INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG 7
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC 1/SC 29/WG 7 m56120
Online – January 2021

Title: [PCC-MESH] Anchor generation for Dynamic Mesh Compression

Author: Danillo B Graziosi, Alexandre Zaghetto, Ali Tabatabai

Abstract

In this contribution we provide a possible anchor for the upcoming Call for Proposal on Mesh Compression for consideration by the PCC group. The anchor is obtained by coding the mesh geometry (vertices connectivity, position and 3D to 2D mapping) using SC3DMC, the MPEG reference software for static mesh compression. Moreover, the texture maps represented by a sequence of PNG files are converted to BGR444 or YUV420 and encoded with HEVC. In addition, we also provide results comparing the anchor to Draco, a state-of-the-art mesh compression software.

Introduction

Dynamic meshes are meshes that have their properties (like vertex position, connectivity, 3D to 2D mapping, attributes, etc.) changing over time. More specifically, current mesh compression activity is also interested in dynamic meshes that have their connectivity changing over time, which imposes greater difficulties for regular mesh compression approaches. For instance, the MPEG standard for animated mesh compression, Frame-based Animated Mesh Compression (FAMC [1]), cannot be used to encode such content, since it assumes that the connectivity does not change over time. One possible solution is to use the MPEG standard for static mesh compression, Scalable Complexity 3D Mesh Compression (SC3DMC, [2]), to encode the geometry of the sequences frame by frame. Even though the coding approach is limited to an ALL INTRA capability, at least it can provide a means for comparison for the upcoming Call for Proposal. In addition to the time-varying geometry, each frame has an associated texture map, usually represented as a PNG image. The sequence of PNG images can be converted into a video sequence (for instance, using ffmpeg [3]), and further encoded with a video compression standard, such as HEVC [4] or VVC [5]. In that case, all-intra and random-access configurations are possible.

We present in the following sections the steps taken to generate the anchor using SC3DMC and HEVC to encode a sequence of dynamic meshes. Even though the proposed coding is limited in the sense that geometry is encoded frame-by-frame, we believe it to be sufficient to provide a good comparison for further proposals on mesh compression. Furthermore, we also provide results for geometry compression using the state-of-the-art in static mesh compression, Draco software from Google [6].

Geometry coding using MPEG AFX

Scalable Complexity 3D Mesh Compression (SC3DMC)

We used a modified version of the MPEG reference software for static mesh compression, SC3DMC [7,8] in our experiments. The reference software had to be modified to accept OBJ as input. We also added some output information in the encoder to report the partial number of bits spent in coding each part of the mesh (connectivity, vertex position, etc.). The modified version can be compiled under Windows or Linux, and the files are available at the public git repository: https://gitlab.com/dangraziosi/tfan_mesh_anchor. The results used for the evaluation considered the TFAN option for encoding mode, Arithmetic Code/Ex-Golomb Code for the binarization mode, fast Parallelogram for the prediction mode, and with various quantization parameters (see Section on Experimental Results for details).
We would like to propose to change the software repository to be under the PCC git repository, if the group accepts this software for the anchor generation.

Texture Map Coding using HEVC

Converting a sequence of PNG images to video and encoding with HEVC

The sequential PNG image files were converted into a video sequence using ffmpeg [3]. The following commands were used:

```
ffmpeg -start_number ${startFr} -i ${seq}_vox10_%04d_uv.png -pix_fmt yuv420p ${outYUV420p}
ffmpeg -start_number ${startFr} -i ${seq}_vox10_%04d_uv.png -pix_fmt gbrp ${outGBRp}
```

The PNG files were converted into BGR444 planar video, for lossless compression, and YUV420 planar video, for lossy compression. The videos were then encoded with HEVC, using the parameters defined in the CTC for V-PCC [9]. The configuration files used for coding point clouds with 1-layer were used.

Experimental Results

Test set and selected metrics

The test set used were the 8i point clouds converted to textured meshes as detailed in [10]. The converted meshes are also available in the MPEG content repository (https://mpegfs.int-evry.fr/mpegcontent/ws-mpegcontent/MPEG-I/Part05-PointCloudCompression/dataSets/8i/8iVSLF/Mesh).

For the metric, we utilize the proposal in [11], using the orthogonal projection sampling method. We then report the D1, D2 and Y-PSNR metrics. For completeness, we also report geometry metric using the metro tool [12].

Quantization parameter selection

For the coding of texture maps using HEVC, we use the same parameters from V-PCC CTC conditions [9]. For the choice of quantization parameters for vertex position and vertex mapping, we selected the following values (*the selection of quantization step for AFX is just a suggestion, the group should discuss the results further*):

Table 1: Parameters used for anchor generation

Coding Parameters	Quality Levels				
	R1	R2	R3	R4	R5
QP	8	9	10	11	12
QT	8	10	10	10	12
QP(HM)	42	37	32	27	22

Figures 1 and 2 show the motivation to select the above quantization steps for AFX.

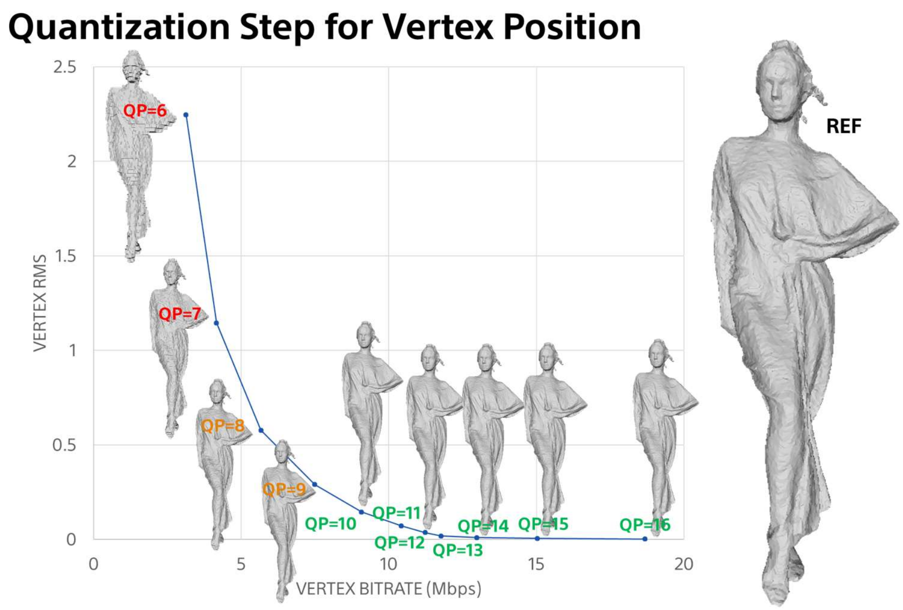

Figure 1: Objective and subjective evaluation of vertex position quantization step (QP)

In Figure 1, the RMS value obtained using the metro tool [12] shows the quality improvement as the bitrate for the vertex position increases. When performing subjective evaluation using Meshlab tool [13], we notice that the quantization becomes imperceptible around QP=12, but it is noticeable for QP<10.

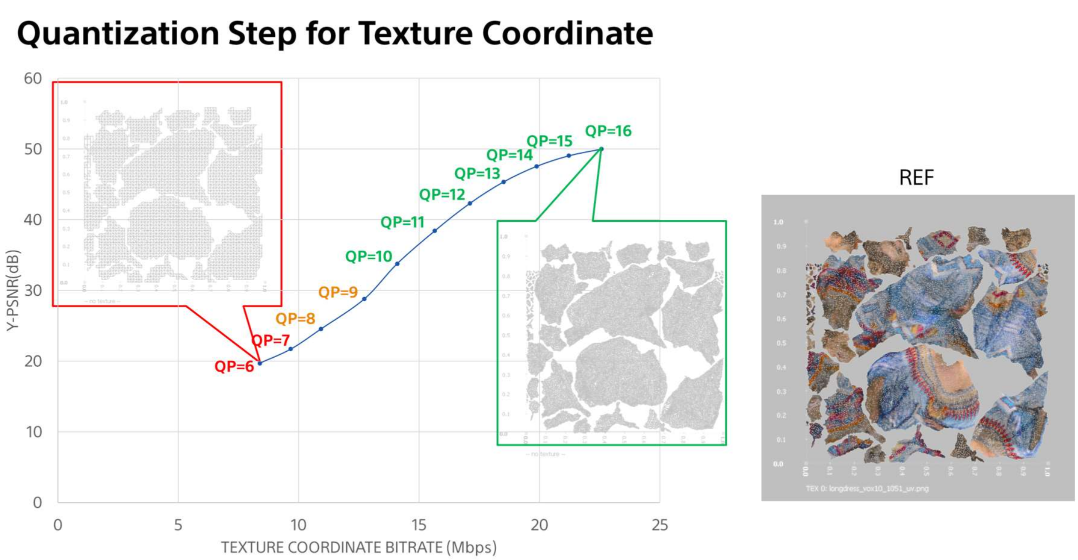

Figure 2: Objective and subjective evaluation of texture coordinate quantization step (QP in the figure, QT in Table 1)

For the evaluation of the texture coordinate quantization QT, the proposed modified pcc_error [11] was used. The graphic in Figure 2 shows the Y-PSNR value obtained by setting the quantization step for vertex position equal to 16 (best quality for geometry), and by using the uncompressed texture map for both reference and compressed mesh. The graphic shows that the quality improves as the bitrate for the texture coordinates increases. When performing subjective evaluation using Meshlab tool [13], we notice that the quantization becomes imperceptible around QT=10, but it is noticeable for QT<8.

Objective Results

For our objective results, we make available together with this contribution an excel file that contains both the SC3DMC performance, as well as the performance of Draco. The excel file is the same used for V-PCC, but only the 8i sequences have results. We produce AI results with the mesh coding tool encoding each frame independently and using the QP's from Table 1. The objective metrics reported in this section are using the extension of the pcc_error software proposed in [11], and the regular surface sampling method to generate the point cloud from the mesh surface sampling. We generated RA results by encoding the video with the RA configuration files, the mesh results are the same ones used for AI condition.

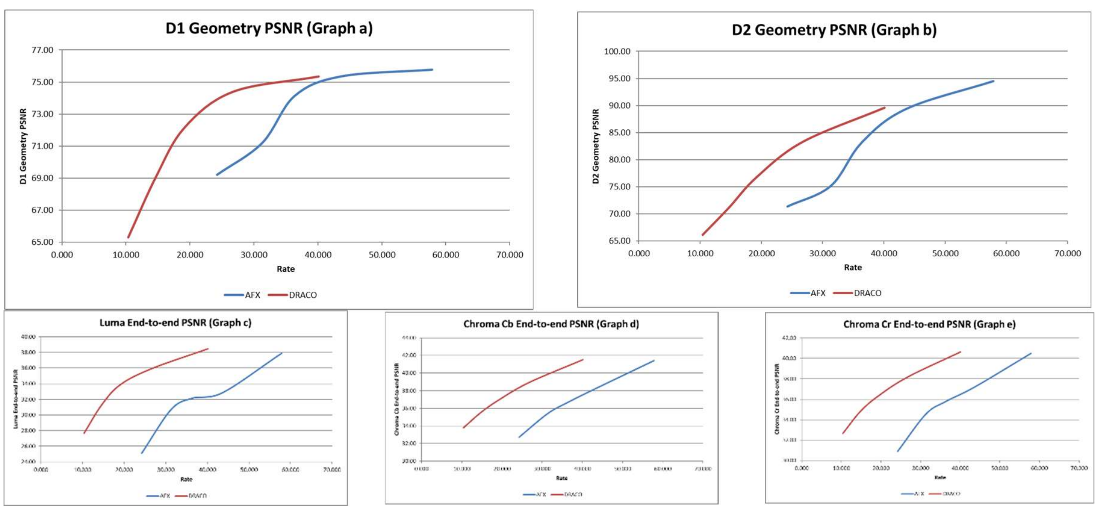

Figure 3: Performance of longdress AI condition

As shown in Figure 3, SC3DMC results are varying between 20-60Mbps, and the quality is increasing accordingly. Around 20-30Mbps is related to geometry coding, while 2-24Mbps is accounted for by color coding. In the case of Draco, the encoder achieves similar quality with less bits, around 7-15Mbps for the considered condition (longdress, AI). Draco is significantly better than SC3DMC is all tested sequences, as shown in Table 2.

Table 2: BD-gains of Draco vs. SC3DMC

Class	Seq.	C2AI					C2RA				
		Geom.		End-to-End BD-TotalRate [%]			Geom.		End-to-End BD-TotalRate [%]		
		D1	D2	Luma	Cb	Cr	D1	D2	Luma	Cb	Cr
A	Loot	-47.7%	-44.8%	-57.4%	-54.1%	-54.4%	-52.4%	-50.1%	-56.9%	-57.5%	-57.6%
	R&B	-48.2%	-44.6%	-56.5%	-50.1%	-55.5%	-53.0%	-50.2%	-56.6%	-55.2%	-56.7%
	Soldier	-43.2%	-39.9%	-57.3%	-49.4%	-49.6%	-50.3%	-47.7%	-56.7%	-55.5%	-56.1%
B	Long.	-37.1%	-31.6%	-55.6%	-47.2%	-49.2%	-46.4%	-42.0%	-54.9%	-52.5%	-53.6%
A avg		-46.4%	-43.1%	-57.1%	-51.2%	-53.2%	-51.9%	-49.3%	-56.7%	-56.1%	-56.8%
B avg		-37.1%	-31.6%	-55.6%	-47.2%	-49.2%	-46.4%	-42.0%	-54.9%	-52.5%	-53.6%
Overall		-44.1%	-40.2%	-56.7%	-50.2%	-52.2%	-50.5%	-47.5%	-56.3%	-55.2%	-56.0%

Subjective results

Here we show some snapshots of the reconstructed textured meshes, using the QP combination as specified in Table 1. Figure 4 shows results for SC3DMC and Figure 5 shows results for Draco.

Figure 4: Subjective comparison of longdress coded with SC3DMC (R1 - left, R3 - center, R5 - right)

Figure 5: Subjective comparison of longdress coded with Draco (R1 - left, R3 - center, R5 - right)

Conclusion

In this contribution, we presented coding results for dynamic mesh compression using MPEG standard software for static mesh compression and HEVC. We suggest the PCC group to evaluate the anchor generation and consider the presented results for the up-coming Call for Proposal on mesh compression.

Bibliography

[1] ISO/IEC 14496-16:2006/AMD 2:2009, “Information technology — Coding of audio-visual objects — Part 16: Animation Framework eXtension (AFX) — Amendment 2: Frame-based Animated Mesh Compression (FAMC)”

[2] ISO/IEC 14496-16:2011, “Information technology — Coding of audio-visual objects — Part 16: Animation Framework eXtension (AFX)”

[3] FFmpeg Developers (2016) ffmpeg tool (Version 4.1.3) [Software]. Available from http://ffmpeg.org/

[4] ISO/IEC 23008-2:2020, “Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding”

[5] ISO/IEC FDIS 23090-3, “Information technology — Coded representation of immersive media — Part 3: Versatile video coding”

[6] “Google Draco,” https://google.github.io/draco/, accessed: 2021-01-03

[7] ISO/IEC 14496 5:2001/Amd 27:2011, “Information technology – Coding of audio-visual objects – Part 5: Reference software - Amendment 27: Scalable complexity 3D mesh coding reference software”, ISO/IEC IS, April 20195. It is worth noting that the electronic attachments of the textual specification of this IS and all its (many!) Amendments and Corrigenda published to date are freely available at http://standards.iso.org/ittf/PubliclyAvailableStandards/: look for all table entries starting with “ISO/IEC 14496 5:2001”.

[8] MPEG SVN repository, http://wg11.sc29.org/svn/repos/MPEG-04/Part16-Animation_Framework_eXtension_(AFX)/trunk/3Dgraphics/SC3DMC/trunk/

[9] ISO/IEC JTC 1/SC 29/WG 7, “Common Test Conditions for V3C and V-PCC”, output document N0038, November 2020

[10] Graziosi, D., Zaghetto, A, and Tabatabai, A. “[V-PCC][EE2.6-related] Mesh generation script update”, input document m55366, October 2020

[11] Graziosi, D., Zaghetto, A, and Tabatabai, A. [PCC-MESH] Metric for mesh compression evaluation”, input document m56119, January 2021

[12] Cignoni, P., Rocchini, C. and Scopigno, R., 1998, June. Metro: measuring error on simplified surfaces. In Computer graphics forum (Vol. 17, No. 2, pp. 167-174). Oxford, UK and Boston, USA: Blackwell Publishers.