Research on Video-based Dynamic Mesh Compression Technology and Proposed Improvements

Liqi Jing

School of Telecommunications Engineering, Xidian University
Xi'an, Shaanxi, 710071, China
senfier_jlq@stu.xidian.edu.cn

ABSTRACT. As Extended Reality (XR) technology evolves, the development of immersive XR content, such as images and videos, has accelerated. The complexity of these products has heightened the demand for high-quality 3D geometric data. Given the vast size and intricacy of 3D geometric data, researchers are focusing on technologies that can optimize this data, which is crucial for creating intricate XR content. Mesh, a fundamental concept in 3D computer image models, plays a pivotal role in this research. Dynamic mesh, in comparison to ordinary mesh, contains significantly more data. Consequently, the efficient, lossless compression, storage, and transfer of this data are paramount. The Moving Picture Experts Group (MPEG) has conducted extensive research to standardize dynamic mesh compression, culminating in the proposal of a new standard known as video-based dynamic mesh compression (V-DMC). This paper elucidates the method for encoding and decoding displacement vector information in 3D mesh based on the V-DMC standard, which features projecting 3D displacement data into 2D video frames. After computation, wavelet transformation, and quantization, the displacement data is packed into a YUV video and encoded to achieve efficient compression according to the standard. Furthermore, this paper introduces three proposed improved schemes for the V-DMC test module, optimizes the displacement codec framework, and enhances compression performance. The paper concludes by summarizing the challenges encountered and proposing potential solutions, providing valuable insights and guidance for future research in this field.

Keywords: 3D Mesh Compression, Displacement Vector, Performance Improvement

1 Introduction

The rapid development of 3D modeling, capture, and other technologies leads to a significant increase in the creation and application of 3D content. This trend is particularly evident in the field of Extended Reality (XR), which includes Augmented Reality (AR) and Virtual Reality (VR). These applications require high-quality 3D geometric data to provide an immersive experience. Due to the complexity and large

Y. Yue (ed.), *Proceedings of the 2024 International Conference on Mechanics, Electronics Engineering and Automation (ICMEEA 2024)*, Advances in Engineering Research 240,
https://doi.org/10.2991/978-94-6463-518-8_6

scale of 3D content, optimization techniques for efficient 3D geometric data are becoming increasingly important.

Mesh is one of the basic concepts in 3D computer graphics and the primary carrier for the research of 3D computer graphics models. It is typically represented as a series of polygons covering the surface of a three-dimensional polyhedron. A mesh usually contains elements at three levels- vertexes, edges, and faces. Vertexes serve as the fundamental component of a 3D mesh. Edges are lines connecting two vertexes in a 3D mesh. Faces are polygons determined by the closed path of edges. In computer graphics models, meshes are usually composed of triangles, because planar polygons can be subdivided into triangles. It proves that applying triangle meshes to represent object surfaces is a general approach. A 3D mesh usually contains information about connectivity, geometry, and attributes. Connectivity information, or topology, refers to the connections between vertexes. Geometric information refers to the position of each vertex in the 3D space, represented as coordinates in the standard coordinate system. Attribute information is optional including color, material information, normal direction, texture coordinates, etc. of the vertexes or the mesh faces. Should any of the three types of information exhibit time-varying characteristics, the mesh is then referred to as a dynamic mesh. Generally speaking, a dynamic mesh contains geometric or connectivity information that varies over time. Additionally, an animated mesh refers to a dynamic mesh containing constant connectivity information. Compared with ordinary mesh, dynamic mesh contains myriad data. Therefore, approaches to compress, store, and transmit mesh data efficiently and losslessly are becoming crucial.

In July 2020, MPEG formulated and released the visual volumetric video-based coding (V3C) standard for volume content compression [1]. In the V3C framework, the video-based point cloud compression (V-PCC) technology can map the geometric information of point cloud data to 2D graphics, and use mature 2D video encoders for encoding, such as H.264/AVC, H.265/HEVC, and H.266/VVC, etc [1]. Inspired by the V3C standard, researchers are trying to extend the functionality of V3C technology in the field of mesh compression, using the V-PCC technology to compress the vertexes of 3D meshes, and using static mesh compression algorithms such as Edgebreaker to compress the connectivity information of the mesh [2, 3]. Although the technology based on the V3C framework to compress the geometric information of vertexes can convert the coordinates of dense point cloud data into a 2D framework to achieve high-efficiency compression, the density of 3D mesh vertexes is too sparse compared to the vertex density of point clouds, so the compression efficiency of geometric information is not ideal.

Considering the shortage of existing technologies in effectively compressing dynamic meshes, it becomes imperative to introduce a more efficient scheme for dynamic mesh compression. Consequently, the Moving Pictures Experts Group (MPEG) conducted in-depth research to standardize the compression of dynamic meshes. A call for proposal (CfP) concerning the compression of dynamic meshes has been proposed by MPEG [4]. Compared with previous frame-based animated mesh compression (FAMC) standard, the new standard can compress meshes with time-varying connectivity information [5]. The CfP has received responses from several major corporations, including Apple, Interdigital, Nokia, Tencent, and Sony. The scheme proposed by Apple, known as the video-based dynamic mesh compression (V-DMC), was adopted as the standard basic reference framework, due to its optimal performance [6]. The core innovation in Apple's reference framework is the introduction of displacement, bridging transformation between base mesh and original mesh, proving to improve compression efficiency enormously. Since the V-DMC test model introduced by Apple exhibits potential for enhancement, several optimization strategies have been proposed, thereby significantly augmenting the efficiency of dynamic mesh compression.

This paper aims to elucidate the underlying principles of the V-DMC standard and its reference framework, along with three proposed enhancement methods. The objective is to equip researchers and practitioners with a thorough understanding of cutting-edge technologies and their prospective applications in this domain.

2 Overview of V-Dmc Framework

This section delves into the encoding framework and technology of V-DMC, mainly focusing on the techniques related to the pre-processing and displacement vector coding modules [6]. As is shown in Fig. 1, the V-DMC framework proposed by Apple can be divided into two modules- the pre-processing module and the encoding module [6].

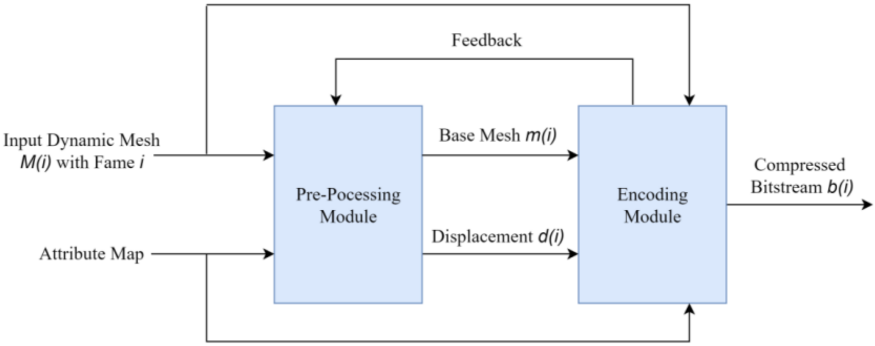

Fig. 1. High-level block diagram of the pre-processing module and the encoding module.

2.1 Pre-Processing Module

In the pre-processing module, the input dynamic mesh, denoted as original mesh $M(i)$ with frame i is converted into base mesh $m(i)$ and a series of displacements $d(i)$ according to attribute map $A(i)$. Fig.2 illustrates the brief principle of this module. The specific procedure includes decimation, re-parameterization, and subdivision surface fitting. The original mesh $M(i)$ is first down-sampled to decimated mesh $dm(i)$ to reduce the number of vertexes, referred to as the 'decimation' scheme. The decimated mesh $dm(i)$ then undergoes re-parameterization through the application of the UV-Atlas tool based on geometry information of $dm(i)$, where base mesh $m(i)$ is generated as output. By leveraging re-parameterization, it becomes feasible to generate fewer patches, thereby reducing discontinuities in parameterization. This approach could potentially enhance the rate-distortion (RD) performance of compression.

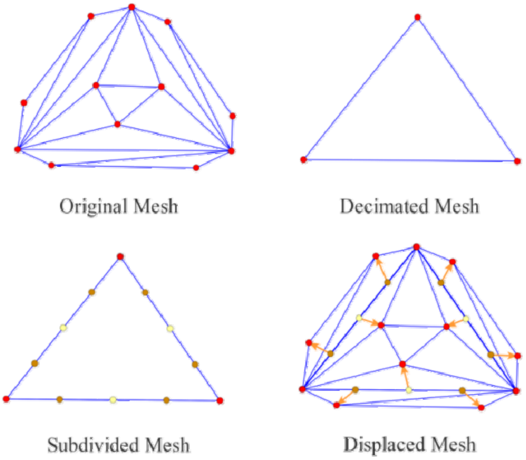

Fig. 2. The brief principle of pre-processing module [7].

Subdivision is then applied to the base mesh $m(i)$, inserting vertexes at each edge of the mesh. Manifold subdivision schemes and algorithms are optional, and the referred scheme utilizes a mid-point subdivision scheme, where a vertex is inserted at the mid-point of two existing vertexes. Note that the mid-point subdivision scheme can be iterative, meaning that in each iteration, a vertex is inserted at the mid-point between two existing vertices. Each vertex in the subdivided mesh needs to match a corresponding vertex which is closest in space in the original mesh, and a series of displacements $d(i)$ are computed after this matching process. To be specific, displacement information is determined by comparing the coordinate difference between the subdivided mesh and the original mesh for each vertex. During the calculation of displacement information, two coordinate systems are optional. One is the classical coordinate system, and the other is the local coordinate system. By establishing a local coordinate system that includes one normal vector and two tangent vectors for each vertex, the displacement vector can be decomposed. This decomposition determines the coordinate components in the local coordinate system. The greatest advantage of calculating displacement in the local coordinate system is that it allows the tangential component of displacement to be compressed more significantly compared to the normal component. This is because the normal component has a greater impact on the reconstruction of the mesh.

2.2 Encoding Module

In the encoding module, the base mesh $m(i)$ and the displacement $d(i)$ are encoded respectively. Fig. 3 shows the procedure of the encoding module. The initial step involves the application of uniform quantization to the base mesh $m(i)$. The quantized data is then encoded through a static mesh encoder, Google Draco for instance [8]. In the encoding framework, the type of static encoder used in the generated bitstream could be defined explicitly in the bitstream or could be predetermined by standards or applications. Since the mesh encoding and quantization are lossy, it is necessary to reconstruct the base mesh $m(i)$ based on the quantized data of the base mesh. The reconstructed base mesh is denoted as $m'(i)$. The difference between the base mesh $m(i)$ and the reconstructed base mesh $m(i)$ would be used to renew the displacement $d(i)$ to obtain $d'(i)$. In other words, the displacement information is determined based on the difference between the original mesh $M(i)$ and the reconstructed base mesh $m'(i)$. A wavelet transform using a lifting scheme is then exerted to $d'(i)$, resulting in the generation of a series of wavelet coefficients. The purpose of the wavelet transform is to filter high-frequency noise of the displacement, thus increasing the accuracy of displacement information. Wavelet transform is carried out in line with the level determined by the subdivision scheme. The displacement $d'(i)$ is decomposed by wavelet, so that the displacement is arranged from a high level of detail (LoD) to a low level of detail. The levels of detail correspond to the insertion order of the subdivision scheme, with higher LoD representing a new displacement of an inserting vertex and lower LoD representing the displacement of an existing vertex in a subdivision iteration. Also, with the escalation of LoD, displacements are capable of representing more intricate geometry information. The displacements at the lower level are used to predict the displacements at the higher level. Afterward, the displacements at the higher level are used to update the displacements at the lower level. The predicted and updated displacements are denoted as wavelet coefficients or transformed displacements.

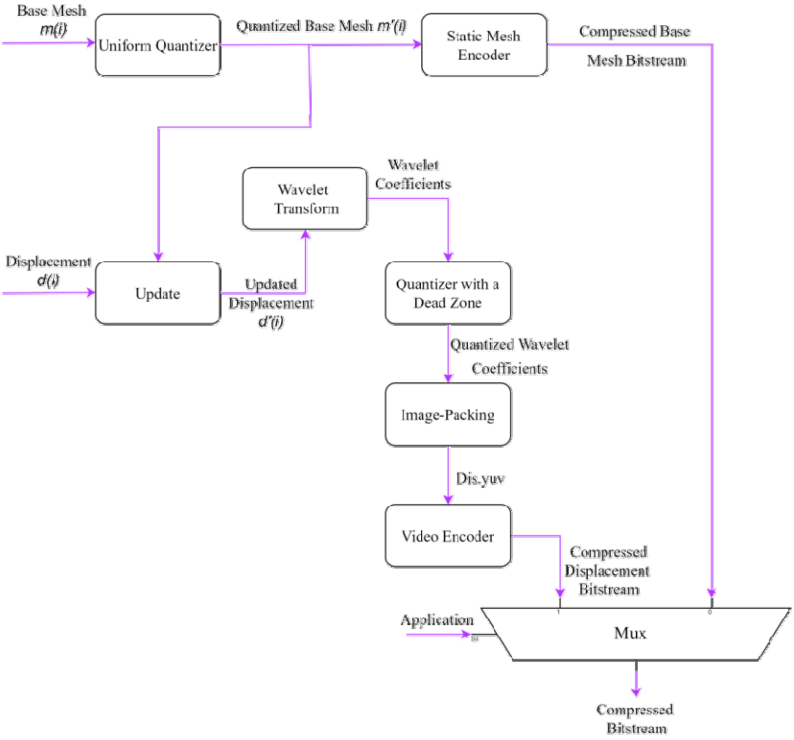

Fig. 3. Block diagram of the encoding module [6].

The transformed displacements then ought to be packed into 2D videos as is shown in Fig. 4. However, since existing codecs prove inefficient in encoding image sequences with sparsely distributed pixels that are implemented with the transformed displacements, a quantization step needs to be applied. The transformed displacements are quantized by a quantizer with a dead zone, helping cluster the sparsely distributed pixels in the images. Transformed displacements are scanned from low LoD to high LoD, which is the same sort of order to be packed later. After being quantized, they are packed into YUV images and encoded by a YUV video encoder to generate the bitstream of displacements. Three components of the displacement (n, t, b) are filled into the YUV components respectively. The pixels storing displacements are divided into blocks in an image. Displacements in pixels are arranged following the Morton order in each block, while blocks are arranged following the raster scan order. The number of pixels in each block is unfixed, which could affect the efficiency of the codec to some extent. The bitstream of displacements and base mesh would be submitted to the multiplexer to generate a compressed bitstream as output.

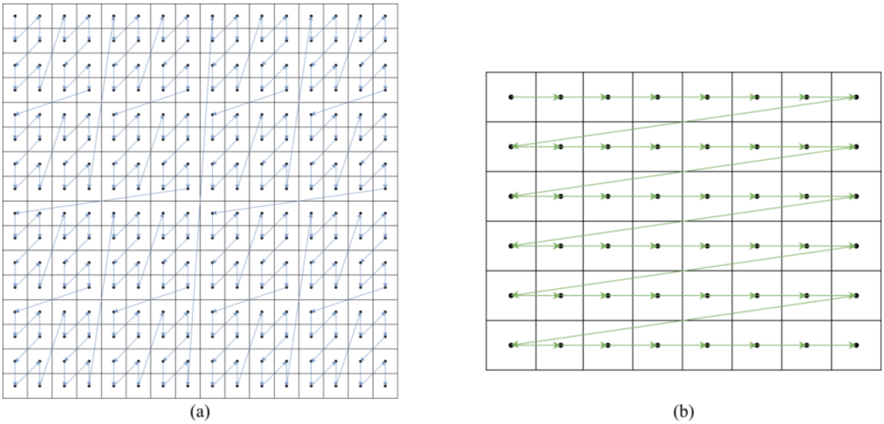

Fig. 4. Displacement arrangement order in a YUV video. (a) Arrangement order in each block (b) Arrangement order of blocks (Photo/Picture credit: Original).

3 Verified Improving Method

This section introduces three verified methods for improving the decline of BD-Rate or time complexity, both of which indicate the performance enhancement of these schemes.

3.1 Displacement Reverse Packing for Dynamic Mesh Compression

In the current V-DMC reference framework, displacements from low LoD to high LoD are packed into a 16*16 block in Morton order. However, such an order of packing might provide sub-optimal coding performance. A verified improving method packs the displacements in reverse Morton order, while the sort order of the displacement remains the same as the reference framework [9]. Besides, blocks are arranged following the reverse raster scan order. The starting point of the displacement arrangement lies at the low right corner of the displacement image, making the image a central symmetric version of the original image. As shown in Fig.5, the level of detail 1 (LoD1) to the bottom of the frame is packed first and then LoD2-3 from the bottom right to the top left of the frame is packed. Frame padding in the rest part of the frame is performed after all LoDs packed. If the height of the current frame is smaller than the max height of other frames, the top of the current frame will be padded.

The proposed method is tested based on the MPEG Vmesh test model with several test sequences offered by MPEG. Packed YUV videos can be visualized and played using a YUV video player, thereby allowing to verify the correctness of the displacement inverse arrangement. Simulation results are generated using this proposed method and compared with the results generated by the original test model under the same coding and decoding configuration. It is shown that BD-Rate based on the point cloud of the proposed method in both random access (RA) mode and all-intra (AI) mode has a comprehensive reduction of 0.1% compared to the original test module, which indicates that the performance of the proposed method has been improved.

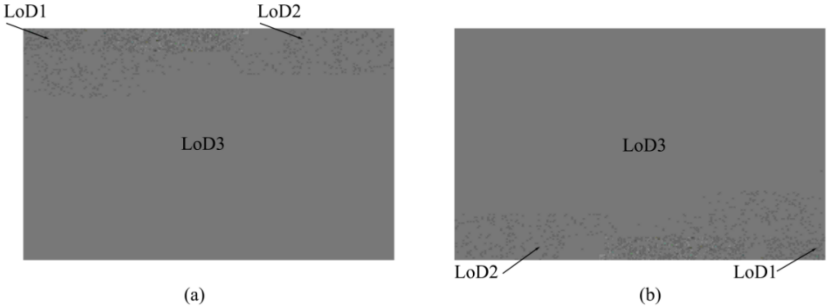

Fig. 5. Displacement arrangement image in original test module and proposed scheme. (a) Image in original test module (b) Image in proposed scheme (Photo/Picture credit: Original).

3.2 Division of Displacements and Binarized Coding for Dynamic Mesh Compression

Quantized displacement coefficients are normally packed into 2D videos to be encoded by existing codecs. However, the exploitation of video codecs aggravates the time complexity and space complexity of the program since video codec proves overly complex when dealing with uncorrelated displacements. Consequently, a verified improving method using division of displacements and binarized coding to substitute the original encoding process is proposed, dividing the LoDs into blocks and smaller subblocks [10]. When the LoDs for all elements within a block or subblock are zero, a flag is encoded to impede the encoding the LoDs.

To be specific, each wavelet coefficient generated from displacement is quantized to a level. The wavelet coefficients are then divided into blocks, each containing four coefficients for instance. A flag is used to record if all the coefficients in one block are zero, avoiding the meaningless encoding of consecutive zeros. Sub-block is also introduced to subdivide the encoding process, making the encoding more precise. If not all levels in one block are zeros, levels are binarized going through five layers of flags called sign, non-zero, greater than one, greater than two, and remaining. Information on the remaining part is then binarized and combined with the other four flags. They are introduced because levels of zero, one, and two appear more frequently than other levels, thus reducing the digits of binarized data. Given that each block features a set of unique components of levels, the coding of each block is performed independently and simultaneously.

The results demonstrate that the proposed method significantly accelerates the displacement encoding time, with a speedup ratio exceeding 300 in five instances of RA mode. This suggests that the encoding process is hundreds of times faster than the original test model, showing a substantially reduce in the encoding time. Conversely, the speedup ratio of displacement decoding time surpasses 2.5 in five instances of RA mode, which indicates that the decoding process is several times faster, showing some reduce in the decoding time. Reduce in encoding/decoding time is the visualized proof of a decline in time complexity, and this performance improvement is because hierarchical arithmetic coding presents a simpler coding method compared to the original V-DMC framework.

3.3 Improvement on Local Coordinate System for Displacement Compression

Calculation of the displacement vector is normally under a local coordinate system, which is established using one normal vector and two tangential vectors for each vertex in a subdivided mesh. The normal vector defines the local coordinate system since two tangential vectors are calculated from the normal vector. A verified improving method, shown in Fig.6, determine the normal vector of high LoD vertex, which is inserted in the middle of two current existing vertexes in subdivision iteration, by considering the normal vector of these two vertexes.

To be more specific, the proposed method only considers the normal vector of every vertex in base mesh $m(i)$. The normal vector of the LoD1 vertex is derived from those of two bilateral vertexes in the base mesh, and its value equals the average of two vectors. Accordingly, the normal vector of a higher LoD vertex is derived from those of two bilateral vertexes in its above LoD mesh. By reducing the number of normal vectors in the coordinate system that require compression, the bit rate can be decreased to some extent, which consequently leads to a proportional reduction in the BD-Rate.

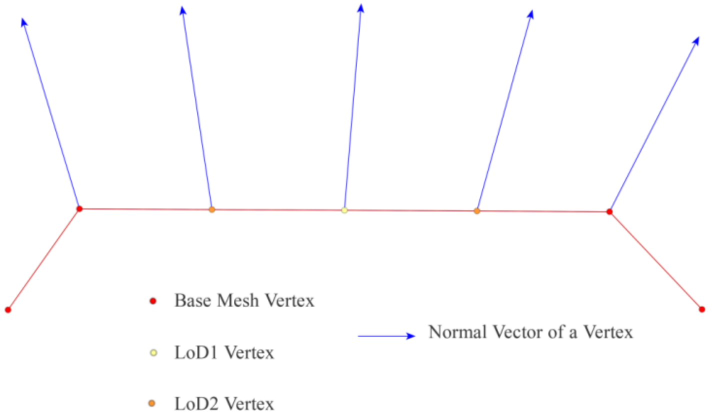

Fig. 6. Schematic diagram of the improvement on the local coordinate system for displacement coding [11].

The result shows that BD-Rate based on the point cloud of the proposed method has an overall reduction of 1.4% in RA mode and an overall reduction of 0.2% in AI mode compared to the original test module. This indicates that the performance of the proposed method is improved.

4 Conclusion

The V-DMC framework and its relevant technology effectively improve the compression efficiency of dynamic meshes.

This technology offers a novel idea and approach to the field of 3D data compression. It surpasses the constraints of traditional mesh coding, which relies on single data types such as images or point clouds. Instead, it achieves effective processing of complex mesh data across multiple dimensions, scales, and modalities. It provides a new solution for the field of mesh transmission, reducing network resource consumption and latency costs, and improving mesh compression performance and user experience. It also provides a new tool for the field of 3D media reconstruction, enhancing reconstruction accuracy and robustness.

This paper introduces the principle of the V-DMC standard and its reference framework. Three validated improving methods are introduced, which could save the BD-Rate or reduce the time complexity to some extent, making the performance of the test module more ideal.

References

1. Boyce, J. M., Doré, R., Dziembowski, A., Fleureau, J., Jung, J., Kroon, B., Yu, L.: MPEG immersive video coding standard. Proceedings of the IEEE 109(9), 1521–1536 (2021).
2. Alface, P. R., Martemianov, A., Ilola, L., Kondrad, L., Bachhuber, C., Schwarz, S.: V3C-Based coding of dynamic meshes. In: 2022 10th European Workshop on Visual Information Processing (EUVIP), pp. 1–6 (2022).
3. Rossignac, J.: Edgebreaker: Connectivity compression for triangle meshes. IEEE Transactions on Visualization and Computer Graphics 5(1), 47–61 (1999).
4. Moving Picture Experts Group.: CfP for Dynamic Mesh Coding. ISO/IEC JTC 1/SC 29/WG 7, N231, Virtual, pp. 1–38 (2021).
5. Mamou, K., Zaharia, T., Preteux, F.: FAMC: The MPEG-4 standard for animated mesh compression. In: 2008 15th IEEE International Conference on Image Processing, pp. 2676–2679 (2008).
6. Mamou, K., Kim, J., Tourapis, A.: Apple's dynamic mesh coding cfp response. ISO/IEC JTC 1/SC 29/WG 7, M59281, Online, pp. 1–16 (2022).
7. Choi, Y., Jeong, J., Lee, S., Ryu, E.: Overview of the video-based dynamic mesh coding (v-dmc) standard work. In: 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), pp. 1–4 (2022).
8. Van Rensburg, B. J., Puech, W., Pedeboy, J.: The first draco 3d object crypto-compression scheme. IEEE Access 10, 10566–10574 (2022).
9. Wang, X., Ma, Z., Wei, H., Yu, Y., Zakharchenko, V., Wang, D.: Displacement reverse packing for dynamic mesh coding. ISO/IEC JTC 1/SC 29/WG 7, M60954, Online, pp. 1–3 (2020).

10. Nishimura, H., Kato, H., Kawamura, K.: Hierarchical arithmetic coding of displacements for dynamic mesh compression. In: 2023 IEEE International Conference on Image Processing (ICIP), pp. 10566–10574 (2023).
11. Zou, W., Huang, H., Trioux, A., Yang, F.: An efficient video-based geometry compression system for 3D meshes. In: 2023 IEEE International Conference on Visual Communications and Image Processing (VCIP), pp. 1–5 (2023).