electronics

MDPI

Article

A Method for Generating Geometric Image Sequences for Non-Isomorphic 3D-Mesh Sequence Compression

Yuan Gao [1], Zhiqiang Wang [2,3] and Jin Wen [2,*]

[1] Department of Electronics and Communications Engineering, Beijing Electronic Science and Technology Institute, Beijing 100070, China; gy@besti.edu.cn
[2] Department of Cyberspace Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China; wangzq@besti.edu.cn
[3] State Information Center, Beijing 100045, China
* Correspondence: 20212818@mail.besti.edu.cn

Abstract: As virtual reality and 3D-modeling technology continue to advance, the amount of digital geometric media data is growing at an explosive rate. For example, 3D meshes, an important type of digital geometric media, can precisely record geometric information on a model's surface. However, as the complexity and precision of 3D meshes increase, it becomes more challenging to store and transmit them. The traditional method of compressing non-isomorphic 3D-mesh sequences through frame-by-frame compression is inefficient and destroys the inter-frame correlations of the sequences. To tackle these issues, this study investigates the generation of time-dependent geometric image sequences for compressing non-isomorphic 3D-mesh sequences. Two methods are proposed for generating such sequences: one through image registration and the other through parametrization-geometry cooperative registration. Based on the experimental compression results of the video-coding algorithms, it was observed that the proposed geometric image-sequence-generation method offers superior objective and subjective qualities, as compared to the traditional method.

Keywords: non-isomorphic 3D-mesh sequence; geometric image sequence; image/video compression; thin-plate spline (TPS)

Citation: Gao, Y.; Wang, Z.; Wen, J. A Method for Generating Geometric Image Sequences for Non-Isomorphic 3D-Mesh Sequence Compression. *Electronics* **2023**, *12*, 3473. https://doi.org/10.3390/electronics12163473

Academic Editors: Xiaoming Li and Henghui Ding

Received: 27 July 2023
Revised: 14 August 2023
Accepted: 15 August 2023
Published: 16 August 2023

1. Introduction

With the rapid development of computer graphics technology and 3D-modeling technology, digital geometric media represented by 3D models simulating real objects and their movements have become one of the new carriers for human beings to obtain information [1,2]. It is widely used in various fields, such as medical imaging, scientific exploration, engineering design, simulation games, movie special effects, etc., and it also provides important technical support for virtual reality (VR) technology.

A 3D model can be used to simulate any physical object that exists in the physical world as well as imaginary objects.It can be artificially synthesized by computer programs, such as 3D-modeling software, or it can be obtained by capturing real objects in the real world using professional perception devices, such as Cyberware 3D scanners [3,4] and multi-view cameras [5]. Although 3D models can be expressed in various ways such as metadata representation, surface representation, entity representation and high-level structures representation (High-level structures) [6], the mesh form (especially the widely used triangular mesh [7]) has become the most popular method for representing 3D models and describing 3D objects, due to its simple structure and universal form.

With the rapid development of virtual reality technology and 3D-scanning technology, 3D-model collection, acquisition, and modeling are becoming more and more mature, thus generating a large number of complex 3D-mesh data across various applications, and the increasing complexity and accuracy has placed tremendous pressure on traditional compression algorithms and processing techniques.

In addition, with the widespread use of dynamic 3D meshes, the size of 3D-mesh data in compression and transmission has grown dramatically. A 3D-mesh sequence is a collection of continuous 3D meshes in the time domain obtained by sampling dynamic meshes at a specific frame rate, and they are highly correlated in the time domain.However, the 3D-mesh sequences generated by the capture of real objects by specialized perception equipment, the way in which the vertices of the frames are connected varies in the time domain, and the temporal correlation of such mesh sequences is hidden. The traditional frame-by-frame compression method cannot effectively extract and exploit the inter-frame correlations of mesh sequences, which reduces the compression efficiency of non-isomorphic 3D-mesh sequences. How to extract and effectively utilize the inter-frame correlation of non-isomorphic 3D-mesh sequences to achieve simple and efficient compression has become a pressing problem in the field of digital geometric media.

Therefore, this study focused on the representation of geometric images, addressing the problem that the traditional frame-by-frame compression of non-isomorphic 3D-mesh sequences destroys the inter-frame correlations of the sequences, and we proposed a geometric image-sequence-generation method for non-isomorphic 3D-mesh sequence compression.

The main contributions of this study was as follows:

- Image registration for geometric-image-sequence generation: A non-rigid registration of geometric image sequences generated by traditional methods using a thin-plate spline function was proposed to generate geometric image registration sequences, thereby improving the correlation between adjacent frames of geometric image sequences and enhancing the compression performance of the system.
- Parametrization-geometry cooperative registration for geometric-image-sequence generation: We proposed a generation scheme of geometric image sequences by parametrization-geometry cooperative registration, of which the core was blending non-rigid registration into the geometric image-sequence generation process. This scheme overcame the shortcoming of traditional parametrization, which damages the temporal correlations of non-isomorphism 3D-mesh sequences, and improves the compression performance.
- Validation and evaluation of effectiveness: We used the H.264 video-coding standard to compress the geometric image sequences generated by our algorithms as well as the traditional algorithm. The experimental results showed that the objective and subjective qualities of the decoding and the reconstruction of the proposed methods were better than those of the traditional method.

The other sections of this paper are organized as follows: Section 2 presents the progress of related research; Section 3 introduces the preliminary knowledge; Section 4 introduces the methods of this study; Section 5 verifies and evaluates the effectiveness of the methods through experiments; and Section 6 concludes this paper.

2. Related Work

With the rapid development of information technology and the wide application of 3D-video-capture equipment, the compression and coding of dynamic 3D-mesh data have become a research hotspot in the field of 3D digital geometry media processing [8–15]. Time-dependent 3D-mesh sequences are taken from dynamic 3D meshes that contain a significant amount of redundant information, similar to 2D-video image sequences [16,17]. Three-dimensional mesh-sequence-compression coding removes mesh-sequence redundancy by methods such as prediction and transformation, specifically through the temporal correlation of each frame of mesh.However, when 3D-mesh sequences are generated by the capture of real objects by specialized perception equipment, the way in which the vertices of each frame are connected varies in the time domain, and the temporal correlations of such topologically non-isomorphic mesh sequences are hidden and difficult to compress and encode.

Previous research on the compression of non-isomorphic 3D-mesh sequences has taken two routes: the traditional frame-by-frame encoding without changing the topology and the uniform encoding of the normalized topology. The current status of research on the two compression routes is briefly reviewed next.

2.1. Frame-by-Frame Encoding without Changing the Topology

Since the vertex connectivity of each frame of topologically non-isomorphic mesh sequences is different, common practice is to use the intra-frame correlation of the 3D mesh to encode the topology structure and the vertex position information of the static 3D mesh, frame by frame [18–30]. In this compression method, each frame of mesh is compressed completely independently, discarding the inter-frame correlations of the mesh sequences, so there is much room for improvement in compression efficiency. In recent years, researchers have tried to make inter-frame predictions of the vertex position information of the mesh without changing the topology of each frame, and they have made some progress.

In 2007, Han et al. [31] studied the encoding of the vertex position information of non-isomorphic 3D-mesh sequences, extended the block-matching-based motion-compensation algorithm to 3D space, with reference to 2D-video-compression methods, and proposed a non-isomorphic 3D-mesh sequence-compression method based on an extended block-matching algorithm (EBMA). The method first decomposes the mesh model into several surfaces based on cubic blocks, takes the previous frame of the mesh sequence as the reference frame, and then searches the matching blocks in a space 4 times the size of the reference frame by the normal direction and an area of the frame blocks to obtain the motion vector of the current block. Since the topological non-isomorphism of each frame makes the number of vertices in the current block different from those of the matching block, the corresponding points of the minimum Euclidean distance are found within the matching block to obtain the predicted residuals of the current block. Finally, differential pulse code modulation (DPCM) is used to encode the motion vector, and Huffman coding is used to encode the predicted residual coefficient after the discrete cosine transformation (DCT). The experimental results have demonstrated that the partial frames of the experimental sequence could achieve 10–18% encoding efficiency.

In 2008, Han et al. [32] studied the quantized encoding of vertex position information of non-isomorphic 3D-mesh sequences. Two quantization levels were proposed for spatiotemporal redundancy: For temporal redundancy, a coarse-level quantization was performed based on the block separation and analyzed patterns of block changes; for spatial redundancy, an efficient fine-level quantization was performed on the intra-frame block vertex position data. Since the generated binary vertex position sequence contained a large number of consecutive zeros, efficient compression could be performed using run-length coding and arithmetic coding. The experimental results demonstrated that the experimental sequences could achieve 1.9–16% encoding efficiency.

In 2010, Yamasaki and Aizawa [33] proposed a block-level-based intra-frame and inter-frame coding method for non-isomorphic 3D-mesh sequences. This method decomposed the mesh model into multiple block levels with approximately the same surface areas in order to effectively exploit the spatial correlations between the block levels. For intra-frame coding, the spectral values of the input data were computed using the Kirchhoff matrix, and then the quantized spectral values were coded using Huffman coding and output. For inter-frame encoding, the first frame of the mesh sequence was used as the reference frame for the block-level matching, using principal component analysis (PCA). The experimental results showed that the intra-frame and inter-frame compression method based on the block level could achieve a code rate of 4–14 bpv (bits per vertex, in bpv) with a root-mean-squared error of 0.01–0.02.

The block-level inter-frame predictive coding method described above was prone to blocking artifacts during block-matching. Therefore, Doumanoglou et al. [34] proposed a vertex-level-based inter-frame predictive coding method for non-isomorphic mesh sequences using a bone-matching and ICP (iterative closest point) algorithm. This method

divided the mesh sequence into the I (intra) frame and EP (enhanced predicted) frame. The I frame of the mesh was used as the reference frame of the EP-frame coding, and it was compressed by the conventional static mesh-coding method. The EP frame used the bone-matching and ICP algorithm to register the reference frame for the input mesh; encoded and output the residual between the original mesh and the predicted mesh; and then quantized and encoded the output through entropy constraints. To improve the prediction, Doumanoglou et al. [35] mined the correlated activity-level properties of non-isomorphic 3D-mesh sequences and proposed 2 I-frame selection methods based on the skeletal-matching principle and the periodic performance-parameter selection.

The above-mentioned compression methods for non-isomorphic 3D-mesh sequences have all adopted the traditional way of encoding, without changing the mesh topology. Although the inter-frame correlation of a mesh sequence was exploited to encode the position information of the mesh predictively, the literature [32–34] has also made some improvements and enhancements to the encoding of the mesh topology. All of these schemes need to encode the topological connection information of each, frame by frame, which affects the overall compression performance of a mesh sequence. Therefore, normalizing the topological structure of each frame of a sequence has become a problem worthy of further study in the field of non-isomorphic 3D-mesh sequence compression.

2.2. Normalized Topology Uniform Code

The normalization of each frame of the topologically non-isomorphic 3D-mesh sequence into the same topology can reduce the code rate of encoding the topological connection information, frame by frame, so that the inter-frame correlation of a mesh can be effectively applied to the topological compression of the non-isomorphic 3D-mesh sequence, and the geometric data can be compressed using the conventional mesh-sequence-coding algorithm, which improves the overall compression efficiency of the sequence. There are two types of methods to unify the topology of the mesh sequence: registering the non-isomorphic 3D-mesh sequence from the representation of the original mesh, and generating the geometric image sequence from the representation of a geometric image.

2.2.1. Registration of Non-Isomorphic 3D Mesh Sequences

Establishing accurate point-to-point data correspondence between 2 completely different 3D models is a very challenging task due to the large structural differences between the individual samples of different models (especially face models) and the presence of non-rigid deformation, etc. At present, in terms of 3D-data registration, registration algorithms based on spatial transformation [36] are the most common. According to the different properties of spatial transformation, 3D-data registration can be divided into registration algorithms based on rigid deformation and registration algorithms based on non-rigid deformation.

Rigid deformation refers to global transformations such as the model's rotation, translation, and scaling. It is mainly used to change different viewing angles of the same model, and the distance between two points in the model, before and after the transformation, remains unchanged. The most typical rigid-deformation-registration algorithm is the 3D-body registration based on ICP (iterative closest point) [37]. This method uses rigid deformation and a closest-point search to establish the point-to-point correspondence of the dataset. Therefore, it can be applied to search matching of artificially synthesized topologically isomorphic 3D-mesh sequence compression [38,39].

For models with large differences in complex surface personality (such as face data, etc.), the global rigid deformation based on rotation, translation, and scaling can only obtain a rough global registration [40]. Therefore, the distance between any 2 points in the rigid deformation remains constant, which is not fully applicable to 3D-mesh sequence registration compression with different topologies and complex surface geometry variations [34,35]. Therefore, the 3D-data-registration method based on non-rigid deformation has gradually become a hotspot in the field of 3D-model processing research. Myronenko

and Song [41] proposed a non-rigid registration algorithm based on coherent point drift (CPD), which used a deterministic annealing EM algorithm to optimize the max-likelihood estimation and thus obtained the point-to-point correspondence of two sample sets. Hu et al. [42,43] conducted a non-rigid deformation study on 3D-face data based on the thin-plate spline (TPS) function and obtained better matching results. After generating a topologically isomorphic 3D-mesh sequence by registering the non-isomorphic mesh sequence, it could be compressed based on common dynamic 3D-mesh-compression methods such as a priori segmentation methods [44,45], principal component analysis (PCA) [46,47], spatiotemporal prediction methods [48,49], wavelet transform methods [50,51], and MPEG [52].

In this study, non-rigid registration was introduced into the field of non-isomorphic 3D-mesh sequence compression, and the coding method based on prediction could make full use of the temporal correlation of the sequence and effectively remove the inter-frame redundancy of the mesh sequence.

2.2.2. Generating Geometric Image Sequences

The use of mesh parametrization methods allows for the mapping of 3D meshes to 2D planes, enabling them to be effectively compressed by image-coding algorithms, such as wavelet transforms [53]. Briceño et al. [54] parameterized the selected reference frame for an artificially synthesized dynamic 3D mesh and then applied the generated parametrization result to other frames of the mesh, thus generating a geometric image sequence (geometric video). Using the geometric image sequence to normalize the 3D-mesh sequence could utilize the existing mature video-coding algorithm effectively. However, their method was only suitable for topologically isomorphic dynamic meshes, and for topologically non-isomorphic mesh sequences, the parametrization method in [54] did not work. Subsequent studies [55,56] for the parametrization of non-isomorphic mesh sequences have primarily been based on the matching of the feature blocks of the original mesh, followed by parametrization to generate geometric image sequences (e.g., typical features of human motion and facial expressions, etc.). Hou et al. [57,58] proposed a two-dimensional geometric video coding and compression framework using geometric image sequences generated by the above parametrization method, based on the premise that the low-rank features of dynamic meshes and the intrinsic structure remained unchanged.

In addition, topologically non-isomorphic 3D-mesh sequences could be parametrically mapped, frame by frame, to the 2D plane using a method suggested in the literature [53] to generate a geometric image sequence of normalized representation, which could then be compressed using existing mature video-coding algorithms, reducing the compression complexity of non-isomorphic 3D-mesh sequences. At the same time, a frame-by-frame parametrization method can make the generated geometric image sequence conform to the geometric characteristics of the original mesh in space. However, because this parametrization method mapped each frame of the non-isomorphic mesh sequence to the 2D plane separately, no correspondence could be established between the same feature vertices in each frame. Therefore, the conventional geometric image sequences generated by frame-by-frame parametrization could not maintain the temporal correlations of the original non-isomorphic mesh sequences, which reduced the compression efficiency.

Therefore, there is an urgent need for a method to generate time-dependent geometric image sequences that can adequately maintain both the original mesh geometric features and the inter-frame correlations of the mesh sequences, in order to effectively improve the compression performance of non-isomorphic 3D-mesh sequences, based on geometric image representation.

3. Preliminary Knowledge

Gu et al. [53] proposed geometric images, a fully normalized 3D-mesh representation, so that the topology of the 3D-mesh was obtained using the pixel adjacencies of the image array, and the vertex coordinates were computed by coordinate-mapping through the 3 color values of the geometric image. Then, Briceño et al. [54] proposed geometry

video, which parameterized the selected reference frame and applied the generated parameterized results to other frames of the mesh, and finally, a sequence of geometric images was generated after resampling and quantization. However, their method was only for synthetic topologically isomorphic dynamic meshes, and for topologically non-isomorphic mesh sequences, the parametrization method of the literature [54] did not work. Therefore, the non-isomorphic 3D-mesh sequence would need to use a frame-by-frame parametrization method to generate a geometric image sequence. The traditional frame-by-frame parametrization [53] to generate a geometric image sequence is described as follows:

Given a sequence of triangular mesh $\{H_1, H_2, \ldots, H_s\}$, the triangular mesh at the kth frame can be represented as $H_k = \{V_k, F_k\}_{k=1,2,\ldots,s}$. The mesh H_k consists of 2 types of geometric data, including the set of 3D coordinates V_k of the vertices and the topological connection relation F_k between them, where $V_k = \{V_{ki} | V_{ki} = (x_{ki}, y_{ki}, z_{ki})\}_{i=1,\ldots,s_k}$, $F_k = \left\{ f_{kj} | f_{kj} = \left(v_{ki_1}, v_{ki_2}, v_{ki_3}\right) \right\}_{j=1,\ldots,t_k}$. In this study, the traditional geometric stretching parametrization method [59] was used to map the mesh H_k onto the parameter domain, frame by frame. The parametric coordinate L of the H_k vertices is given by Equation (1) and was obtained by minimizing the geometric stretching function of all triangular surface pieces in the parametric domain.

$$L^2(H_k) = \frac{\sum_{T_{kj} \in H_k} \left(\frac{\Gamma_{kj}{}^2 + \gamma_{kj}{}^2}{2} area\left(T_{kj}\right) \right)}{\sum_{T_{kj} \in H_k} area\left(T_{kj}\right)} \tag{1}$$

where T_{kj} represents the jth parameter triangle on the mesh H_k at the kth frame. $area\left(T_{kj}\right)$ is the surface area of the triangle T_{kj}, and Γ_{kj}, γ_{kj} represent the max and minimum singular values, respectively, of the Jacobi matrix, obtained by affine mapping of the vertex parameter coordinates of the parameter triangle T_{kj}.

The parameterized mesh was resampled to generate a $m_1 \times m_2$ resolution mesh, which was then quantized into a sequence of n-bit pixel bit-depth geometric images $\{G_1, G_2, \ldots, G_s\}$, where $G_k = \left\{ G_{uv}^k | G_{uv}^k = \left(r_{uv}^k, g_{uv}^k, b_{uv}^k\right) \right\}_{u=1,\ldots,m_1}^{v=1,\ldots,m_2}$, and $r, g, b \in [2^n - 1]$. The vertices of each frame of the reconstructed mesh were represented by the pixel points of each frame of the geometric image sequence, and the coordinates of the vertices could be calculated from the three color values of the geometric image by coordinate mapping. The pixel adjacencies of the geometric image sequence were consistent for each frame of the image array. It had been stipulated that each pixel point G_{uv}^k in each frame had to be connected with its surrounding 6 neighboring pixel points, and each point in the mesh had to be connected with 6 adjacent vertices regularly to form 6 adjacent triangular face slices, and this adjacency relationship represented the topology of the reconstructed mesh sequence.

Taking a set of face-mesh smile-expression sequences as an example (row 1 of Figure 1), the process of traditional frame-by-frame parametrization to generate geometric image sequences and reconstruct mesh sequences is shown in Figure 2, and the generated geometric image sequences are shown in the second row of Figure 1. The 3D-mesh sequence expressed by the unified geometric image sequence had the same topological connection relationship in each frame, thus solving the problem that time-dependent compression could not be directly applied to topologically non-isomorphic 3D-mesh sequences, and at the same time, the burden of compression was reduced because the regular topology did not require channel coding in the compression process. However, it was obvious in Figure 1 that the topological non-isomorphism of each frame led to a large difference between the adjacent frames of the generated geometric image sequence (line 2 of Figure 1). As a result, the feature vertices of the mesh represented by the pixel points of each frame were not aligned, which destroyed the temporal correlation of the non-isomorphic 3D-mesh sequence and limited the performance of the compression.

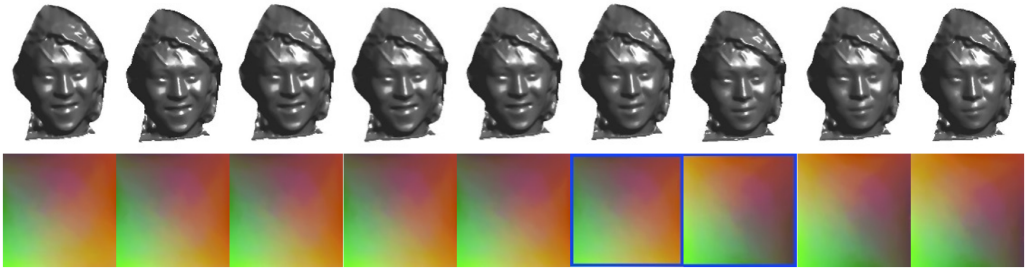

Figure 1. Row 1: the original 3D-mesh sequence. **Row 2**: the geometric image sequence generated by the traditional method.

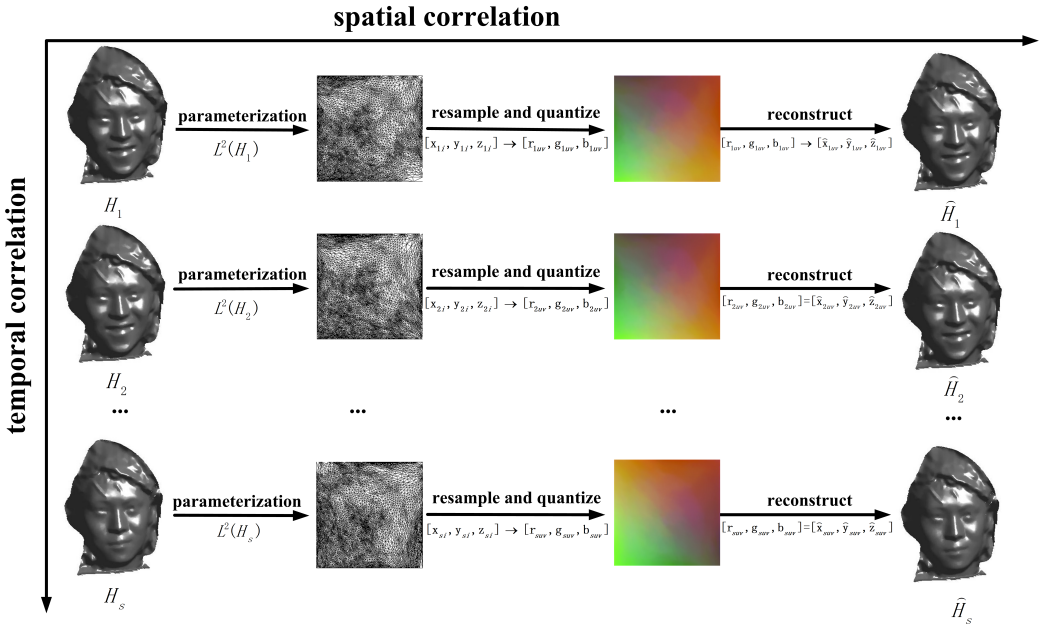

Figure 2. The process of generating geometric image sequences.

Therefore, we proposed to generate time-dependent geometric image sequences for use in the compression task of non-isomorphic 3D-mesh sequences in order to reduce the inter-frame redundancy of the geometric image sequences. The generation method included two schemes of image registration and parametrization-geometry cooperative registration, which used TPS transformation to register geometric image sequences generated by traditional methods [53] and incorporated non-rigid registration into the geometric-image-sequence generation process to generate registered geometric image sequences, which were finally compressed by a mature video-coding algorithm to effectively improve the coding efficiency of non-isomorphic 3D-mesh sequences.

4. Proposed Method

4.1. Image Registration for Geometric Image Sequence Generation

Based on the characteristics of non-isomorphic 3D-mesh sequences, we proposed a geometric-image-sequence generation method based on image registration. The method first generated a geometric image sequence using the traditional mesh parametrization, and then the generated geometric image sequence was non-rigidly registered using the TPS function to generate the registered geometric image sequence.

A 3D-mesh sequence $\{H_1, H_2, \ldots, H_s\}$ was found, and the original geometric image sequence $\{G_1, G_2, \ldots, G_s\}$ was generated by traditional frame-by-frame parametrization and resampling quantization. To facilitate the compression, we divided the geometric image sequences into groups of pictures (GOP), each of which had a TPS-transformed reference I-frame (denoted as G_I) and a current i-frame (denoted as G_i).

Define such a mapping:

$$\Gamma(P) = PA + KW \tag{2}$$

where P is the homogeneous coordinate $(x, y, z, 1)$ of the vertices in 3D space, A and W are the affine transformation matrix of 4×4 and the non-affine transformation matrix of $r \times 4$, respectively, and $k = (k_1(P), \ldots, k_r(P))$ is the kernel function matrix $k_i(P) = \|P - P_i\|, i = 1, 2, \ldots, r$ of the TPS transformation.

Given the reference frame geometry image G_I and the current frame geometry image G_i , we defined the set of homogeneous coordinates of G_I and G_i as L_1 and L_2, respectively:

$$\begin{cases} L_1 = \{P_{uv} | P_{uv} = (x_{uv}, y_{uv}, z_{uv}, 1)\}_{u=1,\ldots,n}^{v=1,\ldots,m} \\ L_2 = \{P_{pq} | P_{pq} = (x_{pq}, y_{pq}, z_{pq}, 1)\}_{p=1,\ldots,n}^{q=1,\ldots,m} \end{cases} \tag{3}$$

We obtained a subset T_1 by uniform sampling in the set L_1 and searched for its closest point set T_2 in the set L_2:

$$\begin{cases} T_1 = \{p_{u_i v_i} | p_{u_i v_i} = (x_{u_i v_i}, y_{u_i v_i}, z_{u_i v_i}, 1), u_i \in \{1, \ldots, n\}, v_i \in \{1, \ldots, m\}, i = 1, \ldots, r\} \\ T_2 = \{p_{p_i q_i} | p_{p_i q_i} = (x_{p_i q_i}, y_{p_i q_i}, z_{p_i q_i}, 1), p_i \in \{1, \ldots, n\}, q_i \in \{1, \ldots, m\}, i = 1, \ldots, r\} \end{cases} \tag{4}$$

We took T_1 and T_2 as the control points of the TPS transformation and brought them into Equation (2), so that the transformation Γ satisfied the interpolation condition at 2 sets of T_1 and T_2 corresponding points. Therefore, we could obtain:

$$p_{p_i q_i} = \Gamma(p_{u_i v_i}) \tag{5}$$

where $i = 1, \ldots, r$. Here, we assumed that Equation (5) held strictly, substituted (2) into (5), and since W was a non-affine transformation matrix, $T_1^T W = 0$:

$$\begin{bmatrix} K & J_1 \\ J_1^T & 0 \end{bmatrix} \begin{bmatrix} W \\ A \end{bmatrix} = \begin{bmatrix} J_2 \\ 0 \end{bmatrix} \tag{6}$$

where, $J_1 = \begin{bmatrix} x_{u_1 v_1} & y_{u_1 v_1} & z_{u_1 v_1} & 1 \\ x_{u_2 v_2} & y_{u_2 v_2} & z_{u_2 v_2} & 1 \\ \cdots & \cdots & \cdots & \cdots \\ x_{u_r v_r} & y_{u_r v_r} & z_{u_r v_r} & 1 \end{bmatrix}$, $J_2 = \begin{bmatrix} x_{p_1 q_1} & y_{p_1 q_1} & z_{p_1 q_1} & 1 \\ x_{p_2 q_2} & y_{p_2 q_2} & z_{p_2 q_2} & 1 \\ \cdots & \cdots & \cdots & \cdots \\ x_{p_r q_r} & y_{p_r q_r} & z_{p_r q_r} & 1 \end{bmatrix}$.

The kernel function K of the TPS transformation was defined by $K = \left\{K_{ij} | K_{ij} = \left\| p_{u_i v_i} - p_{u_j v_j} \right\|\right\}$. Since K was an invertible matrix, A and W could be approximated by: $\begin{bmatrix} W \\ A \end{bmatrix} \approx \begin{bmatrix} K & J_1 \\ J_1^T & 0 \end{bmatrix} \begin{bmatrix} J_2 \\ 0 \end{bmatrix}$. The TPS transformation matrices A and W were thus obtained, and then the TPS transformation was applied to the global $P_{pq} = \Gamma(P_{uv})$, that is to, obtain the set of homogeneous coordinates $L^{'} = \left\{P_{uv}^{'} | P_{uv}^{'} = \left(x_{uv}^{'}, y_{uv}^{'}, z_{uv}^{'}, 1\right)\right\}_{u=1,\ldots,n}^{v=1,\ldots,m}$ after geometric image transformation, where $L^{'}$ is the approximate point set of L_2 . For each pixel point $P_{uv}^{'}$ on $L^{'}$, the pixel point with the minimum Euclidean distance in L_2 was found as the corresponding point $P_{uv}^{'} = P_{p_0 q_0}$ by $P_{p_0 q_0} = arg\ min_{pq} \left\| P_{uv}^{'} - P_{pq} \right\|$, and the search process was accelerated using KD-tree. We referred to this process as closest-point matching. Thus, the set $L_2^{'} = \left\{P_{uv}^{'} | P_{uv}^{'} = \left(x_{uv}^{'}, y_{uv}^{'}, z_{uv}^{'}, 1\right)\right\}_{u=1,\ldots,n}^{v=1,\ldots,m}$, consisting of vertices $P_{uv}^{'}$, was obtained, whose corresponding image $G_i^{'}$ was an approximate result of G_i and, at the same time, maintained a geometric correlation with G_I,

$$\begin{cases} G_I = \{P_{uv} | P_{uv} = (x_{uv}, y_{uv}, z_{uv})\}_{u=1,\ldots,n}^{v=1,\ldots,m} \\ G_i^{'} = \left\{P_{uv}^{'} | P_{uv}^{'} = (x_{uv}^{'}, y_{uv}^{'}, z_{uv}^{'})\right\}_{u=1,\ldots,n}^{v=1,\ldots,m} \end{cases} \tag{7}$$

The framework of the above registration geometric image sequence is shown in Figure 3.

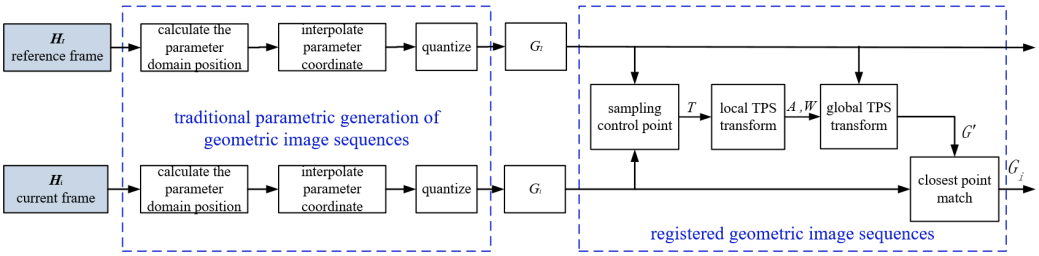

Figure 3. The framework of geometric image sequence registration based on TPS.

4.2. Parametrization-Geometry Cooperative Registration for Geometric Image Sequence Generation

The application of TPS transformation to register the geometric image sequences provided an effective solution for the compression of the non-isomorphic 3D-mesh sequences, but there was still room for improvement in the compression efficiency because the geometric-image-sequence generation and the non-rigid registration process were independent of each other. Therefore, we integrated the idea of geometric image registration based on TPS transformation into the generation process of geometric image sequences and proposed a single-reference frame and a double-reference frame parametrization-geometry cooperative registration method for generating geometric image sequences.

4.2.1. Single-Reference Frame Parametrization

For topologically non-isomorphic 3D-mesh sequences, since the traditional parametrization method only considered the spatial correlation of the mesh sequences (refer to Figure 4a, it could not establish a corresponding relationship between the same feature vertices of the reconstructed mesh sequences, which destroyed the temporal correlations of the original mesh sequences. Therefore, we proposed the integration of the TPS transformation with the geometric-image-sequence generation process (refer to Figure 4b, to fully exploit the inter-frame correlation of the mesh sequence and effectively utilize the temporal correlations of the non-isomorphic 3D-mesh sequence in order to generate a time-dependent geometric image sequence.

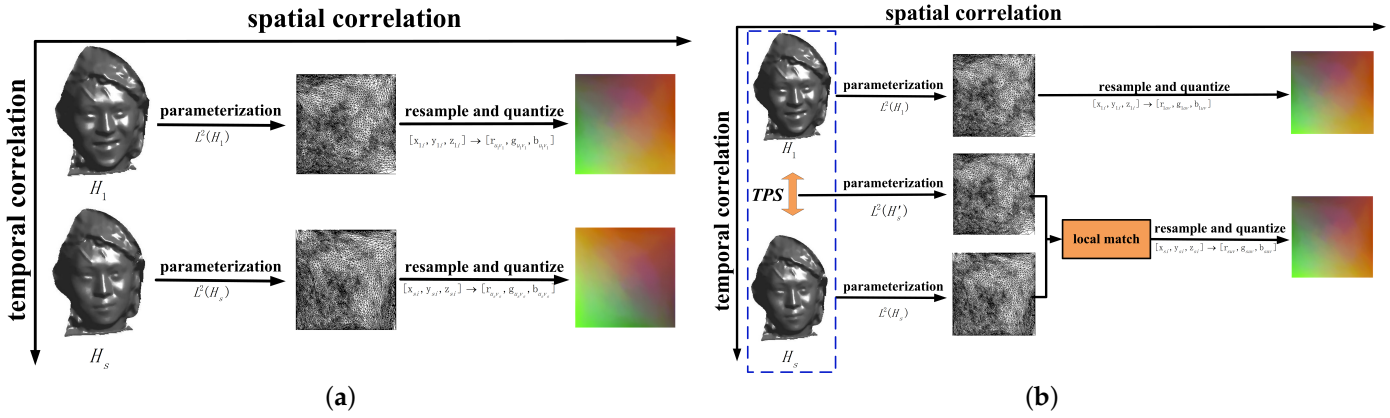

Figure 4. The correlations in the generation procedure of geometry images (**a**) The traditional method (**b**) The proposed method.

Given a non-isomorphic 3D-mesh sequence GOM of length s, $\{H_k = (P_k, F_k)\}_{k=1}^{s}$, where P_k is the set of vertex positions of the kth frame of mesh and F_k is the set of topological triangles of the kth frame of mesh, each GOM included 1 I-frame H_I, $s-1$ P-frames H_P. The I-frame was used as the initial reference frame of the TPS transformation, and the P-frame

was used as the target frame of the TPS transformation. To match the video-coding order, we used the first frame of mesh as the I-frame for the parametric cooperative registration. The order of parametrization is shown in Figure 5, using the I-frame or the previously transformed P-frame as the reference frame.

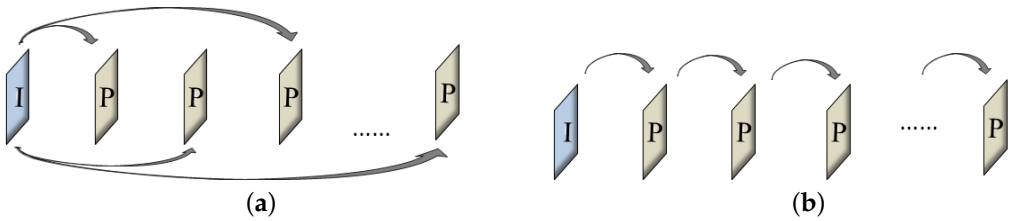

Figure 5. Two structures of single-reference frame parametrization (**a**) Use the I-frame as the reference frame (**b**) Use the I-frame or the previously transformed P-frame as the reference frame.

There was a reference frame of mesh $H_1 = (P_1, F_I)$ and a target frame of mesh $H_2 = (P_2, F_2)$, where H_1 could be either the I-frame of mesh (P_I, F_I) or the previous TPS-transformed mesh $\left(P^{'}, F_I\right)$, where the vertex position sets are $P_1 = \{p_i | p_i = (x_i, y_i, z_i)\}_{i=1,\dots,n_1}$, $P_2 = \{p_j | p_j = (x_i, y_i, z_i)\}_{j=1,\dots,n_2}, n_1 \neq n_2$. First, we selected the control points T_1 uniformly from P_1 and found the corresponding point set T_2 in P_2. We substituted the homogeneous coordinate sets of T_1, T_2 into the TPS transformation $\Gamma(p) = pA + KW$ to obtain the transformation matrixes A, W, and then we applied the TPS transformation to the global P_1, P_2 to obtain the vertex position set $P^{'} = \left\{p_i^{'} | p_i^{'} = \left(x_i^{'}, y_i^{'}, z_i^{'}\right)\right\}_{i=1,\dots,n_1}$ that was globally aligned with P_1.

To map (P_1, F_I), $\left(P^{'}, F_I\right)$, (P_2, F_2) to the parameter domain, their parameter positions needed to be calculated. According to the literature [49], the parameter positions of P_1, P_2 were calculated by Equation (1), $R_1 = \{r_i | r_i = (a_i, b_i)\}_{i=1,\dots,n_1}$ and $R_2 = \left\{r_j^{''} | r_j^{''} = \left(a_j^{''}, b_j^{''}\right)\right\}_{j=1,\dots,n_2}$, where a, b represent the horizontal and vertical axis parameter positions. For $P^{'}$, one could directly apply the parameter position of P_1, so $R^{'} = R_1$, or one could find the parameter coordinates $R^{'} = \left\{r_i^{'} | r_i^{'} = \left(a_i^{'}, b_i^{'}\right)\right\}_{i=1,\dots,n_1}$ of $P^{'}$, according to Equation (1). Experimentally, it was shown that the parameter positions obtained by applying the latter were more suitable for $P^{'}$, which led to the parameter positions $\begin{cases} R_1 = \{r_i | r_i = (a_i, b_i)\}_{i=1,\dots,n_1} \\ R^{'} = \left\{r_i^{'} | r_i^{'} = \left(a_i^{'}, b_i^{'}\right)\right\}_{i=1,\dots,n_1} \end{cases}$ and $R_2 = \left\{r_j^{''} | r_j^{''} = \left(a_j^{''}, b_j^{''}\right)\right\}_{j=1,\dots,n_2}$ for $P_1, P^{'}$ and P_2. Then, P_1 and $P^{'}$ were interpolated into a regular mesh set $\begin{cases} Q_1 = \{q_{uv} | q_{uv} = (x_{uv}, y_{uv}, z_{uv})\}_{u=1,\dots,m_1}^{v=1,\dots,m_2} \\ Q^{'} = \left\{q_{uv}^{'} | q_{uv}^{'} = \left(x_{uv}^{'}, y_{uv}^{'}, z_{uv}^{'}\right)\right\}_{u=1,\dots,m_1}^{v=1,\dots,m_2} \end{cases}$ With $m_1 \times m_2$ resolution and to facilitate subsequent closest point matching, P_2 was up-sampled by a factor of m and interpolated into a mesh set $Q_2 = \left\{q_{st}^{''} | q_{st}^{''} = \left(x_{st}^{''}, y_{st}^{''}, z_{st}^{''}\right)\right\}_{s=1,\dots,(m_1 \times m)}^{t=1,\dots,(m_2 \times m)}$ with $((m_1 \times m) \times (m_2 \times m))$ resolution. For each parameter vertex $q_{uv}^{'}$ on $Q^{'}$ in the parameter domain, the vertex $q_{st}^{''}$ with the minimum Euclidean distance was found as the corresponding vertex in Q_2 by $q_{s_0 t_0}^{''} = arg\ min_{st} \left\| q_{uv}^{'} - q_{st}^{''} \right\|$ so that $q_{uv}^{'} = q_{s_0 t_0}^{''}$, and the search process was accelerated using KD-tree to complete the local closest point matching in the target frame. Thus, the set $Q_2^{'}$, consisting of the vertex $q_{uv}^{'}$, was obtained. Then, Q_1 and $Q_2^{'}$ were quantized into n-bit-deep geometric images $\begin{cases} G_1 = \{G_{uv} | G_{uv} = (r_{uv}, g_{uv}, b_{uv})\}_{u=1,\dots,m_1}^{v=1,\dots,m_2} \\ G_2 = \left\{G_{uv}^{'} | G_{uv}^{'} = \left(r_{uv}^{'}, g_{uv}^{'}, b_{uv}^{'}\right)\right\}_{u=1,\dots,m_1}^{v=1,\dots,m_2} \end{cases}$, which generated time-dependent geometric images G_1, G_2. The above algorithms were sequentially applied to the GOM of the non-isomorphic mesh sequence to obtain the

registered geometric image sequence $\{G_1, G_2, \ldots, G_k\}_{k=1}^{s}$. See Figure 6 for the specific parametrization-geometry cooperative registration framework.

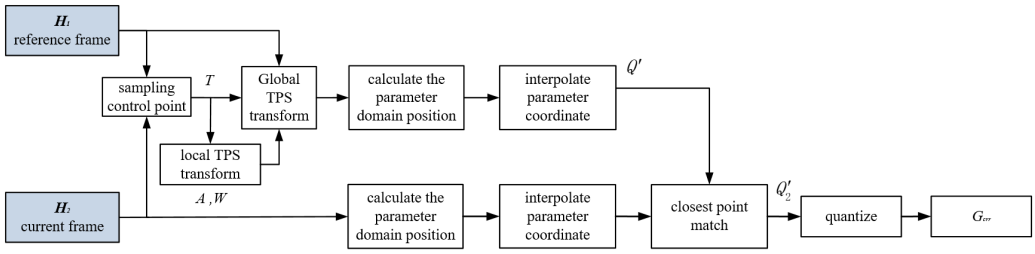

Figure 6. The framework of parametrization-geometry cooperative registration using a single-reference frame based on TPS.

4.2.2. Double-Reference Frame Parametrization

Considering the double-reference frames in the 3D-mesh sequence registration mode could effectively improve the registration accuracy. Inspired by this, we proposed the application of the TPS transformation to the non-isomorphic 3D-mesh sequence for the double-reference frame registration and then to fuse it with the geometric-image-sequence generation process in order to generate a time-dependent geometric image sequence. This scheme made full use of the spatiotemporal correlations of the mesh sequences and effectively maintained the intra-frame and inter-frame correlations of the non-isomorphic 3D-mesh sequences by optimally matching the approximate mesh sets of the double-reference frames.

Given a non-isomorphic 3D-mesh sequence GOM of length s, $\{H_k = (P_k, F_k)\}_{k=1,\ldots,s}$, where P_k is the set of vertex positions of the kth frame and F_k is the set of topological triangles of the kth frame, each GOM consisted of one I-frame H_I, multiple P-frames H_P and B-frames H_B. The I-frame was used as the initial reference frame for TPS transformation, and the P-frame and B-frame were used as the target frames. As shown in Figure 7, the double-reference frame parametrization sequence consisted of 3 structures: structure 1 was a P-frame using an I-frame or the previous parameterized P-frame as the reference frame, and a B-frame using an I-frame and the previous parameterized P-frame or 2 adjacent P-frames together as the reference frame; structure 2 was a P-frame using an I-frame or the previous parameterized P-frame as the reference frame, and a B-frame using an I-frame and the previous parameterized B-frame or the previous parameterized P-frame and B-frame together as the reference frame; structure 3 was a P-frame using an I-frame and the previous parameterized P-frame together as the reference frame.

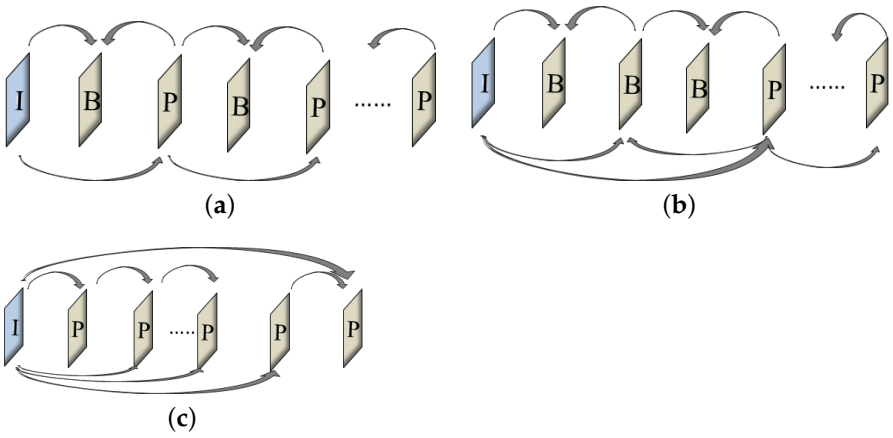

Figure 7. The structures of double-reference frames parametrization (**a**) Structure 1 (**b**) Structure 2 (**c**) Structure 3.

Assuming two reference frames of meshes $H_{ref_1} = (P_1, F_I)$, $H_{ref_2} = (P_2, F_I)$ and a target frame of mesh $H_{crr} = (P_{crr}, F_{crr})$, H_{ref_1} and H_{ref_2} could be the initial reference I-frames of meshes (P_I, F_I), or the P-frame or B-frame of meshes $\left(P^{'}, F_I\right)$, transformed by TPS, yielding sets of mesh vertex positions $P_1 = \{p_{1i}|p_{1i} = (x_{1i}, y_{1i}, z_{1i})\}_{i=1,\dots,n_1}$, $P_2 = \{p_{2i}|p_{2i} = (x_{2i}, y_{2i}, z_{2i})\}_{i=1,\dots,n_1}$, $P_{crr} = \left\{p_j|p_j = \left(x_j, y_j, z_j\right)\right\}_{j=1,\dots,n_2}$, where $n_1 \neq n_2$. Firstly, the control points T_1, T_2 were evenly selected from the reference frames P_1, P_2, and the corresponding point sets T_{crr_1}, T_{crr_2} were found in P_{crr}. The homogeneous coordinate sets of these two sets of control points were substituted into the TPS transformation $\Gamma(p) = pA + KW$, and the transformation matrixes $(A_1, W_1), (A_2, W_2)$ were obtained. Then, the TPS transformation was applied globally to obtain the vertex position set $P_1^{'} = \left\{p_{1i}^{'}|p_{1i}^{'} = \left(x_{1i}^{'}, y_{1i}^{'}, z_{1i}^{'}\right)\right\}_{i=1,\dots,n_1}$, which was globally aligned with P_1, and $P_2^{'} = \left\{p_{2i}^{'}|p_{2i}^{'} = \left(x_{2i}^{'}, y_{2i}^{'}, z_{2i}^{'}\right)\right\}_{i=1,\dots,n_1}$, which was aligned with P_2.

Next, we performed a double-reference frame-optimization-matching on the two corresponding vertex position sets of the current frame and set the balance coefficients γ; then, we used a linear combination:

$$P_{crr}^{'\gamma} = \gamma P_1^{'} + (1-\gamma)P_2^{'} \tag{8}$$

where $\gamma = 0 : 10^{-n} : 1$, n is the number of combination levels. By mapping $\left(P_{crr}^{'\gamma}, F_1\right)$, (P_{crr}, F_2) to the parameter domain through Equation (1), we obtained their parameter positions, $R_{crr}^{'\gamma} = \left\{r_i^{'}|r_i^{'} = \left(a^{'}, b^{'}\right)\right\}_{i=1,\dots,n_1}$ and $R_{crr} = \left\{r_j|r_j = \left(a_j, b_j\right)\right\}_{j=1,\dots,n_2}$, and then interpolated $P_{crr}^{'\gamma}$ into the parameter mesh coordinates of $m_1 \times m_2$ resolution $Q_{crr}^{'\gamma} = \left\{q_{uv}^{'}|q_{uv}^{'} = \left(x_{uv}^{'}, y_{uv}^{'}, z_{uv}^{'}\right)\right\}_{u=1,\dots,m_1}^{v=1,\dots,m_2}$, and P_{crr} was up-sampled by a factor of m and interpolated into a parameter domain mesh coordinate set of $((m_1 \cdot m) \times (m_2 \cdot m))$ resolution $Q_{crr} = \{q_{st}|q_{st} = (x_{st}, y_{st}, z_{st})\}_{s=1,\dots,(m_1 \cdot m)}^{t=1,\dots,(m_2 \cdot m)}$, for each parameter vertex $q_{uv}^{'}$ on $Q_{crr}^{'\gamma} = \left\{q_{uv}^{'}|q_{uv}^{'} = \left(x_{uv}^{'}, y_{uv}^{'}, z_{uv}^{'}\right)\right\}_{u=1,\dots,m_1}^{v=1,\dots,m_2}$. In the parameter domain, we found the vertex q_{st} of the minimum Euclidean distance in Q_{crr} by $q_{s_0t_0} = arg\ min_{st}\left\|q_{uv}^{'} - q_{st}\right\|$ as the corresponding vertex $q_{uv}^{'} = q_{s_0t_0}$, and then we obtained the set $\overset{\frown}{Q}_{crr}^{'\gamma}$, consisting of the new vertex coordinate values $q_{uv}^{'}$. Then, by calculating the Hausdorff distance d^{γ} between $\overset{\frown}{Q}_{crr}^{'\gamma}$ and the original mesh parameter domain coordinate Q_{crr} under different γ, γ at the minimum distance d_{min}^{γ} was the optimal equilibrium coefficient γ_{opt}. The optimal set of vertex coordinates $P_{opt}^{'}$ was obtained by the following equation:

$$P_{opt}^{'\gamma} = \gamma_{opt} P_1^{'} + (1-\gamma_{opt})P_2^{'} \tag{9}$$

Furthermore, the optimal parameter domain coordinate $\overset{\frown}{Q}_{crr}^{'opt}$ could be obtained as well. Then, we quantized $\overset{\frown}{Q}_{crr}^{'opt}$ into an n-bit-deep geometric image $G_{crr} = \left\{G_{uv}^{'}|G_{uv}^{'} = \left(r_{uv}^{'}, g_{uv}^{'}, b_{uv}^{'}\right)\right\}_{u=1,\dots,m_1}^{v=1,\dots,m_2}$. We applied the above algorithms to the entire non-isomorphic mesh sequences GOM to obtain time-dependent geometric image sequences $\{G_1, G_2, \dots, G_k\}_{k=1,\dots,s}$. See Figure 8 for a specific parametrization-geometry cooperative registration framework.

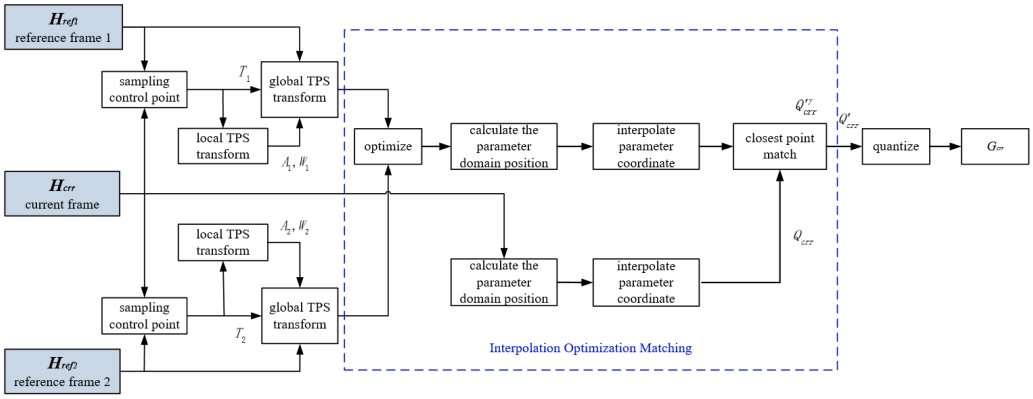

Figure 8. The framework of parametrization-geometry cooperative registration using double-reference frames based on TPS.

5. Experiments and Evaluation

5.1. General Idea of the Experiment

We verified the compression performance of the non-isomorphic 3D-mesh sequence based on the registration-based geometric image sequence through experiments, and the experimental results and related analysis will also be provided. We selected four face-expression sequences (*Happy_M003*, *Fear_M024*, *Surprise_M028*, *Happy_F001*) as experimental samples from the BU-4DFE face database [60], and we set the mesh sequence GOM length to be odd in order to facilitate the double-reference frame parametrization. In the experiment, we randomly selected nine frames in the mesh sequence as a GOM. In the experiment, the traditional parametrization, image registration, and the parametrization-geometry cooperative registration in this study were applied to the experimental sequence. The resolution of the generated geometric image was set to 256 × 256, and the up-sampling factor on the parametric mesh was 4. To the best of our knowledge, the current mature video compression technology HEVC inter-frame compression supported up to 12-bits, and H.264 inter-frame compression supported up to 14-bits, so the experiments quantized the parameter coordinates to 14-bits. After generating the geometric image sequence, it was compressed using the H.264 video-coding standard.

5.2. Time-Dependent Geometric Image Sequence Performance

To illustrate the significance of generating time-dependent geometric image sequences in this study, *Happy_M003* was used as an example to compare the results and performance of the different parametrization. In Figure 9, row 1 is the original non-isomorphic 3D-mesh sequence, row 2 is the parametric position of the traditional parametrization, row 3 is the detail view of the enlarged parametric positions of the traditional parametrization, row 4 is the geometric image sequence generated by the traditional parametrization, row 5 is the geometric image residual image sequence generated by the traditional parametrization, row 6 is a sequence of geometric images generated by image registration, row 7 is a sequence of geometric image residual images generated by image registration, row 8 is a parameterized position for parametrization-geometry cooperative registration, row 9 is the detail view of the enlarged parameterized position for parametrization-geometry cooperative registration, and row 10 is a sequence of geometric images generated by parametrization-geometry cooperative registration.

First, the mesh sequence maps each frame of the 3D-mesh to a 2D plane, frame by frame, through traditional parametrization. The position coordinates of the parameter field of the model *Happy_M003* are shown in the second row of Figure 9. Since the sequence was modeled by real objects scanned by professional equipment, the vertex connectivity of each frame of mesh was different, and it could be clearly seen that the vertices were not aligned, and the details are shown in the third row of Figure 9. Then, the 3 coordinate components (x, y, z) were resampled and interpolated into the parameter coordinates of 256 × 256 and

quantized into 14-bit-deep color geometric image components (r, g, b), thus generating the traditional geometric image sequence (see row 4 of Figure 9). It could be seen that in the geometric image sequence generated by traditional parametrization, although the specifications of each frame were consistent, the 3D-vertex coordinates represented by the pixel values of the corresponding pixels were not aligned.

The next step was to register the geometric image sequences generated by the traditional parametrization method. We selected 200 pixels in the geometric image of the reference frame as the local control points of the TPS transformation, generated the global transformation matrix A, W through the TPS transformation, and then acted on the reference frame image to generate the transformed image of the current frame. Finally, the closest-point local fine-matching was performed in the original image of the current frame to find the corresponding point, and the geometric image sequence that was time-correlated with the reference frame image was generated, and then transformed and registered, frame by frame, to generate the registered geometric image sequence (see row 6 of Figure 9).

To verify the correlation between the adjacent frames of the geometric image sequence after the image registration, the residual images of the adjacent frames were generated in this study. The residual images between the adjacent frames of the original geometric image sequence are shown in row 5 of Figure 9, and the residual images between the adjacent frames of the registered geometric image sequence are shown in row 7 of Figure 9, from which it could be clearly observed that the residual images between the adjacent frames of the registered geometric image sequence were smaller than the residual images between the adjacent frames of the original geometric image sequence. The root-mean square (rms) between the adjacent frames of the geometric image sequence before and after registration was also calculated, given the two geometric images $G = \{Q_{uv} = (x_{uv}, y_{uv}, z_{uv})\}_{u=1,\dots,n}^{v=1,\dots,m}$, $G' = \left\{Q'_{uv} = \left(x'_{uv}, y'_{uv}, z'_{uv}\right)\right\}_{u=1,\dots,n}^{v=1,\dots,m}$. The rms and max pixel-point distance between the adjacent frames were given by Equations (10) and (11):

$$d_{rms} = \frac{1}{nm}\left(\sum_{u=1}^{n}\sum_{v=1}^{m}\left\|Q_{uv} - Q'_{uv}\right\|^2\right)^{\frac{1}{2}} \tag{10}$$

$$d_{max} = \max_{1\le u\le n}\max_{1\le v\le m}\left\|Q_{uv} - Q'_{uv}\right\| \tag{11}$$

The results of calculating the rms between the adjacent frames of the original geometric image sequence of the model *Happy_M003* and the rms between the adjacent frames of the registered geometric image sequence and the max pixel-point distance are shown in Table 1. It could be clearly seen that both the rms and the max pixel-point distance of the adjacent frames of the geometric image sequence after registration were smaller than the original geometric image sequence. By comparing the above residual image and rms, we could clearly see that the correlation between adjacent frames of the registered geometric image sequence was better than that of the original geometric image sequence, which indicated that the registered geometric image sequence could improve the compression performance of the system.

Table 1. The rms and max pixel-wise distance between the adjacent frames of *Happy_M003*.

Datatype	1–2	2–3	3–4	4–5	5–6	6–7	7–8	8–9
rms of the original geometric image sequence	3223	723	581	1522	1856	19,182	1834	2843
rms of the registered geometric image sequence	247	194	129	210	361	340	269	250
max of the original geometric image sequence	11,950	5772	5406	9075	6495	38,980	17,519	15,461
max of the registered geometric image sequence	1843	2077	1200	1777	1657	1771	3334	1817

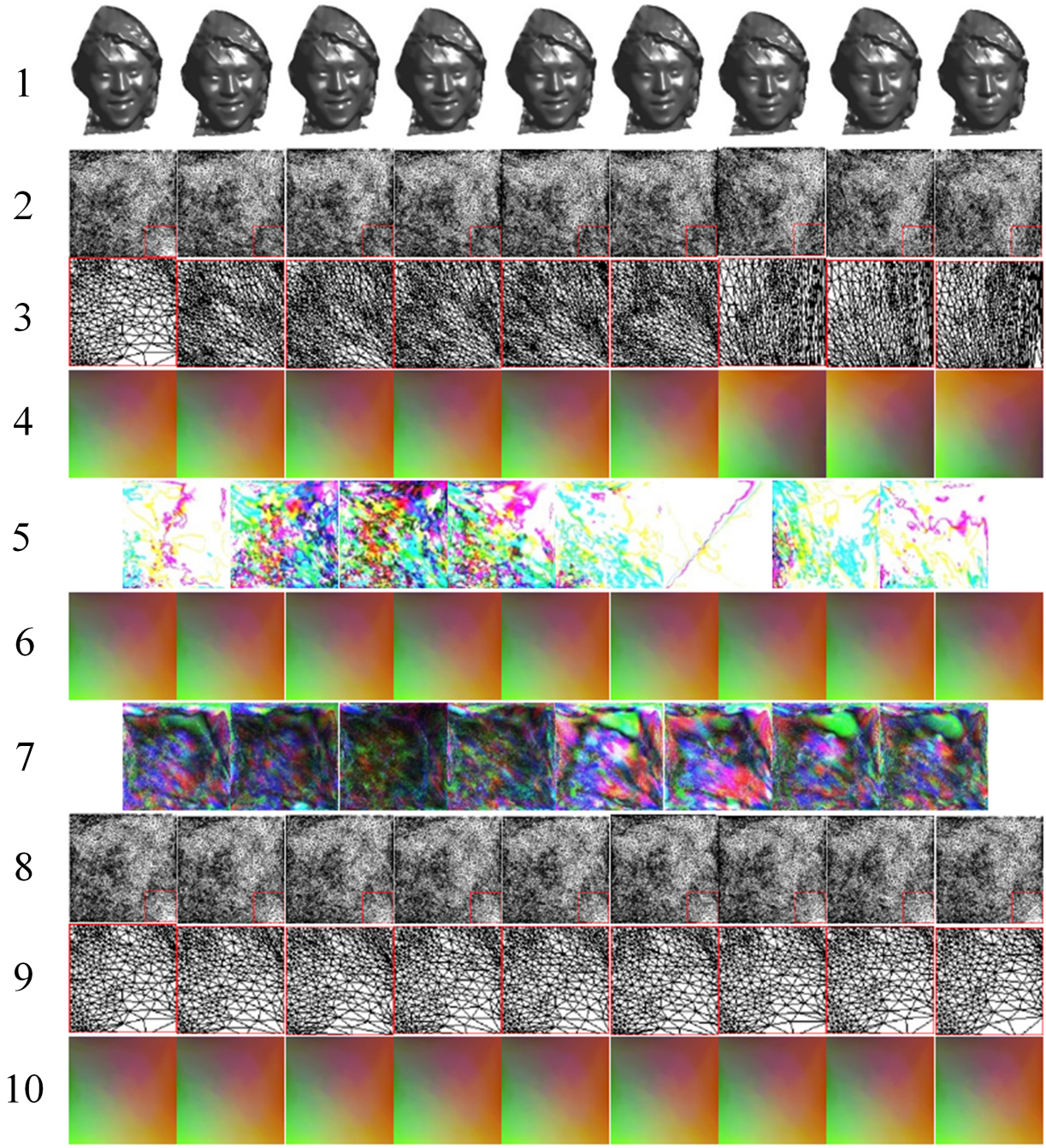

Figure 9. The geometric image sequences generated by different methods (*Happy_M003*). The number on the left of the figure shows the number of rows.

A single-reference frame parametrization was used as an example to compare the geometric image sequence generated by the traditional parametrization when applied to the model *Happy_M003*. The 200 vertices in the first frame of the mesh were selected as the local control points of the TPS transformation, and their corresponding points in the current frame were found, and the TPS transformation was performed to generate the global transformation matrixes A, W. We applied the transformation matrixes globally to generate a set of transformation vertex coordinates for the current frame, mapped them to the 2D-parameter domain, generated the position coordinates of the parameter domain (see row 8 of Figure 9). The details are displayed in row 9. Subsequently, these results were resampled and interpolated to 256×256 parameter coordinates, and the corresponding points were found in the current frame, 1024×1024 parametric coordinates, using

closest-point local-fine-matching, and quantized into color geometric image components (r, g, b) 14-bit-deep, and then we obtained the geometric image sequence generated by parametrization-geometry cooperative registration (see row 10 of Figure 9). By comparing the mesh parameter coordinates of the traditional parametrization (see row 2 of Figure 9) and its detailed images (see row 3 of Figure 9), it was obvious that the inter-frame correlations of the geometric image sequence generated by the parametrization-geometry cooperative registration based on the TPS transformation in this study were significantly better than those of the traditional parametrization method.

In the double-reference-frame parametrization mode, the first parametrization used the first GOM frame as the initial reference I-frame and then selected 200 vertices in the transform vertex parameter coordinate sets of the 2 reference frames as the local control points of the TPS transformation according to different parametrization structures, generated 2 parameter domain position coordinates according to the single-reference frame parametrization mode, and then performed double-reference-frame optimization matching to generate the optimal transform vertex parameter coordinate set, resampled and interpolated into 256×256 parameter coordinates. Subsequently, the closest point local fine matching was used to find the corresponding point in the current frame 1024×1024 parametric coordinates and quantized it into 14-bit-deep color geometric image components (r, g, b), thus generating a time-dependent geometric image sequence by double-reference frame parametrization-geometry cooperative registration.

5.3. Compression Performance

Subsequently, four face expression sequences were used as experimental samples. The geometric image sequences generated by traditional parametrization, the geometric image sequences generated by image registration, and the geometric image sequences generated by parametrization-geometry cooperative single-reference and double-reference frame registration were synthesized into videos with a frame rate of 9 Hz, and the motion-search-window radius of the traditional parametric geometric image sequence was halved and applied to the registration-generated geometric image sequence. The reference software JM19.0 for the H.264 video-coding standard was used for compression, and finally, the decompressed video was decomposed into geometric image sequences and then reconstructed into 3D-mesh sequences. The objective quality of the reconstructed mesh sequences was measured by the PSNR value with the original mesh sequences. Usually, after compression, the 3D-mesh sequence would differ to some extent from the original 3D-mesh sequence. We used PSNR to measure the quality of the 3D-mesh sequence, reconstructed after compression, thereby measuring the compression quality. The higher the PSNR value, the closer the compressed 3D-mesh sequence was to the original mesh sequence, and the smaller the distortion, the higher the compression quality. The code rate indicated the size state of the 3D-mesh sequence after compression, i.e., how many bits, and the smaller the code rate was, the smaller the 3D-mesh sequence was after compression. Table 2 and Figure 10 show the results of PSNR comparison before and after compression of non-isomorphic 3D-mesh sequences at different code rates (bits per vertex, in bpv).

Table 2. The performance data of compression for different models.

Model	Traditional Parametrization [53]		Registered Geometric Image Sequences		Single-Reference Frame Parametrization		Double-Reference Frame Parametrization	
	PSNR [1]	Code Rate	PSNR	Code Rate	PSNR	Code Rate	PSNR	Code Rate
Happy_M003	55.19	8.02	55.63	10.3	57.23	7.92	58.91	7.97
	55.13	5.09	55.58	6.96	57.21	4.51	58.81	4.53
	55.06	3.07	55.41	3.57	57.10	2.35	58.59	2.38
	54.98	1.76	55.03	1.32	56.78	1.24	58.13	1.24
	54.77	0.96	54.57	0.49	56.31	0.67	57.20	0.66
Average Value	55.03	3.78	55.24	4.53	56.93	**3.34** [2]	**58.33**	3.36

Table 2. *Cont.*

Model	Traditional Parametrization [53]		Registered Geometric Image Sequences		Single-Reference Frame Parametrization		Double-Reference Frame Parametrization	
	PSNR [1]	Code Rate	PSNR	Code Rate	PSNR	Code Rate	PSNR	Code Rate
Fear_M024	55.05	7.54	55.34	10.22	55.96	7.23	56.56	7.16
	55.04	4.81	55.31	6.62	55.98	4.25	56.51	4.22
	54.86	2.85	55.20	3.44	55.87	2.30	56.38	2.27
	54.69	1.62	54.8	1.30	55.63	1.22	56.00	1.20
	54.19	0.88	54.31	0.53	55.08	0.64	55.54	0.63
Average Value	54.77	3.54	54.99	4.42	55.70	3.13	**56.20**	**3.10**
Surprise_M028	54.06	8.10	54.56	10.76	54.23	8.26	55.06	8.21
	54.12	5.17	54.53	7.42	54.29	4.98	55.05	4.85
	54.06	3.09	54.40	3.95	54.22	2.67	54.96	2.58
	53.96	1.76	54.14	1.51	54.18	1.41	54.81	1.35
	53.47	0.95	53.44	0.54	53.70	0.75	54.06	0.71
Average Value	53.93	3.81	54.21	4.84	54.13	3.61	**54.79**	**3.54**
Happy_F001	58.26	9.30	58.38	11.02	58.38	9.48	58.42	8.93
	58.25	5.95	58.39	7.86	58.40	5.75	58.45	5.29
	58.16	3.56	58.13	4.35	58.28	3.03	58.30	2.71
	58.05	2.03	57.63	1.61	58.00	1.57	57.98	1.39
	57.49	1.10	56.81	0.58	57.36	0.80	57.27	0.71
Average Value	58.04	4.39	57.87	5.08	58.08	4.13	**58.09**	**3.81**

[1] PSNR: peak signal-to-noise ratio. [2] Values in bold refer to the smallest code rate and the largest PSNR in the average value.

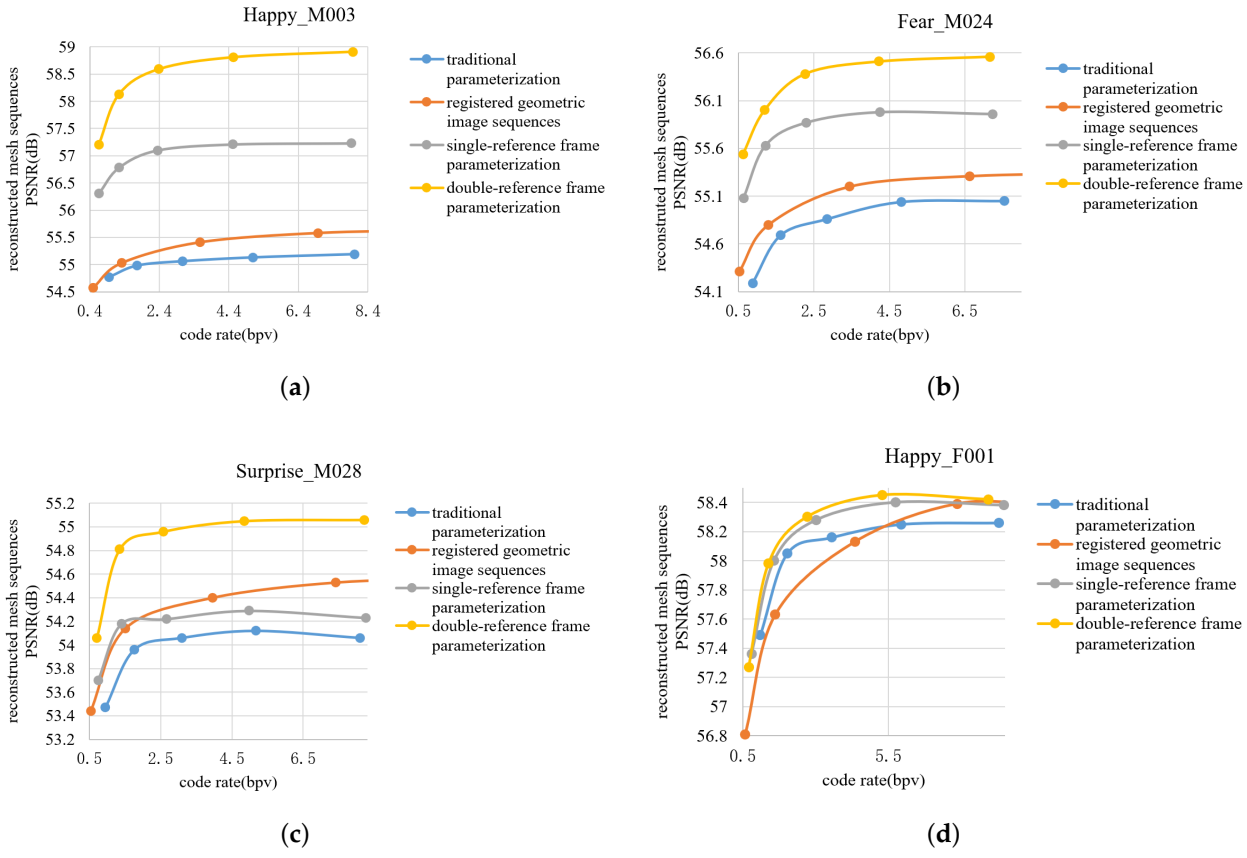

Figure 10. PSNR curves of the reconstructed mesh sequences with different methods. The traditional parameterization is based on [53]. (**a**) Experimental sample *Happy_M003* (**b**) Experimental sample *Fear_M024* (**c**) Experimental sample *Surprise_M028* (**d**) Experimental sample *Happy_F001*.

It could be seen that at different code rates, in the most of the test sequences, the geometric image sequences generated by registration had better compression performance than those generated by the traditional frame-by-frame parametric method [53], where the overall compression performance of the parametrization-geometry cooperative double-reference frame registration method was better than the other methods.

Finally, the subjective quality of the reconstructed meshes of the 2 experimental sequences is shown in Figures 11 and 12. In Figure 11, the first column is the original mesh, the second column is the compressed geometric image sequence reconstruction mesh generated by traditional parametrization [53] (The total code rate of the sequence is 8.02 bpv), and the third column is the compressed geometric image sequence reconstruction mesh generated by parametrization-geometry cooperative single-reference frame registration (the total code rate of the sequence was 7.92 bpv). The first column in Figure 12 shows the original mesh, the second column is the compressed geometric image sequence reconstruction mesh generated by traditional parametrization [53] (The total code rate of the sequence is 7.54 bpv), and the third column is the compressed geometric image sequence reconstruction mesh generated by parametrization-geometry cooperative single-reference frame registration (the total code rate of the sequence was 7.23 bpv). It could be clearly seen that in the case of comparable or slightly better subjective quality, the total code rate of compressing the reconstructed 3D-mesh sequences of the geometric image sequence generated by parametrization-geometry cooperative registration was smaller than the total code rate of the geometric image sequences generated by the traditional method. It showed that the size of the compressed geometric image sequence generated by the parametrization-geometry cooperative registration was smaller than that of the geometric image sequence generated by the traditional method.

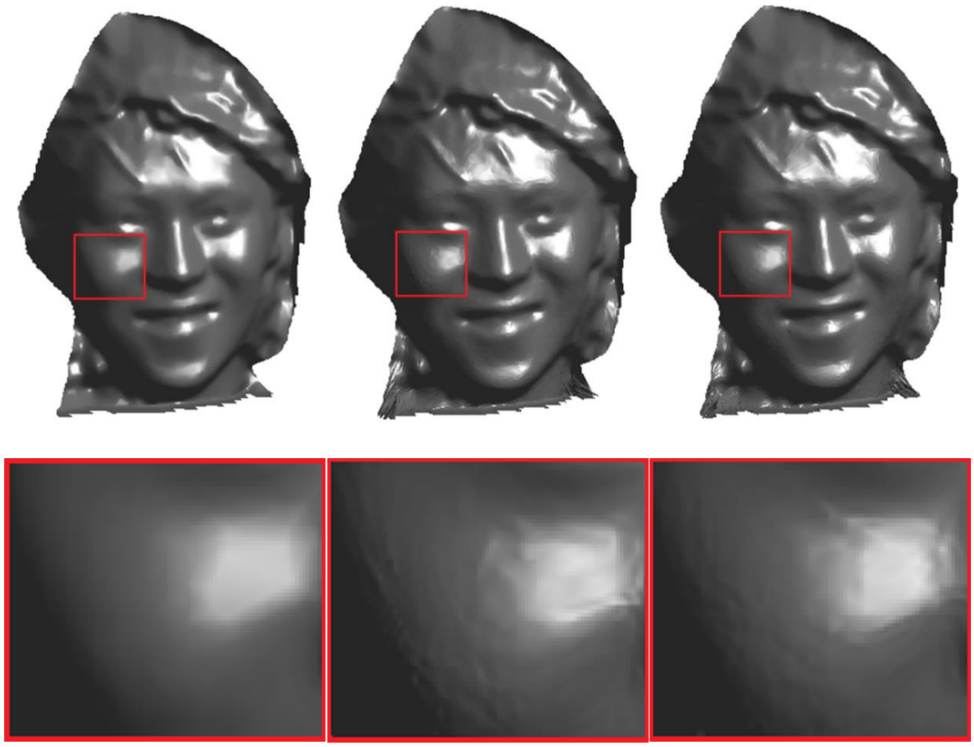

Figure 11. Subjective quality comparison of the fourth frame of *Happy_M003*.

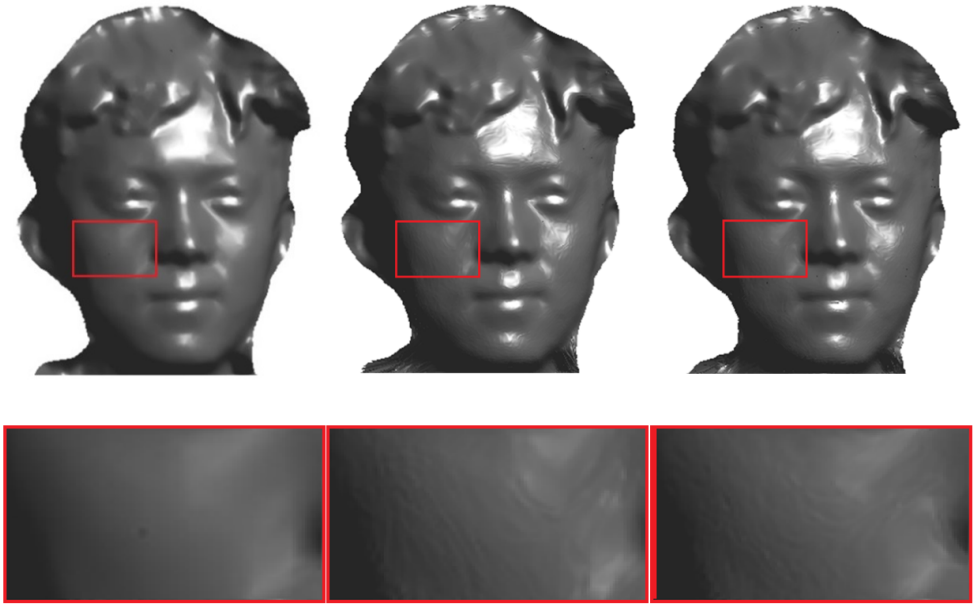

Figure 12. Subjective quality comparison of the second frame of *Fear_M024*.

Overall, the compression performance of the geometric image sequences generated by image registration and the geometric image sequences generated by parametrization-geometry cooperative registration was better than that of the geometric image sequences generated by traditional parametrization, among which the geometric image sequences generated by parametrization-geometry cooperative double-reference frame registration had the best compression performance. Within the context of the explosive growth trend of digital geometric media, it brings a positive impact on the storage and transmission of 3D meshes, an important data type of digital geometric media, and contributes to the development of non-isomorphic 3D-mesh data compression.

6. Conclusions

We focused on the problem of the compression of non-isomorphic 3D-mesh sequences based on geometric image representation and proposed two registered geometric-image-sequence generation methods for the task of compression of non-isomorphic 3D-mesh sequences, including non-rigid registration of geometric image sequences generated by traditional parametrization using TPS function and parametrization-geometry cooperative single-reference or double-reference frame registration by applying TPS transformation. The geometric correlations of topologically non-isomorphic 3D-mesh sequences were effectively maintained. Finally, mature video compression algorithms were applied to compress the geometric image sequences generated by traditional parametrization and those generated by the proposed methods in this paper. The compression experimental results showed that the subjective and objective qualities of the decoding and the reconstruction of the proposed methods in this study were better than those of the traditional methods, which facilitated the storage and transmission of non-isomorphic 3D-mesh data.

Author Contributions: Conceptualization, Y.G., Z.W. and J.W.; methodology, Y.G., Z.W. and J.W.; software, Y.G.; validation, Z.W. and J.W.; formal analysis, Y.G.; writing—original draft preparation, Y.G.; writing—review and editing, Z.W. and J.W.; funding acquisition, Z.W. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Fundamental Research Funds for the Central Universities (Grant number: 328202203, 20230045Z0114, 328202267), China Postdoctoral Science Foundation funded project (Grant number: 2019M650606), and First-class Discipline Construction Project of Beijing Electronic Science and Technology Institute (Grant number: 3201012).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, S.; Li, X.; Fu, Q. Laplace-Based 3D Human Mesh Sequence Compression. *Int. J. Image Graph.* **2022**, 2450027. [CrossRef]
2. Luo, G.; Zhao, X.; Chen, Q.; Zhu, Z.; Xian, C. Dynamic data reshaping for 3D-mesh animation compression. *Multimed. Tools Appl.* **2022**, *81*, 55–72. [CrossRef]
3. Hu, Y.L.; Yin, B.C.; Cheng, S.Q.; Gu, C.L. An improved morphable model for 3D face synthesis. In Proceedings of the 2004 International Conference on Machine Learning and Cybernetics, Shanghai, China, 26–29 August 2004; pp. 4362–4367.
4. Cyberware Rapid 3D Scanners. Available online: http://www.cyberware.com/ (accessed on 15 July 2023).
5. Tejera, M.; Hilton, A. Compression techniques for 3D video mesh sequences. In Proceedings of the Articulated Motion and Deformable Objects: 7th International Conference, Port d'Andratx, Mallorca, Spain, 11–13 July 2012; pp. 12–25.
6. Overview of 3D Object Representations. Available online: https://www.cs.princeton.edu/courses/archive/fall03/cs597D/lectures/reps.pdf (accessed on 15 July 2023).
7. Deering, M. Geometry compression. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 15 September 1995; pp. 13–20.
8. Graziosi, D.B. Video-based dynamic mesh coding. In Proceedings of the 2021 IEEE International Conference on Image, Anchorage, AK, USA, 19–22 September 2021; pp. 3133–3137.
9. Choi, Y.; Jeong, J.B.; Lee, S.; Ryu, E.S. Overview of the Video-based Dynamic Mesh Coding (V-DMC) Standard Work. In Proceedings of the 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Nanjing, China, 6–8 May 2022; pp. 578–581.
10. Alface, P.R.; Martemianov, A.; Ilola, L.; Kondrad, L.; Bachhuber, C.; Schwarz, S. V3C-based Coding of Dynamic Meshes. In Proceedings of the 2022 10th European Workshop on Visual Information Processing (EUVIP), Lisbon, Portugal, 11–14 September 2022; pp. 1–6.
11. Huang, C.; Zhang, X.; Tian, J.; Xu, X.; Liu, S. Boundary-Preserved Geometry Video for Dynamic Mesh Coding. In Proceedings of the 2022 Picture Coding Symposium (PCS), San Jose, CA, USA, 7–9 December 2022; pp. 133–137.
12. Wien, M.; Jung, J.; Baroncini, V. Formal Visual Evaluation and Study of Objective Metrics for MPEG Dynamic Mesh Coding. In Proceedings of the 2022 10th European Workshop on Visual Information Processing (EUVIP), Lisbon, Portugal, 11–14 September 2022; pp. 1–6.
13. Ahmmed, A.; Paul, M.; Murshed, M.; Pickering, M. Dynamic mesh commonality modeling using the cuboidal partitioning. In Proceedings of the 2022 IEEE International Conference on Visual Communications and Image Processing (VCIP), Suzhou, China, 13–16 December 2022; pp. 1–5.
14. Graziosi, D.; Nakagami, O.; Kuma, S.; Zaghetto, A.; Suzuki, T.; Tabatabai, A. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC). *APSIPA Trans. Signal Inf. Process.* **2020**, *9*, e13. [CrossRef]
15. Marvie, J.E.; Krivokuća, M.; Graziosi, D. Coding of dynamic 3D meshes. In *Immersive Video Technologies*; Academic Press: Cambridge, MA, USA, 2023; pp. 387–423.
16. Strümpler, Y.; Postels, J.; Yang, R.; Gool, L.V.; Tombari, F. Implicit neural representations for image compression. In Proceedings of the European Conference on Computer Vision, Tel-Aviv, Israel, 23–27 October 2022; pp. 74–91.
17. Zhang, K.; Liang, Y.; Zhang, J.; Wang, Z.; Li, X. No one can escape: A general approach to detect tam-pered and generated image. *IEEE Access* **2019**, *7*, 129494–129503. [CrossRef]
18. Chow, M.M. Optimized Geometry Compression for Real-Time Rendering. In Proceedings of the IEEE Visualization 97 Conference, Phoenix, AZ, USA, 19–24 October 1997; pp. 347–354.
19. Bajaj, C.L.; Pascucci, V.; Zhuang, G. Single resolution compression of arbitrary triangular meshes with properties. *Comput. Geom.* **1999**, *14*, 167–186. [CrossRef]
20. Taubin, G.; Rossignac, J. Geometric compression through topological surgery. *ACM Trans. Graph. (TOG)* **1998**, *17*, 84–115. [CrossRef]
21. Diaz-Gutierrez, P.; Gopi, M.; Pajarola, R. Hierarchyless simplification, stripification and compression of triangulated two-manifolds. *Comput. Graph. Forum* **2005**, *24*, 457–467. [CrossRef]
22. Li, J.; Kuo, C.C. A dual graph approach to 3D triangular mesh compression. In Proceedings of the 1998 International Conference on Image, Chicago, IL, USA, 4–7 October 1998; pp. 891–894.
23. Gumhold, S.; Straßer, W. Real time compression of triangle mesh connectivity. In Proceedings of the 25th Annual Conference on Computer Graphics and Iinteractive Techniques, New York, NY, USA, 24 July 1998; pp. 133–140.
24. Rossignac, J. Edgebreaker: Connectivity compression for triangle meshes. *IEEE Trans. Vis. Comput. Graph.* **1999**, *5*, 47–61. [CrossRef]
25. King, D.; Rossignac, J. Guaranteed 3.67 vs. bit encoding of planar triangle graphs. In Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG'99), Vancouver, BC, Canada, 15–18 August 1999; pp. 146–149.

26. Gumhold, S. New Bounds on the Encoding of Planar Triangulations; Technical Report WSI-2000-1. 2000. Available online: https://ub01.uni-tuebingen.de/xmlui/handle/10900/48082 (accessed on 26 July 2023).
27. Isenburg, M.; Snoeyink, J. Spirale Reversi: Reverse decoding of the Edgebreaker encoding. *Comput. Geom.* **2001**, *20*, 39–52. [CrossRef]
28. Rossignac, J.; Szymczak, A. Wrap&Zip decompression of the connectivity of triangle meshes compressed with Edgebreaker. *Comput. Geom.* **1999**, *14*, 119–135.
29. Szymczak, A.; King, D.; Rossignac, J. An Edgebreaker-based efficient compression scheme for regular meshes. *Comput. Geom.* **2001**, *20*, 53–68. [CrossRef]
30. Szymczak, A. Optimized edgebreaker encoding for large and regular triangle meshes. *Vis. Comput.* **2003**, *19*, 271–278. [CrossRef]
31. Han, S.R.; Yamasaki, T.; Aizawa, K. Time-varying mesh compression using an extended block matching algorithm. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 1506–1518. [CrossRef]
32. 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 Proceedings of the 2008 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; pp. 1045–1048.
33. Yamasaki, T.; Aizawa, K. Patch-based compression for time-varying meshes. In Proceedings of the 2010 IEEE International Conference on Image, Hong Kong, China, 12–15 September 2010; pp. 3433–3436.
34. Doumanoglou, A.; Alexiadis, D.S.; Zarpalas, D.; Daras, P. Toward real-time and efficient compression of human time-varying meshes. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 2099–2116. [CrossRef]
35. Doumanoglou, A.; Alexiadis, D.; Asteriadis, S.; Zarpalas, D.; Daras, P. On human time-varying mesh compression exploiting activity-related characteristics. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 6147–6151.
36. Wang, C.Z.; Yin, B.C.; Bai, X.M.; Sun, Y. Improved genetic algorithm based model matching method for 3D face synthesis. In Proceedings of the International Symposium on Computing and Its Application in Information Science, Hefei, China, 15–19 August 2005; pp. 381–384.
37. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. *Sens. Fusion IV Control Paradig. Data Struct.* **1992**, *1611*, 586–606. [CrossRef]
38. Gupta, S.; Sengupta, K.; Kassim, A.A. Compression of dynamic 3D geometry data using iterative closest point algorithm. *Comput. Vis. Image Underst.* **2002**, *87*, 116–130. [CrossRef]
39. Gupta, S.; Sengupta, K.; Kassim, A. Registration and partitioning-based compression of 3D dynamic data. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 1144–1155. [CrossRef]
40. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
41. Myronenko, A.; Song, X. Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 2262–2275. [CrossRef]
42. Hu, Y.; Zhou, M.; Wu, Z. A dense point-to-point alignment method for realistic 3D face morphing and animation. *Int. J. Comput. Games Technol.* **2009**, *2009*, 1–9. [CrossRef]
43. Hu, Y.; Zhou, M.; Wu, Z. An automatic dense point registration method for 3D face animation. In Proceedings of the 2009 2nd International Congress on Image and Signal Processing, Tianjin, China, 17–19 October 2009; pp. 1–6.
44. Lengyel, J.E. Compression of time-dependent geometry. In Proceedings of the 1999 Symposium on Interactive 3D Graphics, Atlanta, GA, USA, 26–29 April 1999; pp. 89–95.
45. Ahn, J.H.; Kim, C.S.; Kuo, C.J.; Ho, Y.S. Motion-compensated compression of 3D animation models. *Electron. Lett.* **2001**, *37*, 1445–1446. [CrossRef]
46. Alexa, M.; Müller, W. Representing animations by principal components. *Comput. Graph. Forum* **2000**, *19*, 411–418. [CrossRef]
47. Karni, Z.; Gotsman, C. Compression of soft-body animation sequences. *Comput. Graph.* **2004**, *28*, 25–34. [CrossRef]
48. Ibarria, L.; Rossignac, J. Dynapack: Space-time compression of the 3D animations of triangle meshes with fixed connectivity. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, CA, USA, 26–27 July 2003; pp. 126–135.
49. Ahn, J.K.; Koh, Y.J.; Kim, C.S. Efficient fine-granular scalable coding of 3D-mesh sequences. *IEEE Trans. Multimed.* **2012**, *15*, 485–497. [CrossRef]
50. Guskov, I.; Khodakovsky, A. Wavelet compression of parametrically coherent mesh sequences. In Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Grenoble, France, 27–29 August 2004; pp. 183–192.
51. Payan, F.; Antonini, M. Temporal wavelet-based compression for 3D animated models. *Comput. Graph.* **2007**, *31*, 77–88. [CrossRef]
52. Mamou, K.; Zaharia, T.; Prêteux, F.; Stefanoski, N.; Ostermann, J. Frame-based compression of animated meshes in MPEG-4. In Proceedings of the IEEE International Conference on Multimedia and Expo, Hannover, Germany, 23–26 June 2008; pp. 1121–1124.
53. Gu, X.; Gortler, S.J.; Hoppe, H. Geometry images. *ACM Trans. Graph.* **2002**, *21*, 355–361. [CrossRef]
54. Briceño Pulido, H.M. Geometry videos: A new representation for 3D animations. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, CA, USA, 26–27 July 2003; pp. 136–146.

55. Quynh, D.T.; He, Y.; Chen, X.; Xia, J.; Sun, Q.; Hoi, S.C.; Quynh, D.T.; He, Y.; Chen, X.; Xia, J.; et al. Modeling 3D articulated motions with conformal geometry videos (CGVs). In Proceedings of the 19th ACM International Conference on Multimedia, Scottsdale, AZ, USA, 28 November–1 December 2011; pp. 383–392.
56. Xia, J.; Quynh, D.T.P.; He, Y.; Chen, X.; Hoi, S.C. Modeling and compressing 3-D facial expressions using geometry videos. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *22*, 77–90. [CrossRef]
57. Hou, J.; Chau, L.P.; He, Y.; Magnenat-Thalmann, N. A novel compression framework for 3D time-varying meshes. In Proceedings of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, Australia, 1–5 June 2014; pp. 2161–2164.
58. Hou, J.; Chau, L.P.; Zhang, M.; Magnenat-Thalmann, N.; He, Y. A highly efficient compression frame-work for time-varying 3D facial expressions. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 1541–1553.
59. Sander, P.V.; Snyder, J.; Gortler, S.J.; Hoppe, H. Texture mapping progressive meshes. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 12–17 August 2001; pp. 409–416.
60. Yin, L.; Chen, X.; Sun, Y.; Worm, T.; Reale, M. A high-resolution 3D dynamic facial expression database. In Proceedings of the 8th IEEE International Conference on Automatic Face & Gesture Recognition, Amsterdam, The Netherlands, 17–19 September 2008; pp. 1–6.