ITE Trans. on MTA Vol. 12, No. 3, pp. 182-196 (2024)

Paper

Tracked QEM Algorithm: Adding Temporal Consistency to Dynamic Mesh Simplification Based on Mesh Registration

Yunlong Liu (student member)†, Jianfeng Xu (member)††, Kei Kawamura (member)††, Hiroshi Watanabe (fellow)†

Abstract The evolution of 3D multi-media technology has spurred the need for more effective 3D video storage and transmission methods. Most of the current standardizing 3D volumetric video coding methods in the mesh decimation stage are still limited: the difference of geometric structure in each simplified frame hinders optimal compression. The proposed Tracked QEM Algorithm effectively tracks mesh models across successive frames, offering a tailored solution for dynamic meshes in 3D volumetric videos.

The Tracked QEM Algorithm ensures that the simplified results have better topological consistency and spatial smoothness between consecutive frames than the original QEM algorithm. The evaluation results based on temporal consistency show that the proposed approach is superior to the conventional mesh simplification. The smoother simplified results with similar topology delineate the discontinuous structural information between frames. As a novel pre-processing approach to 3D video compression, this proposal has the potential to improve the compression rate.

Key words: Multi-media Technology, Volumetic Video, Computer Graphic, Mesh Simplification, Mesh Registration, QEM Algorithm

1. Introduction

Over the past few years, Augmented and Virtual Reality (AR/VR) technologies[1)] developed rapidly[2)]. Central to these advanced multi-media technologies, 3D volumetric video[3)] is pivotal in accurately recreating 3D objects and scenes, gaining significant attention[4)]. With the ongoing actions in standardizing 3D volumetric video coding[5)], there is a critical need to enhance storage and transmission efficiency for this particular 3D video format.

Similar to 2D video compression[6)], 3D volumetric video encoding follows a multi-step pipeline[7)], a concise summary can be depicted in Fig.1. The encoding progress begins by taking a series of raw frames as input, transforming them into a continuous bitstream for efficient storage and transmission[8)]. As the red part shown in this figure, an essential component of this pipeline is mesh decimation, which reduces the number of faces in the input mesh to diminish the amount of data. Given that dynamic meshes in 3D videos consist of a series of static mesh frames, current encoding standards[9)] rely on static mesh simplification methods.

Represented by the Quadratic Error Metric (QEM) algorithm[10)], these methods can efficiently simplify the individual static mesh frame[11)]. As shown in Fig.2, the simplification algorithm uses the mesh of each frame as an input to generate corresponding simplified results. However, due to less consideration of the inter-frame continuity, the difference between adjacent frames may be amplified in the simplification process. The simplified frames may differ in appearance and topology, which can be called inconsistency of the mesh decimation step.

Since the discontinuous structural information will lead to higher encoding expenses in bitstream phase[12)], ensuring the consistency of frame sequences as much as possible in pre-processing stages becomes an essential issue in 3D video coding. A novel mesh decimation method is proposed to solve this problem. This method aims to add temporal consistency via the introduction of a reference frame.

As shown in Fig.3, considering the dynamic mesh consists of static mesh frames, the proposed Tracked

Received April 4, 2024; Revised May 25, 2024; Accepted May 28, 2024

† Graduate School of Fundamental Science and Engineering, Waseda University
(3-4-1 Ohkubo, Shinjuku-ku, Tokyo, 169-8555, Japan)
†† KDDI Research, Inc.
(2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502, Japan)

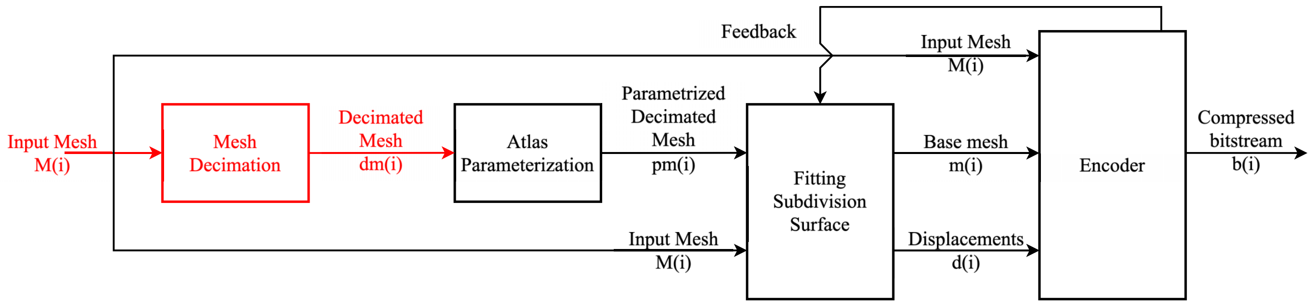


Fig. 1 A typical encoder pipeline for 3D Volumetric video

QEM algorithm tracks registered[13)] mesh model from the previous frame to drastically boost the temporal consistency in dynamic mesh simplification.

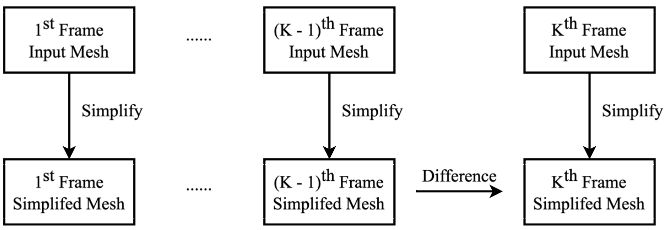


Fig. 2 Current mesh decimation in 3D video coding on individual frames

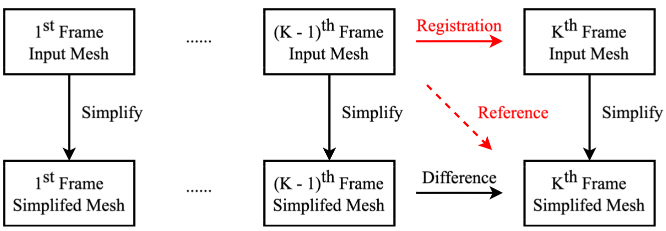


Fig. 3 Proposed dynamic mesh simplification with inter-frame tracking

Experiments are also carried out to compare the effectiveness in dynamic mesh decimation task between the proposed Tracked QEM Algorithm and previous Original QEM Algorithm. The result reveals that the proposal significantly improves the temporal consistency of the traditional approach. As a pre-processing step of 3D video coding, preparing a smoother, more coherent 3D frame sequence helps maintain the consistent topology of input meshes. This merit shows the potential of the Tracked QEM Algorithm to advance the related 3D volumetric video compression research in the future.

The main contributions of the proposed method in this paper are as follows:

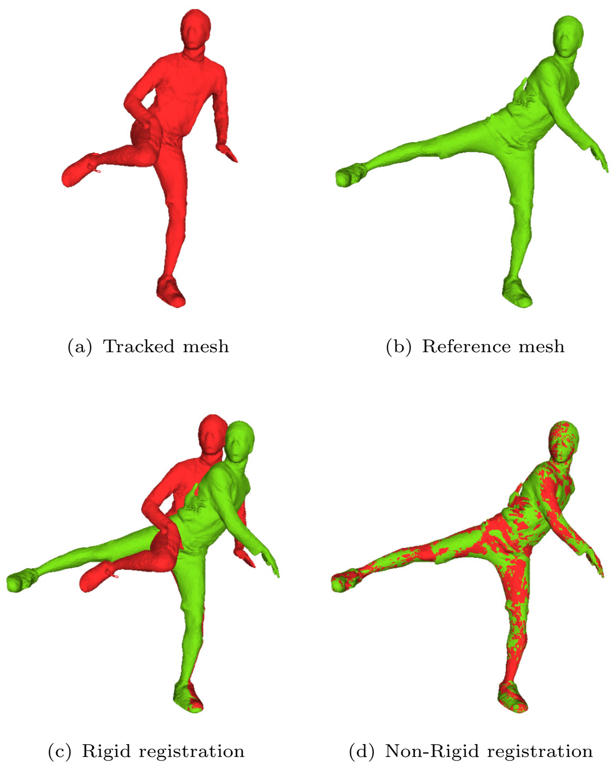
• Innovatively introduces 3D mesh registration to explore the possibility of considering temporal consistency in mesh simplification tasks.

• Proposes a Tracked QEM Algorithm suitable for dynamic mesh in 3D volumetric video.

• Redesigns inter-framed collapsed cost function and optimal strategy, enabling QEM Algorithm to handle reference input.

2. Related work

2.1 Mesh Registration



(a) Tracked mesh (b) Reference mesh

(c) Rigid registration (d) Non-Rigid registration

Fig. 4 Rigid & Non-Rigid Registration

The main work in this paper is a proposal of a novel mesh simplification method that innovatively integrates the principles of mesh registration. Mesh registration is a specific 3D registration in computer vision and computer graphics, which is a beneficial task that focuses on tracking and aligning distinct 3D models into a typical coordinate system[13)]. The registration has evolved from rigid to non-rigid approaches that account for variations and distortions[14]. As shown in Fig.4, mesh registration can be divided into rigid and non-rigid registration.

Rigid Registration: Rigid Registration aligns two 3D image by applying an Euclidean transformations[15], many related research has also made progress in various 3D image formats[16]. The rigid mesh registration evolved from approaches for point set matching[17–19] from point cloud[20], the representative methods[21–24] with many variants[25] is Iterative Closest Point (ICP)[26]. As shown in Eq.(1), for each corresponding point pair $\mathrm{p}_i^{\mathrm{c}}$ and $\mathrm{p}_j^{\mathrm{r}}$, ICP constructs the distance function between two point sets $\mathcal{P}^{\mathrm{c}} = \{\mathrm{p}^{\mathrm{c}}_{1:N^{\mathrm{c}}}\}$ and $\mathcal{P}^{\mathrm{r}} = \{\mathrm{p}^{\mathrm{r}}_{1:N^{\mathrm{r}}}\}$ with the inherent noise Ω_{ij} from 3D sensor, then iterates the minimum value of affine transformation matrix T via least square method to get the most suitable transformation matrix $\mathbf{T}^*$[27]:

$$\mathbf{T}^* = \arg\min_{\mathrm{T}} \sum_{c} \left(\mathrm{p}_i^{\mathrm{c}} - \mathrm{T} \oplus \mathrm{p}_j^{\mathrm{r}}\right)^T \Omega_{ij} \left(\mathrm{p}_i^{\mathrm{c}} - \mathrm{T} \oplus \mathrm{p}_j^{\mathrm{r}}\right) \quad (1)$$

Non-Rigid Registration: In the context of non-rigid registration, it is imperative to account for the multifaceted nature of transformations that surpass the simplicity of rigid body movements[28]. Embedded deformation[29] inspired from modified ICP algorithm[30–32] is a representative approach[33]. As stated in Eq.(2), Like rigid registration, the early methods, aimed to generate the deformation field $\mathbf{X}$ with a transformation point set $\mathbf{G} = \left\{\mathbf{g}_j \middle| \mathbf{g}_{1:N^{\mathrm{X}}}\right\}$ from several transformation $\mathbf{X}_j$ with affine $\mathbf{R}_j$ and displace $\mathbf{t}_j$ matrices. The relationship between each target point $\hat{\mathbf{v}}_i$ and transformed source $\mathbf{v}_i$ can be described by several neighbour transform $\mathbf{X}_j$ weighted by w_j.

$$\begin{aligned} \hat{\mathbf{v}}_i &= \sum_{j=1}^{m} w_j\left(\mathbf{v}_i\right)\left[\mathbf{R}_j\left(\mathbf{v}_i - \mathbf{g}_j\right) + \mathbf{g}_j + \mathbf{t}_j\right], \\ \mathbf{X}_j &= \left(\mathbf{R}_j, \mathbf{t}_j\right), \ \mathbf{X}_j \in \mathbf{X} \end{aligned} \quad (2)$$

Considering generality and usability, the robust iterative optimization-based methods[34–37] are more suitable for combining with upstream or downstream tasks. The approach integrated into proposed method is the Fast-RNRR[38], which constructed the optimal target $\mathbf{X}^*$ from cost function $E(\mathbf{X})$ through three sub-items in Eq.(3): alignment item $E_{\mathrm{align}}(\mathbf{X})$ to penalizes the deviation between aligned $\hat{\mathbf{v}}_i$ and input $\mathbf{u}_{\rho(i)}$, regularization item $E_{\mathrm{reg}}(\mathbf{X})$ to avoid the over-fitting of deformation $\mathbf{D}_{ij}$ with weight α, and rotation item $E_{\mathrm{rot}}(\mathbf{X})$ to make the deviation $\mathbf{A}_i$ close to a projection transformation $\mathrm{proj}_{\mathcal{R}}\left(\mathbf{A}_i\right)$ weighted by β.

$$\begin{aligned} \mathbf{X}^* &= \min_{\mathbf{X}} E_{\mathrm{align}}(\mathbf{X}) + \alpha E_{\mathrm{reg}}(\mathbf{X}) + \beta E_{\mathrm{rot}}(\mathbf{X}), \\ E_{\mathrm{align}}(\mathbf{X}) &= \sum_{i=1}^{n} \psi_{\nu_{\mathrm{a}}}\left(\left\|\hat{\mathbf{v}}_i - \mathbf{u}_{\rho(i)}\right\|\right), \\ E_{\mathrm{reg}}(\mathbf{X}) &= \sum_{i=1}^{r} \sum_{\mathbf{p}_j \in \mathcal{N}(\mathbf{p}_i)} \psi_{\nu_{\mathrm{r}}}\left(\left\|\mathbf{D}_{ij}\right\|\right), \\ E_{\mathrm{rot}}(\mathbf{X}) &= \sum_{i=1}^{r} \left\|\mathbf{A}_i - \mathrm{proj}_{\mathcal{R}}\left(\mathbf{A}_i\right)\right\|_F^2, \\ \psi_\nu(x) &= 1 - \exp\left(-\frac{x^2}{2\nu^2}\right) \end{aligned} \quad (3)$$

This scheme introduces the Welsch function $\psi_\nu(\cdot)$[39] to ensure the smoothness of the optimization. Fast-RNRR uses the Quasi-Newton solver combining MM and L-BFGS to improve optimization efficiency, which has rapid convergence speed without prior data or training.

2. 2 Mesh Simplification

(a) Original input (b) Simplified result

Fig. 5 A visualized case for input and output of typical mesh simplification

The other part of the related work is structured around mesh simplification techniques as the basic knowledge, which helps to contextualize the QEM algorithm used in the proposal. Mesh Simplification is a crucial research direction in 3D computer graphics processing. As shown in Fig.5, it is a procedure to reduce the number of vertices, edges, and faces in a mesh model[40]. Meanwhile, retain the original model's appearance and topological structure with numerous attempts[41]. These researches[42] emerged in the 1990s[43–46], which aims to automatically generate different Levels of Detail (LOD) for 3D models[47].

Early methods focused on the geometric error[48–51], ensuring retaining the original appearance after simplification. Iterative reduction methods[52], such as Progressive Meshes[53], were widely studied, allowing for gradual refinement or simplification of the model[54–56]. In recent years, the trend has shifted towards leveraging advanced computational techniques, such as Parallel Computing[57–59] and Machine Learning[60–62], to achieve more efficient and accurate simplification results.

Among the many approaches mentioned above, the QEM algorithm is a commonly used method for mesh simplification, which inspired the proposal in this paper. Michael Garland and Paul S. Heckbert introduced it in 1997[10], then derived many variable species in the following several years[48–50)63].

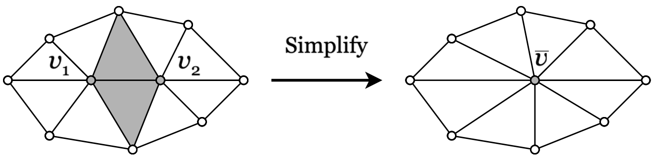


Fig. 6 A typical edge collapse step in QEM Algorithm

As shown in Fig.6, with high-quality model simplification processing, the QEM algorithm minimizes the quadratic error to select and merge vertex pairs efficiently. This merging procedure can also be called edge collapse. QEM algorithm uses Euclidean distance to measure the collapse cost. Specifically, the sum of the squared distances of all neighboring planes $p \in \text{plane}(v_i)$ from the target position $\overline{v}$ to the vertex pair (v_1, v_2) being merged:

$$\overline{v} = \arg\min_{v} \sum_{p \in \text{plane}(v_1) \cup \text{ plane } (v_2)} \text{distance}(v, p)^2 \quad (4)$$

For the merger $(v_1, v_2) \to \overline{v}$ of any given vertex pair (v_1, v_2), the quadratic error $\Delta(\overline{v}) = \overline{\mathbf{v}}^\top \mathbf{Q}\, \overline{\mathbf{v}}$ in target place $\overline{\mathbf{v}} = [v_x, v_y, v_z, 1]^\top$ can be defined as an geometric approximation via a symmetric 4×4 matrix $\mathbf{Q}$:

$$\begin{aligned} \Delta(\overline{v}) = \Delta\big([v_x, v_y, v_z, 1]^\top\big) &= \sum_{\mathbf{p} \in \text{planes}(\overline{v})} \left(\mathbf{p}^\top \overline{\mathbf{v}}\right)^2 \\ &= \sum_{\mathbf{p} \in \text{planes}(\overline{v})} \overline{\mathbf{v}}^\top \left(\mathbf{p}\mathbf{p}^\top\right) \overline{\mathbf{v}} \\ &= \overline{\mathbf{v}}^\top \left(\sum_{p \in \text{planes}(\overline{v})} \mathbf{K}_p \right) \overline{\mathbf{v}} \\ &= \overline{\mathbf{v}}^\top \mathbf{Q}\, \overline{\mathbf{v}} \end{aligned} \quad (5)$$

According Eq.(5), since this error $\Delta(\overline{v})$ is a quadratic function, finding the optimal position $\overline{v}$ is transformed into a linear problem: Assuming the quadratic matrix $\mathbf{Q}$ is a positive definite matrix, the extreme value $\overline{\mathbf{v}}$ of position $\overline{v}$ can be find by partial derivative when $\partial\Delta/\partial x = \partial\Delta/\partial y = \partial\Delta/\partial z = 0$:

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \overline{\mathbf{v}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (6)$$

3. Proposed Method

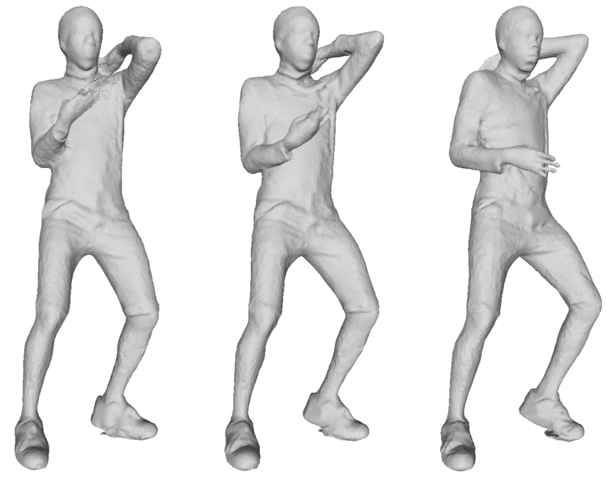
The characteristic of 3D volumetric video inspired the approach in this paper: If an algorithm can consider such movement between adjacent frames as conditions, then this mesh simplification algorithm is more advantageous for dynamic mesh than the traditional existing method. As shown in Fig.7, the Tracked QEM Algorithm is proposed to track this dynamic movement, which can introduce the previous frame as a reference via mesh registration.

Compared with the original QEM algorithm used in current volumetric video coding standard[5], the proposed Tracked QEM Algorithm is mainly improved from three places: introduced mesh registration to generate the reference for simplification; added inter-frame tracking for edge collapse cost function; and improved optimal search strategy when the quadratic error matrix is irreversible.

These modifications ensure that the structure of the simplified mesh has temporal consistency with the previous frame to avoid geometric structures that are too different from those caused by motion and deformation between adjacent frames. The following part of this chapter will comprehensively describe these three main innovation points in this proposed approach.

3.1 Previous frame reference

As discussed in Section 1, the dynamic mesh in 3D volumetric videos is similar to a frame sequence in 2D video, which also consists of static mesh frames. As the example shown in Fig.8, These static mesh frames captured with a constant time interval Δt[64], specific k-th frame in this sequence can also be viewed as a static mesh $mesh^{(k)}$ in traditional computer graph. the subject movement in a specific period $(k-1)\Delta t \sim k\Delta t$ will be recorded by difference between current frame $mesh^{(k)}$ and previous frame $mesh^{(k-1)}$.



(a) (k-1)-th frame (b) k-th frame (c) (k+1)-th frame

Fig. 8 A frame sequence from dynamic mesh in 3D volumetric video fragment

In order to make the simplified results leverage these differences between two adjacent frames, Tracked QEM introduced the information of the previous frame as a reference in the simplification process. However, due to the inter-frame information, including subject movement, there is no significant correspondence between the two frames in the geometric structure of the static mesh. For this reason, mesh registration was introduced to track this inter-frame association.

While the 3D volumetric video focuses on the nuances of posture alterations within the volumetric space, it is equally crucial to consider the displacing of the character's position that may occur[65]. This displacement and deformation can be regarded as a combination of rigid and non-rigid transformation[14], which underscores the complexity of capturing and replicating human motion within volumetric environments.

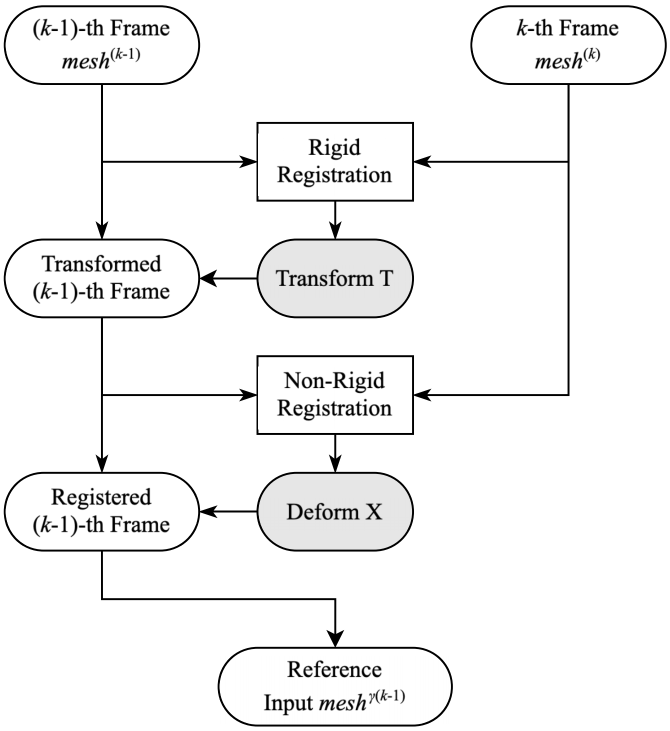


Fig. 9 Tracked Quadric Error Metrics (QEM) Algorithm

As the flowchart shown in Fig.9, rigid and non-rigid registration is used in the proposed method simultane-

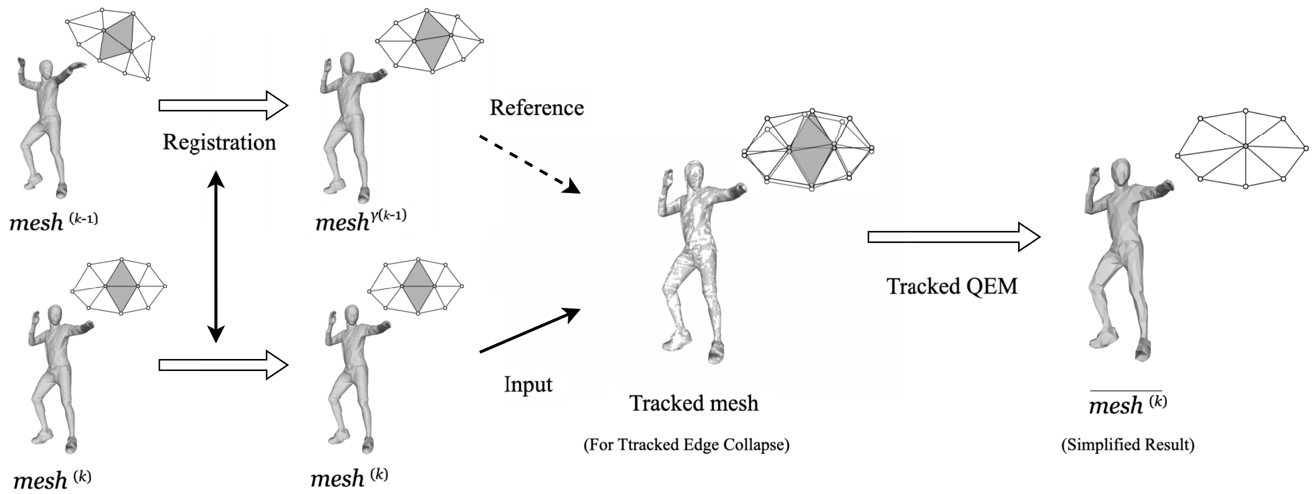


Fig. 7 Flowchart of proposed Tracked QEM Algorithm, including tracked reference from previous frame

ously to track complex human motion[13]. The registered $mesh^{\gamma(k-1)}$ can be obtained by registration $\gamma(\cdot)$ from previous $mesh^{(k-1)}$, which can be used as a reference of current $mesh^{(k)}$ in tracked edge collapse:

1. **Rigid registration:** Rigid registration aims to align inputs to eliminate the possible difficulties caused by displacement in the subsequent processing. The algorithm employed here is the naive Iterative Closest-Point (ICP)[26]. The previous $mesh^{(k-1)}$ will be moved to align with the current $mesh^{(k)}$ to fetch rigid transformation. The optimization target $\mathbf{T}^*$ in Eq.(1) is changed into the specific format $\mathbf{T}^{\gamma(k-1)}$ in Eq.(7) below, which minimum the corresponding point pair transformation $\mathbf{P}_{ij}^{\gamma(k-1)}$. The result will be passed as input for the subsequent non-rigid registration processing.

$$\begin{aligned}\mathbf{T}^{\gamma(k-1)} &= \arg\min_{\mathrm{T}} \sum_{c} \left(\mathbf{P}_{ij}^{\gamma(k-1)}\right)^{T} \Omega_{ij} \left(\mathbf{P}_{ij}^{\gamma(k-1)}\right), \\ \mathbf{P}_{ij}^{\gamma(k-1)} &= \mathrm{p}_i^{(k)} - \mathrm{T} \oplus \mathrm{p}_j^{(k-1)}\end{aligned} \tag{7}$$

2. **Non-rigid registration:** non-rigid deformations will be corrected after rigid alignment. Non-rigid deformations can potentially be in-homogeneous and may occur only in a local region. Therefore, the Fast-RNRR[38] based on iteration was selected to obtain acceptable correspondence. Through this Quasi-Newton's solver, The deformation field $\mathbf{X}^*$ in Eq.(2) will be able to be solved by a two-layer iteration. In the proposed method, the affine $\mathbf{R}_j^{\gamma}$ and displace $\mathbf{t}_j^{\gamma}$ transformation can be described as $\mathbf{X}^{\gamma(k-1)}$ by the formula in Eq.(8).

$$\begin{aligned}\mathbf{X}^{\gamma(k-1)} &= \min_{\mathbf{X}} E_{\text{align}}(\mathbf{X}) + \alpha E_{\text{reg}}(\mathbf{X}) + \beta E_{\text{rot}}(\mathbf{X}), \\ \mathbf{X}^{\gamma(k-1)} &= \left\{ \mathbf{X}_j^{\gamma} \,\middle|\, \mathbf{X}_j^{\gamma} = \left(\mathbf{R}_j^{\gamma}, \mathbf{t}_j^{\gamma}\right) \right\}\end{aligned} \tag{8}$$

Consistent with the flowchart shown in Fig.9, once the rigid transformation $\mathrm{T}^{\gamma(k-1)}(\cdot)$ and non-rigid deformation $\mathrm{X}^{\gamma(k-1)}(\cdot)$ has been obtained, Comibine the defination from Eq.(2), the registered reference $mesh^{\gamma(k-1)}$ can then be expressed by the transformation matrix $\mathbf{T}^{\gamma(k-1)}$ and the deformation field $\mathbf{X}^{\gamma(k-1)}$:

$$\begin{aligned}mesh^{\gamma(k-1)} &= \gamma_{\text{reg}(k)} \left(mesh^{(k-1)}\right) \\ &= \mathrm{X}^{\gamma(k-1)} \left(\mathrm{T}^{\gamma(k-1)} \left(mesh^{(k-1)}\right)\right)\end{aligned} \tag{9}$$

Mesh registration aims to establish tracking. The correspondence of geometric entities between $mesh^{(k-1)}$ and $mesh^{(k)}$ can obtain this inter-frame tracking. As the reference of current frame $mesh^{(k)}$, related vertices in tracked $mesh^{\gamma(k-1)}$ also need to be introduced into the optimization target $\overline{v^{(k)}}$. It changes the optimization goal from "minimize the effect on current mesh appearance" to "minimize the effect on the previous and current mesh."

3.2 Tracked collapse cost function

As implied by the name, Tracked QEM is based on the original QEM algorithm and has added some new features related to tracking. Consider an edge collapsing on the current k-th frame $mesh^{(k)}$, original QEM contracting two endpoints v_1 and v_2 on a specific edge e to be collapsed into a single point $\overline{v}$. With the same meaning as in Eq.(4), the edge collapse function only considers the faces $\mathrm{plane}(v_1)$ and $\mathrm{plane}(v_2)$ adjacent collapsed vertices v_1, v_2 from current mesh surface.

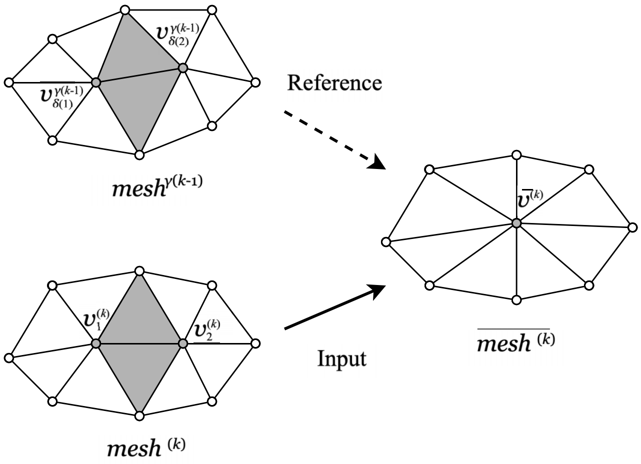


Fig. 10 Tracked Quadric Error Metrics (QEM) Algorithm

Since the inter-frame information was not considered, the original QEM may have led to a lack of temporal consistency in the simplified results. Fortunately, due to the introduction of mesh registration, tracking between two frames was re-established. The Tracked QEM Algorithm redesigned the collapse cost function to utilize this correspondence, as described in Fig.6, above Fig.10 illustrates the proposed design.

Considering the convenience in computing, the similarity between the registration results $mesh^{\gamma(k-1)}$ and the current model $mesh^{(k)}$ should be considered[40]. Which means that when calculating the collapses of edge $e^{(k)}$, optimized items of introduced tracking

$mesh^{\gamma(k-1)}$ in the target $\overline{v^{(k)}}$ should have similar format with $mesh^{(k)}$.

Except for the two vertices $v_1^{(k)}$ and $v_2^{(k)}$ of edge $e^{(k)}$, vertices near to them from tracked $mesh^{\gamma(k-1)}$ will also be considered. Then the optimization target $\overline{v}$ from Eq.(5) changes to $\overline{v^{(k)}}$, where the position has a minimum distance to the adjacent triangle planes $p \in \text{plane}\,(v_i)$. But except merged $v_1^{(k)}$ and $v_2^{(k)}$ in $mesh^{(k)}$, closed vertices $V_{\delta(i)}^{\gamma(k-1)}$ from $mesh^{\gamma(k-1)}$ in a certain neighbourhood $\delta(i)$ centered around $v_i^{(k)}$ also should be considered:

$$\begin{aligned} \overline{v^{(k)}} &= \arg\min_{v} \sum_{p \in \text{plane}(v_i)} \text{distance}(v,p)^2, \\ v_i &\in \{v_1^{(k)}, v_2^{(k)}\} \cup V_{\delta(1)}^{\gamma(k-1)} \cup V_{\delta(2)}^{\gamma(k-1)} \end{aligned} \tag{10}$$

The corresponding rough relationship between the two frames has been established through the mesh registration. The position of $mesh^{\gamma(k-1)}$ and $mesh^{(k)}$ are also aligned into the same space coordinates. Hence the vertices $v_{\delta(i)}^{\gamma(k-1)}$ in neighborhood $\delta(i)$ constructed collection $V_{\delta(i)}^{\gamma(k-1)}$ from $mesh^{\gamma(k-1)}$ for a specific center $v_i^{(k)}$ in $mesh^{(k)}$ can be efficiently built through the K-Nearest Neighbor (K-NN) algorithm. Each neighborhood have a specific search radius $\|e^{(k)}\|$, which means the length of edge $e^{(k)} = \left(v_1^{(k)}, v_2^{(k)}\right)$:

$$\begin{aligned} V_{\delta(i)}^{\gamma(k-1)} &= \left\{ v_{\delta(i)}^{\gamma(k-1)} \middle| \|v_{\delta(i)}^{\gamma(k-1)} - v_i^{(k)}\| < \|e^{(k)}\| \right\}, \\ v_{\delta(i)}^{\gamma(k-1)} &\in mesh^{\gamma(k-1)},\ v_i^{(k)} \in mesh^{(k)} \end{aligned} \tag{11}$$

According to the QEM matrix in Eq.(5), for the given edge collapse $(v_1, v_2) \to \overline{v}$, the sum distance $\Delta(v) = \mathbf{v}^\top \mathbf{Q} \mathbf{v}$ between collapsed vertex $\overline{v}$ and surrounding planes $p \in \text{plane}\,(v_1) \cup \text{plane}\,(v_2)$ can be represent by sum of $\mathbf{Q}_1$ and $\mathbf{Q}_2$. Similarly, the tracked quadric error matrix $\mathbf{Q}^{(k)}$ with certain k-th frame $mesh^{(k)}$ can be obtained:

$$\begin{aligned} \mathbf{Q}^{(k)} &= \mathbf{Q}_{\text{track}(1)}^{(k)} + \mathbf{Q}_{\text{track}(2)}^{(k)} \\ &= \mathbf{Q}_1^{(k)} + \mathbf{Q}_{\delta(1)}^{\gamma(k-1)} + \mathbf{Q}_2^{(k)} + \mathbf{Q}_{\delta(2)}^{\gamma(k-1)} \end{aligned} \tag{12}$$

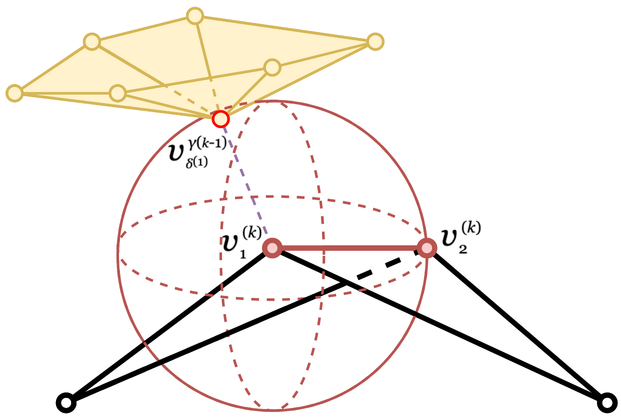
The above formula introduced additional quadric matrices $\mathbf{Q}_{\delta(i)}^{\gamma(k-1)}$, which constructed by the neighbour vertices collection $V_{\delta(i)}^{\gamma(k-1)}$:

$$\mathbf{Q}_{\delta(i)}^{\gamma(k-1)} = \sum_{v \in V_{\delta(i)}^{\gamma(k-1)}} \mathbf{Q}_v^{\gamma(k-1)} \tag{13}$$

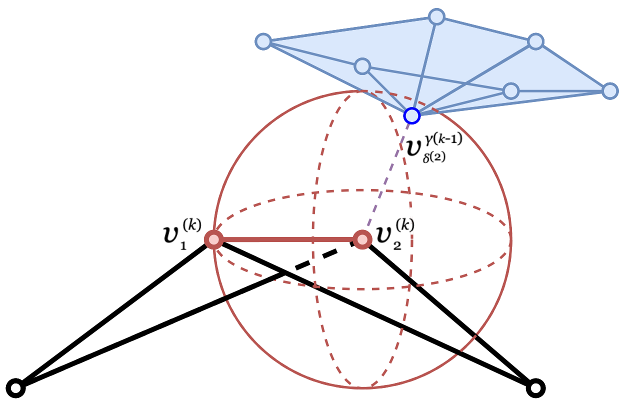
It means that when searching for nearby points, the maximum radius of the search will be limited to the side length of $e^{(k)}$, namely $length(e^{(k)}) = \sqrt{(v_1^{(k)} - v_2^{(k)})^2}$. This restrictive strategy avoids introducing excessive tracking points, resulting in an imbalance in the weight of mesh data between the two frames. The Fig.11 shows the construction process of the neighbour vertices set $V_{\delta(1)}^{(k-1)}$ and $V_{\delta(2)}^{(k-1)}$.

3.3 Approximate optimal search strategy

Another improvement is the search strategy for the approximate optimal collapse position. In the original QEM algorithm, the location of the optimal point $\overline{v}$ is related to the nature of the quadratic matrix $\mathbf{Q}$, which is also the $\mathbf{Q}^{(k)}$ in tracked QEM. According to Eq.(5), suppose the expression of any plane p is $ax + by + cz + d = 0$, the quadratic matrix $\mathbf{Q}_{\text{track}(i)}^{(k)}$ of certain vertex $v_i^{(k)}$ in Eq.(12) can be denoted as the format in Eq.(14).



(a) vertex $v_{\delta(1)}^{\gamma(k-1)}$ in neighbourhood of $v_1^{(k)}$



(b) vertex $v_{\delta(2)}^{\gamma(k-1)}$ in neighbourhood of $v_2^{(k)}$

Fig. 11 Nearest Point Search for Edge in Tracked QEM Based on K-NN Algorithm

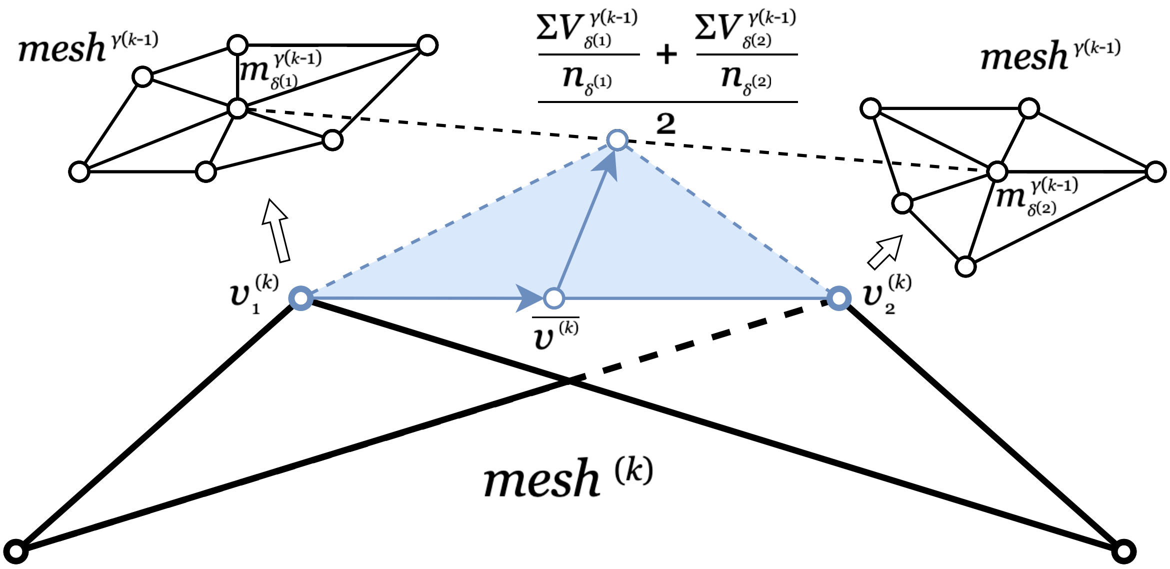


Fig. 12 Approximate solution search on the triangular surface

$$\begin{aligned}
\mathbf{Q}_{\text{track}(i)}^{(k)} &= \mathbf{Q}_i^{(k)} + \mathbf{Q}_{\delta(i)}^{\gamma(k-1)} \\
&= \sum_{v \in V_{\delta(i)}^{\gamma(k-1)} \cup \left\{ v_i^{(k)} \right\}} \sum_{p \in \text{plane}(v)} \mathbf{K}_p \\
&= \sum_p \begin{bmatrix} a_p^2 & a_p b_p & a_p c_p & a_p d_p \\ a_p b_p & b_p^2 & b_p c_p & b_p d_p \\ a_p c_p & b_p c_p & c_p^2 & c_p d_p \\ a_p d_p & b_p d_p & c_p d_p & d_p^2 \end{bmatrix} \\
&= \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix}
\end{aligned} \tag{14}$$

As same as original QEM, according to the Eq.(6), the optimized $\overline{v^{(k)}}$ can be solved by the inverse matrix in the form of Eq.(15), which can get the element value through the matrix $\mathbf{Q}^{(k)}$ stated in Eq.(12) calculate by $\mathbf{Q}_{\text{track}(i)}^{(k)}$ in Eq.(14). However, if the quadric error matrix $\mathbf{Q}$ is irreversible, it is necessary to specify the search method to find approximately optimal solutions.

$$\overline{\mathbf{v}} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{15}$$

In reality, only a simple search strategy is needed since the matrix $\mathbf{Q}$ is invertible in most cases. The original QEM algorithm chooses the midpoint position $(v_1+v_2)/2$ or executes the recursive binary search along the edge e. But according to Eq.(14), accounted neighbour vertices $V_{\delta(i)}^{\gamma(k-1)}$ in tracked QEM increases the probability that the $\mathbf{Q}_{\text{track}(i)}^{(k)}$ is irreversible. For this reason, more precise search strategies must be designed to ensure the quality of the appearance of the simplified model.

In the original QEM, the search area is the segment defined on edge $e^{(k)}$, which is also restricted on an original mesh surface. Since in the tracked QEM algorithm, the introduction of reference frame $mesh^{\gamma(k-1)}$, possible solution should not be limited at the $mesh^{(k)}$ surface anymore. As shown in Fig.12, the proposed search strategy significantly expands the solution space.

The candidate position in the space of the optimized vertex has been expanded into a triangular plane in 3D space. In this proposal, the new search direction is extended by the barycenter $m_\delta^{\gamma(k-1)}$ of tracked neighbor vertices set $V_{\delta(1)}^{\gamma(k-1)}$ and $V_{\delta(2)}^{\gamma(k-1)}$, which defined by the endpoint $v_1^{(k)}$, $v_2^{(k)}$ in current edge $e^{(k)}$:

$$m_\delta^{\gamma(k-1)} = \frac{\frac{\sum V_{\delta(1)}^{\gamma(k-1)}}{n_{\delta(1)}} + \frac{\sum V_{\delta(2)}^{\gamma(k-1)}}{n_{\delta(2)}}}{2} \tag{16}$$

In order to avoid excessive computing expenses, the optimal search will be split into two stages. The first stage is uniform with the original QEM algorithm. Algorithm only proceed search on segment $(v_1^{(k)}, v_2^{(k)})$ to find the optimized result $\overline{v_{1\text{st}}^{(k)}}$ of 1st stage. The second stage is the search on the segment $(\overline{v_{1\text{st}}^{(k)}}, m_\delta^{\gamma(k-1)})$. Therefore, the entire progress to get the final $\overline{v^{(k)}}$ can be considered in staged search on the two line segments:

$\left(v_1^{(k)}, v_2^{(k)}\right)$ and $\left(\overline{v_{1\text{st}}^{(k)}}, m_\delta^{\gamma(k-1)}\right)$.

Algorithm 1 Iteration-based binary search

Require: binary search on segment (v_1, v_2)
Ensure: Approximate $\overline{v}$ with smallest cost
  Initialization: Iteration count $c \leftarrow 0$
  $m \leftarrow \text{middle}(v_1, v_2)$
  while $c < C_{max}$ **do**
    if $\text{cost}(m) < \min(\text{cost}(v_1), \text{cost}(v_2))$ **then**
      $m_1 \leftarrow \text{middle}(v_1, m)$
      $m_2 \leftarrow \text{middle}(m, v_2)$
      if $\text{cost}(m_1) < \text{cost}(m_2)$ **then**
        $v_2 \leftarrow m,\ m \leftarrow m_1$
      else
        $v_1 \leftarrow m,\ m \leftarrow m_2$
      end if
    end if
    Update: $\overline{v} \leftarrow \min(\text{cost}(v_1), \text{cost}(v_2), \text{cost}(m))$
  end while
Ensure: Approximate $\overline{v}$ with smallest cost

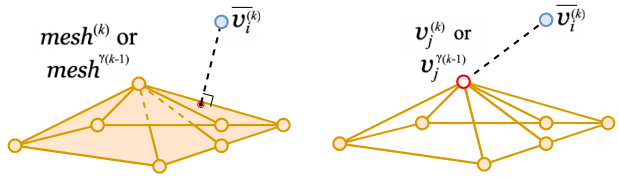
As described in Algorithm 1, the search for both stages uses an iteration-based binary search for performance considerations. The algorithm will first select the middle point m of the search line segment to calculate the cost. If the endpoints are less than both sides, two new middle points m_1 and m_2 will be taken on the two new line segments divided by the current middle point. The segment with the middle point with a more negligible cost will be selected for the next iteration. The maximum number of iterations selected in this item is 4. It is worth noting that when the number of iterations is only 1, the algorithm will be degraded to try to use the center point of triangle $v_1^{(k)}$, $v_2^{(k)}$, $m_\delta^{\gamma(k-1)}$ as the approximate solution.

4. Experiment

4.1 Data and Metrics

In the experiment part, the proposed Tracked QEM was compared with the Original QEM algorithm to evaluate the quality and efficiency of the dynamic mesh decimation task. Both algorithms use the same data and conditions for evaluation to ensure the objectivity of the comparison. A few short 3D volumetric video frame sequences are intercepted as the input. The video totaled 150 frames and recorded a human movement. A total of five frame sequences were cut. Each sequence contains five frames. Each frame has about 20,000 valid vertices, 40,000 triangles, and 60,000 edges.

Due to the tracking QEM algorithm always requiring the previous (k-1)-th frame as the additional reference for each simplified k-th frame, each group conducted four experiments. The algorithm will simplify the mesh from the 2-nd to the 5-th frame, without the 1-st frame. For the same reason, the evaluation will also include the error between each simplified k-th frame and the registered (k-1)-th frame. Simplified results will be visualized and evaluated to analyze the performance of time consistency.



(a) Mesh surface distance (b) Nearest vertex distance

Fig. 13 Indicators based on distance measurement

The evaluation metric of error is designed to measure the inter-frame consistency of the dynamic mesh decimation task in 3D video compression[66]. Inspired by Cloud-Mesh and Cloud-Cloud distance[67], as shown in Fig.13, mesh surface distance and nearest vertex distance were used as main metrics:

1. Mesh surface distance: The shortest distance from each vertex $\overline{v_i^{(k)}}$ in simplified $\overline{mesh^{(k)}}$ to any point on surface of compared original $mesh^{(k)}$ or referenced $mesh^{\gamma(k-1)}$. Which can be used to measure appearance similarity. The smaller value means the result is closer to the input.

2. Nearest vertex distance: The shortest distance from each vertex $\overline{v_i^{(k)}}$ in simplified $\overline{mesh^{(k)}}$ to the corresponding nearest vertex $v_j^{(k)}$ or $v_j^{\gamma(k-1)}$. Which can be used to evaluate the consistency of geometric topology. The smaller value means more approximate to the input.

4.2 Visualized result

Fig.14 shows the visualized results from the Tracked QEM algorithm. A typical simplified frame with good visual effects was selected and visualized. Fig.14.(a) and Fig.14.(b) is the current frame input $mesh^{(k)}$ and previous frame reference $mesh^{(k-1)}$. It is easy to see from Fig.14.(c) that aligned the input $mesh^{(k)}$ from the current k-th frame and tracked $mesh^{\gamma(k-1)}$ from (k-1)-th frame have a high degree of overlapping in appearance. However, the divergence of local details and geometric topology between the two frames still leads to subtle differences in some areas in the registered result. In traditional mesh decimate processing, this difference will affect the consistency of the simplified frame sequence. This conclusion is why the registered reference $mesh^{\gamma(k-1)}$ should be introduced to Tracked QEM.

(a) $mesh^{(k-1)}$ (b) $mesh^{(k)}$ (c) aligned $mesh^{\gamma(k-1)}$ (d) simplified $\overline{mesh^{(k)}}$ (e) overlapping result

Fig. 14 Visualized inputs & simplification result from Tracked QEM Algorithm

In the Tracked QEM Algorithm, both the spatial information in two input $mesh^{\gamma(k-1)}$ and $mesh^{(k)}$ will be leveraged moderately. These highly similar tracked surface information can be used as a good reference part $V_{\delta(i)}^{\gamma(k-1)}$ in Eq.(10). As shown in Fig.10, the new vertices in simplified result will be generated between $mesh^{\gamma(k-1)}$ and $mesh^{(k)}$, which means convert the result $\overline{mesh^{(k)}}$ shown in Fig.14.(d) from single frame to the inter-frame. It can be observed in Fig.14.(e) that if the alignment results are to be superimposed on the final simplified results, the appearance of both frame mesh was retained well in most regions.

(a) $mesh^{\gamma(k-1)}$ (b) $mesh^{(k)}$

Fig. 15 Comparison of inputs between two frames with simplified results

The additional visualized results from the same frame group support this conclusion by comparing the two input meshes with the simplified results separately. As shown in Fig.15.(a) and Fig.15.(b), respectively superimposed with the simplified results $\overline{mesh^{(k)}}$ using $mesh^{\gamma(k-1)}$ and $mesh^{(k)}$, the areas where they overlap are almost identical. The visualized result shows that the proposed method can also ensure that the simplified results have better temporal consistency and spatial smoothing on geometric structure, even if the movement dramatically changes the attitude of the tracked object between two frames. This result is consistent with the expectations in the Tracked QEM algorithm and the results from the next Comparative experiment.

4.3 Comparative experiment

Consistent with the intuitive feelings in visualized results, introducing the reference frame will make the simplified result also numerically closer to the average of two input $mesh^{\gamma(k-1)}$ and $mesh^{(k)}$ than the traditional decimation method, which can be approved by the stochastic comparison on Mesh surface distance and Nearest vertex distance mentioned before.

Considering the simplified result still has several hundreds or thousands of vertices, the measurement of two distances needs to be summarized using statistical metrics. Five metrics were chosen to quantify algorithm performance in this study: average, Mean, Variance, Standard Deviation, Mean Square Error, and Root Mean Square Error. These metrics are used to characterize the distribution of error distances. Since smaller distance values are usually better, smaller statistical values also mean better results.

Table 1 Best result of mesh surface distance

Algorithm	Mesh	Statistical indicator				
		Ave	*Var*	*STD*	*MSE*	*RMSE*
Orig QEM	(k-1)-th	82.73	6143	78.37	56.97	7.547
	k-th	46.23	2129	46.14	21.48	4.635
Track QEM	(k-1)-th	**52.36**	**2448**	**49.47**	**22.42**	**4.734**
	k-th	**45.80**	**1975**	**44.44**	**20.41**	**4.518**

Retain 4 valid digits except variance.

Table 2 Best result of nearest vertex distance

Algorithm	Mesh	Statistical indicator				
		Ave	*Var*	*STD*	*MSE*	*RMSE*
Orig QEM	(k-1)-th	178.2	28798	169.7	**122.4**	**11.06**
	k-th	139.0	22320	149.4	117.3	10.82
Track QEM	(k-1)-th	**144.7**	**24211**	**155.6**	125.2	11.19
	k-th	**134.8**	**20793**	**144.2**	**113.4**	**10.64**

Retain 4 valid digits except variance.

The results in Table.1 and Table.2 are the best results that can be obtained on test data separately for the Tracked and Original QEM algorithm. Table.1 shows that the Tracked QEM achieved better surface distance results in all indicators. This result indicates that unless the gap is not large, the Tracked QEM not only can the simplification of the current frame $mesh^{(k)}$ be achieved without losing the original QEM in the best case, but it can also handle the reference information from previous frame $mesh^{\gamma(k-1)}$ better.

According to the result in Table.2, Tracked QEM can also get the closer vertex distance in the finest of circumstances. Interestingly, although the results of the Original QEM do not include the information from (k-1)-th frame, the distance distribution is better than the Tracked QEM. Considering that the results of Tracked QEM do have a better mean and variance, this difference should be treated as reasonable. The reason is that the QEM algorithm only considers the distance to the neighboring surfaces, and the nearest vertex distance was not part of the optimization objective.

Then is the average performance, which comes from the average of all effective experimental results. The average value of the experimental results is shown in Table.3 and Table.4. Although the best result is better in the value of evaluation metrics, the average performance of the two QEM methods is closer to the real situation in the actual application scenario.

The data in Table.3 and Table.4 shows that Tracked QEM and Original QEM have similar performance at the current k-th frame both on the mesh surface distance and nearest vertex distance. Even on the indicator of mean value, the Tracked QEM algorithm produces a slight degradation.

Table 3 Average value of mesh surface distance

Algorithm	Mesh	Statistical indicator				
		Ave	*Var*	*STD*	*MSE*	*RMSE*
Orig QEM	(k-1)-th	101.4	18424	135.7	95.34	9.764
	k-th	**57.84**	4578	67.66	**48.40**	**6.957**
Track QEM	(k-1)-th	**64.26**	**4246**	**65.16**	**55.25**	**7.433**
	k-th	60.96	**4122**	**64.20**	52.52	7.247

Retain 4 valid digits except variance.

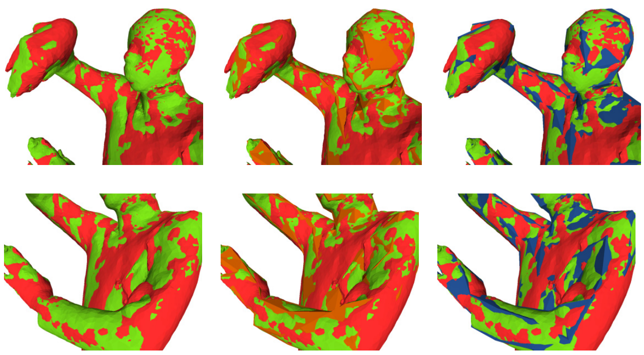
Table 4 Average value of nearest vertex distance

Algorithm	Mesh	Statistical indicator				
		Ave	*Var*	*STD*	*MSE*	*RMSE*
Orig QEM	(k-1)-th	222.4	55731	236.0	220.5	14.85
	k-th	**174.5**	**33238**	**182.3**	176.3	13.27
Track QEM	(k-1)-th	**187.5**	**37425**	**193.4**	**167.7**	**12.95**
	k-th	185.0	35124	187.4	**165.9**	**12.88**

Retain 4 valid digits except variance.

Nevertheless, it is easy to see that the two distances of the Tracked QEM at the previous (k-1)-th frame are optimized well, which can even achieve similar statistic indicators to the current k-th frame. This result shows that when the topology structure between the two frames changes considerably, the Tracked QEM can capture this change well to achieve better temporal consistency than the traditional QEM in the dynamic Mesh decimation task.

4.4 Analysis and discussion



(a) Registration (b) Original QEM (c) Tracked QEM

Fig. 16 Comparison with Traditional QEM Algorithm

If the simplified result $mesh^{\gamma(k-1)}$ is aligned to the exact coordinates as the current frame $mesh^{(k)}$, the surfaces of two meshes should be highly coincident and produce some local crossings, as shown in Fig.16.(a). Accordingly, the simplified result $\overline{mesh^{(k)}}$ should be more flattering in appearance to the new overlapped surfaces after alignment.

Visualization of the comparative experiment can also support conclusions from statistical indicators, result by tracked QEM (Blue mesh in Fig.16.(c)) have a more similar appearance than the original QEM (Orange mesh in Fig.16.(b)). It was evident that the tracked QEM's result is better fitted with finer details like the character's face and palm.

From another perspective, the comparative experiment results show that in most cases, the Tracked QEM algorithm does not significantly improve the simplified performance of the current k-th frame. However, the algorithm's advantage is the temporal consistency between frames; the comparative results prove this opinion.

By introducing reference, Tracked QEM can compress the simplified results from the distance from the registered previous frame $mesh^{\gamma(k-1)}$ to the level similar to the current frame $mesh^{(k)}$, and ensure the simplified $\overline{mesh^{(k)}}$ still have the similar accuracy to the original QEM simultaneously, which proved that the Tracked QEM can significantly improve the time consistency of dynamic mesh decimation without affecting the simplification of the current frame.

For 3D volumetric video compression, mesh decimation is the first stage of the whole encoder pipeline. It aims to reduce the redundant information in the video so that subsequent encoders can compress the video better. The simplification result of track QEM remarkably improves temporal consistency. It makes the simplified mesh sequence more continuous to delineate such redundancies, which helps compress the 3D volumetric video better in the subsequent steps of the encoder pipeline.

5. Conclusion

In this paper, a specialized Tracked QEM Algorithm was proposed, which can use an additional input as a reference from the previous frame to improve temporal consistency between continuous frames. This exploration discussed the potential to extend the conventional mesh simplification algorithm to the dynamic mesh decimation task, offering a tailored solution for mesh decimation in the pre-processing of 3D volumetric video encoding, which seamlessly bridges the difference between consecutive frames.

The most innovative aspect is the application of 3D mesh registration, which facilitates the tracking of mesh models across successive frames, thereby ensuring the temporal consistency of the simplification process. Due to this improvement, the redesigned collapse cost function and the approximate optimal search strategy guarantee the smoothness of simplified mesh sequences.

The experimental results show that tracked QEM emphasizes its excellent performance in maintaining temporal consistency. The error distance to the registered previous frame is reduced to a close approximation to the current frame. This study demonstrated the potential to improve 3D video compression rates at the pre-processing stage. In future work, the method is expected to be integrated into the standard 3D video coding pipeline to evaluate its impact on compression rates further and inspire subsequent research.

Acknowledgment This work was supported by Ministry of Internal Affairs and Communications (MIC) of Japan (Grant no. JPJ000595).

References

1) C. Ziker, B. Truman, H. Dodds : "Cross reality (XR): Challenges and opportunities across the spectrum", Innovative learning environments in STEM higher education: Opportunities, challenges, and looking forward, pp.55-77 (2021)
2) J. Xiong, E.-L. Hsiang, Z. He, T. Zhan, S.-T. Wu : "Augmented reality and virtual reality displays: emerging technologies and future perspectives", Light: Science & Applications, 10,1, pp.216 (2021)
3) O. Schreer, I. Feldmann, S. Renault, M. Zepp, M. Worchel, P. Eisert, P. Kauff : "Capture and 3D video processing of volumetric video", 2019 IEEE International conference on image processing (ICIP), pp.4310-4314 (2019)
4) M. Xu, C. Li, S. Zhang, P. Le Callet : "State-of-the-Art in 360° Video/Image Processing: Perception, Assessment and Compression", IEEE Journal of Selected Topics in Signal Processing, 14,1, pp.5-26 (2020)
5) M. Wien, J.M. Boyce, T. Stockhammer, W.-H. Peng : "Standardization status of immersive video coding", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 9,1, pp.5-17 (2019)
6) T. Sikora : "MPEG digital video-coding standards", IEEE signal processing magazine, 14,5, pp.82-100 (1997)
7) J.M. Boyce, R. Doré, A. Dziembowski, J.

Fleureau, J. Jung, B. Kroon, B. Salahieh, V.K.M. Vadakital, L. Yu : "MPEG immersive video coding standard", Proceedings of the IEEE, 109,9, pp.1521-1536 (2021)

8) K. Mammou, J. Kim, A.M. Tourapis, D. Podborski, D. Flynn : "Video and Subdivision based Mesh Coding", 2022 10th European Workshop on Visual Information Processing (EUVIP), pp.1-6 (2022)

9) Y. Choi, J.-B. Jeong, S. Lee, E.-S. Ryu : "Overview of the Video-based Dynamic Mesh Coding (V-DMC) Standard Work", 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), pp.578-581 (2022)

10) M. Garland, P.S. Heckbert : "Surface simplification using quadric error metrics", Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp.209-216 (1997)

11) A. Maglo, G. Lavoué, F. Dupont, C. Hudelot : "3d mesh compression: Survey, comparisons, and emerging trends", ACM Computing Surveys (CSUR), 47,3, pp.1-41 (2015)

12) I. Viola, P. Cesar : "Chapter 15 - Volumetric video streaming: Current approaches and implementations", Immersive Video Technologies, Academic Press, pp.425-443 (2023)

13) Y. Diez, F. Roure, X. Lladó, J. Salvi : "A qualitative review on 3D coarse registration methods", ACM Computing Surveys (CSUR), 47,3, pp.1-36 (2015)

14) G.K.L. Tam, Z.-Q. Cheng, Y.-K. Lai, F.C. Langbein, Y. Liu, D. Marshall, R.R. Martin, X.-F. Sun, P.L. Rosin : "Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid", IEEE Transactions on Visualization and Computer Graphics, 19,7, pp.1199-1217 (2013)

15) M. Rodrigues, R. Fisher, Y. Liu : "Introduction: Special issue on registration and fusion of range images", Computer Vision and Image Understanding, 87, pp.1-7 (2002)

16) J. Salvi, C. Matabosch, D. Fofi, J. Forest : "A review of recent range image registration methods with accuracy evaluation", Image and Vision computing, 25,5, pp.578-596 (2007)

17) R. Sandhu, S. Dambreville, A. Tannenbaum : "Particle filtering for registration of 2D and 3D point sets with stochastic dynamics", 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp.1-8 (2008)

18) D. Aiger, N.J. Mitra, D. Cohen-Or : "4-points congruent sets for robust pairwise surface registration", ACM SIGGRAPH 2008 papers, pp.1–10 (2008)

19) A. Myronenko, X. Song : "Point set registration: Coherent point drift", IEEE transactions on pattern analysis and machine intelligence, 32,12, pp.2262-2275 (2010)

20) B. Bellekens, V. Spruyt, R. Berkvens, M. Weyn : "A survey of rigid 3d point cloud registration algorithms", AMBIENT 2014: the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy, pp.8-13 (2014)

21) D. Chetverikov, D. Svirko, D. Stepanov, P. Krsek : "The trimmed iterative closest point algorithm", 2002 International Conference on Pattern Recognition, 3, pp.545-548 (2002)

22) D. Chetverikov, D. Stepanov, P. Krsek : "Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm", Image and vision computing, 23,3, pp.299-309 (2005)

23) S. Bouaziz, A. Tagliasacchi, M. Pauly : "Sparse iterative closest point", Computer graphics forum, 32,5, pp.113-123 (2013)

24) J. Zhang, Y. Yao, B. Deng : "Fast and robust iterative closest point", IEEE Transactions on Pattern Analysis and Machine Intelligence, 44,7, pp.3450-3466 (2021)

25) S. Rusinkiewicz, M. Levoy : "Efficient variants of the ICP algorithm", Proceedings third international conference on 3-D digital imaging and modeling, pp.145-152 (2001)

26) P.J. Besl, N.D. McKay : "Method for registration of 3-D shapes", Sensor fusion IV: control paradigms and data structures, 1611, pp.586-606 (1992)

27) J. Serafin, G. Grisetti : "NICP: Dense normal based point cloud registration", 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp.742-749 (2015)

28) O. van Kaick, H. Zhang, G. Hamarneh, D. Cohen-Or : "A Survey on Shape Correspondence", Computer Graphics Forum, 30,6, pp.1681-1707 (2011)

29) R.W. Sumner, J. Schmid, M. Pauly : "Embedded deformation for shape manipulation", ACM

SIGGRAPH 2007 Papers, pp.80–es (2007)
30) H. Pottmann, Q.-X. Huang, Y.-L. Yang, S.-M. Hu : "Geometry and convergence analysis of algorithms for registration of 3D shapes", International Journal of Computer Vision, 67, pp.277-296 (2006)
31) B. Amberg, S. Romdhani, T. Vetter : "Optimal step nonrigid ICP algorithms for surface registration", 2007 IEEE conference on computer vision and pattern recognition, pp.1-8 (2007)
32) H. Hontani, T. Matsuno, Y. Sawada : "Robust nonrigid ICP using outlier-sparsity regularization", 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp.174-181 (2012)
33) S. Monji-Azad, J. Hesser, N. Löw : "A review of non-rigid transformations and learning-based 3D point cloud registration methods", ISPRS Journal of Photogrammetry and Remote Sensing, 196, pp.58-72 (2023)
34) H. Chui, A. Rangarajan : "A new point matching algorithm for non-rigid registration", Computer Vision and Image Understanding, 89,2-3, pp.114-141 (2003)
35) N. Gelfand, N.J. Mitra, L.J. Guibas, H. Pottmann : "Robust global registration", Symposium on geometry processing, 2,3, pp.5 (2005)
36) B. Jian, B.C. Vemuri : "A robust algorithm for point set registration using mixture of Gaussians", Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, 2, pp.1246-1251 (2005)
37) J. Ma, W. Qiu, J. Zhao, Y. Ma, A.L. Yuille, Z. Tu : "Robust L_2E estimation of transformation for non-rigid registration", IEEE Transactions on Signal Processing, 63,5, pp.1115-1129 (2015)
38) Y. Yao, B. Deng, W. Xu, J. Zhang : "Quasi-newton solver for robust non-rigid registration", Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.7600-7609 (2020)
39) P.W. Holland, R.E. Welsch : "Robust regression using iteratively reweighted least-squares", Communications in Statistics-theory and Methods, 6,9, pp.813-827 (1977)
40) D.P. Luebke : "A developer's survey of polygonal simplification algorithms", IEEE Computer Graphics and Applications, 21,3, pp.24-35 (2001)
41) M.-E. Algorri, F. Schmitt : "Mesh Simplification", Computer Graphics Forum, 15,3, pp.77-86 (1996)
42) J.O. Talton : "A short survey of mesh simplification algorithms", University of Illinois at Urbana-Champaign (2004)
43) W.J. Schroeder, J.A. Zarge, W.E. Lorensen : "Decimation of triangle meshes", Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pp.65-70 (1992)
44) P. Lindstrom, D. Koller, W. Ribarsky, L.F. Hodges, N. Faust, G.A. Turner : "Real-time, continuous level of detail rendering of height fields", Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp.109-118 (1996)
45) R. Ronfard, J. Rossignac : "Full-range approximation of triangulated polyhedra.", Computer graphics forum, 15,3, pp.67-76 (1996)
46) W.J. Schroeder : "A topology modifying progressive decimation algorithm", Proceedings. Visualization'97 (Cat. No. 97CB36155), pp.205-212 (1997)
47) S.J. Kim, W.K. Jeong, C.H. Kim : "LOD generation with discrete curvature error metric", Proceedings of Korea Israel Bi-National Conference, pp.97-104 (1999)
48) R. Klein, G. Liebich, W. Straßer : "Mesh reduction with error control", Proceedings of Seventh Annual IEEE Visualization'96, pp.311-318 (1996)
49) P.S. Heckbert, M. Garland : "Optimal triangulation and quadric-based surface simplification", Computational Geometry, 14,1-3, pp.49-65 (1999)
50) S.-J. Kim, C.-H. Kim, D. Levin : "Surface simplification using a discrete curvature norm", Computers & Graphics, 26,5, pp.657-663 (2002)
51) M. Garland, Y. Zhou : "Quadric-based simplification in any dimension", ACM Transactions on Graphics (TOG), 24,2, pp.209-239 (2005)
52) H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle : "Mesh optimization", Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, pp.19–26 (1993)
53) H. Hoppe : "Progressive meshes", Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp.99–108 (1996)
54) H. Hoppe : "View-dependent refinement of progressive meshes", Proceedings of the 24th annual

conference on Computer graphics and interactive techniques, pp.189-198 (1997)

55) A. Khodakovsky, P. Schröder, W. Sweldens : “Progressive geometry compression”, Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp.271-278 (2000)

56) P. Alliez, M. Desbrun : “Progressive compression for lossless transmission of triangle meshes”, Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp.195-202 (2001)

57) C. DeCoro, N. Tatarchuk : “Real-time mesh simplification using the GPU”, Proceedings of the 2007 symposium on Interactive 3D graphics and games, pp.161-166 (2007)

58) A. Papageorgiou, N. Platis : “Triangular mesh simplification on the GPU”, The Visual Computer, 31, pp.235-244 (2015)

59) H. Lee, M.-H. Kyung : “Parallel mesh simplification using embedded tree collapsing”, The Visual Computer, 32,6-8, pp.967-976 (2016)

60) Y. Liang, F. He, X. Zeng : “3D mesh simplification with feature preservation based on whale optimization algorithm and differential evolution”, Integrated Computer-Aided Engineering, 27,4, pp.417-435 (2020)

61) R.A. Potamias, G. Bouritsas, S. Zafeiriou : “Revisiting point cloud simplification: A learnable feature preserving approach”, Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II, pp.586-603 (2022)

62) R.A. Potamias, S. Ploumpis, S. Zafeiriou : “Neural mesh simplification”, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.18583-18592 (2022)

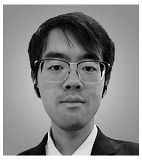
63) H. Hoppe : “New quadric metric for simplifying meshes with appearance attributes”, Proceedings Visualization’99 (Cat. No. 99CB37067), pp.59-510 (1999)

64) T. Matsuyama, X. Wu, T. Takai, S. Nobuhara : “Real-time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualization for 3D video”, Computer Vision and Image Understanding, 96,3, pp.393-434 (2004)

65) G. Pavlakos, J. Malik, A. Kanazawa : “Human mesh recovery from multiple shots”, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.1485-1495 (2022)

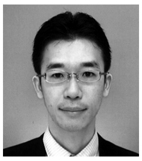
66) Y. Liu, J. Xu, K. Kawamura, H. Watanabe : “Measure The Difference between Frames: Reconsidering The Evaluation Method in Dynamic Mesh Simplification”, 2023 IEEE 12th Global Conference on Consumer Electronics (GCCE), pp.372-375 (2023)

67) A. Guezlec : “Meshsweeper: dynamic point-to-polygonal mesh distance and applications”, IEEE Transactions on visualization and computer graphics, 7,1, pp.47-61 (2001)



Yunlong Liu Yunlong Liu received the B.E. degree from Nanjing University of Posts and Telecommunications in 2019, and got the M.E. from Waseda University in 2022 respectively. Now he is pursuing his Ph.D. degree at Graduation school of Fundamental Science and Engineering, Waseda University. He is major in Computer Science and Communication Engineering, and mainly concerned with research in directions related to Deep Learning, Computer Vision and Multi-media Technology. This paper is an attempt related to 3D video processing.

Jianfeng Xu Jianfeng Xu received the B.S. (with honor) and the M.S. degrees from Tsinghua University, China, in 2001 and 2004 respectively, and the Ph.D. degree from the University of Tokyo, Japan, in 2007. He has worked at KDDI Research, Inc. since 2007 and is now a Core Researcher in the Advanced Visual Communication Laboratory. His research interests include dynamic mesh coding technologies and standardization, computer graphics and computer vision, as well as deep learning technologies. He is a member of IEICE and ITE.



Kei Kawamura Kei Kawamura received his B.E., M.Sc., and Ph.D. degrees in Global Information and Telecommunication Studies from Waseda University, Japan, in 2004, 2005, and 2013, respectively. He joined KDDI in 2010. He has been involved with developing HEVC, VVC, and V-DMC standards under JCT-VC, JVET, and MPEG-3DGH. He is currently engaged in the research and development of a video and dynamic mesh coding system at KDDI Research, Inc. His research interests include image and video processing, video and dynamic mesh coding, and multimedia distribution. He is a rapporteur of Question 1 of ITU-T Study Group 9 from 2020. He is a member of IEEE, IEICE, and ITE.

Hiroshi Watanabe received B.E., M.E. and Ph.D. degrees in Engineering from Hokkaido University in 1980, 1982, and 1985, respectively. He joined Nippon Telegraph and Telephone Corporation (NTT) in 1985, and was engaged in research and development of image and video coding systems at NTT Human Interface Labs. and NTT Cyber Space Labs. until 2000. He has also been involved with the development of JPEG, MPEG standards under JTC 1/SC 29. He is currently a professor at the Department of Communications and Computer Engineering, School of Fundamental Science and Engineering, Waseda University. His research interests include object recognition, deep learning, image processing, video coding, and multimedia distribution. He is a member of IEEE, IEICE, ITE, IPSJ, and IIEEJ.