossless encoding for 5 projection resolutions described in Tab. I, 3 lossy CRF (Constant Rate Factor) for projection resolution 7680x4320 pixels (CRF 5, 10 and 15-lower values result in better quality), pixel format ”gbrp” for color and combined range projections, pixel format ”gray” for separate range projections, preset ”slower”, one pass, input and output 30 fps. Effectively, this creates 8 different compression types (5 lossless and 3 lossy), and in each compression type 2 or 4 video streams. 2 video streams are for RGB color and combined RGB range projection images, while 4 video streams are for RGB color and separate range projection images (3 images per one PC).

A final step might be included, to obtain visually better quality. Firstly, normals for point sets were computed using 16 neighbor points (with other default values) to estimate normals. Afterwards, the screened Poisson surface reconstruction algorithm [10] was used, with reconstruction depth 12 (octree depth) and other default values. The poisson surface reconstruction algorithm was used to create better visual quality of decompressed point clouds. Those point clouds were reconstructed in Meshlab [9] (version 2016 from 23.12.2016).

## III. RESULTS AND DISCUSSION

In this section we will present some results that were made using the procedure described in the previous section. Fig. 1 shows the output PC percentage compared to the input PC number of points, for previously described compression with two video sequences (color and combined RGB range files). Table III shows the compression ratios for the eight previously described cases.

Decompressed point cloud are generally compared with the original point cloud using some objective measure. Generally, objective measures are divided in geometric and projection based objective measures [11]. For example, in [12] the authors compared correlation between different geometric based objective measures and subjective quality of different static point clouds. In [13] the authors proposed novel methodology for static voxelized point cloud quality based on their projections. However, none of those measures have been tested on dynamic point clouds, so their correlation with subjective evaluation has yet to be determined. Because of that, in this paper we have visually inspected decompressed point clouds, to be able to draw conclusions. Fig. 2 shows decompressed tenth point cloud (from longdress_vox10_1060.ply), using different resolutions. In this figure, we showed results for resolutions 1920 x 1080, 2048 x 2160, 4096 x 2160, 4096 x 4320 and 7680 x 4320 with lossless compression, 7680x4320 with lossy compression (crf 5) and Poisson surface reconstructed [10] point clouds with resolutions 1920x1080 and 7680x4320. We did not show results for lossy compression with crf 10 and crf 15 due to the low visual quality - it is similar or somewhat worse that with crf 5. Specific camera settings in Meshlab can be find in Tab. II.

TABLE I
SPECIFIC PARAMETERS IN 3DTK SCAN_TO_PANORAMA AND PANORAMA_TO_SCAN

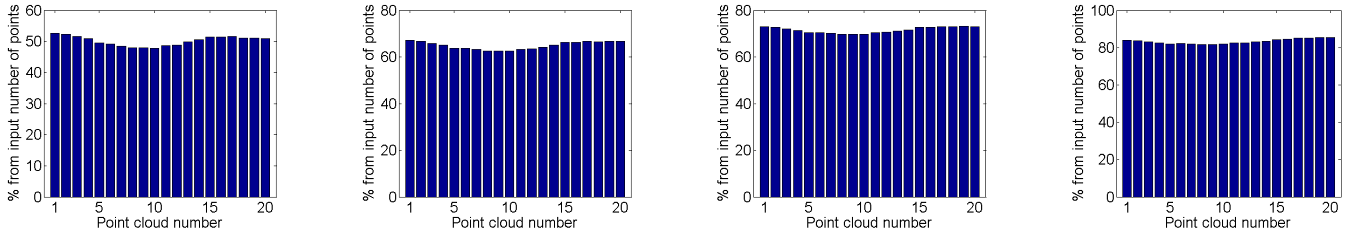
| | scan_to_panorama | panorama_to_scan |
|---|---|---|
| Projection type | equirectangular | equirectangular |
| Projection width x height (pixels) | 1920 x 1080<br>2048 x 2160<br>4096 x 2160<br>4096 x 4320<br>7680 x 4320 | Input defined |
| Vertical scanning angle | -100 100 | -100 100 |
| Horizontal scanning angle | default (0-360) | default (0-360) |
| Scan method | Full | / |
| Range file type | 3 grayscale .png images<br>or<br>1 combined RGB .png image | 3 grayscale .png images<br>or<br>1 combined RGB .png image |
| Color file type | Color .png RGB image | Color .png RGB image |

TABLE II
MESHLAB CAMERA SETTINGS

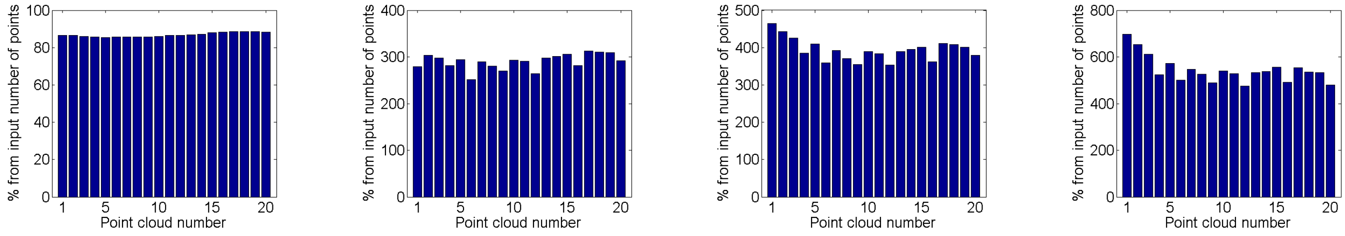
| **Parameter** | **Value** |
|---|---|
| Camera Translation | [4.64991e-5 8.24928e-5 -1.14619 1] |
| Lens Distortion | 0 0 |
| Camera Type | 0 |
| Rotation Matrix | [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1] |
| Viewport Px | 1585 1051 |
| Pixel Size Mm | 0.0369161 0.0369161 |
| Focal Mm | 33.6007 |
| Center Px | 792 525 |
| View Settings Near Plane | 1.66509 |
| FarPlane | 5.18794 |
| TrackScale | 0.909327 |

Fig. 1 shows the percentage between the output number of points and the original number of points, for each of the 20 tested point clouds. Generally, results are stable for lossless video compression, and become unstable for lossy video compression, for all crf factors (5, 10 and 15).

From Tab. III it can be seen that, in the lossless case, as expected, a higher projection resolution will keep a higher number of points in decompressed point cloud. Also, higher resolution means lower compression ratio. However, in the lossy case, we generated decompressed point clouds with more



(a) *1920x1080* (b) *2048x2160* (c) *4096x2160* (d) *4096x4320*



(e) *7680x4320* (f) *7680x4320_crf5* (g) *7680x4320_crf10* (h) *7680x4320_crf15*

Fig. 1. Output point cloud number of points - percentage compared to the input point cloud. Case with 2 video sequences (i.e. color in first and combined range in second RGB file)

points than the original ones. This is explained because lossy video compression introduced additional noise in the projection images, resulting in a noisy decompressed point cloud, Fig. 2f. This means that currently the presented lossy video compression is not well suited for point cloud compression, only lossless mode gives satisfactorily results. Also, from Fig. 2 it is seen that Poisson surface reconstruction deals with missing points from the compression process itself and gives better visual quality, comparing with raw decompressed point cloud.

A possible further step, not done in the paper, might be sampling of surface reconstructed point cloud, to obtain final raw points. This might be analogue to point cloud interpolation as a post processing step. This step is done in the final point cloud, or possibly in the projection image, where many interpolation methods already exist [14].

## IV. CONCLUSION

In this paper we presented a novel dynamic point cloud compression based on equirectangular projection method using 3DTK and H.265/HEVC video compression. Results showed better visual quality with lossless video compression, while lossy compression introduced additional noise in projection images, resulting a in lower visual quality. Additionally, the Poisson surface reconstruction algorithm filled some of the missing points that were not present in the decompressed point clouds, especially with lowest tested projection resolution (1920x1080 pixels). As a result, better visual quality was obtained for the same projection resolution.

Further research includes sampling of surface reconstructed point clouds, to obtain final raw points. Similarly, some of the interpolation methods might be used. Also, the presented methodology is tested using the whole dataset for the tested point cloud "longdress" (with 300 point clouds), other point clouds, different projection types, etc.

## REFERENCES

[1] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG Pleno: Toward an Efficient Representation of Visual Reality," *IEEE MultiMedia*, vol. 23, no. 4, pp. 14–20, Oct 2016.

[2] L. He, W. Zhu, and Y. Xu, "Best-effort projection based attribute compression for 3d point cloud," in *2017 23rd Asia-Pacific Conference on Communications (APCC)*, Dec 2017, pp. 1–6.

[3] H. Houshiar, D. Borrmann, J. Elseberg, and A. Nüchter, "Panorama based point cloud reduction and registration," in *2013 16th International Conference on Advanced Robotics (ICAR)*, Nov 2013, pp. 1–8.

[4] H. Houshiar and A. Nüchter, "3d point cloud compression using conventional image compression for efficient data transmission," in *2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT)*, Oct 2015, pp. 1–8.

[5] http://threedtk.de, "3DTK The 3D Tookit," .

[6] Emin Zerman, Pan Gao, Cagri Ozcinar, and Aljoscha Smolic, "Subjective and objective quality assessment for volumetric video compression," in *Fast track article for IST International Symposium on Electronic Imaging 2019: Image Quality and System Performance XVI proceedings*, 2019.

[7] https://www.ffmpeg.org/, "FFmpeg," .

[8] https://jpeg.org/plenodb/, "Jpeg Pleno database," .

[9] http://www.meshlab.net/, "Meshlab," .

[10] Michael Kazhdan and Hugues Hoppe, "Screened Poisson Surface Reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, July 2013.

[11] E. Dumic, C. R. Duarte, and L. A. da Silva Cruz, "Subjective evaluation and objective measures for point clouds state of the art," in *2018 First International Colloquium on Smart Grid Metrology (SmaGriMet)*, April 2018, pp. 1–5.

[12] E. Alexiou, M. V. Bernardo, L. A. da Silva Cruz, L. Gotal Dmitrovic, R. Duarte, E. Dumic, T. Ebrahimi, D. Matkovic, M. Pereira, A. Pinheiro, and A. Skodras, "Point cloud subjective evaluation methodology based on 2d rendering," in *Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, 2018.

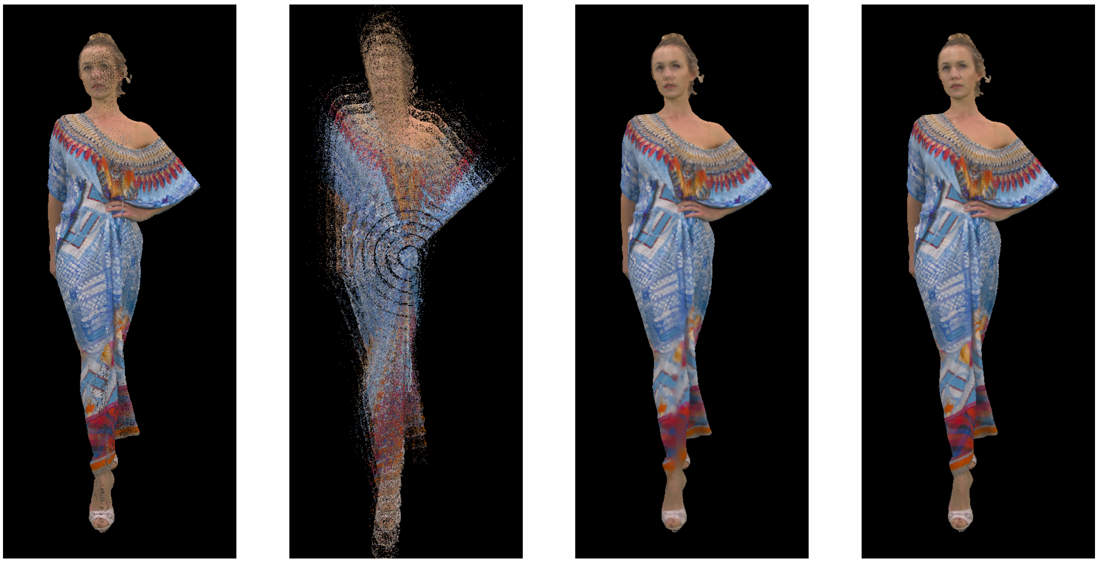
[13] Eric M. Torlig, Evangelos Alexiou, Tiago A. Fonseca, Ricardo L. de Queiroz, and Touradj Ebrahimi, "A novel methodology for quality assessment of voxelized point clouds," in *Proc.SPIE*, 2018, vol. 10752, pp. 10752 – 10752 – 17.

[14] E. Dumic, S. Grgic, and M. Grgic, "Hidden influences on image quality when comparing interpolation methods," in *2008 15th International Conference on Systems, Signals and Image Processing*, June 2008, pp. 367–372.

TABLE III
COMPRESSION RESULTS FOR PREVIOUSLY DESCRIBED CASES

| | 1920x1080 | 2048x2160 | 4096x2160 | 4096x4320 | 7680x4320 | 7680x4320_crf5 | 7680x4320_crf10 | 7680x4320_crf15 |
|---|---|---|---|---|---|---|---|---|
| color (bytes) | 43208605 | 62555497 | 73232555 | 91236063 | 99323578 | 133564028 | 101483407 | 75766885 |
| range (bytes) | 21378775 | 31969752 | 38674480 | 49647651 | 55031165 | 83228018 | 83228018 | 83228018 |
| range1 (bytes) | 6547 | 8242 | 10157 | 14441 | 21231 | 18475 | 18128 | 17943 |
| range2 (bytes) | 5594345 | 9136746 | 11622895 | 16018200 | 18344625 | 32576588 | 22721435 | 11120481 |
| range3 (bytes) | 15351130 | 22191418 | 26341527 | 33103648 | 36273168 | 74737537 | 61859230 | 50298493 |
| 4 video sequences (color+range1,2,3) (bytes) | 64160627 | 93891903 | 111207134 | 140372352 | 153962602 | 240896628 | 186082200 | 137203802 |
| % from input size (254408640 bytes) | 25.22 | 36.91 | 43.71 | 55.18 | 60.52 | 94.69 | 73.14 | 53.93 |
| 2 video sequences (color+range) (bytes) | 64587380 | 94525249 | 111907035 | 140883714 | 154354743 | 216792046 | 184711425 | 158994903 |
| % from input size (254408640 bytes) | 25.39 | 37.15 | 43.99 | 55.38 | 60.67 | 85.21 | 72.60 | 62.50 |

(a) *1920x1080* (b) *2048x2160* (c) *4096x2160* (d) *4096x4320*



(e) *7680x4320* (f) *7680x4320_crf5* (g) *1920x1080_poisson* (h) *7680x4320_poisson*

Fig. 2. Decompressed tenth point cloud (longdress_vox10_1060.ply). Case with 2 video sequences (e.g. color in first and combined range in second RGB file)