PATCH-AWARE AVERAGING FILTER FOR SCALING IN POINT CLOUD COMPRESSION

Keming Cao1, Yi Xu2, Pamela Cosman1
1 Dept. of Electrical and Computer Engineering, UC San Diego; 2 Ovili Inc., Beijing

Abstract

With the development of augmented reality, the delivery and storage of 3D content have become an important research area. Among the proposals for point cloud compression collected by MPEG, Apple’s Test Model Category 2 (TMC2) achieves the highest quality for 3D sequences under a bitrate constraint. However, the TMC2 framework is not spatially scalable. In this paper, we add interpolation components which make TMC2 suitable for flexible resolution. We apply a patch-aware averaging filter to eliminate most outliers which result from the interpolation. Experimental results show that our method performs well both on objective evaluation and subjective quality.

Motivation

3D media gets increasing attention
• Apps such as Pokémon Go, IKEA Place, Google Streetview, Amazon AR shopping
• Companies such as Google and Apple release platforms for smartphone-based AR/VR
• Investment reaches a new record high in 2017

Autonomous driving brings point cloud back
• Easy to capture from depth sensor
• Easy to concatenate

Compression standard is under research in MPEG
• Winner for point cloud sequence compression does not consider spatial scalability

TMC2 Framework

Packing:
• Fitting all projected patches onto an image of pre-fixed size (L_x, L_y)
• Occupancy map to record whether a pixel is occupied or not
• Pixels between patches are filled with value of nearby pixels
• Packing image is shown below (for texture, for geometry and occupancy map)

Encoding:
• Treat projected image sequence as 2D video codec
• Video encoder could be any 2D codec
• Occupancy map is encoded with arithmetic coding

Patch-aware Averaging Filter

• Add scale-down module at encoder side and scale-up module at decoder side
• Simple scaling module will cause noise at edge of path

Evaluation

Evaluation metric:
• 2D: PSNR
• 3D: MSE defined below

\[ \text{MSE} = \frac{1}{L_x L_y} \sum_{i=1}^{L_x} \sum_{j=1}^{L_y} (F(i,j) - S(i,j))^2 \]

\[ \text{MSE} = \frac{1}{N} \sum_{i=1}^{N} e(i) \]

\[ e(i)\text{ is error vector from point } p_i\text{ in } A. N\text{ is number of points in } A. \]

Interpolation combination:
• N2N2: Nearest neighbor at the encoder and decoder
• N2B2: Nearest neighbor at the encoder and bicubic at the decoder
• B2N2: Bicubic at the encoder and nearest neighbor at the decoder
• B2B: Bicubic at encoder and decoder

Filter parameters:
• \( \text{filter} \in \{0, 10, 20, 30, 40\} \)

Scale factors:
• 1.25, 1.5, 1.75, 2.0, 2.25, 2.5

Quantization parameters:
• \( \text{encoder} \in \{6, 8, 10, 12, 14\} \)

\[ \Delta_{\text{PSNR}} = \frac{\text{PSNR} - \text{PSNR}_1}{\text{PSNR}_1} \]

\[ \Delta_{\text{MSE}} = \frac{\text{MSE} - \text{MSE}_1}{\text{MSE}_1} \]

\[ \Delta_{\text{MSE}} \uparrow \Rightarrow \text{denotes smaller MSE with filter} \]

\[ \text{Example for sequence } \text{dancer} \text{with different } S, Q, F, \text{and } \text{filter} \]

\[ \Delta_{\text{PSNR}} = \frac{\text{PSNR} - \text{PSNR}_1}{\text{PSNR}_1} \]

\[ \Delta_{\text{MSE}} = \frac{\text{MSE} - \text{MSE}_1}{\text{MSE}_1} \]

\[ \Delta_{\text{MSE}} \uparrow \Rightarrow \text{denotes smaller MSE with filter} \]

\[ \text{Compare different } F \]

A-M: \( \Delta_{\text{MSE}} = \text{MSE}_1 - \text{MSE}_2 \), \( A-W: \Delta_{\text{MSE}} = \text{MSE}_1 - \text{MSE}_5 \), \( A-L: \Delta_{\text{MSE}} = \text{MSE}_1 - \text{MSE}_6 \)

\[ \text{The averaging filter generates slightly lower MSE than the median filter and a weighted filter. The experimental results on 2D PSNR and 3D MSE as well as visual inspection of image pairs show that our method performs well both on objective evaluation and on subjective visual quality.} \]