Project Proposal: Evaluation of *TVMC – Time-Varying Mesh Compression Using Volume-Tracked Reference Meshes*

Andrew Xie Srotriyo Sengupta
Satrajit Ghosh
VR Technology

Department of Electrical and Computer Engineering
March 2025

1 Introduction

In this project, we aim to review and evaluate *TVMC (Time-Varying Mesh Compression using Volume-Tracked Reference Meshes)*, a novel compression framework proposed by Chen et al. [1]. Time-varying meshes (TVMs), which are sequences of 3D meshes with changing geometry and topology over time, play a key role in applications such as augmented and virtual reality (AR/VR), digital avatars, telepresence, holographic conferencing, and volumetric video streaming. However, TVMs are notoriously difficult to store and transmit due to their large file sizes and the lack of consistent connectivity across frames. Traditional compression solutions like *Google Draco* treat each frame independently, missing temporal redundancies, while approaches like *V-DMC* [4] assume fixed topology, limiting their generalizability. TVMC addresses these limitations by tracking internal motion using volume centers, constructing a self-contact-free reference mesh, and encoding displacements from this reference using Google Draco. It claims to reduce bitrates by over 50% and decoding times by 66.1%, potentially enabling real-time 3D mesh streaming even on constrained devices. Our goal in this project is to reproduce, validate, and critically analyze the performance and limitations of this approach.

2 Literature Review

Compression of time-varying meshes (TVMs) has posed a persistent challenge due to their inherently dynamic structure—each frame may introduce variations in vertex count, mesh connectivity, and surface geometry. Traditional intra-frame compression techniques such as *Google Draco* operate by compressing each mesh individually using vertex quantization and entropy coding. While

Draco is efficient for static meshes, it fails to exploit inter-frame temporal redundancy, resulting in unnecessarily large file sizes when applied to sequences. To address this, inter-frame approaches such as *V-DMC* attempt to predict frame-wise geometry using motion estimation across time. However, these approaches typically assume a fixed mesh topology, which does not hold for real-world data captured using photogrammetry or multi-view systems.

Several prior works have explored ways to reduce the complexity and storage cost of TVMs. Maeda et al. [2] proposed a skeleton-based model that extracts motion information from a sequence of meshes and uses it to animate a single reference frame. This greatly reduces data size—achieving storage of just 300 bytes per frame—but sacrifices detailed surface variation, such as facial expressions or cloth dynamics. Their method also struggles with body rotation, as they restrict motion estimation to prevent unnatural twisting artifacts. Similarly, Han et al. [3] adapted 2D block matching algorithms to the 3D domain for time-varying mesh compression. They estimate motion vectors and transform meshes accordingly, but their method relies on consistent vertex indexing across frames, which is often unavailable in unconstrained TVMs.

These existing solutions are limited either by topology assumptions, loss of geometric detail, or lack of robustness to self-contact regions. This creates the need for a compression framework that can operate on general, real-world mesh sequences. TVMC addresses these gaps by introducing a volume-based motion tracking mechanism using internal centers rather than surface correspondences, followed by multidimensional scaling to create a stable reference space. It avoids self-contact artifacts by explicitly filtering out high-contact frames and builds a self-contact-free reference mesh for efficient displacement-based encoding. This makes TVMC well-suited for compressing TVMs with complex motion and variable topology, where previous methods often fail.

3 Solutions

In this project, we evaluate the compression pipeline proposed in *TVMC (Time-Varying Mesh Compression Using Volume-Tracked Reference Meshes)* by Chen et al. [1]. TVMC was designed to address critical limitations in prior time-varying mesh compression techniques, especially in scenarios with variable topology and self-contact regions—conditions under which traditional codecs like *Google Draco* and methods like *V-DMC* perform poorly. While Draco compresses each frame independently, it ignores temporal redundancy. V-DMC, on the other hand, assumes consistent connectivity across time, limiting its applicability to real-world mesh data. TVMC overcomes both issues by rethinking the problem: instead of compressing the mesh sequence directly, it introduces a new representation based on a *volume-tracked reference mesh* and displacement fields.

The pipeline begins with **ARAP (As-Rigid-As-Possible) volume tracking**, where a set of internal volume centers are uniformly distributed inside each mesh in the sequence. These centers are tracked over time to capture the vol-

umetric motion of the object, rather than relying on surface correspondences that are sensitive to mesh connectivity. The density of these centers significantly impacts tracking quality. As illustrated in Figure 1, using 2000 centers results in far more accurate alignment than using only 500, demonstrating the importance of center resolution in motion tracking.

Figure 1: Effect of volume center density on motion tracking in the TVMC framework. Both subfigures show the same mesh with different numbers of tracked internal volume centers (orange dots). (a) 500 centers result in sparser coverage and less accurate deformation. (b) 2000 centers provide finer-grained tracking, enabling more precise alignment and lower deformation error. This demonstrates the importance of parameter tuning in the ARAP volume tracking stage. Figure adapted from Chen et al. [1].

After tracking, the trajectories of volume centers are optimized using **Multi-dimensional Scaling (MDS)**, which projects their motion into a low-distortion reference space. This forms the basis for generating a reference mesh. However, directly fusing all frames into this space can lead to artifacts if self-contact is present. Therefore, TVMC includes a **self-contact filtering step**, where only frames with low self-contact (i.e., where body parts do not touch or intersect) are selected for reference mesh generation. Figure 2 demonstrates this: the mesh with self-contact results in visible deformation errors, while the one without produces artifact-free alignment.

Once the **volume-tracked reference mesh** is built from these clean frames, the remaining frames in the sequence are reconstructed through a process called **center affinity deformation**. This technique deforms the reference mesh using the motion of the volume centers and calculates a **displacement field** that captures the difference between the deformed reference and the target mesh. These displacement vectors, stored as point clouds, are then compressed—along with the reference mesh itself—using **Google Draco**. The key innovation here is that TVMC transforms the original input into a format that is *highly compressible by standard codecs*, thus outperforming Draco even while using Draco

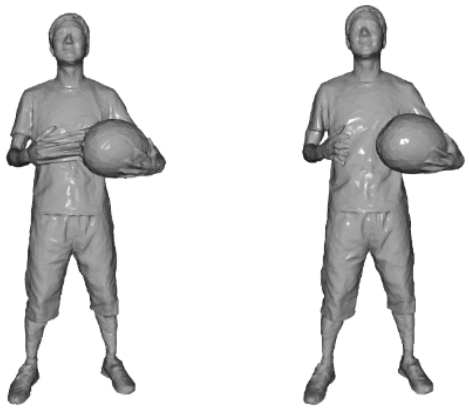

Figure 2: Impact of self-contact regions on deformation quality in time-varying mesh sequences. The left mesh contains regions where the hand touches the basketball, causing deformation artifacts during alignment. The right mesh separates fingers and ball, producing a cleaner and more accurate deformation. This illustrates why TVMC filters out frames with high self-contact before constructing the reference mesh, as it significantly improves final reconstruction fidelity. Figure adapted from Chen et al. [1].

as the final encoder.

Figure 3 summarizes this process. From input meshes to center tracking, global optimization, filtering, reference mesh extraction, and finally displacement-based compression, TVMC delivers a modular pipeline that integrates motion understanding and topology-agnostic representation.

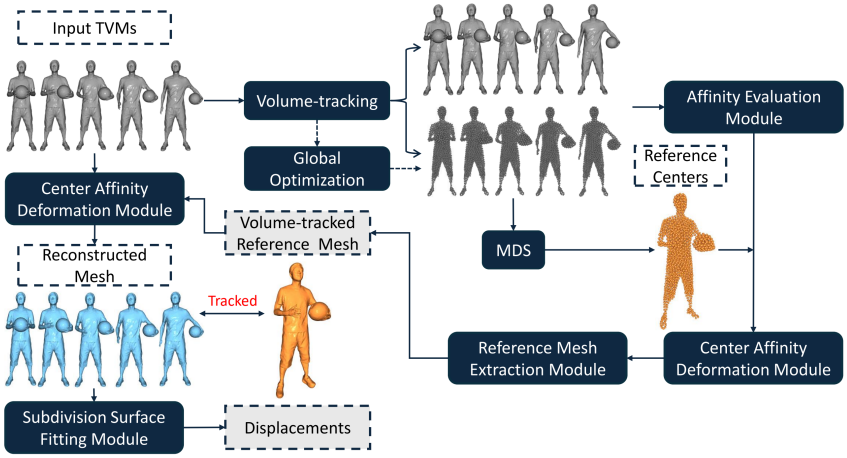

Figure 3: Overview of the complete TVMC pipeline for compressing time-varying meshes. The process begins with a sequence of mesh inputs, from which a set of internal volume centers is tracked across time using ARAP-based volume tracking. These centers are optimized using multidimensional scaling (MDS) to create a reference space. Low self-contact frames are then deformed into this space and fused to generate a clean reference mesh. Each input mesh is encoded as a set of displacements from the reference, which are then compressed using Google Draco. The pipeline ensures both geometric fidelity and efficient temporal compression. Figure adapted from Chen et al. [1].

4 Evaluation Plan

Our evaluation of the TVMC framework will follow a comprehensive, multi-dimensional strategy focused on assessing both compression performance and reconstruction quality. As a point of comparison, we will consider the performance of *Google Draco*—a widely used intra-frame mesh compression library that processes each frame independently, without accounting for temporal coherence. Depending on resource availability and time constraints, we may also attempt to implement a minimal Draco-only evaluation pipeline to establish a consistent benchmarking baseline. This would allow us to quantify the added value of TVMC's temporal encoding and preprocessing steps under controlled conditions.

Our quantitative evaluation will be based on three core metrics: (1) **bitrate**, measured in Mbps, to assess compression efficiency; (2) **geometry distortion**, using metrics such as *D2-PSNR* (point-to-surface Peak Signal-to-Noise Ratio) and *root mean square error (RMSE)* to measure reconstruction fidelity; and (3) **decoding latency per frame**, to evaluate whether the approach meets the requirements for real-time 3D streaming (targeting less than 33ms per frame).

We will begin our testing on the `basketball_player` sequence provided

in the official TVMC repository. This dataset features complex motion patterns, including articulated limbs and self-contact interactions, making it ideal for stress-testing deformation quality and temporal consistency. In addition, we plan to introduce multiple external mesh sequences—either from synthetic datasets or real-world 3D captures—to evaluate the generalizability of the pipeline to previously unseen data.

For qualitative assessment, we will use tools such as *Open3D* and *MeshLab* to visualize reconstructed meshes frame-by-frame and manually inspect surface accuracy, deformation artifacts, and temporal coherence. We will also replicate the bitrate-versus-distortion plots from the original TVMC paper, varying internal parameters such as the number of volume centers and Group-of-Frames (GoF) size to evaluate their impact on performance. These experiments will help us understand not only the overall effectiveness of TVMC, but also the contribution of each individual component in the compression pipeline.

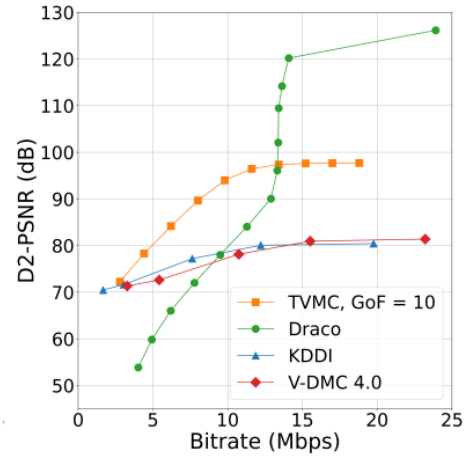

Figure 4: A performance claim we aim to validate: rate-distortion comparison of TVMC (with GoF = 10) against Google Draco, KDDI, and V-DMC 4.0. The plot shows bitrate (Mbps) on the x-axis and geometric fidelity (D2-PSNR in dB) on the y-axis. TVMC is reported to achieve higher PSNR at lower bitrates, converging to a performance plateau with better efficiency. Our experiments will test whether this advantage holds in practice. Figure adapted from Chen et al., *TVMC: Time-Varying Mesh Compression Using Volume-Tracked Reference Meshes*, ACM MMSys 2025 [1].

Through this evaluation process, we hope to not only reproduce the core results reported in the TVMC paper, but also assess the framework's real-world utility and robustness under varied input conditions. Our analysis will inform whether TVMC delivers practical advantages for real-time mesh streaming in immersive applications such as AR/VR, telepresence, or holographic video systems.

5 Progress So Far

As of the current milestone, we have successfully cloned, built, and executed the full TVMC pipeline using the official GitHub repository and Docker-based environment provided by the authors. The pipeline was validated end-to-end on the `basketball_player` dataset, where we generated all key outputs including volume center trajectories, transformation logs, reference mesh reconstructions, displacement fields, and final Draco-compressed files. We verified the functionality of each major module—*ARAP volume tracking*, *MDS-based reference center generation*, *reference mesh extraction*, and *displacement encoding*—by comparing visual and numerical results to those reported in the paper.

Figure 5: Reconstructed mesh sequence visualized in Blender using frames from the `basketball_player` dataset. Each mesh corresponds to a time step within the Group-of-Frames. The output matches the qualitative deformation seen in the TVMC paper, confirming that the pipeline is producing consistent and reproducible results.

We further reproduced the cumulative distribution function (CDF) of deformation distances between the tracked volume centers and their corresponding mesh surfaces, as shown in Figure 6. Our results follow the same trend as the TVMC paper, where static regions exhibit tighter deformation bounds and moving regions show greater variance. This confirms that the volume-tracking module is operating correctly.

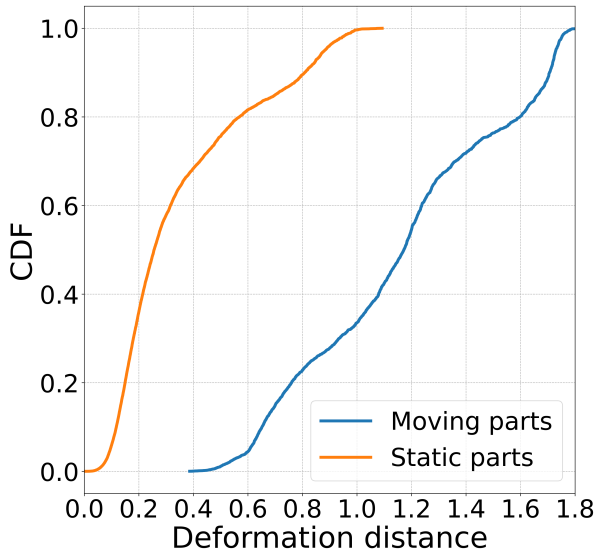

Figure 6: CDF of deformation distances for static and moving parts within the `basketball_player` sequence. The orange curve (static regions) shows lower average deformation, while the blue curve (moving limbs) exhibits wider variance. This plot reproduces the deformation behavior reported in the TVMC paper and supports its claim regarding the importance of motion-aware volume tracking.

We also replicated the rate-distortion curve that compares TVMC against other mesh compression methods, shown in Figure 7. Our plot closely matches the original in terms of D2-PSNR and bitrate, supporting the paper's reported claims and verifying the reproducibility of their performance benchmarks.

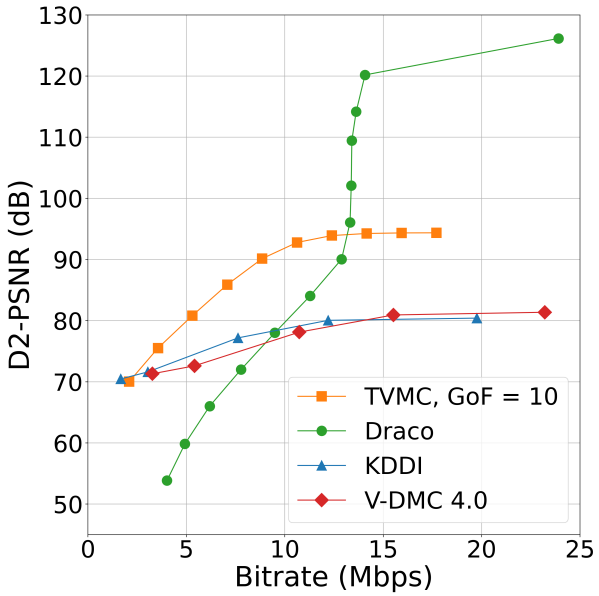

Figure 7: Replicated rate-distortion curve comparing TVMC (GoF = 10) with Google Draco, KDDI, and V-DMC 4.0. The D2-PSNR (in dB) is plotted against bitrate (Mbps). Our results reproduce the relative performance trends observed in the original TVMC paper, confirming both its claims and the functional correctness of the source code.

Challenges Encountered and Resolution

The initial setup of the TVMC environment posed several technical challenges that we had to overcome before successfully running the full pipeline. One major issue stemmed from the outdated Docker environment provided by the authors, which relied on deprecated versions of Conda and the .NET SDK. These caused repeated build errors during CMake compilation, particularly for the volume-tracking and mesh deformation modules. To resolve this, we manually installed a newer version of the .NET SDK (7.0) inside the container and reconfigured environment variables to ensure compatibility with the build system. Additionally, we encountered compatibility issues with Python packages such as Open3D, NumPy, and PyTorch, leading to segmentation faults during visualization and processing stages. We addressed this by setting up a fresh Conda environment within Docker and carefully aligning package versions with those used in the paper's original codebase. Another significant challenge involved limited file access between the container and host system—output files such as reconstructed meshes and point clouds were not visible outside the container due to missing volume mappings. We fixed this by mounting persistent directories via the `-v` flag in Docker, allowing seamless synchronization of results. Lastly, several scripts contained hardcoded file paths and assumptions about directory layout, which caused execution errors when running

stages independently. We revised these scripts (including `evaluation.py` and `generate_reference_mesh.py`) to use relative paths and accept dynamic input arguments. These debugging efforts were necessary to get the system running and have provided us with a stable and extensible setup that is now ready for additional experiments and evaluations.

References

[1] G. Chen, F. Hácha, L. Váša, and M. Dasari, *TVMC: Time-Varying Mesh Compression Using Volume-Tracked Reference Meshes*, in Proceedings of ACM MMSys, 2025. `https://github.com/SINRG-Lab/TVMC`

[2] T. Maeda, T. Yamasaki, and K. Aizawa, *Model-Based Analysis and Synthesis of Time-Varying Mesh*, Lecture Notes in Computer Science, vol. 5098, Springer, pp. 112–121, 2008.

[3] S. Han, T. Yamasaki, and K. Aizawa, *Motion Estimation from Time-Varying Mesh Sequences for 3D Geometry Compression*, in Advances in Visual Computing, Springer, pp. 39–50, 2021.

[4] Y. Choi, J.-B. Jeong, S. Lee, and E.-S. Ryu, *Overview of the Video-based Dynamic Mesh Coding (V-DMC) Standard Work*, in Proceedings of the 2022 International Conference on Information and Communication Technology Convergence (ICTC), IEEE, pp. 578–581, 2022. DOI: `10.1109/ICTC55196.2022.9952734`