



Survey of Inter-Prediction Methods for Time-Varying Mesh Compression

Jan Dvořák,¹ Filip Hácha,¹ Gerasimos Arvanitis,² David Podgorelec,³ Konstantinos Moustakas² and Libor Váša¹

¹Department of Computer Science and Engineering, University of West Bohemia in Pilsen, Faculty of Applied Sciences, Pilsen, Czech Republic
{jdvorak, hachaf, lvasa}@kiv.zcu.cz

²Department of Electrical and Computer Engineering, University of Patras, Patras, Greece
arvanitis@ece.upatras.gr, moustakas@upatras.gr

³Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia
david.podgorelec@um.si

Abstract

Time-varying meshes (TVMs), that is mesh sequences with varying connectivity, are a greatly versatile representation of shapes evolving in time, as they allow a surface topology to change or details to appear or disappear at any time during the sequence. This, however, comes at the cost of large storage size. Since 2003, there have been attempts to compress such data efficiently. While the problem may seem trivial at first sight, considering the strong temporal coherence of shapes represented by the individual frames, it turns out that the varying connectivity and the absence of implicit correspondence information that stems from it makes it rather difficult to exploit the redundancies present in the data. Therefore, efficient and general TVM compression is still considered an open problem. We describe and categorize existing approaches while pointing out the current challenges in the field and hint at some related techniques that might be helpful in addressing them. We also provide an overview of the reported performance of the discussed methods and a list of datasets that are publicly available for experiments. Finally, we also discuss potential future trends in the field.

Keywords: compression algorithms, data compression, modelling, polygonal mesh reduction

CCS Concepts: • Computing methodologies → Animation; Mesh models; • Theory of computation → Data compression

1. Introduction

Representing the shape of a surface is a long-standing problem in computer graphics, which has been approached from various directions, and to this day, new ideas are being tested that overcome some of the drawbacks of the popular choices. Out of those, it is hard to think of one that has gained more popularity than *triangle meshes*: they are versatile, comparatively easy to work with and widely supported by modelling software and rendering hardware. Moreover, with the recent advances in consumer grade 3D scanning technology and automatic photogrammetric reconstruction including robust methods of conversion from point clouds to triangle meshes, they are even easy to obtain.

So easy in fact, that it is possible to reconstruct a sequence of triangle meshes that capture a shape that develops in time, at a framerate of 30–60 fps, which allows for a smooth playback. Such data

structure is an attractive alternative to a traditional approach to representing 3D animations: model rigging and skinning. While creating a skinned animation is usually an expensive process, which requires work of a skilled 3D artist, often combined with actors whose movements are captured via motion-capture systems and applied to a rigged mesh, obtaining a mesh sequence is a matter of proper equipment and can in general be cheaper while achieving natural realism of motion and appearance. Still, at current time, mesh sequences exhibit more than a few drawbacks as well. For one, they are rather difficult to manipulate and edit in order to adjust them for a particular purpose, and perhaps more importantly, they are usually associated with a much bigger memory footprint, especially when compared with common rigged models combined with a skeletal representation of motion. For each frame, typically at least a set of vertex positions and a connectivity, represented by a set of triangles, must be stored. Even for scenes of moderate resolution and duration,

Table 1: Overview of existing methods for TVM compression. Methods are in the order in which they appear in the survey. Highlighted methods are considered best performing. Ref. column contains the most relevant reference for the given method. Columns Category and Subcategory show the classification of the methods according to the type of inter-prediction they use. Columns Versatility and Designed for show the type of input data the method handles (see Section 9.2). Columns Iso. and Conn. indicate whether the method preserves structure (Isomorphism) and whether it addresses connectivity coding (see Section 9.1). The colour in the Conn. column indicates efficiency (green = efficient, orange = inefficient, red = not addressed). Symbol meaning: \checkmark = Yes, \times = No, * = Optional, - = Not relevant.

| Method | Ref. | Category | Subcategory | Versatility | Designed for | Iso. | Conn. |
|---------------------------------|-----------|-------------|------------------|----------------|--------------|--------------|--------------|
| Patch ICP | [GSK03] | ME | — | General | Synthetic | \checkmark | \checkmark |
| EBMA | [HYA07] | ME/Struct. | — | General | General | \checkmark | \times |
| PCA-aligned patches | [YA10] | ME | — | General | General | \checkmark | \checkmark |
| Grid occupancy XOR | [HYA08] | Struct. | Grid | General | General | \checkmark | \times |
| Semi-regular representation | [YKL06] | Model | Tracked surface | Const. GT Top. | Const. Top | \times | — |
| Reeb graph matching | [TSM07] | Model | Reeb graph | Const. GT Top. | Human | \times | — |
| Topology dictionary | [TM12] | Model | Reeb graph | General | Human | \times | — |
| Skinned mesh | [MYA08] | Model | Tracked skeleton | Const. GT Top. | Human | \times | — |
| Per-bone ICP | [DAZD14] | Model | Tracked skeleton | Const. GT Top. | Human | * | \checkmark |
| Occupancy network | [ZGT23] | Model | Neural model | General | General | \times | — |
| Optimized embedded deformation | [HCNC23] | Model | ED nodes | General | Human | \times | — |
| SkinOff | [HKM04] | Video | Geometry video | Textured | Textured | \times | — |
| EIP geometry video | [XHQ*10] | Video | Geometry video | Face | Face | \times | — |
| Cut over local extrema | [TM13] | Video | Geometry video | General | Human | \times | — |
| Polycube geometry video | [HCHMT14] | Video | Geometry video | Const. Top. | Const. Top. | \times | — |
| Registered geometry images | [GWW23] | Video/ME | Geometry video | General | General | \times | — |
| V-PCC + Edgebreaker | [FJB20] | Video | Projection | General | General | \checkmark | \checkmark |
| V-DMC Nokia | [AMI*22] | Video | Projection | General | General | \times | — |
| V-DMC Tencent | [HZT*22] | Video | Geometry video | General | General | \times | — |
| V-DMC Apple | [MKT*22] | Model/Video | Geometry video | General | General | \times | — |
| Inter wavelet coefficients | [NKK23a] | Model/Video | Geometry video | General | General | \times | — |
| Temporally-consistent remeshing | [JXK23] | Model/Video | Geometry video | General | General | \times | — |
| Tracked base mesh | [JXK24] | Model/Video | Geometry video | General | General | \times | — |
| Low-/High-precision split | [ZHTY23] | Video | Projection | General | General | \times | — |

this quickly leads to data sizes that barely fit into the RAM of common workstations, making the processing, storage and transmission of this kind of data problematic.

Having an efficient compression algorithm could lead to a breakthrough in the applicability of this data format, as has been the case in the past with many other multimedia. When considering the problem, it seems obvious, that there is indeed a lot of data redundancy, not unlike that observed in video and audio data. In particular, there is a strong temporal coherence of the shapes that make up the sequence, which should be exploitable in order to reduce the data rates significantly. For a special case of temporal sequences, where the connectivity remains unchanged throughout the sequence (known as dynamic meshes), exploiting the temporal coherence has indeed led to highly efficient compression algorithms that are able to compress data down to data rates of less than 1 bit per frame per vertex (bpfv) without a visually perceivable loss in quality. However, it turns out, that replicating this success with more general data, where the connectivity is different for each frame, is going to be rather difficult.

The main problem stems from the fact that each frame carries not only information about the shape but also about the sampling of the shape. Even very similar shapes may have a rather different sampling, as seen in Figure 1, and while there is a strong temporal coherence in shape, there is usually little or none in the sampling. Moreover, it is quite difficult to actually distinguish the two compo-

nents (shape, sampling) in the data, since they are inherently mixed together in the 3D coordinates. From a rather abstract point of view, it can be expected that the sampling may take up to two thirds of the data, since placing a vertex on a surface essentially requires storing its 2D (tangential) coordinates, while the shape is captured mainly in the third, normal coordinate. This indicates that in fact, the temporally coherent data takes up a minor part of the overall data structure. In real datasets, however, this issue is naturally more complex, at least because the true information load of the tangential and normal components of the 3D coordinates varies with data character.

It is probably due to this difficulty that to this day, no general efficient compression algorithm has been presented, that is not limited to a narrow type of input data, preserves the input connectivity and outperforms a frame-by-frame compression. In the past, many attempts to achieve this goal have been made; however, only with partial success. In this state-of-the-art report, we provide an overview of the methods and analyse their properties.

We will discuss only the methods performing *inter-frame* prediction of mesh geometry, that is the methods that exploit the temporal coherence between the frames of the mesh sequence. On the other hand, some methods perform prediction only inside a single frame. These are referred to as *intra-only* methods. The simplest example of an intra-only method is applying a static mesh codec, such as Google Draco [GHS*18], to each frame separately.

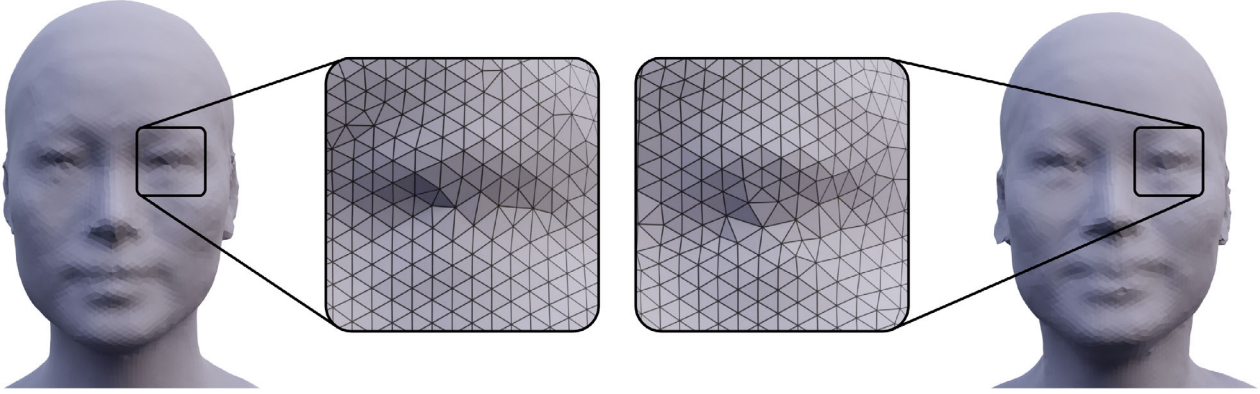


Figure 1: Similarity of appearance of individual frames of time-varying meshes does not imply coherence in sampling and connectivity between subsequent frames. This makes it rather difficult to exploit the temporal redundancy when compressing such data.

Nevertheless, intra-only methods designed specifically for the compression of mesh sequences also exist (e.g. the method proposed by Mekuria et al. [MCB14]). We also exclude all the methods that only consider the compression of dynamic meshes since this field is considered well-studied and has already been discussed in the past, for example in the survey of Maglo et al. [MLDH15].

The methods are classified into four main groups based on the paradigm they facilitate for inter-prediction. Methods that incorporate multiple paradigms are placed in a category where we subjectively consider them to fit in the context of the other methods in that category. The main criteria we use to evaluate the methods are their versatility (i.e. the ability to process general input data) and their ability to preserve the original structure of individual mesh frames. We also discuss existing relative compression performance evaluations between individual methods.

2. Preliminaries

A triangle mesh is currently considered the most popular representation of 3D surfaces due to its simplicity, approximation quality, and native support on graphics cards. It is a piecewise planar surface usually defined as $\mathcal{M} = (V, T)$, where V is a set of vertices, and T is a set of triangles that connect them. A vertex is usually represented by its geometry (position) and properties (normal, colour, texture coordinate, etc.), while a triangle is represented as an ordered triplet of indices, and the order of vertices induces its orientation (the direction of a normal). When talking about mesh *geometry*, we will refer to the positions of vertices, *connectivity* will refer to the combinatorial information (vertex indices, triangles, or edges) and the *topology* will refer to the overall structure (the number of connected components, handles and boundary curves).

To represent shapes evolving in time, one can utilize a sequence of triangle meshes $\mathcal{S} = (\mathcal{M}_1, \dots, \mathcal{M}_n)$, where each individual mesh \mathcal{M}_i denoted a *frame* represents the state of the shape at a certain discrete point in time t_i , and where n is the number of frames. The frames are in an ascending order of the times they were sampled. We assume the sequence to be temporally coherent in terms of the geometry, meaning that two subsequent frames are often visu-

ally nearly indistinguishable. The temporal coherence between the frames can be described by a *correspondence* function $f_{ij} : \mathcal{M}_i \mapsto \mathcal{M}_j$, which for any point $\mathbf{x} \in \mathcal{M}_i$ assigns a corresponding point $f_{ij}(\mathbf{x}) \in \mathcal{M}_j$ if it exists. Correspondences not only can be used for inter-frame prediction but also allow temporally coherent mapping of values on the surfaces, for example texture [BHLW12].

Dynamic meshes (DMs) are a special class of triangle mesh sequences. The distinguishing property of a dynamic mesh is a *common connectivity* T shared by all frames:

$$T_0 = T_1 = \dots = T_{n-1} = T,$$

where n is the number of frames. This also indicates that the number of vertices and their order are constant through time. Only the geometry and properties change between frames, and thus the connectivity needs to be encoded only once. One of the main advantages of a dynamic mesh is that the vertex correspondences are explicitly coded in the connectivity:

$$\forall v_k \in V : f_{ij}(\mathbf{x}_k^i) = \mathbf{x}_k^j$$

for any pair of frames (i, j) , where \mathbf{x}_k^i is the position of k -th vertex in the i -th frame. The correspondence function for any point $\mathbf{x}^i \in \mathcal{M}_i$ can be directly generalized from vertex correspondences using barycentric coordinates $(\lambda_a, \lambda_b, \lambda_c) : \mathbf{x}^i = \lambda_a \mathbf{x}_a^i + \lambda_b \mathbf{x}_b^i + \lambda_c \mathbf{x}_c^i$ in triangle $t = (v_a, v_b, v_c)$, which contains \mathbf{x}^i :

$$f_{ij}(\mathbf{x}^i) = \lambda_a f_{ij}(\mathbf{x}_a^i) + \lambda_b f_{ij}(\mathbf{x}_b^i) + \lambda_c f_{ij}(\mathbf{x}_c^i) = \lambda_a \mathbf{x}_a^j + \lambda_b \mathbf{x}_b^j + \lambda_c \mathbf{x}_c^j.$$

For dynamic meshes, the function f_{ij} is always an isomorphism. Instead of treating the geometry of each frame separately by assigning each vertex v_k a position $\mathbf{x}_k^i \in \mathbb{R}^3$, a static mesh $\mathcal{M} = (V, T)$ can be considered, where for each vertex v_k the geometry is represented as a trajectory $\mathbf{t}_k \in \mathbb{R}^{3n}$. These properties have allowed a straightforward application of various redundancy reduction techniques, which have led to many efficient algorithms. As a result, the field of DM compression is currently considered well-studied. On the other hand, while simple and easy to work with, the DM lacks representation versatility. Not only is the time-evolving topology of the represented surface not allowed, but since the connectivity complexity directly influences the ability to represent fine details, any

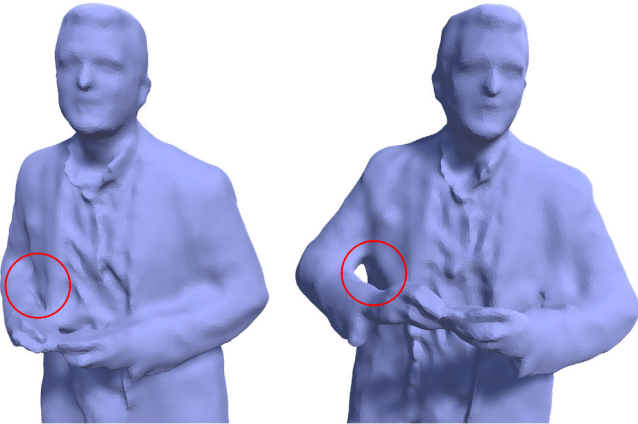


Figure 2: Example of bijection loss of correspondences. There are no correspondences for vertices at the back of the arm in the highlighted area.

fine detail must also be accounted for in the shared connectivity and in the geometry of every frame, even when it appears only in a single frame.

Any mesh sequence in which the number of vertices and/or connectivity changes over time is denoted a *time-varying mesh* (TVM). While the temporal coherence of the connectivity might be present (e.g. in synthetic data), it cannot be generally assumed. Not only we cannot directly derive the correspondences of time-varying mesh frames from the connectivity, but they are difficult to establish in general. This is caused by the fact that the bijective property of the correspondence function is lost with the merging and separation of parts (see Figure 2). This renders exploiting the temporal coherence a much more difficult problem.

When considering topology changes in mesh sequences, it is important to distinguish between the change in the topology of the underlying shape (i.e. the ground-truth topology) and the change of topology between the actual mesh frames. The latter often occurs due to errors introduced during the process of obtaining the data, even when the underlying shape is of constant topology (e.g. a human actor).

2.1. Problem definition

This survey considers the problem of TVM compression. Compression is the process of transforming input data X , which is represented by a certain sequence of m bits, into a reduced representation of n bits, where $m \gg n$, from which one can obtain a reconstruction \hat{X} by a reverse process called decompression. If the compression is lossless, X and \hat{X} must be identical. In lossy compression, \hat{X} is distorted and must resemble X in a defined sense. In our case, the input data is a sequence of triangle meshes represented by geometry, connectivity and other properties of each mesh.

In terms of mesh geometry compression, the majority of methods are lossy, meaning the reconstructed positions are different from the original ones. This is because positions are represented by vectors of floating point values, and such data is quite difficult to efficiently

encode in a lossless manner. Loss of the information in this case is also less likely to be detected by a viewer than if it occurred in connectivity.

For connectivity, the key criterion to classify a method as lossless is whether a map exists between the connectivity graph of the original and reconstructed mesh, which is an isomorphism. Such a definition permits lossless methods to reorder vertices, as long as the original structure (number of vertices and connections between them) is preserved. In this survey, TVM compression methods that perform lossless connectivity compression will be denoted *structure-preserving*. Lossy methods usually perform remeshing, simplification or filtering, which also implies loss of information in geometry. Unless the only purpose of the compressed mesh sequence is to be rendered (e.g. in entertainment or tele-immersion), it is desirable to preserve its original structure.

2.2. Evaluating distortion

In lossy data compression, one of the most important parameters is the ratio between the resulting bitrate and the data distortion. Evaluating such distortion is not straightforward, even in the case of static meshes, since the metric for evaluating mesh quality should reflect a human's visual perception of the distortion. In the case of static meshes, there are already several perceptual metrics that achieve reasonably good results, but comparing TVMs presents additional challenges since, in addition to the shape distortion itself, it is necessary to consider the possible *temporal artifacts*, which cannot be perceived when comparing frames separately, but appear during playback of the sequence, such as shaking and other types of temporally inconsistent distortion.

2.3. Related surveys

TVM compression has already been discussed in the 3D mesh compression survey of Maglo et al. [MLDH15]. The survey contains a whole section dedicated to mesh sequence compression. However, the authors only mention two existing methods for TVM compression and comment on the field being much more challenging than DM compression due to a lack of explicit temporal correspondences in TVMs. The rest of the section is dedicated solely to DM compression.

Cao et al. [CPZ19] presented a survey on 3D point cloud compression, which also considers the compression of dynamic point clouds (DPCs). This field is closely related to TVM compression since DPCs also lack explicit temporal correspondence information. Therefore, DPC compression methods often use similar techniques to exploit temporal coherence.

Related to this work is also the recent book on immersive video technologies [imm23]. The book covers a broad range of topics, out of which the compression of mesh sequences and their perceptual comparison are the most related to the topic of this survey. Since we focus on a much narrower topic, we provide a more in-depth discussion.

Finally, the survey of perceptual metrics presented by Corsini et al. [CLL*13] is also relevant to this paper, since it discusses why

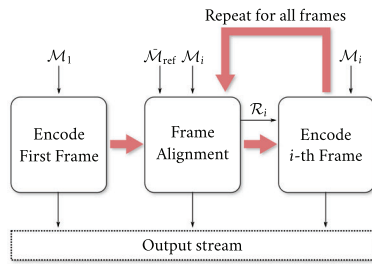


Figure 3: Example pipeline of a TVM compression based on motion estimation. Thin arrows represent the data flow, while bold arrows represent the order of operations. M_i denotes a mesh at the i -th frame and M_{ref} denotes a certain mesh (previous frame or keyframe) used to obtain a reference shape \mathcal{R}_i . The bar above a symbol denotes data distorted by compression.

it is important to consider perceptual distortion in the dynamic setting. Nevertheless, the authors only consider distortion metrics for static and dynamic meshes.

2.4. Paper structure

The rest of the paper is structured as follows: the compression methods facilitating the four main paradigms are discussed in Section 3 (Motion estimation), Section 4 (Prediction of data structures), Section 5 (Temporal-model-based methods) and Section 6 (Video-based methods) respectively. Next, Section 7 discusses the available TVM datasets, Section 8 provides an overview of the relative performance of the methods and Section 9 highlights the current research challenges in the field of TVM compression. Section 10 provides some practical recommendations that stem from the analysis of the state of the art, and finally Section 11 concludes the paper.

3. Motion Estimation

Motion estimation (ME) is a technique often used in video compression. It assigns so-called *motion vectors* to blocks of pixels of the previous frame describing how these pixels moved. This information is then used for the prediction (or replacement) of the current frame. It is possible to apply an analogous process for mesh sequence compression by aligning parts of two subsequent frames. The alignment process is also referred to as *registration*. While motion estimation is often used in approaches across our specified categories, the methods discussed in this section rely solely on it. In general, these methods proceed as follows (see also Figure 3): The first frame is encoded in an intra fashion. For each subsequent frame, a reference shape is obtained by aligning the current frame with the previous frame (or the previous keyframe). Parameters of the alignment are encoded as well so that the reference shape can also be reconstructed by the decoder. This reference shape is then used to predict (or replace) the coded frame. The main advantage of such approaches is their simplicity which comes at the cost of compression performance.

The first method to address TVM compression, although only for synthetic sequences, was proposed by Gupta et al. [GSK03]. It

uses the iterative closest point (ICP) algorithm [BM92] to rigidly map patches of the coded frame onto the whole mesh of the previous frame. In each iteration of the ICP algorithm, correspondences between source and target objects are found using the closest point search. Then, a transformation that best aligns corresponding points is found and applied to the source object. After alignment, the patches are refined to merge segments with a similar motion. The vertices are divided into three sets based on the residual of the estimated rigid transformations. The first group can be represented solely by the rigid transformation, the second group requires correction vectors to be encoded and the rest is encoded without exploiting temporal information. The method, however, relies on a certain temporal coherence in the connectivity, since the authors assume it changes in simple updates (e.g. vertex insertion/removal, subdivision), which, in general, does not occur in real-world data.

Han et al. [HYA07] proposed a method based on the block matching algorithm (BMA) [JJ81], a widely used ME technique in video compression. The method divides the bounding box of the coded frame into grid of cubic blocks of a specified size. The surface patch in each block is then translated for its centroid to lie at the centre of the block. For each block, a corresponding block in a defined search area of the previous frame is found by matching the weighted average normal vectors of the patches. Positions of vertices are predicted using correspondences derived from the nearest neighbour search.

Another method based on the fitting of patches was proposed by Yamasaki and Aizawa [YA10]. Instead of registration, their approach first transforms all the patches into a common local coordinate frame obtained by principal component analysis. Each coded patch is predicted by the closest patch in the reference frame (i.e. the one minimizing the lengths of the correction vectors). To reconstruct such a patch, the decoder needs the index of the corresponding reference patch, the vertex correspondences between patches, the correction vectors, and a rigid transform which moves the patch from the local coordinate frame into the correct global position. Instead of encoding the vertex correspondences explicitly, for each reference vertex, the number of corresponding coded vertices is encoded followed by the matching correction vectors. This way, the decoder can reconstruct the original set of vertices, albeit in a different order.

4. Prediction of Data Structures

Instead of working directly with the geometry, the methods in this category construct spatial data structures over each frame and proceed with the inter-prediction of such structures. Although there is only one method in this category for time-varying meshes, it is fairly popular in dynamic point cloud compression due to the input data usually being voxelized point clouds stored in an octree. Compared to motion estimation, the prediction of data structures is often much simpler, since no local matching between subsequent frames is usually required.

Han et al. [HYA08] based their method on cubic binary grids. The method first transforms all the frames so that their centroid lies at the origin to maximize their overlap. Then, a coarse grid of cubic blocks is constructed over the sequences bounding box. For each frame, a binary function on the grid is evaluated, which indicates whether a block contains any frame vertices. The

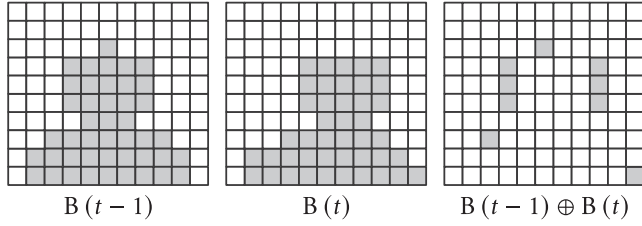


Figure 4: Exploiting coherence of binary grids between subsequent frames using XOR operation. Source: [HYA08].

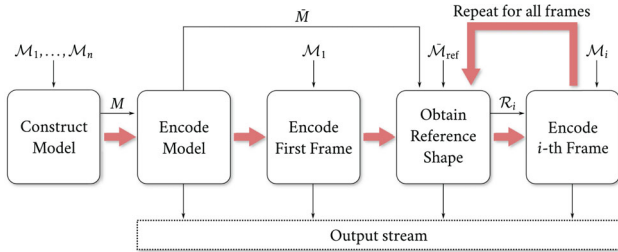


Figure 5: Example pipeline of a TVM compression based on a temporal model M . Thin arrows represent the data flow, while bold arrows represent the order of operations. M_i denotes a mesh at the i -th frame and M_{ref} denotes a certain mesh (previous frame or keyframe) used to obtain a reference shape \mathcal{R}_i . The bar above a symbol denotes data distorted by compression.

temporal coherence is exploited using XOR operation on two subsequent frames (see Figure 4). This information is then encoded using the run-length encoding (RLE). Each block that contains mesh geometry is then further subdivided at a finer scale, ideally so that each sub-block contains at most a single vertex. This finer information is also encoded using RLE without further exploiting the temporal coherence. In subsequent work, Ferreira et al. [FHYA10] extended this approach for progressive coding using multiple levels of finer subdivisions.

5. Temporal-Model-Based Methods

The most efficient way to exploit temporal coherence in the geometry is by employing temporal models capturing the dynamic behaviour of the sequence. These models are usually constructed with a particular assumption on the properties of the represented surface, such as static topology, articulated motion, or the sequence's representation of a human performance. The model-based methods are fairly similar to the ME-based; however, the main difference is that instead of using only the information from the previous frame (or previous keyframe), they exploit global temporal information to some extent. An example model-based compression pipeline is shown in Figure 5. Usually, the method first constructs the model considering all the frames (although some methods construct the model on the fly while encoding individual frames). The model and the first frame of the sequence are then encoded. For each subsequent frame, the model is used to construct a specific reference shape \mathcal{R} (not necessarily a mesh or a point cloud) from the pre-

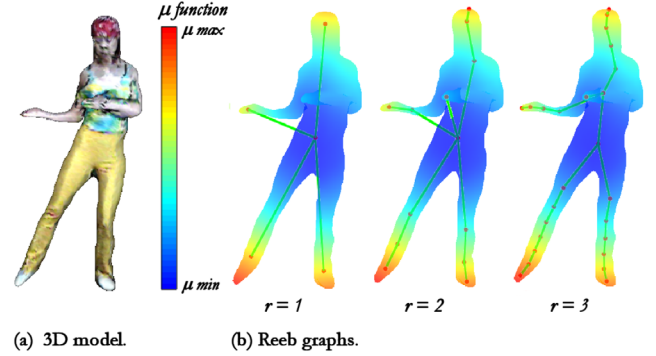


Figure 6: Example of the geodesic integral function $\mu(v)$ on a human 3D model and its corresponding multi-resolution Reeb graphs. Source [TSM07].

vious frame (or previous keyframe). \mathcal{R} is then used for geometry prediction or as a replacement of the current frame.

The model-based methods can be further divided into categories based on the temporal model they use. The most prominent approach for obtaining a temporal model in TVM compression is *tracking*. Its goal is to obtain a temporally consistent representation of the original surface. This is usually done by gradually deforming some template (e.g. a *surface* [LAGP09] or a *skeleton* [MYA08]) to align it with subsequent frames. The models obtained by tracking provide a certain insight into temporal correspondences. However, different models might capture different underlying information that could be used to reduce the redundancy of the data, for example, the shape or the topological structure, without describing the relations between the frames [ZGT23]. Note that many temporal models (e.g. a tracked surface) are themselves a reduced representation of the mesh sequence. For the sake of simplicity, only the methods that directly address compression are discussed in this section.

The first model-based method for TVM compression was proposed by Yang et al. [YKL06]. It replaces the original sequence with a dynamic mesh obtained by tracking the remeshed first frame. The remeshing is performed by simplification using the quadric error metric [GH97] followed by butterfly subdivision projected on the original surface. This creates multiple levels of detail. The sequence is then encoded progressively by exploiting the temporal coherence in an uplifting scheme.

Tung et al. [TSM07] utilized augmented multi-resolution Reeb graphs [TS05] of geodesic integral function

$$\mu(v) = \int_{\mathcal{S}} g(v, s) dS, \quad (1)$$

where g is the geodesic distance between two points on a surface \mathcal{S} , to capture the topological properties of the frames. Figure 6 shows $\mu(v)$ alongside its corresponding multi-resolution Reeb graphs visualized for a human 3D model. Considering how the graph nodes are connected and whether they are located at the local extrema of the generating function, the authors show that it is possible to manually devise rules which allow their tracking if the underlying ground truth topology is known. The original mesh frames are replaced by

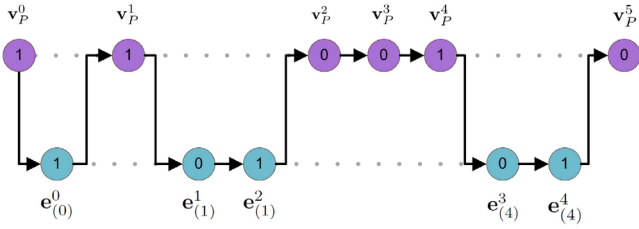


Figure 7: Directed graph used to encode temporal correspondences between reference vertex v_p^j and residual vectors $e_{(i)}^j$ in the method proposed by Doumanoglou et al. [DAZD14]. Values inside the nodes denote the emitted symbols for a given node. Source: [DAZD14].

the first frame deformed by this tracked graph structure. To accommodate for the fact that the ground truth topology is usually unknown, the approach was later generalized [TM12] by incorporating a so-called *topology dictionary*. It clusters frames according to the similarity of the underlying Reeb graphs. Frames in each cluster are then replaced by a deformed representative frame (one that is the most similar to all the frames in the cluster).

Maeda et al. [MYA08] proposed a compression method for human TVMs that replaces the sequence with the first frame deformed by a tracked skeleton. In a subsequent work [NYA10], the method was further improved using adaptive simplification and prediction of the first frame by a reference human triangle mesh to allow transmission and displaying of such data on consumer-grade mobile devices at the time (the year 2010).

The method proposed by Doumanoglou et al. [DAA*14, DAZD14] also uses a tracked skeleton as a temporal model. In each frame f of the sequence, each vertex is assigned to the closest bone. Each bone b_f^j has a rotation matrix \mathbf{R}_f^j assigned that describes its orientation. Its position is determined by the position of its parent joint $\mathbf{j}_f^{(i,0)}$. The method aligns the skinned meshes of the current frame t and a selected intra-coded keyframe k . First, the frames are partially aligned by applying orientations and positions of the predicted frame to the keyframe. Any vertex \mathbf{x}_k from the keyframe assigned to bone b_k^i is transformed as follows:

$$\hat{\mathbf{x}}_k = \mathbf{R}_t^i (\mathbf{R}_k^i)^{-1} (\mathbf{x}_k - \mathbf{j}_k^{(i,0)}) + \mathbf{j}_t^{(i,0)}.$$

Then, the alignment is improved by applying the ICP algorithm [BM92] on each bone separately. Resulting rigid transformations are encoded. The aligned intra-coded frame is then used for vertex position prediction similarly to the prediction of Yamasaki and Aizawa [YA10]; however, the authors proposed an improved technique to encode correspondences during a traversal of a directed graph formed over reference and predicted vertices (see Figure 7). The method can optionally preserve the original connectivity or the compression rate can be further improved by replacing inter-coded frames with aligned intra-only encoded keyframes. In the original work, the last intra-only frame was always selected as a reference. The authors have also experimented with selecting the intra-only frames for prediction based on skeleton matching criteria and periodicity detection [DAA*14], which in some cases led to an improvement in terms of rate-distortion performance.

Zaghetto et al. [ZGT23] have recently filed a patent on a method that replaces the mesh sequence by a neural representation using occupancy networks [MON*19]. Given a coarse point cloud \mathcal{P} sampled over the original surface, the neural occupancy function $f_\theta(\mathbf{x}|\mathcal{P})$ estimates the probability of a point \mathbf{x} lying inside the volume enclosed by the surface. The surface is reconstructed using a surface extraction algorithm, such as *marching cubes* [LC87]. Thus, a coarse point cloud \mathcal{P}^f is sampled over each frame. On the encoder side, the point clouds and the frames are used to train the occupancy network. Only the point clouds and the weights of the occupancy network are encoded. The proposition of the method seems promising; however, since it was published through a patent, no performance evaluation was given.

Hoang et al. [HCNC23] introduced a method for compression of human TVMs based on Embedded deformation technique [SSP07]. They first divide the TVM into small subsequences. The first frame in each subsequence (denoted I-frame) is encoded using Draco [GHS*18]. Any other frame is then replaced by the previous frame deformed using the Embedded deformation model. Having a set of key nodes of the Embedded deformation model, each assigned a position \mathbf{n}_i , a rotation matrix \mathbf{R}_i and a translation vector \mathbf{t}_i , the deformed position $\hat{\mathbf{x}}_i$ of a vertex \mathbf{x}_i from the previous frame can be found as

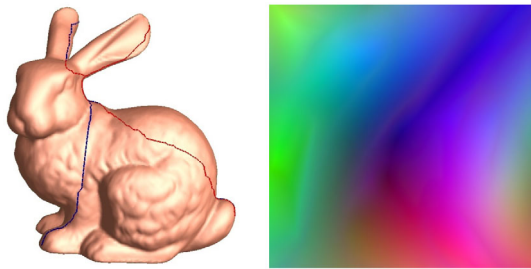
$$\hat{\mathbf{x}}_i = \sum_{j \in N(i)} w_{ij} (\mathbf{R}_j (\mathbf{x}_i - \mathbf{n}_j) + \mathbf{t}_j + \mathbf{n}_j),$$

where $N(i)$ is set of k nearest deformation nodes and w_{ij} is the influence weight based on the Euclidean distance between the node and the deformed point. The authors correctly highlight that the resulting compressed data size depends primarily on the number of key nodes. To address this, they have proposed to optimize both the number of key nodes and the reconstruction quality simultaneously using an optimization technique called *Alternating direction method of multipliers* (ADMM) [PB14, Section 4.4].

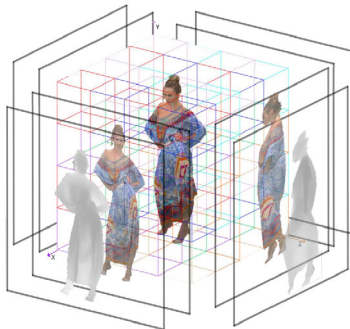
6. Video-Based Methods

Video compression methods are well-known for their efficiency in terms of exploiting temporal coherence. For this reason, several methods for TVM compression have focused on mapping the information contained in each frame into the image domain and encoding such data as a video. These methods perform notably well on textured mesh sequences since the video is a natural representation of time-varying texture information. The main challenge in video-based TVM compression is to find a mapping that produces the most temporally coherent images. Based on the type of mapping approaches, the video-based methods can be further divided into two categories.

- *Parameterization-based methods* usually store the geometry as a sequence of *Geometry images* (GI) [GGH02] (positions in 3D as RGB values in the texture, see Figure 8a), often also referred to as *Geometry videos* (GV) (although GV was initially a term coined to sequence of GIs used to compress dynamic meshes [BnSM*03]).
- *Orthogonal-projection-based methods* project surface patches onto certain projection planes and then represent the geometry using occupancy information (which pixel is part of a certain patch)



(a) Parameterisation. Source: [GGH02]



(b) Projection. Source: [ZMXLL21]

Figure 8: The two main approaches of mapping mesh data onto the 2D domain.

and the distance from the plane, similarly to depth images (see Figure 8b) [ZMXLL21].

To achieve temporal coherence of images, the parameterization methods focus on the temporally consistent cutting of the mesh frames before mapping them onto the 2D domain, while projection-based approaches focus on the temporally consistent placement of patches. Note that the placement of geometry is not required to be perfectly consistent, since the video codec usually also utilizes ME in the video domain.

The first authors, who considered video compression for TVMs, were Habe et al. [HKM04]. Their parameterization-based method attempts to place the cut path in places with lower texture complexity. The complexity for a certain surface patch s is computed as

$$T(s) = \frac{1}{A_s} \int_s \sqrt{d_x(p)^2 + d_y(p)^2} dp,$$

where A_s denotes the area of the patch, p is a point on the patch, and $d_x(p)$, $d_y(p)$ are the spatial differentials of the texture function at p . When considering whether to include a certain edge to a cut path, the method examines the texture complexity of its adjacent triangles. Their main motivation is that the largest texture distortion arises near the cut path. If the cut path leads through parts of complex texture, the distortion would be more visible. Although not verified experimentally, the authors also hint that such a cut path could be to some extent temporally consistent.

For compression of TVMs representing facial expressions extracted from motion data (see Figure 9(a)), Xia et al. [XHQ*10] proposed a so-called expression-invariant parameterization (EIP). The

method first cuts the mesh \mathcal{M} along geodesics between the eyes and the mouth to obtain a surface with two boundary curves (an outer boundary $\partial\mathcal{M}_0$ and an inner boundary $\partial\mathcal{M}_1$). These curves are mapped consistently into the parametric domain using arc-length parameterization. Then, harmonic function f on the mesh is constructed by solving the Laplace equations with Dirichlet boundary conditions:

$$\Delta f = 0$$

$$f(u) = 0, u \in \partial\mathcal{M}_0,$$

$$f(v) = 1, v \in \partial\mathcal{M}_1.$$

Analogously, a harmonic function g is also obtained in the parametric domain. For each point $v \in \partial\mathcal{M}_1$, an integral curve α_v is traced by following the gradient flow ∇f from it. A corresponding integral curve β_v can be traced by following gradient flow ∇g from a point in the parametric domain corresponding to v . Finally, points alongside α_v and β_v can be identified with each other, yielding the resulting parameterization. This process is depicted in Figure 9. The authors have proposed multiple methods based on EIP, each differing only in the way, how the resulting geometry images are encoded [XHQ*10, XQH*12, HCH*12, HCHMT13, HCH*13, HCZ*14].

A more sophisticated way of cutting the mesh to obtain temporal coherence was proposed by Tung and Matsuyama [TM13]. They track the local extrema of the geodesic integral function (see Equation 1) and construct the cut as the shortest path between them (see Figure 10). Authors claim that if selected properly, such points should remain consistent unless a change in topology occurs.

A slightly different approach to compressing TVM as a sequence of GIs was proposed by Hou et al. [HCHMT14]. Instead of direct planar mapping, the method maps frames to a polycube [GXH*13], which is cut at predefined edges and flattened. Although the structure of the polycube must be known before encoding, temporal coherence can be achieved by mapping consistently tracked salient points (e.g. features of the human body) onto the same positions in the 2D domain (see Figure 11). In the original method, the video was decomposed into a low-rank approximation representation. Some improvements were achieved by representing the sequence as a linear interpolation between a set of keyframes, which were also reordered to increase their temporal coherence [HCMTH15].

Gao et al. [GWW23] studied ways to incorporate registration to improve the temporal coherence of GIs. They proposed two approaches. The first approach performs the registration directly on GIs. It uniformly samples a subset of pixels of the reference frame and for each sampled pixel, it finds a corresponding pixel in the current frame as the one with the closest (x,y,z) value. These samples then serve as control points for Thin plate spline (TPS) transformation [HZW09] that maps their positions to the ones in the current frame. Once the parameters for such a transform are found, it is applied globally on all pixel values. The registered GI replaces the original GI of the current frame. The second approach performs the TPS-transformation-based registration on original mesh frames. It then constructs the coded GI by parameterizing the reference mesh. The second approach can also be extended to support two reference meshes.

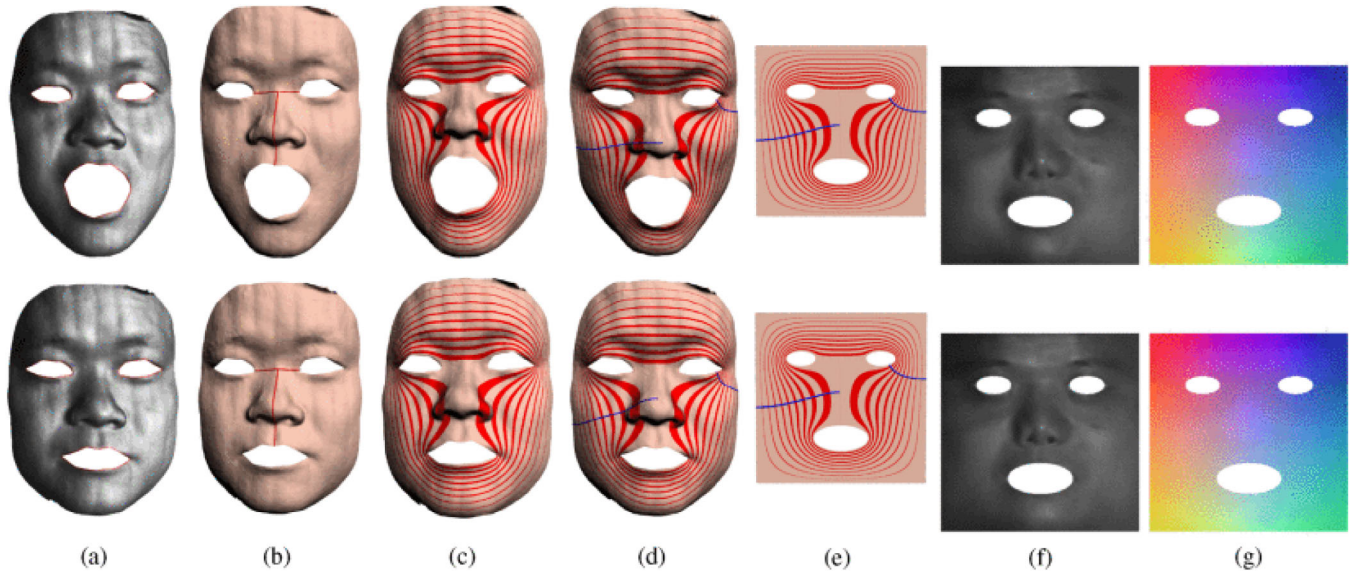


Figure 9: Expression-invariant parameterization (EIP). (a) Input data, (b) Performed cut, (c) Harmonic function with Dirichlet boundary conditions, (d) Example integral curves, (e) Corresponding curves in the parametric domain, (f) Parameterization, (g) Geometry image. Source: [XQH*12].

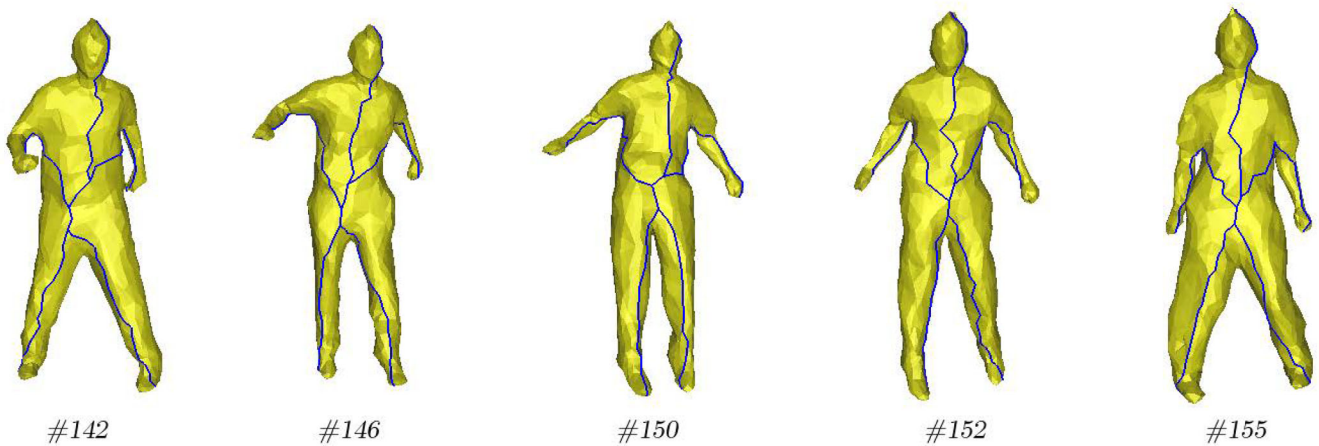


Figure 10: Temporally consistent cut between local extrema of the geodesic integral function. Source: [TM13].

6.1. MPEG standardization efforts

The most recent advancements in TVM compression can be mainly attributed to standardization efforts of the *Moving Picture Experts Group* (MPEG), specifically the *MPEG Coding of 3D Graphics and Haptics* working group (ISO/IEC JTC 1/SC 29/WG 7 - MPEG 3DG). The interest of MPEG in the compression of dynamic surfaces dates back to as early as 2007 when they adopted the MPEG-4 FAMC standard [MSK*08] for dynamic mesh compression. After the successful standardization efforts in the compression of dynamic point clouds, the MPEG switched focus to TVMs. Since the development of MPEG point cloud codecs is crucial for understanding the choices made when standardizing TVM compression, we will briefly discuss it in the following section.

6.1.1. Point cloud compression

Before the MPEG standardization efforts in point cloud compression, DPC compression was considered an open problem, with only a few methods proposed. Given the classification considered in this survey, most of the DPC compression methods were based on the prediction of spatial data structures [KBR*12, TCF16, GdQ17] or motion estimation [DFS*12, CK12, MBC16].

MPEG 3DG identified the need for an efficient compression method for point cloud data in general (including dynamic point clouds). Exploratory work started in 2014 [SPB*19]. This led to the call for proposals in early 2017 [ISO17]. As a result, 13 proposed solutions were collected from various industry and research

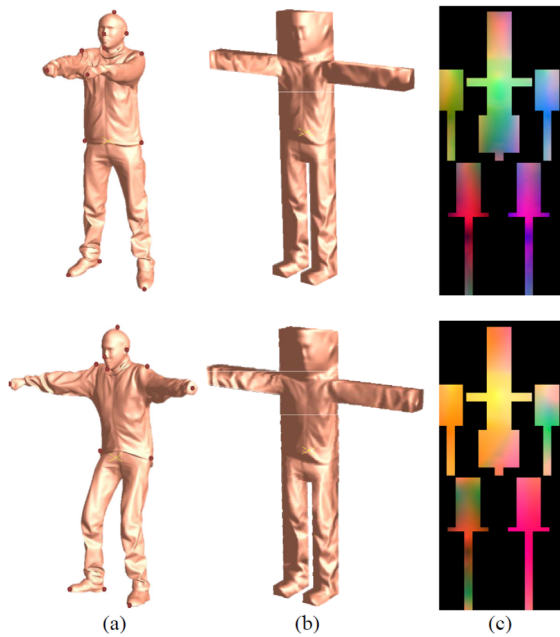


Figure 11: The process of obtaining a temporally coherent geometry video using polycube parameterization. (a) Salient points. (b) Polycube parameterization. (c) Geometry video. Source: [HCHMT14].

contributors and three different test model cases were identified: TMC1 for static data (e.g. cultural heritage), TMC2 for dynamic data, and TMC3 for dynamically acquired data (e.g. LiDAR). Eventually, due to the similarities in the approaches, TMC1 and TMC3 were merged to form TMC13, which led to the development of *geometry-based point cloud compression* (MPEG G-PCC), the ISO/IEC 23090-9 standard published in March 2023 [ISO22]. The

method evolved from TMC2 is called *video-based point cloud compression* (MPEG V-PCC), the ISO/IEC 23090-5 standard published in June 2021 [ISO20].

An encoding diagram of MPEG V-PCC [ISO20] is shown in Figure 12. The standard uses a video-based approach using orthogonal geometry projection onto planar patches. First, the point normals are estimated. Each point is assigned to one of the predefined planes around the frame (e.g. planes that form the axis-aligned bounding box) with the closest normal. The points are then clustered by grouping neighbouring points with similar orientations of normals to create patches. Patches are then projected onto the corresponding plane and assigned positions in the parametric domain by *patch packing*. In the first frame in a group of frames, the patches are sorted by size and incrementally inserted into the image in the first empty place found in the raster scan order, considering eight different rotations of the patch. For the rest of the frames, patches are matched using intersection over union (IOU) and placed in similar positions to achieve temporal coherence. Three different images are generated: depth (distance of a point to the projection plane), occupancy (information on whether the pixel contains geometry information, which is required due to the complex shape of the patch) and attribute (e.g. colour). Note that two points in the same patch can be projected onto the same position. In this case, V-PCC allows encoding multiple images to allow lossless coding. Occupancy information is usually encoded with reduced resolution, while depth and attribute images are padded with smooth transitions between values. The method allows additional smoothing in post-processing to reduce the gaps between patches caused by quantization.

An encoding diagram of MPEG G-PCC [ISO22] is shown in Figure 13. The MPEG G-PCC exploits the fact that point clouds are usually stored in voxelized form – original point positions are replaced by a cubic voxel grid (often represented by an octree) sampling occupancy values (a voxel is occupied if it contains at least one point). The standard supports three coding modes: octree entropy coding, trisoup and geometry prediction. In octree entropy

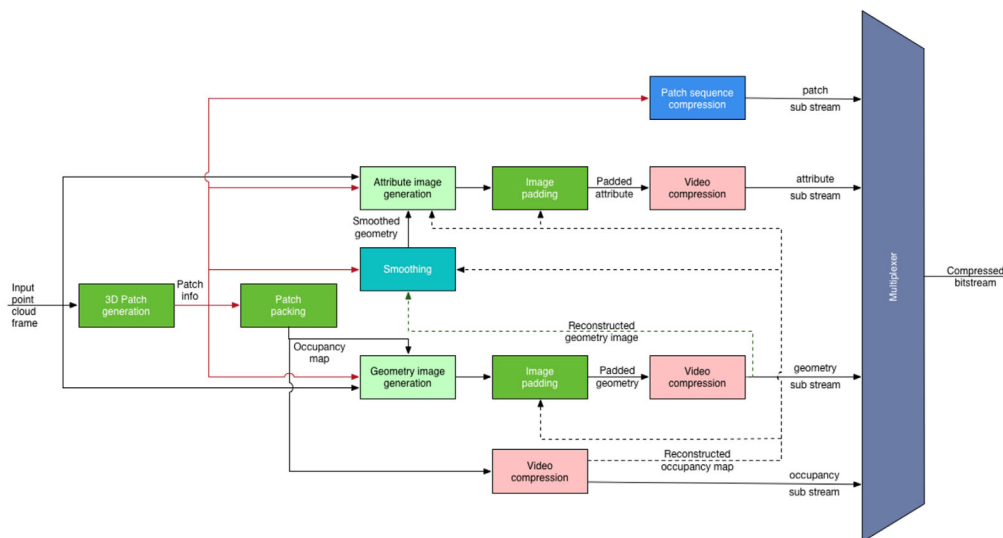


Figure 12: An encoding diagram of the MPEG V-PCC standard. Source: [ISO20].

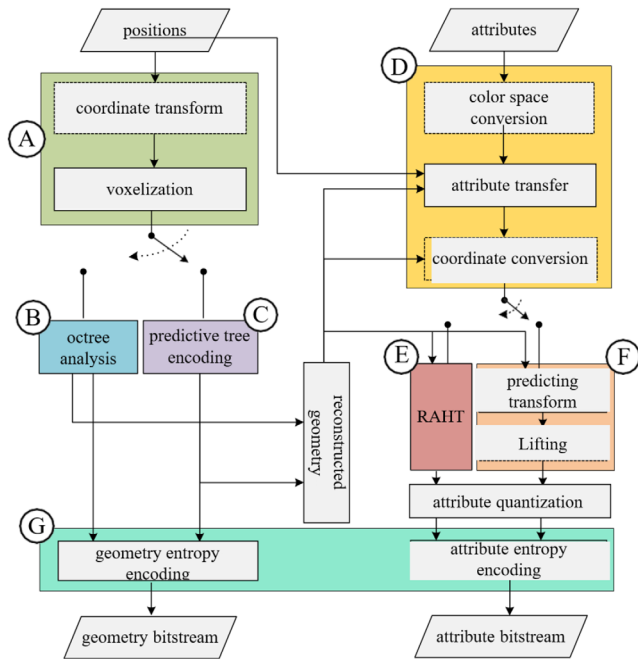


Figure 13: An encoding diagram of the MPEG G-PCC standard. Source: [ISO23a].

coding mode, the occupancy values of the octree are encoded during a breadth-first traversal using *Optimal Binarisation with Update On-the-Fly* (OBUF) strategy, which selects an optimal entropy coder from a list of coders based on neighbouring occupancy values. The neighbouring occupancy values are also used for prediction. The trisoup mode also uses octree coding but its tree has a limited depth. Leaf nodes are replaced by a series of unconnected triangles that approximate the surface in multiple nodes. In the geometry prediction mode, a prediction tree is constructed over the point cloud, where the position of a point is predicted by its hierarchical predecessors. Although the first edition of MPEG G-PCC did not contain any means of inter-frame prediction, there is currently a second edition of the standard under construction that will include inter-frame techniques such as motion compensation, temporal context modelling and inter-frame occupancy prediction [ISO23b].

Simultaneously with the MPEG efforts in point cloud coding, there was also an ongoing standardization in the field of immersive video coding (multi-view video + depth) which resulted in the *MPEG Immersive Video* (MIV) standard [ISO21b]. Given the similarities with the data coded in V-PCC (depth + occupancy + attributes), a generic volumetric video coding standard called *Visual volumetric video coding* (V3C) was established, covering the common parts of the two standards.

The development of both MPEG point cloud coding standards has significantly shifted the landscape of DPC compression. Either participating in the initial call for proposals [SSFSH19] or proposing various improvements over MPEG V-PCC [LYTC19, LLZ*20, KIRK20, CTPZ20, SL22, WWCF24], the field saw an unexpected increase in popularity of video-based methods. Initially, the MPEG G-PCC method served as a baseline for many inter-frame

prediction methods for coarsely sampled point clouds [PMR20, RPM21, KT22]. More recently, the inclusion of inter-prediction in the codec has also inspired multiple research works [JLS*21, LLLL22, GYWG22].

6.1.2. MPEG V-DMC

The idea that MPEG V-PCC could also be used to code TVMs was initially explored by Faramarzi et al. [FJB20]. They used it to encode the geometry of the frames, which was interpreted as a point cloud. For connectivity coding, they incorporated Edgebreaker [Ros99] or TFAN [MZP09] followed by encoding a vertex permutation map. Graziosi [Gra21] later improved this approach by adapting it for sparse meshes and storing the connectivity as a 2D mesh. A similar pipeline is also described in Sony's recent patent [GZT22], which also describes an additional way of storing connectivity using triangle rasterization.

Having the technology for storing dynamic geometric data using video at hand, the focus of MPEG 3DG eventually started to shift towards other fields, where it could also be employed. At the 136th meeting of the MPEG in October 2021, a call for proposals on TVM compression was issued [ISO21c]. The potential authors were strongly encouraged to incorporate the V3C standard [ISO21a]. Five companies contributed to this call for proposals: InterDigital, Nokia, Tencent, Sony and Apple [CJLR22b].

In terms of geometry, the approach proposed by InterDigital [MKG*22] is intra-only. It simplifies the frames using quadric error metric [GH97] and encodes them using Google Draco [GHS*18]. The temporal coherence is exploited only in texture data. Each mesh is segmented into patches which are parameterized using *Boundary First Flattening* [SC17] to obtain local UV coordinates. All the patches are then organized into a regular grid in a global UV coordinate system. Intra- and inter-frame reorganization of patches follows to minimize sharp transitions between neighbouring tiles and to place patches of similar RGB values in time at the same tiles.

The response from Nokia [AMI*22] encodes geometry quite similarly to the MPEG V-PCC [ISO20] method. Their patch-based approach performs temporal patch alignment inside a group of pictures by packing patches of similar average positions in 3D to similar areas in the image domain. They also combine the depth and occupancy images into a single image in YCbCr 4:2:0 format, where the depth is contained in the luma channel and occupancy in the chrominance.

The approach proposed by Tencent [HZT*22] uses a *Multi-chart Geometry Image* [SWG*03] representation. They assume that a temporally coherent parameterization is already known before coding and focus mainly on the means of preserving the watertightness of the input meshes. To this end, they find boundary vertices of all the patches and encode them separately including their UV and XYZ coordinates (in predictive coding) and the information to identify the corresponding vertices at neighbouring patches.

As of the time this paper was being written (December of 2024), to the best of our knowledge, no document directly describing the pipeline proposed by Sony was released to the public. It was only

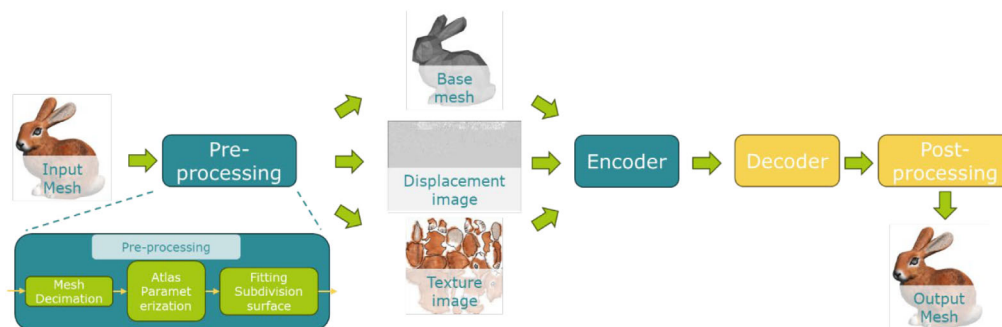


Figure 14: A diagram of the MPEG V-DMC standard. Source: [Cao23].

briefly summarized in an overview paper by Choi et al. [CJLR22a]. We can only assume from this text, that the method is probably very similar to the approach proposed in Sony’s recent patent [GZT22].

The method that was proposed by Apple [MKT*22] best fits into the class of model-based approaches when considering our classification, although it also incorporates the V3C technology. Their preprocessing of the sequence consists of a simplification followed by a midpoint subdivision projected to the original surface. The simplified mesh (called *base mesh*) is then encoded using Google Draco [GHS*18]. The displacements between the subdivided mesh and the original surface are encoded using wavelet transform. The wavelet coefficients are encoded as a video. The temporal coherence can be exploited by replacing the current base mesh with the base mesh of a reference frame if possible. Then, instead of coding the whole base mesh, only a motion field is required.

Out of the five proposals, the best compression performance was achieved by Apple and InterDigital [CJLR22a]. At the 138th meeting, the proposal by Apple was selected as a basis for the subsequent standardization process [CJLR22b]. In April of 2024, at the 146th MPEG meeting, the standard reached the *Committee Draft* stage [MPE24]. This implies that the standard’s initial draft is completed and sent for the first external review. The diagram of the standard is shown in Figure 14. Considering exploiting temporal coherence, the standard does not differ significantly from the original proposal. While the original proposal heavily relied on external tools (e.g. for base mesh compression, subdivision or parameterization), in the standard, most of them were incorporated into the coding method [CJLR22b]. The current version of the V-DMC standard test model that can be used for compression performance comparison is available at <http://mpegx.int-evry.fr/software/MPEG/dmc/mpeg-vmesh-tm> (to access, one must be accredited by a standardization National Body).

Individual contributions to MPEG standards are usually private. However, some contributions or methods inspired by MPEG V-DMC were published as scientific papers. Most focused on encoding displacements between the base mesh and encoded frame [BKP*23, KKK23, NKK23b, SRR*23, KBS24]. Particularly, Nishimura et al. [NKK23a] proposed to perform inter-prediction of the wavelet coefficients (note that temporal correspondences between displacements are given implicitly by shared structure between the last intra-coded frame and inter-coded frames). Zou et al. [ZZCY23] pro-

posed a method to predict texture coordinates in V-DMC given the already known mesh geometry.

Jin et al. [JXK23] developed a technique to achieve temporally-consistent simplification. Their method first performs an ARAP-based deformation [SA07] to remesh the coded frame so that its structure matches that of the reference frame. Then, both the reference and the remeshed frames are simultaneously decimated considering the combined error quadrics from both frames. The authors also proposed a different method [JXK24] that instead performs frame-by-frame tracking driven by the embedded deformations approach [SSP07] to find a mapping between the frames. Next, this deformation is used to propagate the base mesh of the first frame in time. Both techniques do not require any additional changes in the V-DMC pipeline, since simplification is performed in preprocessing.

As a competing method to the one implemented in the standard, Zou et al. [ZHTY23] proposed an MPEG V-PCC-based approach addressing its compression performance issues on sparse data. It splits the vertex position data into low- and high-precision parts. The low-precision data is stored using the MPEG V-PCC projection approach, while high-precision data is stored in raw V3C patches.

7. Existing TVM Datasets

Time-varying meshes, despite all their benefits, are still quite uncommon, mainly due to their size. Consequently, there is only a limited number of publicly available datasets. Table 2 contains a summary of this section.

To the best of our knowledge, the earliest publicly available TVM sequences were published to showcase the capabilities of dynamic shape reconstruction systems. Methods of Han et al. [HYA07, HYA08, FHYA10], Yamasaki-Aizawa [YA10] and Maeda et al. [MYA08, NYA10] all used results of the system designed by Tomiyama et al. [TOKI04]. Tung et al. [TSM07, TM12, TM13] used data obtained by the system of Matsuyama et al. [MWTN04]. Although not used in any TVM compression method, Vlasic et al. [VPB*09] made their reconstruction results publicly available alongside the multi-view image data from which they were generated. There are also commercial mesh sequences, that human performance capture companies, namely

Table 2: Overview of existing TVM datasets. The datasets are in the order in which they appear in Section 7. Column Ref. contains a reference for the given dataset. Column Purpose shows the main purpose of the given dataset. Column # Seq. shows the number of sequences in the given dataset. Columns Tex., resp. Col. show whether the dataset contains texture, resp. vertex colour data. Column Availability shows whether the dataset is available online or not, or getting it requires sending a request to its authors. Symbol meaning: \checkmark = Yes, \times = No.

| Dataset | Ref. | Purpose | # Seq. | Tex. | Col. | Availability |
|------------------|-------------------|-----------------|--------|--------------|--------------|--------------|
| Tomiyama et al. | [TOKI04] | Capture results | ? | \times | \checkmark | Unavailable |
| Matsuyama et al. | [MWTN04] | Capture results | ? | \checkmark | \times | Unavailable |
| Vlasic et al. | [VPB*09] | Capture results | 4 | \times | \times | Public |
| Volograms | [PZA*21, PAO*22] | Commercial | 3 | \checkmark | \times | Public |
| Renderpeople | [Ren21] | Commercial | 1 | \checkmark | \times | Public |
| Tung | [TM12, Tun12] | Computer vision | 2 | \checkmark | \times | On request |
| DFAUST | [BRPMB17] | Computer vision | 129 | \times | \times | On request |
| CAPE | [PMPHB17, MYR*20] | Computer vision | ? | ? | ? | On request |
| Hi4D | [YGK*23] | Computer vision | 100 | \checkmark | \times | On request |
| 4DHumanOutfit | [ABB*23] | Computer vision | 1617 | \checkmark | \times | On request |
| VCL ITI dataset | [Vis13] | Compression | 20 | \times | \checkmark | Public |
| V-DMC data | [ISO21d] | Compression | 8 | \checkmark | \checkmark | On request |
| TDMD | [YJD*23] | Perception | 303 | \checkmark | \times | Public |

Volograms [PZA*21, PAO*22] and Renderpeople [Ren21], made publicly available mainly to spark the interest of potential customers. Sequences in both datasets have already been preprocessed to introduce temporal coherence in structure; however, in general, the frames do not have the same connectivity.

Another purpose of available TVM datasets is to serve as a benchmark for shape analysis and computer vision tasks. To our knowledge the earliest dataset for computer vision was made available alongside the topology dictionary method of Tung-Matsuyama [TM12, Tun12] and its purpose is to test methods for motion understanding. To benchmark registration, tracking and temporal models, Bogo et al. [BRPMB17] have published the *D-FAUST* dataset. It contains 129 TVMs representing the human performances of multiple people, as well as tracked data. The *CAPE* dataset [PMPHB17, MYR*20] of human models in various clothing contains raw scans the authors promise to send upon request. The *Hi4D* dataset [YGK*23] consists of 100 sequences, each capturing a pair of subjects under close interaction. This is quite useful for testing the robustness of methods against topological noise. The *4DHumanOutfit* [ABB*23] dataset consists of 20 actors, dressed in 7 outfits each, and performing 11 motions exhibiting large displacements in each outfit.

Finally, there are datasets created mainly to benchmark TVM compression. The dataset published at VCL-ITI website [Vis13] consisted of overall 20 sequences, obtained by 3 different capture setups and reconstructed using either *Zippering* [AZD13] or *Poisson surface reconstruction* [KBH06]. It was used by Doumanoglou et al. [DAA*14, DAZD14] and Hoang et al. [HCNC23]. Unfortunately, as of June 2024, links to individual sequences are no longer working. As part of the call for proposals for MPEG V-DMC, MPEG 3DG has compiled a dataset of TVMs used to evaluate the compression performance of individual proposals [ISO21d]. This dataset consists of eight sequences. Most were constructed from point cloud data [dHMC17, SARG21], but it also contains sequences with partially tracked structure that were provided by Volucap [SARG21] and Vologram [PZA*21]. Most recently, Tencent has

created another dataset called TDMD by applying different types of distortion to MPEG testing sequences to test their impact on human perception [YJD*23]. Alongside the data, they also published the results of a large-scale subjective experiment. In the future, this dataset will very likely serve as a benchmark for newly proposed human perception-oriented distortion metrics.

8. Relative Comparison of Compression Performance

Comparing the efficiency of compression algorithms is not a straightforward task. With lossy compression, the performance cannot be evaluated by a single number, such as the compression ratio, but is expressed by a rate/distortion (RD) curve, which records the amount of distortion at various data rates, since lossy compression algorithms usually allow influencing the data rate via so-called quantization parameters (QPs). When competing algorithms are compared, the relative position of their RD curves must be investigated. It reveals which of the algorithms provides a lower data rate at a comparable level of distortion; however, since the RD curves may cross, it is possible that one algorithm performs better than the other at a certain range of data rates and worse at others. Moreover, the relative position of the RD curves is strongly influenced by the character of the input data and by the particular metric used to evaluate the amount of distortion. It is well known that mechanistic metrics, such as Mean Squared Error or Hausdorff distance, correlate poorly with human perception of shape distortion, and therefore are mostly inappropriate when used to evaluate the performance of a lossy mesh compression algorithm, nevertheless, they are still commonly used for this purpose because of their simplicity.

It is difficult to infer an order of the methods in terms of compression performance only from the experimental results presented alongside individual methods. If the authors performed any compression performance evaluation, there were varying types of input data (e.g. synthetic, textured, manifold/non-manifold sequences), varying baseline methods and the results were reported in varying ways. See Tables 3, resp. 5, for details about how the compression

Table 3: Overview of reported experimental compression performance evaluations of structure-preserving methods. Column Data shows the type of data tested in experiments. Column Dataset shows datasets used in experiments. Column Competitor(s) shows which methods the given method was compared to. Column Distortion metrics shows metrics used to evaluate distortion.

| Method | Ref. | Data | Dataset | Competitor(s) | Distortion metrics |
|---------------------|----------|----------------|-------------------|------------------------------------|--------------------------------------|
| Patch ICP | [GSK03] | Synth. TVM, DM | Own data, Chicken | [TR98], [Len99, AKKH01, AM00] (DM) | ✗ |
| EBMA | [HYA07] | TVM | [TOKI04] | No motion vectors | RMSE, PSNR-Image |
| PCA-aligned patches | [YA10] | TVM | [TOKI04] | [HYA07, HYA08] | RMSE-Geom, RMSE-Col |
| Grid occupancy XOR | [HYA08] | TVM | [TOKI04] | [HYA07] | RMSE |
| | [FHYA10] | TVM | [TOKI04] | [HYA08] | RMSE |
| Per-bone ICP | [DAZD14] | TVM | [Vis13] | [MZP09] | RMSE-Chamfer, RMSE-METRO, PSNR-Image |
| | [DAA*14] | TVM | [Vis13] | [DAZD14] | RMSE-Chamfer |
| VPCC + Edgebreaker | [FJB20] | TVM | [ISO21d] | [GHS*18] | (lossless) |
| | [Gra21] | TVM | [ISO21d] | [GHS*18] | PSNR-Geom, PSNR-Y |

Table 4: Compression performance of structure-preserving methods. Reported data rates and distortions are for a single selected result. Columns Conn., Col., and Tex. show whether the reported data rate also considers connectivity, colour or texture information.

| Method | Ref. | Data rate | Distortion | Conn. | Col. | Tex. |
|---------------------|----------|-----------|-----------------------|-------|------|------|
| Patch ICP | [GSK03] | CR 143:1 | — | ✓ | ✗ | ✗ |
| EBMA | [HYA07] | 10bpfv | RMSE 0.29 cm | ✗ | ✗ | ✗ |
| PCA-aligned patches | [YA10] | 10bpfv | RMSE-Geom 0.05 cm | ✗ | ✗ | ✗ |
| Grid occupancy XOR | [HYA08] | 10bpfv | RMSE 0.07 cm | ✗ | ✗ | ✗ |
| | [FHYA10] | 10bpfv | RMSE 0.09 cm | ✗ | ✗ | ✗ |
| Per-bone ICP | [DAZD14] | 10bpfv | RMSE-Chamfer 0.44 cm | ✓ | ✗ | ✗ |
| | [DAA*14] | 10bpfv | RMSE-Chamfer 0.725 cm | ✓ | ✗ | ✗ |
| VPCC + Edgebreaker | [FJB20] | CR 7.59:1 | — | ✓ | ✓ | ✗ |
| | [Gra21] | 10bpfv | PSNR-Geom 67dB | ✓ | ✗ | ✓ |

was evaluated for individual structure-preserving, resp. structure-discarding methods and Tables 4, resp. 6, for selected experimental results. Nevertheless, some conclusions on the relative performance between the methods can be deduced.

For many TVM compression methods discarding the original structure of the frames, there were no comparing experiments conducted in terms of compression performance [HKM04, TSM07, MYA08, NYA10, TM12, ZGT23], or they were compared to their altered version with some part of the pipeline disabled [YKL06, XHQ*10, ZHTY23].

Although at the time of proposing their ME-based method, Han et al. [HYA07] did not perform any comparison with competing approaches, it was later shown to be outperformed by their method based on prediction of spatial structures [HYA08]. That approach was later compared with its multi-rate extension [FHYA10]. Although the multi-rate version performs slightly worse, it brings better control over the distortion of data. Finally, both single-rate methods proposed by Han et al. [HYA07, HYA08] were shown to be outperformed by the approach of Yamasaki and Aizawa [YA10]

and by one of the versions of EIP-based methods proposed by Xia et al. [XQH*12] in terms of geometry compression.

Video-based methods that use parameterization [TM13, HCZ*14, HCHMT14, GWW23] are often compared to the sequence of geometry images [GGH02] or an experiment is conducted on dynamic meshes and the results are compared to geometry video method [BnSM*03]. Although they usually outperform these approaches, it is difficult to use these results to infer their relative performance to other TVM compression methods.

The best way to demonstrate the effectiveness of the inter-prediction is by comparing the compression results to an intra-only approach (e.g. parallelogram prediction [TG98] or weighted parallelogram [VB13]). The first approach that was shown to outperform an intra-only method was already the earliest method for TVM compression [GSK03], which was compared to the approach of Taubin and Rossignac [TR98], albeit only on synthetic TVMs and only in terms of data rates. On real-world data, this approach is expected to be inefficient. The MPEG V-PCC-based methods proposed by Faramarzi et al. [FJB20] and Graziosi [Gra21] were

Table 5: Overview of reported experimental compression performance evaluations of structure-discarding methods. Column Data shows the type of data tested in experiments. Column Dataset shows datasets used in experiments. Column Competitor(s) shows which methods the given method was compared to. Column Distortion metrics shows metrics used to evaluate distortion.

| Method | Ref. | Data | Dataset | Competitor(s) | Distortion metrics |
|---------------------------------|-----------|----------------|--------------|------------------------------------|----------------------------|
| Semi-regular representation | [YKL06] | TVM, DM | Own data | Inter off, [YKL02, GK04] (DM) | Normalized METRO |
| Reeb-graph matching | [TSM07] | TVM, DM | [MWTN04] | — | Surface overlap error |
| Topology dictionary | [TM12] | TVM | [MWTN04] | — | Hausdorff MSE and PSNR |
| Skinned mesh | [MYA08] | TVM | [TOKI04] | — | ✗ |
| | [NYA10] | TVM | [TOKI04] | [MYA08] | ✗ |
| Per-bone ICP | [DAZD14] | TVM | [Vis13] | [MZP09] (lossless struct.) | PSNR-Image |
| Occupancy network | [ZGT23] | TVM | [ISO21d] | — | ✗ |
| Optimized embedded deformation | [HCNC23] | TVM, DM | [Vis13] | [DAZD14, MZP09] (lossless struct.) | RMSE-METRO |
| EIP geometry video | [XHQ*10] | Facial TVM, DM | Own data | Different video codec | PSNR-GI |
| | [XQH*12] | Facial GI | Own data | [XHQ*10, HYA07, HYA08] | PSNR-GI |
| | [HCH*12] | Facial GI | Own data | [XQH*12] | PSNR-GI |
| | [HCHMT13] | Facial GI | Own data | H.264 Intra | RMSE |
| | [HCH*13] | Facial GI | Own data | H.264 Intra, [XQH*12] | RMSE |
| | [HCZ*14] | Facial GI | Own data | [XQH*12, HCH*13, BnSM*03] | KG Error (DM metric) |
| Cut over local extrema | [TM13] | TVM | Remeshed DMs | [GGH02] | Hausdorff |
| Polycube geometry video | [HCHMT14] | DM | [VBMP08] | [BnSM*03] (DM) | KG Error (DM metric) |
| | [HCMTH15] | DM | [VBMP08] | [BnSM*03, QHC*11] (DM) | Rel. RMSE-METRO, MSDM2 |
| Registered geometry images | [GWW23] | Facial TVM | Own data | [GGH02] | PSNR |
| V-DMC Nokia | [AMI*22] | TVM | [ISO21d] | [GHS*18] | V-DMC metrics |
| V-DMC Tencent | [HZT*22] | TVM | [ISO21d] | [GHS*18] | V-DMC metrics |
| V-DMC Apple | [MKT*22] | TVM | [ISO21d] | [GHS*18] | V-DMC metrics |
| Inter wavelet coefficients | [NKK23a] | TVM | [ISO21d] | [MKT*22, NKK23b] | V-DMC metrics |
| Temporally-consistent remeshing | [JXK23] | TVM | [ISO21d] | MPEG V-DMC (unkn. version) | D2 of V-DMC metrics |
| Tracked base mesh | [JXK24] | TVM | [ISO21d] | MPEG V-DMC TM 4.0, [JXK23] | V-DMC metrics |
| Low-/High-precision split | [ZHTY23] | TVM | [ISO21d] | Low/High split turned off | D1 and D2 of V-DMC metrics |

both compared to the approach implemented in the Google Draco library [GHS*18]. Both achieve worse results, although the method of Graziosi [Gra21] achieved comparable performance in some experiments. The only current structure-preserving approach to outperform the intra-only method on real-world data is the one proposed by Doumanoglou et al. [DAZD14]. It was compared with the MPEG TFAN algorithm [MZP09] on non-manifold mesh sequences. Although it was not compared to the ME-based method of Yamasaki and Aizawa [YA10], we can consider this approach state-of-the-art in terms of structure-preserving TVM compression, since the ME-based method does not efficiently encode mesh connectivity. Both MPEG TFAN and the Per-bone ICP approach served as baseline for the Embedded-deformation-based method of Hoang et al. [HCNC23]. The method performs better at low data rates (5–8 bpfv). However, while their method discards structure, the comparison was done with results where the structure was preserved.

To this date, the most comprehensive compression performance comparison in this field was done during the process of evaluating the proposals for the MPEG V-DMC standard. As a baseline method, MPEG 3DGC selected Google Draco [GHS*18] for the mesh geometry and connectivity and HEVC [SOHW12] for textures. Both mechanistic (geometry – D1, D2, colour - Luma, Cb, Cr) [ISO21c] and perceptual distortions were considered, although the perceptual metric was based on metrics for images [WJB22]. For

each metric, relative performance was measured using a so-called Bjontegaard-Delta (B–D) rate [Bjo01], which computes the average percentual difference of rate-distortion curves over a specified interval of data rates. Out of the five proposals [ISO21c], only those proposed by Apple [MKT*22] and InterDigital [MKG*22] outperformed the baseline method on all criteria. Additionally, the approach of Tencent [HZT*22] performed better considering the distortion of colour information. The best performance was achieved by the proposal of Apple [MKT*22], which was selected as a base method for the future standard.

The proposed improvements over MPEG V-DMC [NKK23b, JXK23, JXK24] have also considered the same performance evaluation benchmark, except the one performing temporally-consistent remeshing [JXK23], where authors considered only a single geometry-based metric of the original ones and only reported bitrates for geometry. Unfortunately, since some report only B-D rates, it is impossible to infer relative performance between them, as each used a different MPEG V-DMC test model as a baseline.

8.1. Computational performance

Compression and decompression times and complexity are also important aspects to consider when comparing individual methods. Based on the scenario in which the method is used, there are

Table 6: Compression performance of structure-discarding methods. Reported data rates and distortions are for a single selected result. Columns Conn., Col., and Tex. show whether the reported data rate also considers connectivity, colour or texture information.

| Method | Ref. | Data rate | Distortion | Conn. | Col. | Tex. |
|---------------------------------|-----------|-------------------------------------|-------------------------------|-------|------|------|
| Semi-regular representation | [YKL06] | 1000 bytes/frame | 0.001 | ✓ | ✗ | ✗ |
| Reeb-graph matching | [TSM07] | 1.3 Mb, 600 frames 23k tris | 5% overlap error | ✓ | ✓ | ✗ |
| Topology dictionary | [TM12] | CR 5:1 | PSNR 75 dB | ✓ | ✗ | ✓ |
| Skinned mesh | [MYA08] | 1/# frames | — | ✓ | ✓ | ✗ |
| | [NYA10] | CR 5000:1 for 96 frames | — | ✓ | ✓ | ✗ |
| Per-bone ICP | [DAZD14] | 12 bpfv | PSNR 20.5 dB | ✓ | ✓ | ✗ |
| Occupancy network | [ZGT23] | — | — | ✗ | ✗ | ✗ |
| Optimized embedded deformation | [HCNC23] | 10bpfv | RMSE-METRO 0.2cm | ✓ | ✗ | ✗ |
| EIP geometry video | [XHQ*10] | CR 212:1 | PSNR 58.31 dB | ✗ | ✗ | ? |
| | [XQH*12] | CR 239:1 | PSNR 60.9 dB | ✗ | ✗ | ? |
| | [HCH*12] | CR 372.8:1 | PSNR 58.64 dB | ✗ | ✗ | ? |
| | [HCHMT13] | 1 bpfv | RMSE 0.16 [unkn. units] | ✗ | ✗ | ? |
| | [HCH*13] | 1 bpfv | RMSE 0.055 [unkn. units] | ✗ | ✗ | ? |
| | [HCZ*14] | 1 bpfv | KG Error 0.075% | ✗ | ✗ | ? |
| Cut over local extrema | [TM13] | 0.91 kBpf | Hausdorff 0.085 [unkn. units] | ✗ | ✗ | ✗ |
| Polycube geometry video | [HCHMT14] | 20 kbpf | KG Error 0.25% | ✗ | ✗ | ✗ |
| | [HCMTH15] | 20 kbpf | Rel. RMSE-METRO 0.0006 | ✗ | ✗ | ✗ |
| Registered geometry images | [GWW23] | 3.10 bpfv | PSNR 56.20 dB | ✗ | ✗ | ✗ |
| V-DMC Nokia | [AMI*22] | 1000 kbps | PSNR-Geom 49.5 dB | ✓ | ✗ | ✓ |
| V-DMC Tencent | [HZT*22] | 1000 kbps | PSNR-Geom 53.9 dB | ✓ | ✗ | ✓ |
| V-DMC Apple | [MKT*22] | 1000 kbps | PSNR-Geom 58 dB | ✓ | ✗ | ✓ |
| Inter wavelet coefficients | [NKK23a] | BD-rate (D2) -0.2% against [MKT*22] | — | ✓ | ✗ | ✓ |
| Temporally-consistent remeshing | [JXK23] | 5 Mbps | D2-PSNR 72.1 dB | ✓ | ✗ | ✓ |
| Tracked base mesh | [JXK24] | 5 Mbps | D2-PSNR 72.5 dB | ✓ | ✗ | ✓ |
| Low-/High-precision split | [ZHTY23] | BD-rate (D2) -58.65% | — | ✗ | ✗ | ✗ |

various requirements on how fast the method must encode or decode the data. For tele-immersion (teleconferencing in virtual or augmented reality), the encoder and the decoder must run in real-time to allow an interactive experience. For streaming TVM content over a network, compression can be performed in advance with a high-complexity algorithm, but decompression is still required to run in real time. This is often referred to as *asymmetric compression*. The least requirements on compression and decompression times are posed for methods when used for archiving, that is the data is stored in a compact form, to be rarely decompressed. Even so, a faster method is preferable over a slower one if their compression performance is comparable.

Simply comparing the times is not enough to infer relative performance. A method with a running time of around a minute reported 15 years ago would very likely outperform a method running around a minute on modern hardware. For this reason, we have also investigated, whether the authors reported the testing hardware alongside the compression and decompression times. Unfortunately, not every publication contains reported times or complexity analysis. To this end, we have attempted to identify probable bottlenecks of individual methods. We also investigated whether the authors claimed to utilize GPU computing or parallelism, which is also a key factor in the compression and decompression times. The findings of our investigation are summarized in Tables 7 and 8, respectively.

Among the structure-preserving TVM compression methods, only Gupta et al. [GSK03] and Doumanoglou et al. [DAZD14] in their first work reported compression and decompression times. The experiments on the computational performance of structure-preserving methods are so limited because these methods are best suited for archiving and usually do not aim at real-time performance. The algorithm of Gupta et al. [GSK03] relies on the ICP algorithm, requiring around 6–7 s to encode a single frame. However, the decoder runs considerably fast, allowing real-time decompression even on older hardware. The method proposed by Doumanoglou et al. [DAZD14, DAA*14] is fairly complex; however, in their first work, the authors claim to use both parallelism and GPU computing, resulting in nearly interactive encoding and decoding times. The authors claim the method achieves real-time transmission of TVM data even in the structure-preserving mode. It is, in fact, the only work to explicitly claim real-time encoding. The method presented in their subsequent work likely has comparable decoding times, and slightly slower encoding times since it also performs keyframe selection, but no direct comparison was provided.

Since the rest of ME-based methods [HYA07, YA10] and the ones based on the prediction of spatial structures [HYA08, FHYA10] are quite simple, we can expect these methods to run in real-time on modern hardware, although we must also consider additional compression/decompression time expected for connectivity compression. Han et al. [HYA07] listed testing hardware for their

Table 7: Computational performance of structure-preserving methods. Column Bottleneck shows the most computationally expensive parts of the method. Column Testing hardware shows the hardware setup used to measure the times. Column Compression contains compression times and whether the authors claimed the method performs compression in real-time. Column Decompression contains decompression times and whether the authors claimed the method performs decompression in real time. Column Acceler. shows whether authors claim the method supports or uses GPU computing or parallelism (Par.).

| Method | Ref. | Bottleneck | Testing hardware | Compression | | Decompression | | Acceler. | |
|---------------------|----------|--|--------------------------------|-------------|----|---------------|----|----------|------|
| | | | | Times | RT | Times | RT | GPU | Par. |
| Patch ICP | [GSK03] | Segmentation, ICP, Conn. coding | Pentium III (550 MHz) | 0.15 fps | ✗ | 400 fps | ✓ | ✗ | ✗ |
| EBMA | [HYA07] | Block matching, DCT | Pentium 4 (3.4 GHz), 2 GB RAM | — | ✗ | — | ✗ | ✗ | ✗ |
| PCA-aligned patches | [YA10] | Patch segmentation, PCA | — | — | ✗ | — | ✗ | ✗ | ✗ |
| Grid occupancy XOR | [HYA08] | Sampling | — | — | ✗ | — | ✗ | ✗ | ✗ |
| | [FHYA10] | Sampling | — | — | ✗ | — | ✗ | ✗ | ✗ |
| Per-bone ICP | [DAZD14] | Registration, Skinning, Conn. coding | i7-2700K, 8 GB RAM, GTX560 GPU | 5 fps | ✓ | 12 fps | ✓ | ✓ | ✓ |
| | [DAA*14] | Registration, Skinning, Keyframe selection, Conn. coding | — | — | ✗ | — | ✗ | ✗ | ✗ |
| VPCC + Edgebreaker | [FJB20] | VPCC, Conn. coding | — | — | ✗ | — | ✗ | ✗ | ✗ |
| | [Gra21] | VPCC, Conn. coding, Post-processing | — | — | ✗ | — | ✗ | ✗ | ✗ |

EBMA-based method but provided no times. The method of Ferreira et al. [FHYA10] is expected to run slightly slower than the original XOR-based method [HYA08] since it processes more levels of detail. The most computationally expensive part of video-based structure-preserving methods [FJB20, Gra21] seems to be the MPEG V-PCC codec. Its current version is reported to achieve real-time performance [GAM*21]. Thus, these methods possibly allow real-time or near-real-time compression and decompression. Of the two, the one proposed by Graziosi [Gra21] is expected to have slightly slower decompression times due to the post-processing of decoded frames.

The structure-discarding methods are more suited for streaming and tele-immersion as there are lower distortion quality requirements in these settings. This comes at the cost of stricter requirements on computational performance. As a result, the authors of these methods were more often concerned about compression and decompression times.

Model-based structure-discarding methods are usually asymmetric. Yang et al. [YKL06] did not report any compression or decompression times for their method. Its main bottleneck is the creation and tracking of the semi-regular shape. As a result, it is probably not capable of real-time encoding, but decoding should be fairly fast making it suitable for streaming. Another model-based method with no complexity discussion is the one proposed by Zaghetto et al. [ZGT23]. It is also asymmetric since it requires training the occupancy network. Although the authors did not make any claims about acceleration, they almost certainly use GPU computation for the training process. The main bottleneck of methods proposed by Tung et al. [TSM07, TM12] is the construction of Reeb graphs. It took around 25 s per frame to compute in their initial work. The subsequent tracking of the graph takes around 0.2 s per frame. The decoding process is much faster, probably achieving real-time per-

formance on modern hardware. Since the purpose of Topology dictionary [TM12] is not directly compression, their reported times are not directly about encoding/decoding. While the authors were able to improve on the speed of computation of the Reeb graph (around 15 s per frame), the method also needs 10 ms per pair of frames for similarity computation. The methods replacing the sequence with skinned mesh [MYA08, NYA10] have no reported encoding times. Regarding decompression, the first method achieves nearly interactive times (~ 10 fps including rendering), while the second runs in real-time even on fairly old hardware. The structure-discarding mode of the method proposed by Doumanoglou et al. [DAZD14] achieves slightly faster decoding times than the structure-preserving mode. Hoang et al. [HCNC23] claim their method is suitable for real-time processing, which, however, does not indicate real-time performance, but the ability to encode the data given only a few previous frames. They also claim some parts of the method can be parallelized.

Most of the video-based methods using parameterization do not report decoding times. This can be explained by the fact that the proposals focus on how the geometry is mapped onto the parametric domain or how the video is encoded, but the reconstruction of geometry from the image data is more or less the same as was in the original GV method [BnSM*03], which was reported to decode a geometry video of resolution 256 x 256 at the rate of 10.57 fps. The information on the computational performance of expression-invariant-parameterization-based methods is limited. The parameterization itself does not allow real-time encoding, since just one of its parts, the computation of geodesics, takes more than a few seconds per frame, according to Xia et al. [XHQ*10]. The only times for the whole encoding process were reported by Hou et al. [HCH*13]. In their experiments, they compared their iteration of the method with the second iteration of Xia et al. [XQH*12]. Encoding of a sequence of 100 frames took 1457 s for Hou et al. [HCH*13], and

Table 8: Computational performance of structure-discarding methods. Column Bottleneck shows the most computationally expensive parts of the method. Column Testing hardware shows the hardware setup used to measure the times. Column Compression contains compression times and whether the authors claimed the method performs compression in real-time. Column Decompression contains decompression times and whether the authors claimed the method performs decompression in real time. Column Accelerator shows whether authors claim the method supports or uses GPU computing or parallelism (Par.).

| Method | Ref. | Bottleneck | Testing hardware | Compression | | Decompression | | Accelerator | |
|---------------------------------|-----------|--|----------------------------------|-------------------|----|---------------|----|-------------|------|
| | | | | Times | RT | Times | RT | GPU | Par. |
| Semi-regular representation | [YKL06] | Remeshing, Registration | — | — | — | × | × | × | × |
| Reeb-graph matching | [TSM07] | Reeb graph, matching | Pentium M (1.60 GHz), 512 MB RAM | 25.2 s per frame | × | × | × | × | × |
| Topology dictionary | [TMI2] | Reeb graph, frame similarity | Core 2 Duo (3.00 GHz), 4 GB RAM | > 15 s per frame | × | × | × | × | × |
| Skinned mesh | [MYA08] | Skeleton extraction | Pentium D (3.40 GHz), 3 GB RAM | — | × | × | × | × | × |
| | [NYA10] | Skeleton extraction, morphing | Core 2 Duo (2.00 GHz) | — | × | × | × | × | × |
| Per-bone ICP | [DAZD14] | Registration, skinning, conn. coding | i7-2700K, 8 GB RAM, GTX560 GPU | 4 fps | ✓ | ✓ | ✓ | ✓ | ✓ |
| Occupancy network | [ZGT23] | Occupancy network training | — | — | × | × | × | × | × |
| Optimized embedded deformation | [HCNC23] | Embedded graph optimization | — | — | × | × | × | × | ✓ |
| EIP geometry video | [XHQ*10] | Laplace equations, Video coding | — | > few s per frame | × | × | × | × | × |
| | [XQH*12] | Laplace equations, Video coding | — | 21.49 s per frame | × | × | × | × | × |
| | [HCH*12] | Laplace equations, Low-rank and sparse decomposition | — | — | × | × | × | × | × |
| | [HCHMT13] | Laplace equations, Sparse representation | — | — | × | × | × | × | × |
| | [HCH*13] | Laplace equations, Bit-allocation problem | — | 14.57 s per frame | × | × | × | × | × |
| | [HCZ*14] | Laplace equations, Low-rank and sparse decomposition, Bit-allocation problem | — | — | × | × | × | × | ✓ |
| Cut over local extrema | [TMI3] | Feature point detection, parameterization | Dual-core CPU | ~ few s per frame | × | × | × | × | × |
| Polycube geometry video | [HCHMT14] | Feature point tracking, Truncated SVD | — | — | × | × | × | × | × |
| | [HCMTH15] | Feature point tracking, keyframe extraction, optimal parameter selection | 3.1 GHz CPU, 4 GB RAM | 14.5 s per frame | × | × | ✓ | × | × |
| Registered geometry images | [GWW23] | TPS-based registration, parameterization | — | — | × | × | × | × | × |
| V-DMC Nokia | [AMI*22] | Segmentation, Temporal patch alignment, Post-processing | — | — | × | × | × | × | ✓ |
| V-DMC Tencent | [HZT*22] | Identifying boundary vertices, Constrained delaunay triangulation | — | — | × | × | × | × | × |
| V-DMC Apple (VSMC) | [MKT*22] | Remeshing (Decimation + Subdivision) | — | — | × | × | ✓ | × | × |
| Inter wavelet coefficients | [NKK23a] | — | — | 0.37% of V-DMC | × | × | × | × | ✓ |
| Temporally-consistent remeshing | [JXK23] | Inter-surface mapping, Remeshing | — | 118.4% of V-DMC | × | × | ✓ | × | × |
| Tracked base mesh | [JXK24] | Base mesh tracking, Subdivision | — | 155% of V-DMC | × | × | × | × | × |
| Low-/High-precision split | [ZHTY23] | VPCC | — | — | × | × | × | × | × |

2149 s for Xia et al. [XQH*12]. Only the last iteration of the method is reported to use parallel computing [HCZ*14]. The video-based method proposed by Tung et al. [TM13] has a comparable complexity to their other methods [TSM07, TM12] as the detected feature points are related to nodes of Reeb graphs. They claim the method to encode a 1000 vertex mesh in orders of seconds. Hou et al. reported times only for their later version of the polycube-parameterization-based method [HCMTH15]. The encoding requires 14.5 s per frame, the authors claim the decoding process runs in real time. Gao et al. [GWW23] did not report any timings. Their rather complex encoding process is unlikely to run in real time. Zou et al. [ZHTY23] also did not discuss computational performance. Since their method is based on MPEG V-PCC it is expected to run in real time or near-real time, although it does not focus on meshing the reconstructed geometry which may also be computationally expensive.

Although the original call for proposals for MPEG V-DMC listed compression/decompression times as one of the criteria on which the methods will be evaluated, as of the time this paper is being written (December of 2024), to the best of our knowledge there is no publicly available document discussing the evaluation of computational performance of the individual proposals. In the proposal of Nokia [AMI*22], the authors noted that the patch segmentation process can be parallelized. The approach of Apple [MKT*22] is claimed to achieve real-time decompression. The proposed improvements over V-DMC [NKK23b, JXK23, JXK24] did not report absolute times but related their computational performance to the anchor software of the standard. Nishimura et al. [NKK23b] reported significant improvements, however, it seems that they report times only for displacement coding. Their method allows parallelism. The improvement based on temporally consistent remeshing [JXK23] results in slightly slower compression but comparable decompression times to V-DMC. For this reason, the authors also claim real-time-capable decompression. The improvement based on tracking of the base mesh [JXK24] results in a more complex encoding process that is 55% slower, but faster decoding.

9. Current Challenges

Although it is apparent that there exists a redundancy of temporal information in TVMs, for a long period, it was difficult for TVM compression methods to outperform intra-only approaches such as Google Draco [GHS*18] or weighted parallelogram [VB13]. For this reason, the baseline method used to evaluate the performance of MPEG V-DMC proposals was intra-only [ISO21c]. In the last 10 years, the advances in video- and model-based approaches led to a few quite effective methods being proposed, which were or should be able to outperform intra-only methods; however, at the cost of sacrificing the original mesh structure [MKT*22, ZGT23, HCNC23, NKK23a, NKK23b, JXK23, NKK23a, JXK24], restriction of the type of input sequences [DAZD14, DAA*14] or both [XHQ*10, HCHMT14]. For future research, we believe that it is important to focus on minimizing such sacrifices and develop a method which can handle general input data while preserving the original structure of the mesh. In the following subsections, we will focus on ways of addressing these challenges individually.

9.1. Structure preservation

As already discussed in Section 2, preserving the structure means that for each frame, there exists an isomorphic map between the connectivity of the original and the decoded mesh. Whether any of the methods described in this chapter preserves the structure or not and whether it also addresses compression of the connectivity, is summarized in columns Iso. and Conn. of Table 1 (the colour indicates connectivity coding efficiency).

The main reason why many approaches discard the original structure (number of vertices and connectivity) is that it usually contains no temporal coherence. The model-based approaches usually encode the original meshes only at keyframes (encoded intra-only) which are then gradually deformed to replace the rest of the frames [YKL06, TSM07, MYA08, NYA10, TM12, MKT*22, HCNC23, NKK23a, NKK23b, JXK23, NKK23a, JXK24]. The video-based approaches on the other hand deduce the connectivity from the decoded data [HKM04, XHQ*10, XQH*12, HCH*12, HCHMT13, HCH*13, HCZ*14, TM13, HCHMT14, HCMTH15, AMI*22, HZT*22, GWW23]. This is because the frames are densely resampled in the image domain, and thus vertices reconstructed from neighbouring pixels can be connected by an edge. A surface extraction from the occupancy network representation [ZGT23] also deduces a different connectivity from the original. There is also a second motivation for not preserving the original connectivity: remeshing to obtain a more exploitable structure. This is mainly used in progressive approaches [YKL06, MKT*22, NKK23a, NKK23b, JXK23, NKK23a, JXK24], but for example, the proposal by InterDigital for the MPEG V-DMC [MKG*22] also performs simplification as a means of further reducing the data rate.

The approaches based on ME [GSK03, HYA07, YA10] and prediction of data structures [HYA08, FHYA10] can preserve the original set of mesh vertices. However, most of these methods did not directly address the encoding of the connectivity information [HYA07, HYA08, FHYA10].

The earliest method for TVM compression proposed by Gupta et al. [GSK03] is the only one that assumes temporal coherence of connectivity between frames. It thus encodes the connectivity using simple update operations. This, however, works only on synthetic sequences and is highly impractical for real-world data (e.g. 3D-scanned human actors).

Yamasaki and Aizawa [YA10] were the first to acknowledge that the connectivity information of TVM occupies a considerable amount of the compressed data stream and should be considered if one proposes an efficient TVM compression method that preserves the mesh structure. Prior methods preserving the original set of vertices [HYA07, HYA08, FHYA10] focused solely on compression of TVM geometry. It is unclear whether their authors limited the scope of their research due to their awareness of how challenging the task of TVM connectivity compression is. In static mesh compression approaches, the mesh connectivity is usually encoded first and then used to drive the geometry coding. Unfortunately, to the best of our knowledge, no one was ever able to propose a connectivity-driven geometry coding method for TVMs that efficiently exploits temporal coherence. Faramarzi et al. [FJB20] attempted to do this but failed in comparison to intra-only approaches. Instead, the

vertex positions can be encoded more efficiently separately, which leads to their reordering. Conventional connectivity coding methods (e.g. Edgebreaker [Ros99] or TFAN [MZP09]), however, also reorder vertices. Some approaches used permutation maps to relate these two reordered sets [YA10, FJB20], but this requires additional $\sum_{i=0}^{n-1} \log_2(m_i!)$ bits of data, where n is the number of frames and m_i is the number of vertices of the frame \mathcal{M}_i , to be transmitted. Graziosi [Gra21] proposed to store the connectivity as a part of a 2D mesh. The mesh shares the connectivity with the original frame, but the (x,y) coordinate of each 2D vertex stores the (i,j) coordinate of the corresponding original vertex in the depth image. This 2D mesh is then encoded using an intra-only method (e.g. Google Draco). In Sony's patent [GZT22], the connectivity is rasterized into an image and then reconstructed by the decoder using segmentation. These two approaches are, however, still quite impractical.

9.1.1. Connectivity coding for known geometry

From all the existing structure-preserving methods, the most effective way of storing the connectivity of a TVM was proposed by Doumanoglou [DAZD14]. They encode the geometry first and then use this information to predict the connectivity based on the fact that if two vertices are close enough, they are very likely to be connected by an edge. Their method uses a modified TFAN [MZP09] algorithm, which instead of the conventional TFAN symbols encodes indices to the list of k nearest neighbours.

Although not used in TVM compression so far, there are quite a few other connectivity coding methods given a fixed geometry. The field, in which the motivation for this type of connectivity coding first arose, is progressive compression. Gandoin and Devillers [GD02] have pointed out that for non-manifold meshes, it is better to encode the geometry independently before the connectivity. Their method is based on kd-tree decomposition. Subdividing a tree cell is equivalent to splitting a vertex into two, with a certain connectivity update, which is encoded using the geometry as a prediction. A similar connectivity update was also used by Peng and Kuo [PK05].

GEncode, a single-rate general mesh (e.g. surface or volume mesh) compression scheme proposed by Lewiner et al. [LCL*06] encodes connectivity during a traversal through the mesh. To signal which vertex should be connected to the currently coded cell (e.g. a face in a surface mesh) the method encodes an index in the list of candidate vertices given a certain geometric function and a selected range of its values. The method works for meshes of arbitrary topology and dimension embedded in spaces of an arbitrary dimension.

Marais et al. [MGS07] claimed that due to advancements in point cloud compression, it is possible to encode the geometry separately as a point cloud and then exploit the global vertex position information to improve the performance of the connectivity encoding. The triangles are encoded in a slightly modified fixed traversal similar to the one used in the Edgebreaker algorithm [Ros99]. A position of a tip vertex of the currently coded triangle is predicted using the parallelogram or the midpoint scheme, which is then rotated and scaled to better match the vertices ahead of the current gate, and the rank of the correct tip vertex in the list of nearest neighbours around the prediction is encoded.

Dvořák et al. [DKVV22] also proposed a rank-based approach to encode the connectivity. However, they order the list of candidate vertices by a combination of various geometric properties measuring the feasibility of a potential triangle formed by connecting the given vertex. They also use a more sophisticated connectivity traversal driven by the certainty of the prediction of the tip vertex, which emits symbols in an order that can be exploited by context-adaptive coding.

For highly regular data, Chaîne et al. [CGR09] proposed a connectivity coding approach based on surface reconstruction. Both the encoder and the decoder perform an iterative surface reconstruction algorithm, and the encoder signals the differences between the actual and the reconstructed surface. The method applies only to triangles that are part of a Delaunay tetrahedralization of points. The rest of the triangles must be encoded less efficiently.

9.2. Versatility

By dropping versatility (i.e. the ability to process general input), one can make more assumptions about the character of the encoded data. It is also important to distinguish between whether the method fails to process more general data or is merely inefficient when compressing such data. Both of these properties for each method are summarized in columns *Versatility* and *Designed for* of Table 1.

The most limiting assumption on input data was done by Gupta et al. [GSK03]. They expected synthetic data on input, which allowed them to efficiently encode the connectivity, as was described in the previous section. The method does not fail for general input sequences, but its performance is very poor on such data.

All the methods based on expression invariant parameterization [XHQ*10, XQH*12, HCH*12, HCHMT13, HCH*13, H CZ*14] can handle only facial data and only in a very specific form, which contains one outer boundary and three inner boundaries located at both the eyes and the mouth. This is due to the method being incorporated directly into the data acquisition pipeline.

The polycube-parameterization-based approach [HCHMT14, HCMTH15] is limited to sequences of constant topology, since it uses a static polycube provided by the user, which reflects this topology. Although there are algorithms for obtaining the polycube parameterization automatically even for surfaces of general topology [YZWL14], to the best of our knowledge, there is no approach that also achieves temporal coherence. Additionally, the authors only theorize that the method works on TVMs and for experiments, they used dynamic meshes.

Since currently, the TVMs are primarily used for representing human actors, there have been quite a few methods (mainly model-based) that are optimized to work on human data. Assuming the TVM captures a performance of a human actor, the method can expect a specific ground truth underlying structure (head, arms and legs connected to the torso) and type of movement (articulated around joints), even though the noise introduced into data, for example, during scanning, might distort the actual topology and motion. Also, dynamic behaviour of loose clothing may be hard to capture by a general humanoid structure.

These approaches can be divided into two classes: those using tracked skeletons [MYA08, NYA10, DAA*14, DAZD14] and those using Reeb graphs [TSM07, TM12, TM13]. While not stated in the original papers, it is theoretically possible to adapt the skeleton-based approaches to sequences representing articulated surfaces of different constant ground-truth topologies (*Const. GT Top.* in Tab. 1) (e.g. animals, articulated robots), but this ground-truth structure must be known before encoding. The Reeb-graph-based approaches can handle general sequences of varying topology, although not efficiently, except for the original approach of Tung et al. [TSM07]. This approach uses heuristic rules (designed specifically for human sequences) to detect spurious self-contact in the topology of the graph. Theoretically, a different set of rules could be designed for a different surface of constant ground-truth topology, but this adaptation is much more difficult than the adaptation of skeleton-based approaches for such data. The optimized embedded deformation key nodes of Hoang et al. [HCNC23] can also handle general sequences of varying topology; however, the efficiency of the approach relies on surface correspondences being an isomorphism, which is often not the case, especially in the presence of topology changes.

Constant ground-truth topology (*Const. GT Top.* in Tab. 1) is also assumed by Yang et al. [YKL06]. Although their method is agnostic of the character of the underlying represented data, it is, to some extent, more limited than the approaches discussed in the previous paragraph. Since it replaces the sequence by the first frame propagated in time, it is crucial for the performance of the method that the first frame reflects the underlying topology. For this reason, it works best on data of constant actual topology of the frames.

The rest of ME-based [HYA07, YA10], prediction of data structure-based [HYA08] and video-based [HKM04, FJB20, Gra21, GZT22, MKG*22, AMI*22, HZT*22] methods put no assumptions on the shape and structure of the represented data other than the assumption of the presence of temporal coherence.

9.2.1. Versatile temporal models

While most efficient in terms of geometry compression, the model-based approaches are notorious for limiting the type of data they can handle. This is, however, not necessarily an issue of the used models, but usually of the approaches themselves. There are model-based approaches that can efficiently handle general input [TM12, MKT*22, ZGT23], unfortunately, none of the models used in these methods is directly suitable for structure-preserving compression.

It is certainly possible to extend the versatility of a constrained model. One good example is the tracked template used by Yang et al. [YKL06]. Bojsen-Hansen et al. [BHLW12] presented a tracking pipeline for surfaces of evolving topology, which can update the tracked template if a change in topology is detected and record a mapping for the updated parts to preserve inter-frame one-to-one correspondences. The result is a Time-varying mesh with temporally coherent connectivity. Unfortunately, for structure-preserving compression, this model still has a large footprint, which makes it impractical for structure-preserving TVM compression, since it must be encoded alongside the data.

Dvořák et al. [DVV21, DKV*22, DHV23] proposed to track points inside the volume (denoted centres) enclosed by the surface.

Since there is no structure of the centres, only a particular notion of their neighbourhood, they can represent surfaces of arbitrarily changing topology, as long as the surface provides a sufficient notion of the inside/outside distinction. The model was also designed with data footprint in mind since the centre trajectories can be encoded efficiently using principal component analysis.

Although, to the best of our knowledge, there is only a single method for TVM compression based on deep learning techniques [ZGT23], in the future, this approach will likely become increasingly popular. There are many possible places in the compression pipeline, where deep learning could be utilized, and it already is used in different compression domains, for example, for context modelling of entropy coder in point cloud compression [BLW*20], but the most probable part is once again the temporal model. We will not go into depth on this topic, but examples of deep learning methods that could be potentially incorporated as a versatile temporal model are the deformation field of *OccupancyFlow* [NMOG19], *Neural Deformation Graphs* proposed by Božič et al. [BPZ*21] and *CaDeX* [LD22], to name a few. These methods typically produce a compact representation of the shape and a deformation model represented by a neural network. However, their use for compression with preservation of the structure of the original sequence is itself a complex problem.

9.3. Perceptual metrics

Important for the development of effective compression methods is the availability of perceptual metrics for the quality assessment of TVMs. The most straightforward approach, which is often used in practice, for comparing the original sequence and the sequence distorted by compression is a frame-by-frame comparison using mechanistic metrics such as Mean Squared Error, Hausdorff Distance, or Chamfer Distance. However, these metrics, as shown above, do not sufficiently reflect the effect of distortion on human visual perception.

The most reliable way to verify that the proposed metric evaluates the perceptual quality of TVM well is to validate it against the results of subjective experiments. These experiments are conducted as user studies in which data are presented to the participants with varying levels and types of distortion. Participants provide responses on how intensely different distortions are perceived. The outputs of these studies are used as ground-truth values to which a perceptual metric should correlate.

A number of perceptual metrics have been proposed for comparing static meshes, such as Dihedral Angle Mesh Error [VR12], Fast Mesh Perceptual Distance [WTM12] or Tensor-based Perceptual Distance Measure [TWC14] which better correlate with ratings of distortion perceived by humans better than mechanistic metrics. Using these metrics provides better results in terms of evaluating shape distortion. However, comparing sequences in a frame-by-frame manner cannot evaluate the distortion of the temporal component.

The mentioned metrics are used exclusively for mesh shape comparison and do not consider mesh texture or its possible distortion. Graphics-LPIPS (Learned Perceptual Image Patch Similarity) has been proposed for textured meshes by Nehmé et al. [NDD*23]. It

is based on an image-based approach and uses convolutional neural networks to extract features from patches, based on which the quality of the mesh is predicted.

One of the drawbacks of image-based, video-based, and also point-based metrics is that they fail to detect the connectivity change caused by TVM compression when lossless connectivity compression is needed. For this purpose, one can use, for example Mmetric++ [ZZCY23], which allows mesh connectivity comparison based on the comparison of the configurations of the triangle fan (TF) defined in TFAN connectivity compression [MZP09] that are generated when traversing the original and distorted meshes.

While for animated meshes with constant connectivity, explicit temporal vertex correspondences can be used to detect temporal distortion artifacts [VS11], in the case of TVMs, these correspondences are unknown, making the construction of potential model-based metrics difficult. For this reason, existing methods for perceptual quality assessment of TVMs rely on image-based approaches, where a TVM is first rendered into typically several videos, which are then compared instead of mesh sequences. Such a method for comparing textured TVMs was proposed by Marvie et al. [MNGL23]. For a more detailed overview of state-of-the-art methods, we refer the reader to the survey by Alexiou et al. [ANZ*23]. A comparison of commonly used point-based, image-based, and video-based metrics can be found in Wien et al. [WJB22] or Yang et al. [YJD*23].

10. Recommendations

Despite extensive research and development, there is no universal compression method for time-varying meshes that meets all desirable properties. These include efficiency in terms of compression/decompression speed and in terms of compression ratio, universal applicability to any kind of input data and full structure preservation, that is lossless compression of the connectivity of each input frame.

When focusing on the group of structure-preserving methods, some of the current state-of-the-art approaches allow exploiting the temporal coherence of the data, albeit only for specific types of input. If the input data represents a movement of a human or humanoid figure, the method of [DAZD14] provides compression ratios that outperform intra-only encoding, albeit not by a substantial margin. The method can be potentially generalized to a broader class of inputs where the movement can be expressed as a skeletal animation; however, such extension is quite demanding and requires a deep understanding of the method.

If the character of the input is not known a-priori, then currently the best approach to structure-preserving compression of TVMs is using some state-of-the-art compression tool/method for static meshes, such as Google Draco [GHS*18] or laplacian-based encoding with error propagation control [VD18] to encode each frame separately. The particular choice of algorithm can in turn be driven by other application-specific requirements, such as ease of implementation, encoding/decoding time or constraints on input qualities (manifoldness etc.).

In the case of textured TVMs or densely sampled TVMs with vertex colours, it is theoretically possible to achieve better compression performance than that achieved with intra-only approaches by using structure-preserving video-based approaches, such as the one proposed by Faramarzi et al. [FJB20], albeit a different way of encoding connectivity must be used, for example the rank-based algorithm proposed by Dvořák et al. [DKVV22] for manifold frames, or the approach used by Doumanoglou et al. [DAZD14] if the encoded frames are non-manifold.

If preserving the original connectivity is not required, different methods can be chosen to achieve efficient compression. With these approaches, however, it is much harder to quantify the distortion caused by the compression, since there is no vertex–vertex correspondence between the original and the distorted meshes. The resulting performance is therefore influenced by both the loss in connectivity reproduction (which sometimes can be controlled to some degree) and the loss in geometry precision. The MPEG V-DMC standard in its current state [MKT*22] works for general input, which is converted into a connectivity that results from a regular subdivision of a general base mesh. Rate-distortion performance improvement over intra-only methods was reported using an image-based distortion metric. The standard is expected to be finalized and published soon, possibly with a reference implementation.

If the application allows discarding the connectivity and the encoding time is not of critical importance, it is possible to represent the TVM data more compactly by replacing it with a temporal model. For example, human TVMs can be replaced by human body appearance and pose models, such as SMPL [LMR*15] or STAR [OBB20]. In the case of sequences of constant ground truth topology, the compression can be approached as a sequence of more general tasks: first, identification of the ground truth topology of the input and construction of a canonical connectivity, second, remeshing of each frame into the shared connectivity, and third, encoding the sequence using some dynamic mesh compression algorithm. Such an approach can be very efficient in terms of compression ratio, since algorithms for encoding of dynamic meshes commonly achieve data rates well below 1 bpfv; however, the actual rate-distortion is heavily influenced by the first two steps of the procedure, for which currently no universal and generally robust solutions exist.

11. Conclusions

The simplest methods for TVM compression are based on ME or on the prediction of spatial structures. These approaches are simple, can be easily adjusted to preserve the original mesh connectivity and can handle general input. Their main drawback is their inefficiency in terms of compression ratio.

Current model-based methods are the most efficient in terms of compression of TVM geometry. Both of the top performing methods for TVM compression can be considered model-based: the structure-preserving method of Doumanoglou et al. [DAZD14] and the future MPEG V-DMC method based on the proposal by Apple [MKT*22].

For textured TVMs, it is best to use a method which uses video compression. These also work well for densely sampled TVMs with

colours as vertex attributes. They can mostly handle general input, but they are difficult to modify to preserve the original connectivity.

We believe the research area of TVM compression still has a lot of gaps, mainly the structural preservation and versatility, which were discussed in Section 9. Other than that, there is also a motivation for real-time compression [OERF*16], where these two challenges can be potentially ignored, but current methods are either too complex or inefficient to be used instead of intra-only approaches.

Acknowledgements

This work was supported by projects 23-04622L of the Czech Science Foundation and J2-4458 of the Slovenian Research and Innovation Agency and project PANOPTIS of H.F.R.I. Project Number: 16469. Filip Hácha was also supported by the university specific research project SGS-2022-015.

Conflict of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [ABB*23] ARMANDO M., BOISSIEUX L., BOYER E., FRANCO J.-S., HUMENBERGER M., LEGRAS C., LEROY V., MARSOT M., PANSIOT J., PUJADES S., REKIK R., ROGEZ G., SWAMY A., WUHRER S.: 4DHumanOutfit: A multi-subject 4D dataset of human motion sequences in varying outfits exhibiting large displacements. *Computer Vision and Image Understanding* 237 (Dec. 2023), 103836. ISSN: 1077-3142. <https://doi.org/10.1016/j.cviu.2023.103836>.
- [AKKH01] AHN J.-H., KIM C.-S., KUO C.-C. J., HO Y.-S.: Motion-compensated compression of 3D animation models. *Electronics Letters* 37, 24 (2001), 1445. ISSN: 0013-5194. <https://doi.org/10.1049/el:20010993>.
- [AM00] ALEXA M., MÜLLER W.: Representing animations by principal components. *Computer Graphics Forum* 19, 3 (Sept. 2000), 411–418. ISSN: 1467-8659. <https://doi.org/10.1111/1467-8659.00433>.
- [AMI*22] 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)* (Sept. 2022), IEEE. <https://doi.org/10.1109/euvip53989.2022.9922839>.
- [ANZ*23] ALEXIOU E., NEHMÉ Y., ZERMAN E., VIOLA I., LAVOUÉ G., AK A., SMOLIC A., LE CALLET P., CESAR P.: Chapter 18 – Subjective and objective quality assessment for volumetric video. In *Immersive Video Technologies*, Valenzise G., Alain M., Zerman E., Ozcinar C., (Eds.). Academic Press, 2023, pp. 501–552. ISBN: 978-0-323-91755-1. <https://doi.org/10.1016/B978-0-323-91755-1.00024-9>.
- [AZD13] ALEXIADIS D. S., ZARPALAS D., DARAS P.: Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras. *IEEE Transactions on Multimedia* 15, 2 (Feb. 2013), 339–358. ISSN: 1941-0077. <https://doi.org/10.1109/TMM.2012.2229264>.
- [Bjo01] BJONTEGAARD G.: Calculation of average PSNR differences between RD-curves. *ITU SG16 Doc. VCEG-M33* (2001).
- [BKP*23] BYEON J., KWON N., PARK H., SUH J., SIM D.: Scalable video-based dynamic mesh coding. In *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)* (Dec. 2023), IEEE. <https://doi.org/10.1109/vcip59821.2023.10402714>.
- [BHLW12] BOJSEN-HANSEN M., LI H., WOJTAN C.: Tracking surfaces with evolving topology. *ACM Transactions on Graphics* 31, 4 (July 2012). ISSN: 0730-0301. <https://doi.org/10.1145/2185520.2185549>.
- [BLW*20] BISWAS S., LIU J., WONG K., WANG S., URTASUN R.: Muscle: Multi sweep compression of lidar using deep entropy models. *Advances in Neural Information Processing Systems 2020-December* (2020). 34th Conference on Neural Information Processing Systems, NeurIPS 2020; 06-12-2020 to 12-12-2020. ISSN: 1049-5258. <https://doi.org/10.48550/arXiv.2011.07590>. eprint: 2011.07590 20.
- [BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (Feb. 1992), 239–256. <https://doi.org/10.1109/34.121791>.
- [BPZ*21] BOŽIČ A., PALAFOX P., ZOLLHÖFER M., THIES J., DAI A., NIEBNER M.: Neural deformation graphs for globally-consistent non-rigid reconstruction. *CVPR* (2021).
- [BRPMB17] BOGO F., ROMERO J., PONS-MOLL G., BLACK M. J.: Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (July 2017), IEEE. <https://doi.org/10.1109/CVPR.2017.591>.
- [BnSM*03] BRICEÑO H. M., SANDER P. V., McMILLAN L., GORTLER S., HOPPE H.: Geometry videos: A new representation for 3D animations. SCA '03, The Eurographics Association, pp. 136–146. ISBN: 1581136595. <https://doi.org/10.2312/SCA03/136-146>.
- [Cao23] CAO C.: *The MPEG Dynamic Mesh Coding Standard*. White paper, Ofinno, Sept. 2023. <https://ofinno.com/whitepaper/the-mpeg-dynamic-mesh-coding-standard/>.
- [CGR09] CHAINE R., GANDOIN P.-M., ROUDET C.: Reconstruction algorithms as a suitable basis for mesh connectivity compression. *IEEE Transactions on Automation Science and Engineering* 6, 3 (July 2009), 443–453. <https://doi.org/10.1109/TASE.2009.2021336>.
- [CJLR22a] CHOI Y.-H., JEONG J.-B., LEE S., RYU E.-S.: MPEG Dynamic Mesh Coding (DMC) standardization trend for volumetric video. *Proceedings of the Korea Broadcasting Media Engineering Conference Conference* (2022), 225–228.

- [CJLR22b] CHOI Y.-H., JEONG J.-B., LEE S., RYU E.-S.: Overview of the video-based dynamic mesh coding (V-DMC) standard work. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)* (Oct. 2022), IEEE. <https://doi.org/10.1109/ictc55196.2022.9952734>.
- [CK12] CHAMPAWAT Y., KUMAR S.: Online point-cloud transmission for tele-immersion. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry* (Dec. 2012), ACM. <https://doi.org/10.1145/2407516.2407540>.
- [CLL*13] CORSINI M., LARABI M. C., LAVOUÉ G., PETŘÍK O., VÁŠA L., WANG K.: Perceptual metrics for static and dynamic triangle meshes. *Computer Graphics Forum* 32, 1 (Jan. 2013), 101–125. <https://doi.org/10.1111/cgf.12001>.
- [CPZ19] CAO C., PREDA M., ZAHARIA T.: 3D point cloud compression. In *The 24th International Conference on 3D Web Technology* (July 2019), ACM. <https://doi.org/10.1145/3329714.3338130>.
- [CTPZ20] CAO C., TULVAN C., PREDA M., ZAHARIA T.: Skeleton-based motion estimation for point cloud compression. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)* (Sept. 2020), IEEE, pp. 1–6. <https://doi.org/10.1109/MMSP48831.2020.9287165>.
- [DAA*14] DOUMANOGLU A., ALEXIADIS D., ASTERIADIS S., ZARPALAS D., DARAS P.: On human time-varying mesh compression exploiting activity-related characteristics. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2014), IEEE. <https://doi.org/10.1109/icassp.2014.6854785>.
- [DAZD14] DOUMANOGLU A., ALEXIADIS D. S., ZARPALAS D., DARAS P.: Toward real-time and efficient compression of human time-varying meshes. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 12 (Dec. 2014), 2099–2116. <https://doi.org/10.1109/TCSVT.2014.2319631>.
- [DFS*12] DARIBO I., FURUKAWA R., SAGAWA R., KAWASAKI H., HIURA S., ASADA N.: Efficient rate-distortion compression of dynamic point cloud for grid-pattern-based 3D scanning systems. *3D Research* 3, 1 (Jan. 2012), 2. [https://doi.org/10.1007/3DRes.01\(2012\)2](https://doi.org/10.1007/3DRes.01(2012)2).
- [dHMC17] D'EON E., HARRISON B., MYERS T., CHOU P. A.: 8i voxelized full bodies – A voxelized point cloud dataset, Jan. 2017. ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006.
- [DHV23] DVOŘÁK J., HÁCHA F., VÁŠA L.: Global optimisation for improved volume tracking of time-varying meshes. In *Computational Science – ICCS 2023*. Springer Nature Switzerland, 2023, pp. 113–127. https://doi.org/10.1007/978-3-031-36027-5_9.
- [DKV*22] DVOŘÁK J., KÁČEREKOVÁ Z., VANĚČEK P., HRUDA L., VÁŠA L.: As-rigid-as-possible volume tracking for time-varying surfaces. *Computers & Graphics* 102 (Feb. 2022), 329–338. ISSN: 0097-8493. <https://doi.org/10.1016/j.cag.2021.10.015>.
- [DKVV22] DVOŘÁK J., KÁČEREKOVÁ Z., VANĚČEK P., VÁŠA L.: Priority-based encoding of triangle mesh connectivity for a known geometry. *Computer Graphics Forum* 42, 1 (Nov. 2022), 60–71. <https://doi.org/10.1111/cgf.14719>.
- [DVV21] DVOŘÁK J., VANĚČEK P., VÁŠA L.: Towards understanding time varying triangle meshes. In *Computational Science – ICCS 2021* (Cham, 2021), Paszynski M., Kranzlmüller D., Krzhizhanovskaya V. V., Dongarra J. J., Sloot P. M. A., (Eds.), Springer International Publishing, pp. 45–58. ISBN: 978-3-030-77977-1. https://doi.org/10.1007/978-3-030-77977-1_4.
- [FHVA10] FERREIRA R. U., HAN S.-R., YAMASAKI T., AIZAWA K.: Mixed spatial and SNR scalability for TVM geometry coding. In *2010 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video* (June 2010), IEEE. <https://doi.org/10.1109/3dtv.2010.5506299>.
- [FJB20] FARAMARZI E., JOSHI R., BUDAGAVI M.: Mesh coding extensions to MPEG-I V-PCC. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)* (Sept. 2020), IEEE, pp. 1–5. <https://doi.org/10.1109/MMSP48831.2020.9287057>.
- [GAM*21] GUEDE C., ANDRIVON P., MARVIE J.-E., RICARD J., REDMANN B., CHEVET J.-C.: V-pcc performance evaluation of the first MPEG point codec. *SMPTE Motion Imaging Journal* 130, 4 (May 2021), 36–52. ISSN: 2160-2492. <https://doi.org/10.5594/jmi.2021.3067962>.
- [GD02] GANDOIN P.-M., DEVILLERS O.: Progressive lossless compression of arbitrary simplicial complexes. *ACM Transactions on Graphics* 21, 3 (July 2002), 372–379. ISSN: 0730-0301. <https://doi.org/10.1145/566654.566591>.
- [GdQ17] GARCIA D. C., DE QUEIROZ R. L.: Context-based octree coding for point-cloud video. In *2017 IEEE International Conference on Image Processing (ICIP)* (Sept. 2017), IEEE, IEEE, pp. 1412–1416. <https://doi.org/10.1109/ICIP.2017.8296514>.
- [GGH02] GU X., GORTLER S. J., HOPPE H.: Geometry images. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2002), SIGGRAPH '02, Association for Computing Machinery, pp. 355–361. ISBN: 1581135211. <https://doi.org/10.1145/566570.566589>.
- [GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '97* (1997), ACM Press, pp. 209–216. <https://doi.org/10.1145/258734.258849>.
- [GHS*18] GALLIGAN F., HEMMER M., STAVA O., ZHANG F., BRETTELE J.: Google/draco: A library for compressing and decompressing 3D geometric meshes and point clouds, 2018.

- [GK04] GUSKOV I., KHODAKOVSKY A.: Wavelet compression of parametrically coherent mesh sequences. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '04* (2004), SCA '04, ACM Press. <https://doi.org/10.1145/1028523.1028547>.
- [Gra21] GRAZIOSI D. B.: Video-based dynamic mesh coding. In *2021 IEEE International Conference on Image Processing (ICIP)* (2021), pp. 3133–3137. <https://doi.org/10.1109/ICIP42928.2021.9506298>.
- [GSK03] GUPTA S., SENGUPTA K., KASSIM A.: Registration and partitioning-based compression of 3-D dynamic data. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 11 (Nov. 2003), 1144–1155. ISSN: 1051-8215. <https://doi.org/10.1109/tcsvt.2003.817625>.
- [GWW23] GAO Y., WANG Z., WEN J.: A method for generating geometric image sequences for non-isomorphic 3D-mesh sequence compression. *Electronics* 12, 16 (Aug. 2023), 3473. ISSN: 2079-9292. <https://doi.org/10.3390/electronics12163473>.
- [GXH*13] GARCIA I., XIA J., HE Y., XIN S.-Q., PATOW G.: Interactive applications for sketch-based editable polycube map. *IEEE Transactions on Visualization and Computer Graphics* 19, 7 (July 2013), 1158–1171. <https://doi.org/10.1109/tvcg.2012.308>.
- [GYWG22] GUO T., YUAN H., WANG T., GAO W.: Graph filter-based fast motion matching for inter frame coding of MPEG G-PCC. In *2022 IEEE International Conference on Image Processing (ICIP)* (Oct. 2022), IEEE. <https://doi.org/10.1109/icip46576.2022.9897391>.
- [GZT22] GRAZIOSI D., ZAGHETTO A., TABATABAI A.: Video based mesh compression, Apr. 2022. US Patent App. 17/322,662. <https://patents.google.com/patent/US20220108483A1>.
- [HCHMT13] HOU J., CHAU L.-P., HE Y., MAGNENAT-THALMANN N.: Expression-invariant and sparse representation for mesh-based compression for 3-D face models. In *2013 Visual Communications and Image Processing (VCIP)* (Nov. 2013), IEEE. <https://doi.org/10.1109/vcip.2013.6706442>.
- [HCHMT14] HOU J., CHAU L.-P., HE Y., MAGNENAT-THALMANN N.: A novel compression framework for 3D time-varying meshes. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)* (June 2014), IEEE, IEEE, pp. 2161–2164. <https://doi.org/10.1109/ISCAS.2014.6865596>.
- [HCH*12] HOU J., CHAU L.-P., HE Y., QUYNH D. T. P., MAGNENAT-THALMANN N.: Dynamic 3-D facial compression using low rank and sparse decomposition. In *SIGGRAPH Asia 2012 Technical Briefs* (Nov. 2012), ACM. <https://doi.org/10.1145/2407746.2407768>.
- [HCH*13] HOU J., CHAU L.-P., HE Y., ZHANG M., MAGNENAT-THALMANN N.: Rate-distortion model based bit allocation for 3-D facial compression using geometry video. *IEEE Transactions on Circuits and Systems for Video Technology* 23, 9 (Sept. 2013), 1537–1541. <https://doi.org/10.1109/tcsvt.2013.2248971>.
- [HCMTH15] HOU J., CHAU L.-P., MAGNENAT-THALMANN N., HE Y.: Compressing 3-D human motions via keyframe-based geometry videos. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 1 (Jan. 2015), 51–62. <https://doi.org/10.1109/tcsvt.2014.2329376>.
- [HCNC23] HOANG H., CHEN K., NGUYEN T., COSMAN P.: Embedded deformation-based compression for human 3D dynamic meshes with changing topology. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (Oct. 2023), IEEE. <https://doi.org/10.1109/iccvw60793.2023.00239>.
- [HCZ*14] HOU J., CHAU L.-P., ZHANG M., MAGNENAT-THALMANN N., HE Y.: A highly efficient compression framework for time-varying 3-D facial expressions. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 9 (Sept. 2014), 1541–1553. <https://doi.org/10.1109/tcsvt.2014.2313890>.
- [HKM04] HABE H., KATSURA Y., MATSUYAMA T.: Skin-off: Representation and compression scheme for 3D video. In *Proceedings of Picture Coding Symposium (PCS '04)* (2004), pp. 301–306.
- [HYA07] HAN S.-R., YAMASAKI T., AIZAWA K.: Time-varying mesh compression using an extended block matching algorithm. *IEEE Transactions on Circuits and Systems for Video Technology* 17, 11 (Nov. 2007), 1506–1518. <https://doi.org/10.1109/tcsvt.2007.903810>.
- [HYA08] HAN S.-R., YAMASAKI T., AIZAWA K.: Geometry compression for time-varying meshes using coarse and fine levels of quantization and run-length encoding. In *2008 15th IEEE International Conference on Image Processing* (2008), IEEE, IEEE, pp. 1045–1048. <https://doi.org/10.1109/ICIP.2008.4711937>.
- [HZT*22] HUANG C., ZHANG X., TIAN J., XU X., LIU S.: Boundary-preserved geometry video for dynamic mesh coding. In *2022 Picture Coding Symposium (PCS)* (Dec. 2022), IEEE. <https://doi.org/10.1109/pcs56426.2022.10018051>.
- [HZW09] HU Y., ZHOU M., WU Z.: A dense point-to-point alignment method for realistic 3D face morphing and animation. *International Journal of Computer Games Technology* 2009 (2009), 1–9. ISSN: 1687-7055. <https://doi.org/10.1155/2009/609350>.
- [ISO17] ISO/IEC JTC 1/SC 29/WG 11: Call for proposals for point cloud compression v2, Apr. 2017. MPEG2017/N16763.
- [ISO20] ISO/IEC JTC 1/SC 29/WG 7: V-PCC codec description, June 2020. MPEG/N00100.
- [ISO21a] ISO/IEC 23090-5:2021: Information technology – coded representation of immersive media – part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC), 2021. International Organization for Standardization, Geneva, Switzerland.
- [ISO21b] ISO/IEC JTC 1/SC 29/AG 03: White paper on MPEG immersive video, Jan. 2021. MPEG/N59.

- [ISO21c] ISO/IEC JTC 1/SC 29/WG 7: Cfp for dynamic mesh coding, Oct. 2021. MPEG/N231.
- [ISO21d] ISO/IEC JTC 1/SC 29/WG 7: Sequences for mesh coding evaluation, Apr. 2021. MPEG/N00114.
- [ISO22] ISO/IEC JTC 1/SC 29/WG 7: G-PCC codec description v12, Apr. 2022. MPEG/N00271.
- [ISO23a] ISO/IEC JTC 1/SC 29/AG 03: White paper on G-PCC, Apr. 2023. MPEG/N0111.
- [ISO23b] ISO/IEC JTC 1/SC 29/WG 7: G-PCC 2nd edition codec description, Oct. 2023. MPEG/N720.
- [JJ81] JAIN J., JAIN A.: Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications* 29, 12 (Dec. 1981), 1799–1808. ISSN: 0096-2244. <https://doi.org/10.1109/tcom.1981.1094950>.
- [JLS*21] JIN J., LI G., SHAO Y., SONG F., ZHANG R.: An improved coarse-to-fine motion estimation scheme for lidar point cloud geometry compression. In *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (July 2021), IEEE. <https://doi.org/10.1109/icmew53276.2021.9456018>.
- [JXK23] JIN X., XU J., KAWAMURA K.: Inter-frame coding for dynamic meshes via temporally-consistent re-meshing. In *2023 IEEE International Conference on Image Processing (ICIP)* (Oct. 2023), IEEE. <https://doi.org/10.1109/icip49359.2023.10222073>.
- [JXK24] JIN X., XU J., KAWAMURA K.: Embedded graph representation for inter-frame coding of dynamic meshes. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Apr. 2024), IEEE. <https://doi.org/10.1109/icassp48485.2024.10447762>.
- [KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2006), SGP '06, Eurographics Association, p. 61–70. ISBN: 3905673363
- [KBR*12] KAMMERL J., BLODOW N., RUSU R. B., GEDIKLI S., BEETZ M., STEINBACH E.: Real-time compression of point cloud streams. In *2012 IEEE International Conference on Robotics and Automation* (May 2012), IEEE, IEEE, pp. 778–785. <https://doi.org/10.1109/ICRA.2012.6224647>.
- [KBS24] KIM K., BYEON J., SIM D.: Valence based lifting wavelet transform for video-based dynamic mesh compression. *Journal of Broadcast Engineering* 29, 1 (Jan. 2024), 42–56. ISSN: 2287-9137. <https://doi.org/10.5909/jbe.2024.29.1.42>.
- [KIRK20] KIM J., IM J., RHYU S., KIM K.: 3D motion estimation and compensation method for video-based point cloud compression. *IEEE Access* 8 (2020), 83538–83547. <https://doi.org/10.1109/ACCESS.2020.2991478>.
- [KKK23] KISHIMOTO K., KAWAMURA K., KATO H.: 1D displacement coding for the displaced subdivision surface. In *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)* (Dec. 2023), IEEE. <https://doi.org/10.1109/vcip59821.2023.10402731>.
- [KT22] KAYA E. C., TABUS I.: Lossless compression of point cloud sequences using sequence optimized CNN models. *IEEE Access* 10 (Aug. 2022), 83678–83691. ISSN: 2169-3536. <https://doi.org/10.1109/access.2022.3197295>.
- [LAGP09] LI H., ADAMS B., GUIBAS L. J., PAULY M.: Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics* 28, 5 (Dec. 2009), 1–10. ISSN: 0730-0301. <https://doi.org/10.1145/1618452.1618521>.
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (Aug. 1987), ACM. <https://doi.org/10.1145/37401.37422>.
- [LCL*06] LEWINER T., CRAIZER M., LOPES H., PESCO S., VELHO L., MEDEIROS E.: GEncode: Geometry-driven compression for general meshes. *Computer Graphics Forum* 25, 4 (2006), 685–695. <https://doi.org/10.1111/j.1467-8659.2006.00990.x>.
- [LD22] LEI J., DANIILIDIS K.: CaDeX: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 6614–6624. <https://doi.org/10.1109/CVPR52688.2022.00651>.
- [Len99] LENGUEL J. E.: Compression of time-dependent geometry. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics* (New York, NY, USA, Apr. 1999), I3D '99, Association for Computing Machinery, pp. 89–95. <https://doi.org/10.1145/300523.300533>.
- [LLLL22] LI L., LI Z., LIU S., LI H.: Motion estimation and coding structure for inter-prediction of lidar point cloud geometry. *IEEE Transactions on Multimedia* 24 (2022), 4504–4513. ISSN: 1941-0077. <https://doi.org/10.1109/tmm.2021.3119872>.
- [LLZ*20] LI L., LI Z., ZAKHARCHENKO V., CHEN J., LI H.: Advanced 3D motion prediction for video-based dynamic point cloud compression. *IEEE Transactions on Image Processing* 29 (2020), 289–302. <https://doi.org/10.1109/TIP.2019.2931621>.
- [LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16. ISSN: 0730-0301. <https://doi.org/10.1145/2816795.2818013>, <https://smpl.is.tue.mpg.de/>.
- [LYTC19] LIU J., YAO J., TU J., CHENG J.: Data-adaptive packing method for compression of dynamic point cloud sequences. In *2019 IEEE International Conference on Multimedia and Expo (ICME)* (July 2019), IEEE. <https://doi.org/10.1109/icme.2019.00160>.
- [MBC16] MEKURIA R., BLOM K., CESAR P.: Design, implementation, and evaluation of a point cloud codec for tele-immersive

- video. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 4 (Apr. 2016), 828–842. <https://doi.org/10.1109/TCSVT.2016.2543039>.
- [MCB14] MEKURIA R., CESAR P., BULTERMAN D.: Low complexity connectivity driven dynamic geometry compression for 3D tele-immersion. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2014), IEEE. <https://doi.org/10.1109/icassp.2014.6854788>.
- [MGS07] MARAIS P., GAIN J., SHREINER D.: Distance-ranked connectivity compression of triangle meshes. *Computer Graphics Forum* 26, 4 (June 2007), 813–823. <https://doi.org/10.1111/j.1467-8659.2007.01026.x>.
- [MKG*22] MARVIE J.-E., KRIVOKUCA M., GUEDE C., RICARD J., MOCQUARD O., TARIOLLE F.-L.: Compression of time-varying textured meshes using patch tiling and image-based tracking. In *2022 10th European Workshop on Visual Information Processing (EUVIP)* (Sept. 2022), IEEE. <https://doi.org/10.1109/euvip53989.2022.9922890>.
- [MKT*22] MAMMOU K., KIM J., TOURAPIS A. M., PODBORSKI D., FLYNN D.: Video and subdivision based mesh coding. In *2022 10th European Workshop on Visual Information Processing (EUVIP)* (Sept. 2022), IEEE. <https://doi.org/10.1109/euvip53989.2022.9922888>.
- [MLDH15] MAGLO A., LAVOUÉ G., DUPONT F., HUDELLOT C.: 3D mesh compression: Survey, comparisons, and emerging trends. *ACM Computing Surveys* 47, 3 (Feb. 2015), 1–41. ISSN: 0360-0300. <https://doi.org/10.1145/2693443>.
- [MNGL23] MARVIE J.-E., NEHMÉ Y., GRAZIOSI D., LAVOUÉ G.: Crafting the MPEG metrics for objective and perceptual quality assessment of volumetric videos. *Quality and User Experience* 8, 1 (June 2023), 4. ISSN: 2366-0147. <https://doi.org/10.1007/s41233-023-00057-4>.
- [MON*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: Occupancy networks: Learning 3D reconstruction in function space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019), IEEE, pp. 4455–4465. <https://doi.org/10.1109/CVPR.2019.00459>.
- [MPE24] MPEG: Mpeg 146, Apr. 2024. <https://www.mpeg.org/meetings/mpeg-146/>. (visited on 06/25/2024)
- [MSK*08] MAMMOU K., STEFANOSKI N., KIRCHHOFFER H., MULLER K., ZAHARIA T., PRÉTEUX F., MARPE D., OSTERMANN J.: The new MPEG-4/FAMC standard for animated 3D mesh compression. In *2008 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video* (May 2008), IEEE. <https://doi.org/10.1109/3dtv.2008.4547817>.
- [MWTN04] MATSUYAMA T., WU X., TAKAI T., NOBUHARA S.: Real-time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualization for 3D video. *Computer Vision and Image Understanding* 96, 3 (Dec. 2004), 393–434. ISSN: 1077-3142. <https://doi.org/10.1016/j.cviu.2004.03.012>.
- [MYA08] MAEDA T., YAMASAKI T., AIZAWA K.: Model-based analysis and synthesis of time-varying mesh. In *Articulated Motion and Deformable Objects* (Berlin, Heidelberg, 2008), Perales F. J., Fisher R. B. (Eds.), Springer Berlin Heidelberg, pp. 112–121. ISBN: 978-3-540-70517-8. https://doi.org/10.1007/978-3-540-70517-8_12.
- [MYR*20] MA Q., YANG J., RANJAN A., PUJADES S., PONS-MOLL G., TANG S., BLACK M. J.: Learning to dress 3D people in generative clothing. In *Computer Vision and Pattern Recognition (CVPR)* (June 2020).
- [MZP09] MAMMOU K., ZAHARIA T., PRÉTEUX F.: TFAN: A low complexity 3D mesh compression algorithm. *Computer Animation and Virtual Worlds* 20, 2-3 (June 2009), 343–354. <https://doi.org/10.1002/cav.319>.
- [NDD*23] NEHMÉ Y., DELANOY J., DUPONT F., FARRUGIA J.-P., LE CALLET P., LAVOUÉ G.: Textured mesh quality assessment: Large-scale dataset and deep learning-based quality metric. *ACM Transactions on Graphics* 42, 3 (June 2023). ISSN: 0730-0301. <https://doi.org/10.1145/3592786>.
- [NKK23a] NISHIMURA H., KATO H., KAWAMURA K.: Arithmetic coding of displacements in dynamic meshes with bypass mode for complexity reduction. In *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)* (Dec. 2023), IEEE. <https://doi.org/10.1109/vcip59821.2023.10402693>.
- [NKK23b] NISHIMURA H., KATO H., KAWAMURA K.: Hierarchical arithmetic coding of displacements for dynamic mesh compression. In *2023 IEEE International Conference on Image Processing (ICIP)* (Oct. 2023), IEEE. <https://doi.org/10.1109/icip49359.2023.10222117>.
- [NMOG19] NIEMEYER M., MESCHEDER L., OECHSLE M., GEIGER A.: Occupancy flow: 4D reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2019), IEEE. <https://doi.org/10.1109/ICCV.2019.00548>.
- [NYA10] NAKAGAWA S., YAMASAKI T., AIZAWA K.: Deformation-based data reduction of time-varying meshes for displaying on mobile terminals. In *2010 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video* (June 2010), IEEE, pp. 1–4. <https://doi.org/10.1109/3DTV.2010.5506509>.
- [OBB20] OSMAN A. A. A., BOLKART T., BLACK M. J.: Star: Sparse trained articulated human body regressor. In *European Conference on Computer Vision (ECCV)* (2020), pp. 598–613. ISBN: 9783030585396. https://doi.org/10.1007/978-3-030-58539-6_36, <https://star.is.tue.mpg.de/>.
- [OERF*16] ORTS-ESCOLANO S., RHEMANN C., FANELLO S., CHANG W., KOWDLE A., DEGTYAREV Y., KIM D., DAVIDSON P. L., KHAMIS S., DOU M., TANKOVICH V., LOOP C., CAI Q., CHOU P. A., MENNICKEN S., VALENTIN J., PRADEEP V., WANG S., KANG S. B., KOHLI P., LUTCHYN Y., KESKIN C., IZADI S.: Holoportation: Virtual 3D teleportation in real-time. *UIST '16, Association*

- for Computing Machinery, pp. 741–754. ISBN: 9781450341899. <https://doi.org/10.1145/2984511.2984517>.
- [PAO*22] PAGÉS R., AMPLIANITIS K., ONDREJ J., ZERMAN E., SMOLIC A.: Volograms & V-SENSE Volumetric Video Dataset. 5. <https://doi.org/10.13140/RG.2.2.24235.31529/1>.
- [PB14] PARIKH N., BOYD S.: Proximal algorithms. *Foundations and trends® in Optimization* 1, 3 (2014), 127–239. ISSN: 2167-3918. <https://doi.org/10.1561/2400000003>.
- [PK05] PENG J., KUO C.-C. J.: Geometry-guided progressive lossless 3D mesh coding with octree (ot) decomposition. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, Association for Computing Machinery, p. 609–616. ISBN: 9781450378253. <https://doi.org/10.1145/1186822.1073237>.
- [PMR20] PEIXOTO E., MEDEIROS E., RAMALHO E.: Silhouette 4d: An inter-frame lossless geometry coder of dynamic voxelized point clouds. In *2020 IEEE International Conference on Image Processing (ICIP)* (Oct. 2020), IEEE, pp. 2691–2695. <https://doi.org/10.1109/ICIP40778.2020.9190648>.
- [PMPHB17] PONS-MOLL G., PUJADES S., HU S., BLACK M.: Clothcap: Seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 36, 4 (2017). Two first authors contributed equally. <https://doi.org/10.1145/3072959.3073711>.
- [PZA*21] PAGÉS R., ZERMAN E., AMPLIANITIS K., ONDŘEJ J., SMOLIC A.: Volograms & V-SENSE Volumetric Video Dataset. *ISO/IEC JTC1/SC29/WG07 MPEG2021/m56767* (2021).
- [QHC*11] QUYNH D. T., HE Y., CHEN X., XIA J., SUN Q., HOI S. C.: Modeling 3D articulated motions with conformal geometry videos (CGVS). In *Proceedings of the 19th ACM International Conference on Multimedia* (Nov. 2011), MM '11, ACM. <https://doi.org/10.1145/2072298.2072349>.
- [Ren21] Renderpeople: Free 3D people for Max, maya, C4D & more, 2021. <https://renderpeople.com/free-3d-people/>. (visited on 07/04/2024)
- [Ros99] ROSSIGNAC J.: Edgebreaker: Connectivity compression for triangle meshes. *IEEE Transactions on Visualization and Computer Graphics* 5, 1 (1999), 47–61. <https://doi.org/10.1109/2945.764870>.
- [RPM21] RAMALHO E., PEIXOTO E., MEDEIROS E.: Silhouette 4D with context selection: Lossless geometry compression of dynamic point clouds. *IEEE Signal Processing Letters* 28 (Aug. 2021), 1660–1664. <https://doi.org/10.1109/lsp.2021.3102525>.
- [SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Geometry Processing* (2007), Belyaev A., Garland M., (Eds.), The Eurographics Association. ISBN: 978-3-905673-46-3. <https://doi.org/10.2312/SGP/SGP07/109-116>.
- [SARG21] SCHAEFER R., ANDRIVON P., RICARD J., GUEDE C.: Volucap and XD productions datasets, Jan. 2021.
- [SC17] SAWHNEY R., CRANE K.: Boundary first flattening. *ACM Transactions on Graphics* 37, 1 (Dec. 2017), 1–14. <https://doi.org/10.1145/3132705>.
- [SL22] SHI H., LI F.: Patch re-segmentation and packing for dynamic point cloud compression via back-and-forth structure. *IEEE Open Journal of Signal Processing* 3 (2022), 155–168. <https://doi.org/10.1109/ojsp.2022.3160392>.
- [SOHW12] SULLIVAN G. J., OHM J., HAN W., WIEGAND T.: Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (Dec. 2012), 1649–1668. <https://doi.org/10.1109/TCSVT.2012.2221191>.
- [SPB*19] SCHWARZ S., PREDI M., BARONCINI V., BUDAGAVI M., CESAR P., CHOU P. A., COHEN R. A., KRIVOKUCA M., LASSERRE S., LI Z., LLACH J., MAMMOU K., MEKURIA R., NAKAGAMI O., SIAHAAN E., TABATABAI A., TOURAPIS A. M., ZAKHARCHENKO V.: Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (Mar. 2019), 133–148. ISSN: 2156-3365. <https://doi.org/10.1109/jetcas.2018.2885981>.
- [SRR*23] SEO Y., RYU G., RHEE C. E., JUNG H., NAM D., KIM H., LIM S.: Improving the compression efficiency of displacement using morton-ordered micro-image in video-based dynamic mesh coding. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)* (May 2023), IEEE. <https://doi.org/10.1109/iscas46773.2023.10182003>.
- [SFSH19] SCHWARZ S., SHEIKHIPOUR N., FAKOUR SEVOM V., HANNUKSELA M. M.: Video coding of dynamic 3D point cloud data. *APSIPA Transactions on Signal and Information Processing* 8, 1 (2019), e31. <https://doi.org/10.1017/ATSIP.2019.24>.
- [SSP07] SUMNER R. W., SCHMID J., PAULY M.: Embedded deformation for shape manipulation. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, July 2007), SIGGRAPH '07, Association for Computing Machinery, pp. 80–es. ISBN: 9781450378369. <https://doi.org/10.1145/1275808.1276478>.
- [SWG*03] SANDER P. V., WOOD Z. J., GORTLER S. J., SNYDER J., HOPPE H.: Multi-chart geometry images. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (Goslar, DEU, 2003), SGP '03, Eurographics Association, pp. 146–155. ISBN: 1581136870.
- [TCF16] THANOU D., CHOU P. A., FROSSARD P.: Graph-based compression of dynamic 3D point cloud sequences. *IEEE Transactions on Image Processing* 25, 4 (2016), 1765–1778. <https://doi.org/10.1109/TIP.2016.2529506>.
- [TG98] TOUMA C., GOTSMAN C.: Triangle mesh compression. In *Proceedings of the Graphics Interface 1998 Conference, June 18-20, 1998, Vancouver, BC, Canada* (June 1998), pp. 26–34.
- [TM12] TUNG T., MATSUYAMA T.: Topology dictionary for 3D video understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 8 (Aug. 2012), 1645–1657. <https://doi.org/10.1109/tpami.2011.258>.

- [TM13] TUNG T., MATSUYAMA T.: Invariant surface-based shape descriptor for dynamic surface encoding. In *Computer Vision – ACCV 2012* (Berlin, Heidelberg, 2013), Lee K. M., Matsushita Y., Rehg J. M., Hu Z. (Eds.), Springer Berlin Heidelberg, pp. 486–499. ISBN: 978-3-642-37331-2. https://doi.org/10.1007/978-3-642-37331-2_37.
- [TOKI04] TOMIYAMA K., ORIHARA Y., KATAYAMA M., IWADATE Y.: Algorithm for dynamic 3D object generation from multi-viewpoint images. In *Three-Dimensional TV, Video, and Display III* (Oct. 2004), Javidi B., Okano F. (Eds.), SPIE. <https://doi.org/10.1117/12.571117>.
- [TR98] TAUBIN G., ROSSIGNAC J.: Geometric compression through topological surgery. *ACM Transactions on Graphics* 17, 2 (Apr. 1998), 84–115. <https://doi.org/10.1145/274363.274365>.
- [TS05] TUNG T., SCHMITT F.: The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modeling* 11, 01 (June 2005), 91–120. <https://doi.org/10.1142/s0218654305000748>.
- [TSM07] TUNG T., SCHMITT F., MATSUYAMA T.: Topology matching for 3D video compression. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), IEEE. <https://doi.org/10.1109/cvpr.2007.383294>.
- [Tun12] TUNG T.: Downloads, 2012. <https://sites.google.com/site/tony2ng/download>. (visited on 07/04/2024)
- [TWC14] TORKHANI F., WANG K., CHASSERY J.-M.: A curvature-tensor-based perceptual quality metric for 3D triangular meshes. *Machine Graphics & Vision* 23, 1/2 (June 2014), 59–82. <https://doi.org/10.22630/MGV.2014.23.1.4>.
- [imm23] Valenzise G., Alain M., Zerman E., Ozcinar C., eds.: *Immersive video technologies*, 2023. <https://doi.org/10.1016/C2021-0-00404-1>.
- [VB13] VÁŠA L., BRUNETT G.: Exploiting connectivity to improve the tangential part of geometry prediction. *IEEE Transactions on Visualization and Computer Graphics* 19, 09 (Sept. 2013), 1467–1475. ISSN: 1941-0506. <https://doi.org/10.1109/TVCG.2013.22>.
- [VBMP08] VLAŠIĆ D., BARAN I., MATUSIK W., POPOVIĆ J.: Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, Aug. 2008), SIGGRAPH '08, Association for Computing Machinery. ISBN: 9781450301121. <https://doi.org/10.1145/1399504.1360696>.
- [VD18] VÁŠA L., DVOŘÁK J.: Error propagation control in laplacian mesh compression. *Computer Graphics Forum* 37, 5 (Aug. 2018), 61–70. <https://doi.org/10.1111/cgf.13491>.
- [Vis13] Visual Computing Lab - Information Technologies Institute: Datasets of multiple kinects-based 3d reconstructed meshes, 2013. <https://vcl.iti.gr/dataset/reconstruction/>. (visited on 07/04/2024)
- [VPB*09] VLAŠIĆ D., PEERS P., BARAN I., DEBEVEC P., POPOVIĆ J., RUSINKIEWICZ S., MATUSIK W.: Dynamic shape capture using multi-view photometric stereo. In *ACM SIGGRAPH Asia 2009 Papers* (Dec. 2009), SA09, ACM. <https://doi.org/10.1145/1661412.1618520>.
- [VR12] VÁŠA L., RUS J.: Dihedral angle mesh error: A fast perception correlated distortion measure for fixed connectivity triangle meshes. *Computer Graphics Forum* 31, 5 (Aug. 2012), 1715–1724. <https://doi.org/10.1111/j.1467-8659.2012.03176.x>.
- [VS11] VÁŠA L., SKALA V.: A perception correlated comparison method for dynamic meshes. *IEEE Transactions on Visualization and Computer Graphics* 17, 2 (Feb. 2011), 220–230. <https://doi.org/10.1109/TVCG.2010.38>.
- [WJB22] WIEN M., JUNG J., BARONCINI V.: Formal visual evaluation and study of objective metrics for MPEG dynamic mesh coding. In *2022 10th European Workshop on Visual Information Processing (EUVIP)* (Sept. 2022), IEEE. <https://doi.org/10.1109/euvip53989.2022.9922894>.
- [WTM12] WANG K., TORKHANI F., MONTANVERT A.: A fast roughness-based approach to the assessment of 3D mesh visual quality. *Computers & Graphics* 36, 7 (Nov. 2012), 808–818. ISSN: 0097-8493. Augmented Reality Computer Graphics in China. <https://doi.org/10.1016/j.cag.2012.06.004>.
- [WWCF24] WANG Y., WANG Y., CUI T., FANG Z.: Occupancy map-based low complexity motion prediction for video-based point cloud compression. *Journal of Visual Communication and Image Representation* 100 ISSN: 1047-3203. (Apr. 2024), 104110. <https://doi.org/10.1016/j.jvcir.2024.104110>.
- [XHQ*10] XIA J., HE Y., QUYNH D. P. T., CHEN X., HOI S. C. H.: Modeling 3D facial expressions using geometry videos. In *Proceedings of the 18th ACM International Conference on Multimedia* (Oct. 2010), ACM. <https://doi.org/10.1145/1873951.1874010>.
- [XQH*12] XIA J., QUYNH D. T. P., HE Y., CHEN X., HOI S. C. H.: Modeling and compressing 3D facial expressions using geometry videos. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 1 (Jan. 2012), 77–90. <https://doi.org/10.1109/tcsvt.2011.2158337>.
- [YA10] YAMASAKI T., AIZAWA K.: Patch-based compression for time-varying meshes. In *2010 IEEE International Conference on Image Processing* (Sept. 2010), IEEE, IEEE, pp. 3433–3436. <https://doi.org/10.1109/ICIP.2010.5652911>.
- [YQG*23] YIN Y., GUO C., KAUFMANN M., ZARATE J., SONG J., HILLIGES O.: Hi4D: 4D instance segmentation of close human interaction. In *Computer Vision and Pattern Recognition (CVPR)* (2023).
- [YJD*23] YANG Q., JUNG J., DESCHAMPS T., XU X., LIU S.: TDMD: A database for dynamic color mesh subjective and objective quality explorations, 2023. arXiv: 2308.01499 [cs.CV]. <https://arxiv.org/abs/2308.01499>.

- [YKL02] YANG J.-H., KIM C.-S., LEE S.-U.: Compression of 3-D triangle mesh sequences based on vertex-wise motion vector prediction. *IEEE Transactions on Circuits and Systems for Video Technology* 12, 12 (Dec. 2002), 1178–1184. ISSN: 1051-8215. <https://doi.org/10.1109/tcsvt.2002.806814>.
- [YKL06] YANG J.-H., KIM C.-S., LEE S.-U.: Semi-regular representation and progressive compression of 3-D dynamic mesh sequences. *IEEE Transactions on Image Processing* 15, 9 (Sept. 2006), 2531–2544. <https://doi.org/10.1109/TIP.2006.877413>.
- [YZWL14] YU W., ZHANG K., WAN S., LI X.: Optimizing poly-cube domain construction for hexahedral remeshing. *Computer-Aided Design* 46 (Jan. 2014), 58–68. <https://doi.org/10.1016/j.cad.2013.08.018>.
- [ZGT23] ZAGHETTO A., GRAZIOSI D., TABATABAI A.: Task-oriented dynamic mesh compression using occupancy networks, Jan. 2023. US Patent App. 17/861,033. <https://patents.google.com/patent/US20230016302A1>.
- [ZHTY23] 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)* (Dec. 2023), IEEE. <https://doi.org/10.1109/vcip59821.2023.10402678>.
- [ZMXLL21] Zhu, W., Ma, Z., Xu, Y., Li, L., Li, Z.: View-Dependent Dynamic Point Cloud Compression. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 2 (Feb. 2021), 765-781. <https://doi.org/10.1109/tcsvt.2020.2985911>.
- [ZZCY23] ZOU W., ZHOU J., CHEN J., YANG F.: Texture coordinate prediction using structural information for MPEG V-DMC. In *2023 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)* (July 2023), IEEE. <https://doi.org/10.1109/icmew59549.2023.00029>.