**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2019** **m49588**

**July 2019, Gothenburg, Sweden**

|  |  |
| --- | --- |
| **Source** | **Samsung Electronics** |
| **Status** | **Input contribution** |
| **Title** | **[V-PCC] EE2.6 Report on mesh coding with V-PCC** |
| **Author** | Esmaeil Faramarzi, Keming Cao, Hossein Najaf-Zadeh, Rajan Joshi, Madhukar Budagavi, Sungryeul Rhyu, and Jaeyeon Song |

# Abstract

This contribution is the report of EE2.6 on mesh coding with V-PCC. In the MPEG-126 meeting, Samsung proposed an architectural extension to V-PCC to support mesh coding [1]. This contribution shows the implementation results of the proposed framework compared to a publically available mesh codec software.

# Proposal

Figures 1 and 2 show the extensions to the V-PCC encoder and decoder to support mesh encoding and mesh decoding, respectively, as proposed in [1]. In the encoder extension (Figure 1), the input mesh data is demultiplexed into vertex coordinate+attributes and vertex connectivity. The vertex coordinate+attributes data is coded using MPEG-I V-PCC, whereas the vertex connectivity data is coded as auxiliary data. Both of these are multiplexed to create the final compressed output bitstream. Vertex ordering is carried out on the reconstructed vertex coordinates at the output of MPEG-I V-PCC to reorder the vertices for optimal vertex connectivity encoding. In the decoder, the input bitstream is demultiplexed to generate the compressed bitstreams for vertex coordinates+attributes and vertex connectivity. The vertex coordinates+attributes data is decompressed using MPEG-I V-PCC decoder. Vertex ordering is carried out on the reconstructed vertex coordinates at the output of MPEG-I V-PCC decoder to match the vertex order at the encoder. The vertex connectivity data is also decompressed and everything is multiplexed to generate the reconstructed mesh.

# Experimental results

We implemented the proposed framework on top of TMC2 v5.0 using the TFAN mesh coding algorithm from the MPEG SC3DMC standard (Scalable Complexity 3D Mesh Coding) [2] for coding vertex indices (mesh connectivities). TFAN compresses vertex indices losslessly based on partitioning mesh faces into a set of triangle fans, as shown in Figure 3.

To test out the implementation, we used the first frame of Basketball, Dancer, Exercise, and Model meshes from the Owlii mesh datasets. We mapped the geometry values into the 10-bit range and voxelized them. Then we did re-meshing of the new 3D points using the MeshLab software. The generated meshes have no color since the original meshes have texture atlas which cannot be converted to point colors in a straightforward way. Figure 4 shows a sample mesh generated from the original floating-point mesh.

Picture1

Figure 1: V-PCC extension for mesh encoding.

Picture2

Figure 2: V-PCC extension for mesh decoding.

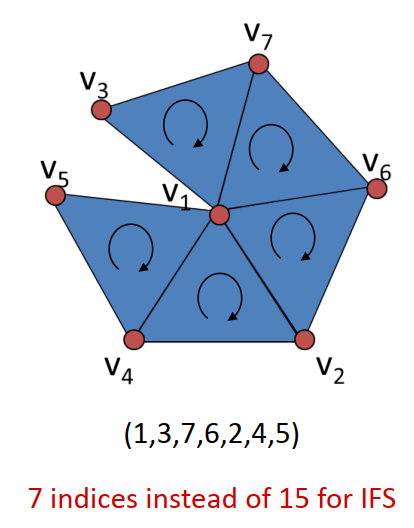


Figure 3. An example of TFAN partitioning of mesh faces. Here, 7 indices are encoded instead of 15 indices in the 5 triangles.

Figure 4. A mesh from the Owlii dataset (left) and the generated 10-bit voxelized mesh used in our simulations (right)

For the voxelized Owlii meshes, the input BPP (bits per point) or BPV (bits per vertex) is calculated as:

**Input BPP** **= 30 bits for geo + N bits per vertex index x num\_vertex\_indices / num\_vertices**

where . Table 1 shows the input BPP values for 1 frame of the voxelized mesh.



Table 1. Input BPP values for voxelized 10-bit Owlii meshes

Table 2 shows lossless simulation results of TMC2v5+TFAN versus near-lossless (using the highest-quality compression level) simulation results of a publicly available codec for 1 frame of the voxelized 10-bit Owlii meshes which contain geometry and vertex indices but no color. In case of TMC2v5+TFAN, both the encoded sizes in bytes and the output BPP values are shown in Table 2a for individual components as well as the total data written into disk. Unfortunately, the public codec does not print out the individual sizes for geometry and vertex indices. Therefore, Table 2b only shows the bytes and BPP values for the total data written into disk in case of the public codec. The results show the superiority of TMC2v5+TFAN over the public codec for the Owlii meshes. Theoutput BPP is calculated as:

**Output BPP = total size of encoded bitstream / num\_vertices**



(a)



(b)

Table 2. Lossless simulation results of TMC2v5+TFAN (a) versus near-lossless (using the highest-quality compression level) simulation results of a public codec (b) for 1 frame of voxelized 10-bit Owlii meshes

We also generated meshes from the CTC point clouds, snapshots of which are shown in Figure 5. The generated meshes have exactly the same geometry and color values as the CTC point clouds. The generated faces are not perfect as there are some holes in them, but they are sufficient for our test purpose.





Figure 5. Snapshots of generated meshes from the CTC point clouds

The input BPP values for the meshes in Figure 5 are shown table 3 which are calculated as follows:

**Input BPP** **= 30 bits for geo + 24 bits for color + N bits per vertex index x num\_vertex\_indices / num\_vertices**

where again .



Table 3. Input BPP values for the meshes created from the CTC point clouds

Table 4 shows the lossless simulation results of TMC2v5+TFAN versus near-lossless simulation results of the public codec for 1 frame of voxelized 10-bit meshes generated from the CTC point clouds which contain geometry, color, and vertex indices. These results show the superiority of TMC2v5+TFAN over the public codec for the meshes generated from the CTC point clouds.



(a)



(b)

Table 4. Lossless simulation results of TMC2v5+TFAN (a) versus near-lossless (using the highest-quality compression level) simulation results of the public codec (b) for 1 frame of voxelized 10-bit meshes generated from the CTC point clouds

The point clouds in Figure 5 which are generated from the CTC point clouds are dense. To compare TMC2v5+TFAN and the public mesh codec for different point densities, we downsampled the meshes in Figure 5 uniformly as shown in Figure 6. The lowest density has about 3%-5% of the number of points in the original density.

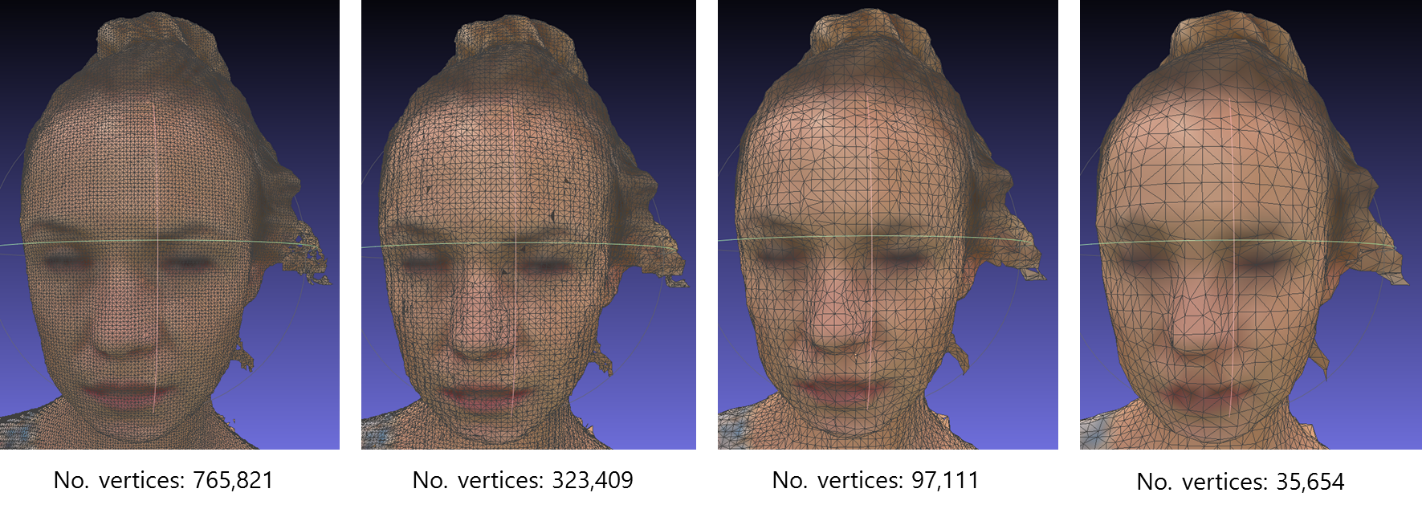


Figure 6. Uniform downsampling of the meshes created from the CTC point clouds

Figure 7 shows the performances of TMC2v5+TFAN (left) and the public mesh codec (right) for the meshes generated from the CTC point clouds and uniformly downsampled using box-grid filtering to result in different point densities. In case of TMC2v5+TFAN, the BPP values increase with reduction in the point density, whereas in the public mesh codec the BPP values changes within a smaller range.

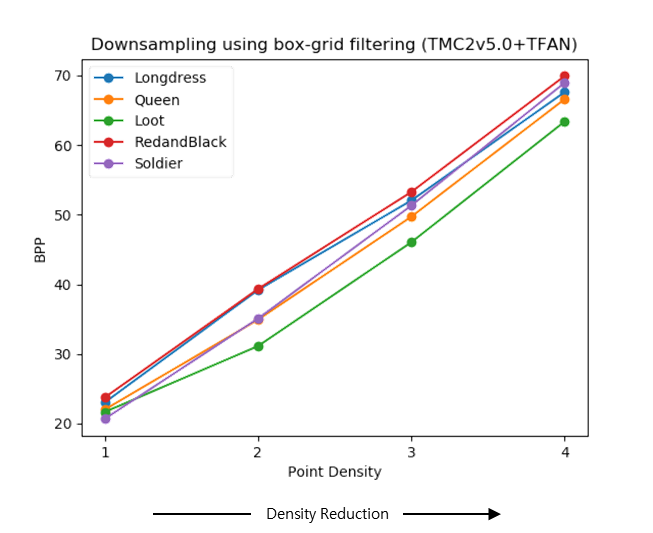
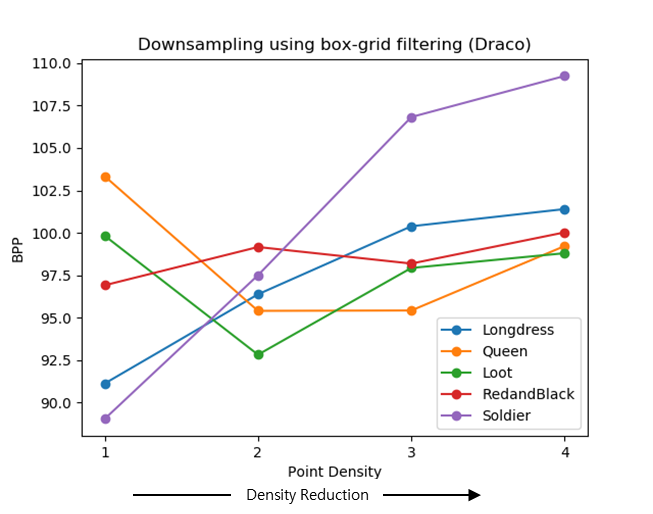
 

Figure 7. Performances of TMC2v5+TFAN (left) and the public mesh codec (right) for the meshes created from the CTC point clouds and uniformly downsampled.

Figure 8 demonstrates the average BPP values of all five sequences at different point densities. The TMC2v5+TFAN generates lower BPP values than the public mesh codec at all densities, but its BPP values increase more considerably with the density reduction. It should be noted that these results are related to uniformly subsampled meshes, and the results for random subsampling might be different.

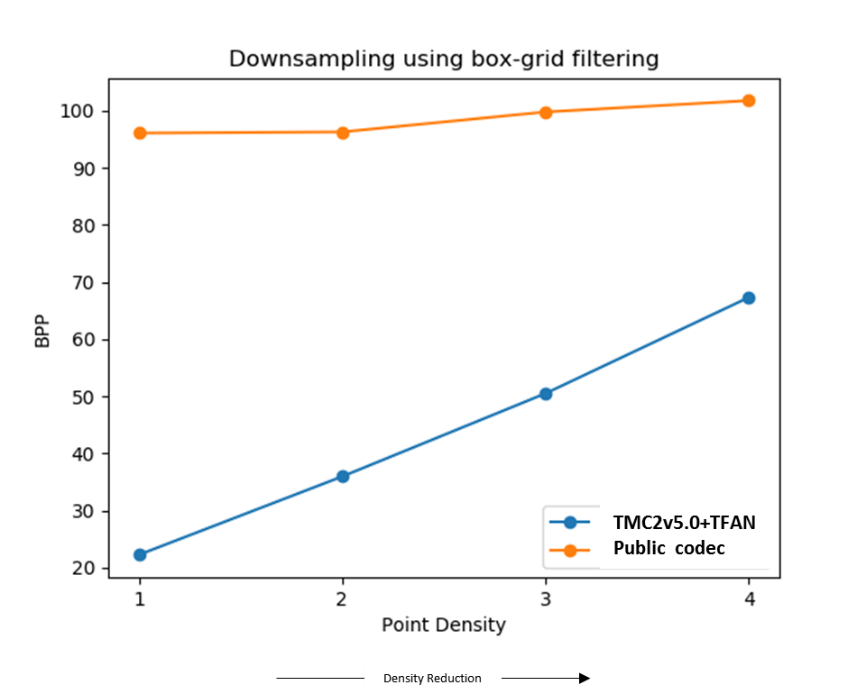


Figure 8. Average BPP of five sequences in Figure 7

# Conclusion

In this report, we show the performance of TMC2v5+TFAN compared to a public mesh codec for different mesh sequences. Based on the obtained results, the proposed framework to support mesh coding in V-PCC looks promising, so we recommend for further investigation.

# References

[1] m47608, “[V-PCC] [New Proposal] V-PCC extension for mesh coding,” ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, March 2019.

[2] ISO/IEC JTC 1/SC 29/WG 11, Information technology — Coding of audio-visual objects — Part 16: Animation Framework eXtension (AFX).