Dynamic Point Cloud Geometry Compression via Patch-wise Polynomial Fitting

Yingzhan Xu, Wenjie Zhu, Yiling Xu；Zhu Li
Shanghai Jiao Tong University, China；University of Missouri, Kansas City, USA

Abstract

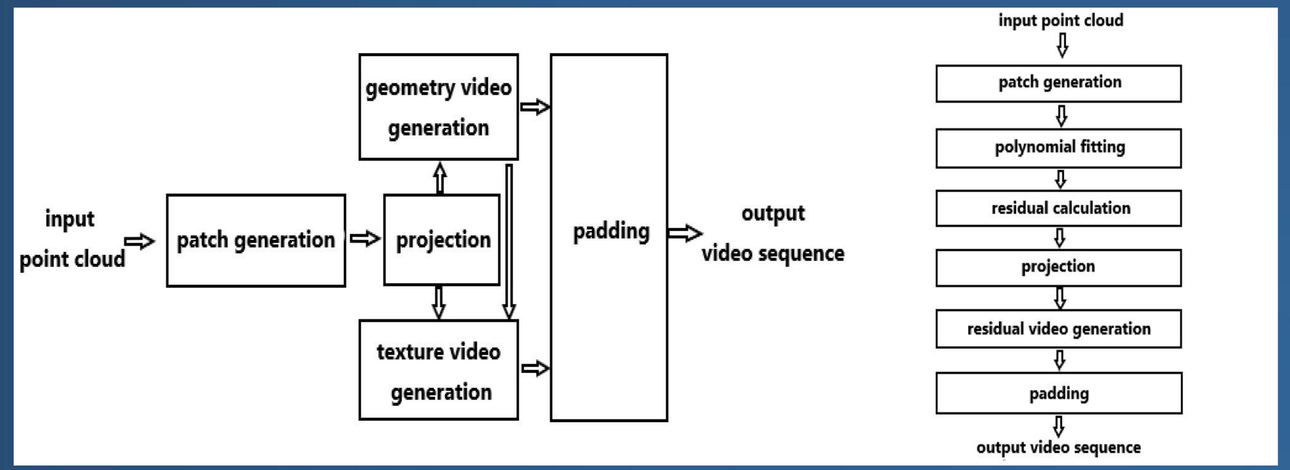
With the boosting requirements of realistic 3D modeling for immersive applications, advent of the newly-developed 3D point cloud has attracted great attention. Frankly, immersive experience using high data volume affirms the importance of efficient compression. Inspired by the video-based point cloud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

Introduction

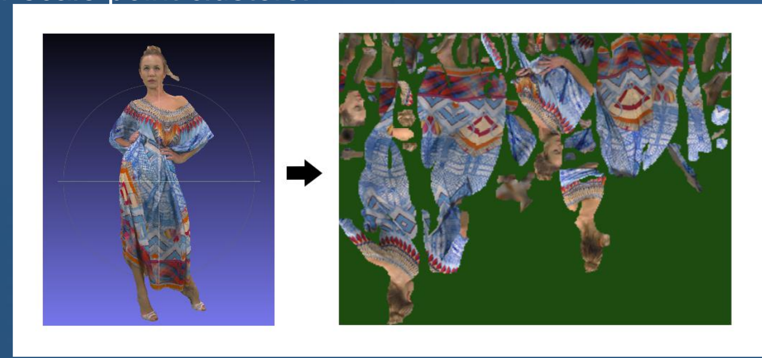
- Point cloud has succeeded in recording and describing three-dimensional objects and scenes, based on its newly-developed 3D media format that assists in recording the point geometry information related to attribute information.
- A high precision of the point cloud offers a high data volume, and when point cloud information is transmitted and processed, the data efficiency usually does not perform well. Moreover, the irregular and scattered point cloud data increases the complexity of the processing algorithms and takes up a lot of computing space. Therefore, a proper compression algorithm is quite essential in the point cloud applications to overcome these problems.
- Numerous studies have explored point cloud compression. Geometry and attribute information are the main properties of the point cloud, which usually coded separately. And the geometry information is the basis for attribute rendering
- For attribute compression, the genetic algorithm based intra prediction is introduced. Moreover, a global projection algorithm that maintains a correlation of the color attribute associated with the nearby points in the 3D space is also introduced for the accuracy of the attribute.
- For geometry cpmpression, the octree structure can be utilized to separate the point clouds and then entropy encode the corresponding leaf nodes to compress the geometry information.Moreover, a binary tree structure can also be used to segregate the unorganized points into block structure and eliminate the redundancy of geometry information through residual coding.Besides, the points can be clustered into a series of hierarchical point clusters and traverse each cluster from top to bottom. The residual between the traversed point and the top point will be encoded to perform the geometry compression. Moreover, a global projection algorithm can convert the 3D data into 2D data and compress geometry information using 2D codecs.
- The Moving Picture Experts Group (MPEG) organization has proposed a standardized compression test model, V-PCC(Video based Point cloud compression) for the dynamic point cloud. The main principle is to convert 3D point clouds to 2D video sequences and employ existing video coding algorithms, such as HEVC, for further compression.

Measures

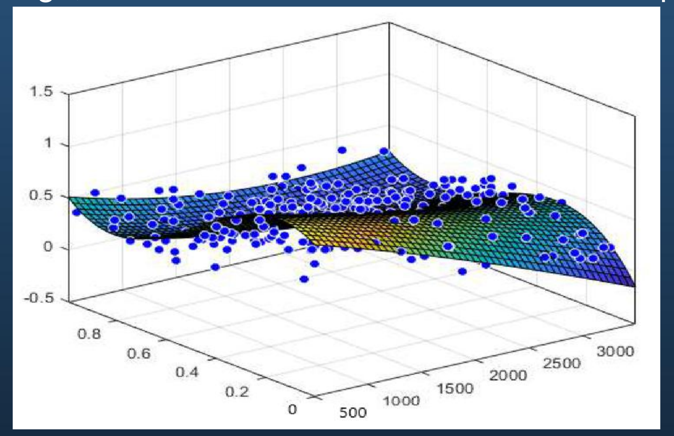
- We utilize existing video codecs to compress the geometry and the texture information of dynamic point cloud sequences. At first, the 3D object is divided into patches of different sizes mainly based on the normal vectors of the points that belong to these patches. Then these patches are projected in different directions according to the main normal vector of each patch to generate a padded depth map. After the projecting and padding processes, two video sequences that record the geometry and texture information from point cloud are generated and compressed using existing video codecs.



- The patch generation process is to reduce the amount of data involved in every process and to improve the efficiency of compression obtained by dividing the complete point cloud objects into smaller scale point clusters.



- The projection process aims at mapping the extracted patches onto a 2D grid for video coding. Via projection, the point cloud object is transformed into depth maps. However, the depth values are directly obtained through a spatial coordinate projection of the point cloud. As the depth values are high, these are not convenient for video compression.



- Polynomial fitting is a surface fitting method to represent each fitting polynomial using several parameters. Through least squares, we can fit these parameters by constantly inserting real point values in our assumed expressions so as to minimize the sum of the squares of the differences between the real depth values and the estimated depth values. In this way, we consider that the three-dimensional coordinates approximately satisfy the geometry relationship.

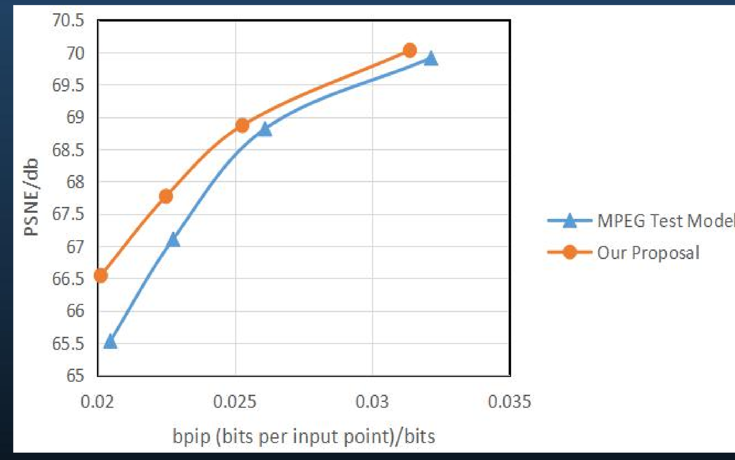
Measures (continued)

- As described above, patches of different sizes are generated after patch generation. Generally, these point cloud segments are quite regular and we can use polynomial fitting to predict the point coordinate information. The number of polynomial terms depends on the complexity of the point cloud segment. Generally, a segment is small and a polynomial with 9 terms is enough for prediction.
- After polynomial fitting, we calculate the residuals between the predicted value and the original value. Further, we project residuals to generate residual video sequence, and the new video sequence is compressed instead of the original depth values to conspicuously reduce the data volume of depth map and improve the compression ratio. In addition, we compress the fitting coefficients by employing the open source PAQ project to encode these parameters owing to its lossless coding. Figure 4 shows the method used to generate the video sequence.
- The residual values are decoded through the video decoder on the decoding side, and the predicted depth values are calculated by satisfying the expression described using the fitting parameters. Moreover, we reconstructed the original depth value by calculating the difference between the predicted depth value and the corresponding residual value. Through this process, we reconstruct the point cloud object using the calculated original depth values.

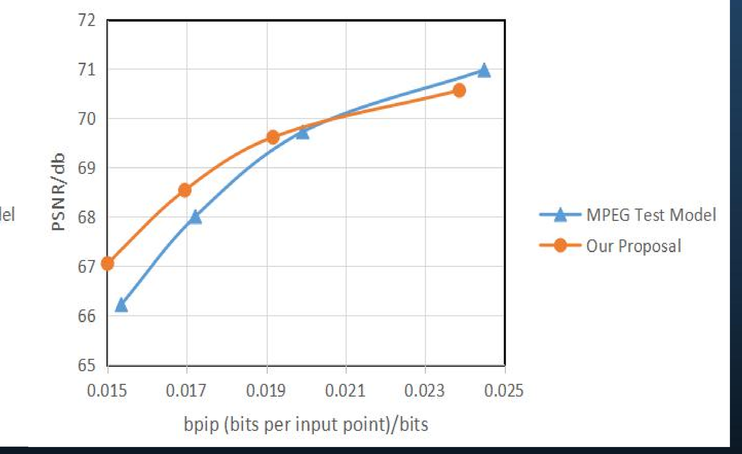
Results

- We perform geometry experiments on four dynamic point cloud sequences. Moreover, we test the compression performance on 32 frames of dynamic point clouds in each point cloud sequence. We compare the compression effect of the V-PCC test model and the proposed method on the same data. We use the bit per input point (bpip) after compression to represent the bit rate cost in compression. The peak signal-to-noise ratio (PSNR) is calculated based on the widely-applied point-to-point distortion in MPEG. We test 4 data sequence provided by 8i company.

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

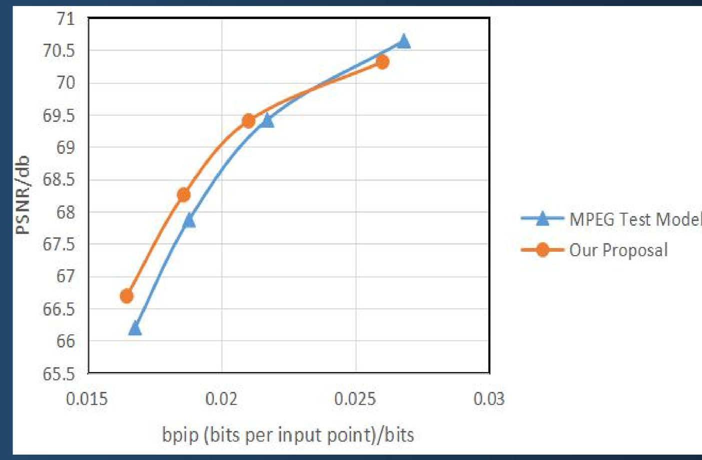


Geometry R-D Curve for dataset 'Longdress'

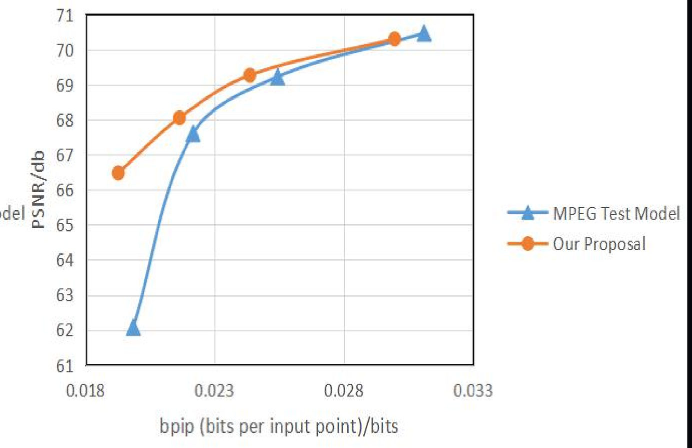


Geometry R-D Curve for dataset 'Soldier'

Results (continued)



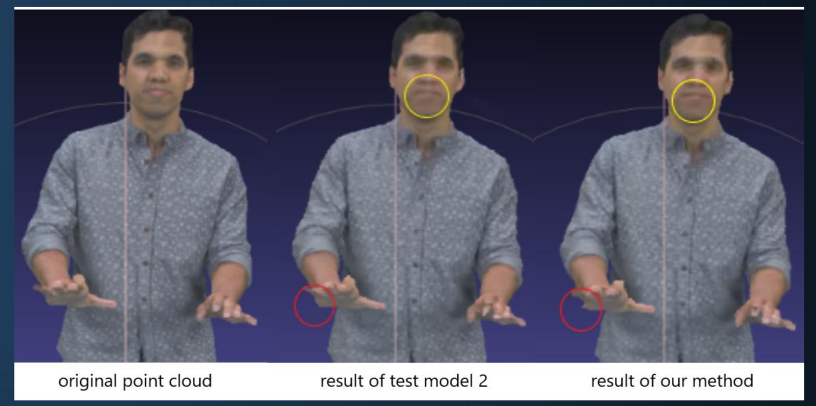
Geometry R-D Curve for dataset 'Red'andbalck



Geometry R-D Curve for dataset 'Boot'

- As shown in the rate-distortion curves (R-D curves), the proposed algorithm improves the performance of a V-PCC, reduces the bit cost, and maintains the quality. When the bpip is low, the bpip of our proposal performs better than that of V-PCC with the same PSNR. An improvement is achieved by decreasing the depth values by polynomial fitting. Moreover, our method's PSNR is little lower when the test is on high bpip.
- When the compression is close to distortion-free, the PSNR will be more sensitive to the errors introduced by the fit. As a result, the effect of PSNR is more pronounced than the reducing of bpip. This is shown by the R-D curve. With the test of the database "reaandblack_vox10", the optimization effect can also be maintained when bpip is high.
- We used HM16.16 decoder as a video compression tool and set the code model to All-intra. Our operating environment was a Windows 10 64-bit operating system with a memory RAM of 8.0 GB and an Intel (R) Core (TM) i5-4590 CPU processor. The program was implemented using C++ and Matlab2015a programming languages.

Discussion



original point cloud　result of test model 2　result of our method

- We chose a frame of the database "loot_vox10" as a test sample. From the results of our proposal, we obtained a better performance that maintains the geometry and texture information in the yellow circle. And for some positions, some distortion arose in the red circle. The result shows that we can render and code attributes even better in some locations.