Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2017, Article ID 2724184, 15 pages
https://doi.org/10.1155/2017/2724184

Research Article

Adaptive Geometry Images for Remeshing

Lina Shi,[1] Dehui Kong,[1] Shaofan Wang,[1] and Baocai Yin[2]

[1]*Beijing Key Laboratory of Multimedia & Intelligent Software Technology, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2]*Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China*

Correspondence should be addressed to Shaofan Wang; wangshaofan@bjut.edu.cn

Received 6 March 2017; Accepted 30 April 2017; Published 2 August 2017

Academic Editor: Yifeng He

Geometry images are a kind of completely regular remeshing methods for mesh representation. Traditional geometry images have difficulties in achieving optimal reconstruction errors and preserving manually selected geometric details, due to the limitations of parametrization methods. To solve two issues, we propose two adaptive geometry images for remeshing triangular meshes. The first scheme produces geometry images with the minimum Hausdorff error by finding the optimization direction for sampling points based on the Hausdorff distance between the original mesh and the reconstructed mesh. The second scheme produces geometry images with higher reconstruction precision over the manually selected region-of-interest of the input mesh, by increasing the number of sampling points over the region-of-interest. Experimental results show that both schemes give promising results compared with traditional parametrization-based geometry images.

1. Introduction

Triangular meshes are important tools for representing geometric data in computer graphics, due to the ease of generation procedure from point clouds and simple manipulation. However, many applications of meshes such as mesh morphing and mesh compression tend to use triangular meshes with regular structure. Traditional triangular meshes, which have irregular connectivity due to the generation procedure, need to be remeshed. This technique is referred to as *remeshing*. Geometry images are a completely regular remeshing method, which represents a triangular mesh as an image array, where the vertex-set of the mesh is stored as the pixels of the image and where the connectivity of the mesh is intrinsically embedded in the image array. Such a regular structure of meshes is helpful for reducing the representation of geometric data and coworking well with many image-based applications such as image compression and rendering process.

In general, geometry images include three steps: mesh parametrization, resampling, and quantification. The first step maps 3D vertices of the input mesh to regular parametrization domain (square, rectangle, or sphere), the second step imposes sampling over the parametrization domain via interpolation methods, and the third step transforms the coordinates of sampling points to pixel values of an image array. To reconstruct a mesh for geometry images, the vertex-set is obtained from the pixels of the image array, and the edge-set is obtained from the connectivity of the adjacent pixels of the array.

Although fruitful research work was proposed for geometry images, many of them focuses on the mesh parametrization and ignores the importance of the resampling step, which increases the burden of parametrization technique, as the parametrization leads to a complicated and nonconvex optimization which heavily depends on the connectivity of the input mesh (while the resampling scheme depends more on the connectivity of the regular sampling fashion instead of the input mesh). In particular, traditional geometry images have difficulties in achieving optimal reconstruction errors, or in preserving manually selected geometric details. To solve such two issues, we propose two adaptive geometry images. The first scheme produces geometry images with the minimum Hausdorff error, by finding the optimization direction for sampling points based on the Hausdorff distance between the original mesh and the reconstructed mesh.

The second scheme produces geometry images with higher reconstruction precision over the manually selected region-of-interest of the input mesh, by increasing the number of sampling points over the region-of-interest. We compare our schemes with traditional geometry images using state-of-the-art mesh parametrization scheme and adaptive sampling scheme in terms of both reconstruction error and mesh compression. Experimental results on both qualitative comparison and quantitative comparison show that our schemes outperform traditional geometry images.

2. Related Work

2.1. Geometry Images. Gu et al. [1] propose the pioneering work of geometry images, which maps a triangular mesh onto a square domain by using a minimizing-geometric-stretch parametrization and gives a regular sampling for surface geometry. Praun and Hoppe [2] propose spherical parametrization for geometry images, which facilitates the representation of genus-zero closed meshes. Gauthier and Poulin [3] fill nonzero genus meshes and propose spherical parametrization for treating meshes of arbitrary genus. Zhou et al. [4] propose an adaptive sampling scheme for geometry images, which keeps most details of models. Gauthier and Poulin [5] propose another sampling scheme for geometry images to maintain both edge features and sharp features. Meng et al. [6] adopt differential coordinates to correct the vector direction of the reconstruction model, which makes the reconstruction model accurately preserve the detailed features of the original model.

The aforementioned work of geometry images maps models into single-chart geometry images, which tends to produce high geometric stretch and ignore details of models. Alternatively, Tewari et al. [7] propose multichart geometry images by cutting the model into some irregular subslices, but it required a lot of space to store the information of subslices. Carr et al. [8] convert the irregular subslices to quadrilateral subslices. Yao and Lee [9] decompose a mesh into square GIM charts with different resolutions, each of which is adaptively determined by a local reconstruction error. Feng et al. [10] propose geometry images for generating triangular patches based on a curvilinear feature. The feature preserves salient features and supports GPU-based LOD representation of meshes.

2.2. Other Remeshing Methods. Alliez et al. [11] propose an interactive remeshing of irregular geometry, which represents the original mesh as a series of 2D parametrization maps. The algorithm facilitates the real-time interaction and intricate control using a map which controls the sampling density over the surface patch. Alliez et al. [12] propose a polygonal remeshing method using the intrinsic anisotropy of natural or man-made geometry. The authors use curvature directions to drive the remeshing process and determine appropriate edges for the remeshed version in anisotropic regions. The method provides the flexibility to produce meshes ranging from isotropic to anisotropic, from coarse to dense, and from uniform to curvature adapted. Dong et al. [13] propose a new quadrilateral remeshing method for manifolds of arbitrary genus. The method computes the gradient of smooth harmonic scalar fields defined over the mesh and forms the polygons of the output mesh using two nets of integral lines. Huang et al. [14] propose a quadrangulation method, by extending the spectral surface quadrangulation approach with the coarse quadrangular structure derived from the Morse-Smale complex of an eigenfunction of the Laplacian operator on the input mesh. The quadrilateral mesh is reconstructed from the Morse-Smale complex by computing a globally smooth parametrization. Zhang et al. [15] propose a new method for remeshing a surface into a quadrangle, by constructing a special standing wave on the surface to generate the global quadrilateral structure, which controls the quad size in two directions and precisely aligning the quads with feature lines.

3. Hausdorff Error Driven Geometry Images

We propose Hausdorff error driven geometry images in this section. The key step of our scheme is to find the points, edges, or faces of the original mesh and reconstruction mesh which achieve the maximum Hausdorff distance and then compute the gradient direction of the Hausdorff distance. Our scheme consists of three phases: an initial adaptive sampling, approximate representation of Hausdorff distance, and the adjustment of sampling vertices, which are described in the following three subsections and illustrated in Figure 1. We shall use the calligraphy letter $\mathscr{M}$ for representing a mesh and denote $\mathscr{V}_{\mathscr{M}}$, $\mathscr{F}_{\mathscr{M}}$, $\mathscr{P}_{\mathscr{M}}$ to be the vertex-set, the face-set, and the point-set (i.e., all points within each face) of a mesh $\mathscr{M}$, respectively, and denote $\mathscr{E}_f$, $\mathscr{V}_e$, $\mathscr{V}_f$ to be the edge-set of a face f, the set of two end-vertices of an edge e, and the set of three end-vertices of a face f, respectively.

3.1. Vertex Density Equalization Based Adaptive Sampling. We propose an initial adaptive sampling scheme using a vertex density equalization metric in this subsection. The equalization is adopted along both x-axis direction and y-axis direction direction over the parametrization domain $\Omega := [0,1]^2$. The algorithm is illustrated as follows:

(1) Employ a $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$ uniform partition over Ω, that is, $\Omega = \bigcup_{i,j=1}^{\lfloor n/2 \rfloor} \Omega_{ij}$ with $\Omega_{ij} = [(i-1)/\lfloor n/2 \rfloor, i/\lfloor n/2 \rfloor] \times [(j-1)/\lfloor n/2 \rfloor, j/\lfloor n/2 \rfloor]$, $i, j = 1, \ldots, \lfloor n/2 \rfloor$.

(2) Denote $\Omega_i^x = \bigcup_j \Omega_{ij}$ to be the x-axis partition and find the partition set $\Omega_{i^*}^x$ which contains the greatest number of parametrization vertices.

(3) Divide the partition set $\Omega_{i^*}^x$ into two sets, that is, $\Omega = \bigcup_{i,j=1}^{\lfloor n/2 \rfloor + 1, \lfloor n/2 \rfloor} \Omega_{ij}$ with $\Omega_{i^*,j} = [(i^*-1)/\lfloor n/2 \rfloor, (i^*-1/2)/\lfloor n/2 \rfloor] \times [(j-1)/\lfloor n/2 \rfloor, j/\lfloor n/2 \rfloor]$, and $\Omega_{i^*+1,j} = [(i^*-1/2)/\lfloor n/2 \rfloor, i^*/\lfloor n/2 \rfloor] \times [(j-1)/\lfloor n/2 \rfloor, j/\lfloor n/2 \rfloor]$.

(4) Repeat Steps (2) and (3) until the x-axis partition contains $n-1$ partition sets.

(5) Apply Steps (2), (3), and (4) on the y-axis partition in a similar fashion.

The algorithm yields $2(n-1)$ partition sets $\{\Omega_i^x\}_{i=1}^{n-1}$ along x-axis and the other $n-1$ sets $\{\Omega_i^y\}_{i=1}^{n-1}$ along y-axis. The location

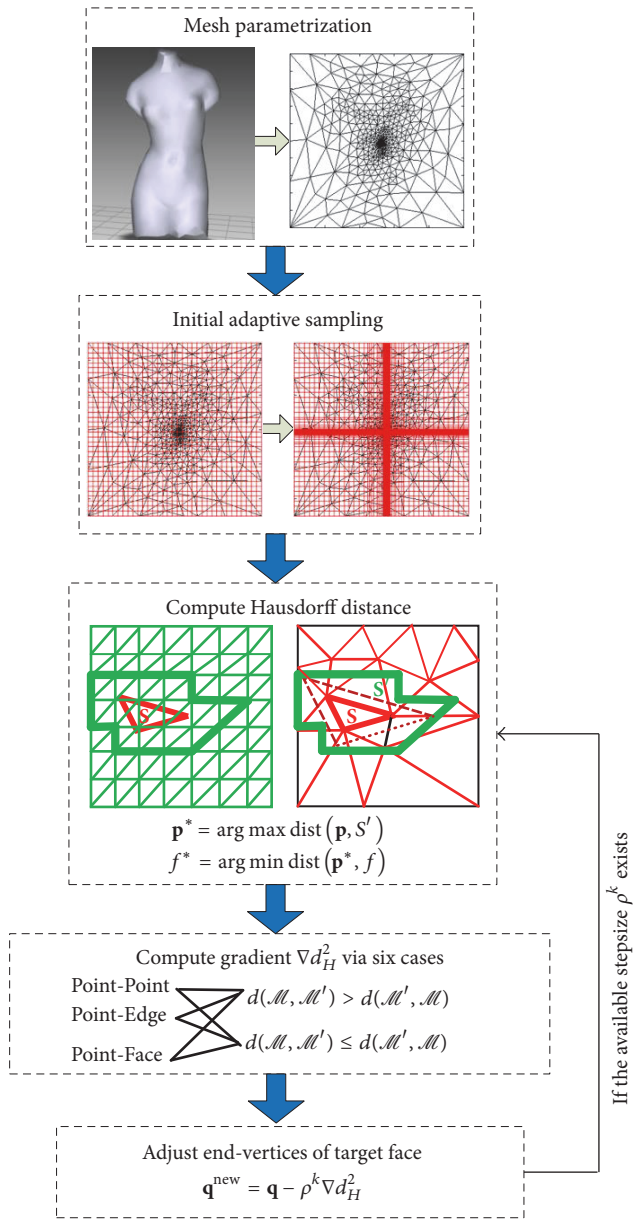


Figure 1: A framework for Hausdorff error driven geometry images.

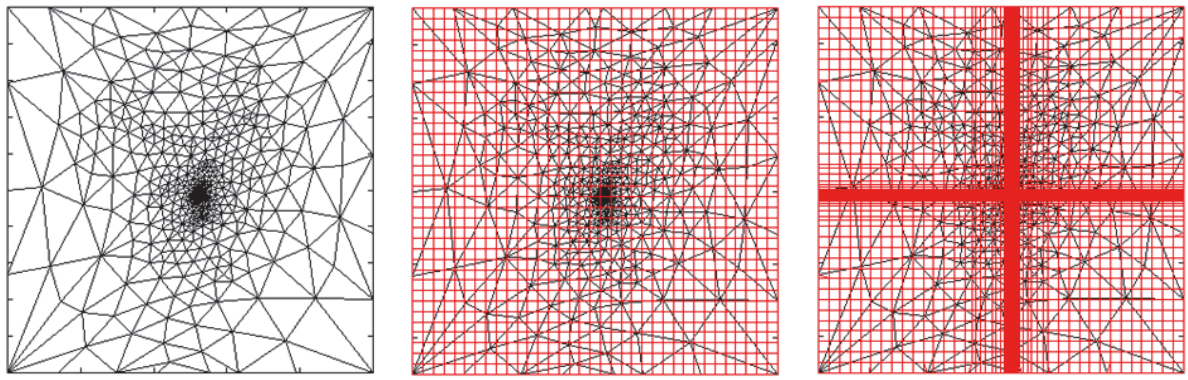
of sampling vertices is then given by the intersections of bounding lines of a partition set along x-axis and a partition set along y-axis, and the sampling rate is $n \times n$. We show the sampling result of venus in Figure 2.

3.2. Representation of Hausdorff Distance between Meshes. Hausdorff distance measures the distance between two 3D meshes (see [16]), which is defined by

$$d_H\left(\mathcal{M}, \mathcal{M}'\right) = \max\left\{d\left(\mathcal{M}, \mathcal{M}'\right), d\left(\mathcal{M}', \mathcal{M}\right)\right\}, \quad (1)$$

$$\begin{aligned} d\left(\mathcal{M}, \mathcal{M}'\right) &= \max_{\mathbf{p}\in\mathcal{P}_{\mathcal{M}}} \operatorname{dist}\left(\mathbf{p}, \mathcal{M}'\right) \\ &= \max_{\mathbf{p}\in\mathcal{P}_{\mathcal{M}}} \min_{\mathbf{p}'\in\mathcal{P}_{\mathcal{M}'}} \left\|\mathbf{p} - \mathbf{p}'\right\|. \end{aligned} \quad (2)$$

According to the definition, the Hausdorff distance depends only on each face of a mesh together with a small number of faces of the other mesh relatively closed to it. Therefore, in order to speed up the computation of the Hausdorff distance, for each target face of the target mesh, we only choose a few number of faces from the other mesh (i.e., we enlarge the parametrization domain of the target face of the target mesh twice, and the faces are chosen as the faces of the other mesh which share common points with the enlarged target face over the parametrization domain). We illustrate the idea in Figure 3, where $\mathcal{S}$, $\mathcal{S}'$ denote a target face of the target mesh and the selected faces of the sampling mesh, respectively. During the computation of Hausdorff distance from a face $\mathcal{S}$ of original mesh to reconstruction mesh, we need to record three pieces of data: the Hausdorff distance



(a) (b) (c)

Figure 2: The vertex density equalization based adaptive sampling over venus with sampling rate $n \times n$. From (a) to (c): the parametrization mesh, the $[n/2] \times [n/2]$ regular sampling mesh, and the vertex density equalization based adaptive sampling mesh, with $n = 70$.

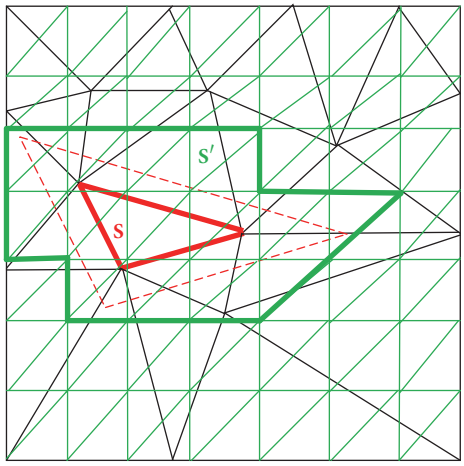


Figure 3: Selection of the associated faces with respect to the target face. Black triangles denote the parametrization mesh, green triangles denote the sampling mesh, and red solid triangle denotes a target face $\mathcal{S}$. The red dash triangles denote the enlarged face, and the green bold lines denote the collection $\mathcal{S}'$ of the selected faces of the sampling mesh.

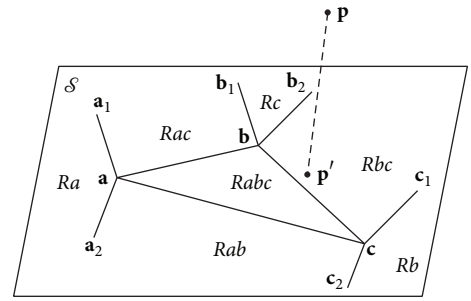


Figure 4: Different types of distances between a point **p** and a triangle **abc**.

$d(\mathcal{S}, \mathcal{S}')$ from $\mathcal{S}$ to $\mathcal{S}'$, the point $\mathbf{p}^*$ of $\mathcal{S}$ which achieves the Hausdorff distance $\mathbf{p}^* = \arg\max_{\mathbf{p}\in\mathcal{P}_S} \mathrm{dist}(\mathbf{p}, \mathcal{S}')$, and the face f^* of $\mathcal{S}'$ which achieves the Hausdorff distance together: $f^* = \arg\min_{f\in\mathcal{F}_{\mathcal{S}'}} \mathrm{dist}(\mathbf{p}^*, f)$. Such process is repeated during the computation of Hausdorff distance from a face $\mathcal{S}$ of reconstruction mesh to original mesh.

3.3. Iterative Adjustment of Sampling Vertices. The key step of our scheme is the iterative adjustment of sampling vertices. We first give the distance type judgement between the point and the face which achieve the maximum Hausdorff distance obtained in Section 3.2 and then compute the gradient direction of the Hausdorff distance with respect to the vertex/vertices associated with the maximum Hausdorff distance. The adjustment of the vertices is then applied using gradient descent method with suitable step length. The whole algorithm stops when no vertices can be adjusted.

(1) Determination of the Distance Type between the Target Point and the Target Face. Similar to [17], the Hausdorff distance between a point and a triangle includes three types: Point-Face, Point-Edge, and Point-Point. We describe the main idea in Figure 4 (see also [17]). We denote $\mathbf{p}'$ to be the projection of **p** over the plane containing **abc**. We first judge whether $\mathbf{p}'$ and **a** lie in the same side or different sides of line **bc** and judge the relationship of $\mathbf{p}'$ and **b** with respect to the line **ca**, as well as the relationship of $\mathbf{p}'$ and **c** with respect to the line **ab**. Three judgements give eight cases of results with one of them (i.e., $\mathbf{p}'$ and each of the end-vertices of **abc** lie in different sides of the corresponding edge) always being invalid. Thus the seven cases of results correspond to the seven cases of locations of $\mathbf{p}'$ corresponding to seven types of Hausdorff distance from **p** to **abc**. In detail, we denote

$$l_{\mathbf{bc}}(x, y) = 0,$$
$$l_{\mathbf{ca}}(x, y) = 0, \quad (3)$$
$$l_{\mathbf{ab}}(x, y) = 0$$

to be the equations of edges **bc**, **ca**, **ab**, respectively. Then we denote

$$\lambda_1 = l_{\mathbf{bc}}(\mathbf{a})\, l_{\mathbf{bc}}(\mathbf{p}'),$$
$$\lambda_2 = l_{\mathbf{ca}}(\mathbf{b})\, l_{\mathbf{ca}}(\mathbf{p}'), \quad (4)$$
$$\lambda_3 = l_{\mathbf{ab}}(\mathbf{c})\, l_{\mathbf{ab}}(\mathbf{p}')$$

to be three real numbers for distance type determination. The distance type between the point $\mathbf{p}$ and the face $\mathbf{abc}$ is given by

$$\begin{aligned}
&\text{Point-Face } \mathbf{p}\text{-}\mathbf{abc} \quad \text{if } \max(\lambda_1,\lambda_2,\lambda_3) > 0,\\
&\text{Point-Edge } \mathbf{p}\text{-}\mathbf{bc} \quad \text{if } \min(\lambda_2,\lambda_3) > 0 \geq \lambda_1,\\
&\text{Point-Edge } \mathbf{p}\text{-}\mathbf{ca} \quad \text{if } \min(\lambda_3,\lambda_1) > 0 \geq \lambda_2,\\
&\text{Point-Edge } \mathbf{p}\text{-}\mathbf{ab} \quad \text{if } \min(\lambda_1,\lambda_2) > 0 \geq \lambda_3,\\
&\text{Point-Point } \mathbf{p}\text{-}\mathbf{a} \quad \text{if } \min(\lambda_2,\lambda_3) \leq 0,\\
&\text{Point-Point } \mathbf{p}\text{-}\mathbf{b} \quad \text{if } \min(\lambda_3,\lambda_1) \leq 0,\\
&\text{Point-Point } \mathbf{p}\text{-}\mathbf{c} \quad \text{if } \min(\lambda_1,\lambda_2) \leq 0.
\end{aligned} \quad (5)$$

Finally, we obtain the following two distance sets:

$$\mathscr{D}_1 = \left\{ d\left(f, \mathscr{M}'\right) : f \in \mathscr{F}_{\mathscr{M}} \right\},$$
$$\mathscr{D}_2 = \left\{ d\left(f', \mathscr{M}\right) : f' \in \mathscr{F}_{\mathscr{M}'} \right\}, \quad (6)$$

where $\mathscr{D}_1$ ($\mathscr{D}_2$, resp.) is the collection of the directed Hausdorff distances from each face of $\mathscr{M}$ ($\mathscr{M}'$, resp.) to the reconstruction mesh $\mathscr{M}'$ (original mesh $\mathscr{M}$, resp.), with the directed Hausdorff distances $d(f, \mathscr{M}')$, $d(f', \mathscr{M})$ defined by (1).

(2) Selection of the Gradient Direction. There are three kinds of objective functions: Point-Face, Point-Edge, and Point-Point and two cases of Hausdorff distance relationship: $\max \mathscr{D}_1 > \max \mathscr{D}_2$ and $\max \mathscr{D}_1 \leq \max \mathscr{D}_2$, which produces six kinds of gradient direction. We introduce them in the following three paragraphs.

Although verbose equations shall be listed, the main idea is simple. The three cases (Point-Point, Point-Edge, and Point-Face) determine the distance equation [(8), (11), and (14)], and the two cases ($\max \mathscr{D}_1 > \max \mathscr{D}_2$ or $\max \mathscr{D}_1 \leq \max \mathscr{D}_2$) determine which point(s) can be moved: when $\max \mathscr{D}_1 > \max \mathscr{D}_2$, we update the location of the target end-vertex (edge, face) on $\mathscr{M}'$ according to the Point-Point (Point-Edge, Point-Face) case; when $\max \mathscr{D}_1 \leq \max \mathscr{D}_2$, we update the locations of the end-vertices of the face containing the target point (i.e., the point achieving the greatest Hausdorff distance) on $\mathscr{M}'$.

Within the following three paragraphs, we denote $\mathbf{p}^* = (x_{\mathbf{p}^*}, y_{\mathbf{p}^*}, z_{\mathbf{p}^*})$ to be the target point, and denote $f^* = \mathbf{q}_1\mathbf{q}_2\mathbf{q}_3$ to be the target face; that is, f^* is the face of the corresponding mesh of $\mathbf{p}^*$ which achieves the maximum distance from $\mathbf{p}^*$. Note that $\mathbf{p}^*$, f^* belong to different meshes of $\mathscr{M}$, $\mathscr{M}'$. We also denote $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$ to be the face containing $\mathbf{p}^*$ and denote α, β, γ to be the barycentric coordinate of $\mathbf{p}^*$ with respect to $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$; that is, $\mathbf{p}^* = \alpha\mathbf{p}_1 + \beta\mathbf{p}_2 + \gamma\mathbf{p}_3$. We denote $(x_\square, y_\square, z_\square)$ to be the coordinates of the point $\square$ and denote

$$\begin{aligned}
x_{ij} &= x_{\mathbf{q}_i} - x_{\mathbf{q}_j},\\
y_{ij} &= y_{\mathbf{q}_i} - y_{\mathbf{q}_j},\\
z_{ij} &= z_{\mathbf{q}_i} - z_{\mathbf{q}_j},\\
&\qquad i, j = 1, 2, 3
\end{aligned} \quad (7)$$

to be the differences of coordinates of $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$.

(a) Point-Point Optimization. The Point-Point distance from $\mathbf{p}^*$ to an end-vertex $\mathbf{q}$ of f^* is given by

$$d_{\mathrm{pp}}^2 = \left(x_{\mathbf{q}} - x_{\mathbf{p}^*}\right)^2 + \left(y_{\mathbf{q}} - y_{\mathbf{p}^*}\right)^2 + \left(z_{\mathbf{q}} - z_{\mathbf{p}^*}\right)^2. \quad (8)$$

We compute the gradient of d_{pp}^2 in two cases. If $\max \mathscr{D}_1 > \max \mathscr{D}_2$, then $\mathbf{p}^* \in \mathscr{P}_{\mathscr{M}}$, $\mathbf{q} \in \mathscr{V}_{\mathscr{M}'}$ and we update the location of $\mathbf{q}$ using the gradient of d_{pp}^2 with respect to $\mathbf{q}$, which is given by

$$\nabla_{\mathbf{q}} d_{\mathrm{pp}}^2 = 2\left[x_{\mathbf{q}} - x_{\mathbf{p}^*} \quad y_{\mathbf{q}} - y_{\mathbf{p}^*} \quad z_{\mathbf{q}} - z_{\mathbf{p}^*}\right]^\top. \quad (9)$$

Otherwise, $\max \mathscr{D}_1 \leq \max \mathscr{D}_2$ holds; then we have $\mathbf{p}^* \in \mathscr{P}_{\mathscr{M}'}$, $\mathbf{q} \in \mathscr{V}_{\mathscr{M}}$ and we shall update the locations of $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$. Thus we compute the gradient of d_{pp}^2 with respect to $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ by

$$\nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d_{\mathrm{pp}}^2 = 2\begin{bmatrix}\alpha\mathbf{M}_1\\ \beta\mathbf{M}_1\\ \gamma\mathbf{M}_1\end{bmatrix} \quad \text{with } \mathbf{M}_1 = \begin{bmatrix}x_{\mathbf{p}^*} - x_{\mathbf{q}}\\ y_{\mathbf{p}^*} - y_{\mathbf{q}}\\ z_{\mathbf{p}^*} - z_{\mathbf{q}}\end{bmatrix}, \quad (10)$$

where we use the notation $\nabla_{\{\mathbf{q}_1;\ldots;\mathbf{q}_m\}} d$ for concatenating the gradient of $d(\mathbf{q}_1, \ldots, \mathbf{q}_m)$ with respect to m points $\{\mathbf{q}_1, \ldots, \mathbf{q}_m\} \subseteq \mathbb{R}^3$ to form a vector of $\mathbb{R}^{3m}$.

(b) Point-Edge Optimization. Let $\mathbf{q}_1 = (x_1, y_1, z_1)$, $\mathbf{q}_2 = (x_2, y_2, z_2)$ be two end-vertices of the triangle $\mathbf{q}_1\mathbf{q}_2\mathbf{q}_3$. The Point-Edge distance from $\mathbf{p}^*$ to the edge $\mathbf{q}_1\mathbf{q}_2$ is given by

$$d_{\mathrm{pe}}^2 = \frac{\left\{\left[\left(y_{\mathbf{p}^*} - y_{\mathbf{q}_1}\right) z_{12} - \left(z_{\mathbf{p}^*} - z_{\mathbf{q}_1}\right) y_{12}\right]^2 + \left[\left(z_{\mathbf{p}^*} - z_{\mathbf{q}_1}\right) x_{12} - \left(x_{\mathbf{p}^*} - x_{\mathbf{q}_1}\right) z_{12}\right]^2 + \left[\left(x_{\mathbf{p}^*} - x_{\mathbf{q}_1}\right) y_{12} - \left(y_{\mathbf{p}^*} - y_{\mathbf{q}_1}\right) x_{12}\right]^2\right\}}{x_{12}^2 + y_{12}^2 + z_{12}^2}. \quad (11)$$

Input. A mesh $\mathscr{M}$ and stepsize threshold $K = 100$
Output. The 3D location of vertices of GIM mesh $\mathscr{M}'$
(1) Apply vertex density equalization on $\mathscr{M}$ (Section 3.1);
(2) Compute $\mathscr{D}_1$, $\mathscr{D}_2$ using (6) (Section 3.2);
(3) **If** $\max \mathscr{D}_1 > \max \mathscr{D}_2$ **then**
(4) Compute the target point $\mathbf{p}^* \leftarrow \arg\max_{\mathbf{p}\in\mathscr{P}_{\mathscr{M}}} \operatorname{dist}(\mathbf{p}, \mathscr{M}')$;
(5) Compute the target face $f^* \leftarrow \arg\max_{f\in\mathscr{F}_{\mathscr{M}'}} \operatorname{dist}(\mathbf{p}^*, f)$;
(6) Judge the distance type between $\mathbf{p}^*$ and f^* using (5);
(7) Update the position of the end-vertices of f^* using Algorithm 2;
(8) **else**
(9) Compute the target point $\mathbf{p}^* \leftarrow \arg\max_{\mathbf{p}\in\mathscr{P}_{\mathscr{M}'}} \operatorname{dist}(\mathbf{p}, \mathscr{M})$;
(10) Compute the target face $f^* \leftarrow \arg\max_{f\in\mathscr{F}_{\mathscr{M}}} \operatorname{dist}(\mathbf{p}^*, f)$;
(11) Judge the distance type between $\mathbf{p}^*$ and f^* using (5);
(12) Update the position of the end-vertices of f^* using Algorithm 3;
(13) **end**

Algorithm 1: Optimal Hausdorff error driven geometry images.

If $\max \mathscr{D}_1 > \max \mathscr{D}_2$, then $\mathbf{p}^* \in \mathscr{P}_{\mathscr{M}}$, $\mathbf{q}_1\mathbf{q}_2 \in \mathscr{E}_{\mathscr{M}'}$ hold, and we shall update the locations $\mathbf{q}_1$, $\mathbf{q}_2$ with the gradient of d^2_{pe} given by (16). Otherwise, we have $\mathbf{p}^* \in \mathscr{P}_{\mathscr{M}'}$, $\mathbf{q}_1\mathbf{q}_2 \in \mathscr{E}_{\mathscr{M}}$, and we update all the end-vertices of the face containing $\mathbf{p}^*$. The gradient of d^2_{pe} with respect to all the end-vertices of the face is given by

$$\nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\mathrm{pe}} = \frac{2}{x_{12}^2 + y_{12}^2 + z_{12}^2} \begin{bmatrix} \alpha\mathbf{M}_2 \\ \beta\mathbf{M}_2 \\ \gamma\mathbf{M}_2 \end{bmatrix} \begin{bmatrix} x_{12} \\ y_{12} \\ z_{12} \end{bmatrix} \quad \text{with } \mathbf{M}_2 = \begin{bmatrix} 0 & \left(x_{\mathbf{p}^*} - x_{\mathbf{q}_1}\right) y_{12} & \left(x_{\mathbf{p}^*} - x_{\mathbf{q}_1}\right) z_{12} \\ \left(y_{\mathbf{p}^*} - y_{\mathbf{q}_1}\right) x_{12} & -\left(y_{\mathbf{p}^*} - y_{\mathbf{q}_1}\right) x_{12} & -\left(z_{\mathbf{p}^*} - z_{\mathbf{q}_1}\right) x_{12} \\ -\left(x_{\mathbf{p}^*} - x_{\mathbf{q}_1}\right) y_{12} & 0 & \left(y_{\mathbf{p}^*} - y_{\mathbf{q}_1}\right) z_{12} \\ \left(z_{\mathbf{p}^*} - z_{\mathbf{q}_1}\right) x_{12} & \left(z_{\mathbf{p}^*} - z_{\mathbf{q}_1}\right) y_{12} & -\left(z_{\mathbf{p}^*} - z_{\mathbf{q}_1}\right) y_{12} \\ -\left(x_{\mathbf{p}^*} - x_{\mathbf{q}_1}\right) z_{12} & -\left(y_{\mathbf{p}^*} - y_{\mathbf{q}_1}\right) z_{12} & 0 \end{bmatrix}. \quad (12)$$

(c) Point-Face Optimization. We denote

$$\begin{aligned} a &= y_{\mathbf{q}_1} z_{23} + y_{\mathbf{q}_2} z_{31} + y_{\mathbf{q}_3} z_{12}, \\ b &= z_{\mathbf{q}_1} x_{23} + z_{\mathbf{q}_2} x_{31} + z_{\mathbf{q}_3} x_{12}, \\ c &= x_{\mathbf{q}_1} y_{23} + x_{\mathbf{q}_2} y_{31} + x_{\mathbf{q}_3} y_{12}, \\ g &= x_{\mathbf{q}_1}\left(y_{\mathbf{q}_3} z_{\mathbf{q}_2} - y_{\mathbf{q}_2} z_{\mathbf{q}_3}\right) + x_{\mathbf{q}_2}\left(y_{\mathbf{q}_1} z_{\mathbf{q}_3} - y_{\mathbf{q}_3} z_{\mathbf{q}_1}\right) \\ &\quad + x_{\mathbf{q}_3}\left(y_{\mathbf{q}_2} z_{\mathbf{q}_1} - y_{\mathbf{q}_1} z_{\mathbf{q}_2}\right) \end{aligned} \quad (13)$$

to be the coefficients of the equation of the plane where the face $\mathbf{q}_1\mathbf{q}_2\mathbf{q}_3$ lies, and the Point-Face distance from $\mathbf{p}^*$ to the face $\mathbf{q}_1\mathbf{q}_2\mathbf{q}_3$ is given by

$$d^2_{\mathrm{pf}} = \frac{\left(a x_{\mathbf{p}^*} + b y_{\mathbf{p}^*} + c z_{\mathbf{p}^*} + g\right)^2}{a^2 + b^2 + c^2}. \quad (14)$$

If $\max \mathscr{D}_1 > \max \mathscr{D}_2$, then $\mathbf{p}^* \in \mathscr{P}_{\mathscr{M}}$, $\mathbf{q}_1\mathbf{q}_2\mathbf{q}_3 \in \mathscr{F}_{\mathscr{M}'}$ and we shall update the locations of $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$ using the gradient of d^2_{pf} given by (17). Otherwise, we update the locations of $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ with the gradient of d^2_{pf} given by

$$\nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\mathrm{pf}} = \frac{2}{a^2 + b^2 + c^2} \begin{bmatrix} \alpha\mathbf{M}_3 \\ \beta\mathbf{M}_3 \\ \gamma\mathbf{M}_3 \end{bmatrix} \begin{bmatrix} x_{\mathbf{p}^*} \\ y_{\mathbf{p}^*} \\ z_{\mathbf{p}^*} \\ 1 \end{bmatrix} \quad \text{with } \mathbf{M}_3 = \begin{bmatrix} a^2 & ab & ac & ag \\ ab & b^2 & bc & bg \\ ac & bc & c^2 & cg \end{bmatrix}. \quad (15)$$

In summary, we show Algorithm 1 for generating the Hausdorff distance geometry images, where we compute the gradient of the directional Hausdorff distance square d^2 in six cases according to the distance type and the quantitative relationship of $\max \mathscr{D}_1$ and $\max \mathscr{D}_2$. Note that when $\max \mathscr{D}_1 > \max \mathscr{D}_2$, we update the end-vertices of the target face f^* on $\mathscr{M}'$ (see Algorithm 2); when $\max \mathscr{D}_1 \le \max \mathscr{D}_2$, we update the end-vertices of the face $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$ containing $\mathbf{p}^*$ on $\mathscr{M}'$ (see Algorithm 3). For each of the six cases, we compute the stepsize ρ^k, $k = 0, \ldots, K$ which admits decreasing of the Hausdorff distance; the algorithm terminates provided that no such stepsize can be found.

$$\nabla_{[\mathbf{q}_1;\mathbf{q}_2]} d_{\mathrm{pe}}^2 = \frac{2}{x_{12}^2 + y_{12}^2 + z_{12}^2} \begin{bmatrix} z_{\mathbf{p}^*} - z_{\mathbf{q}_2} & y_{\mathbf{p}^*} - y_{\mathbf{q}_2} & x_{21} \\ x_{\mathbf{p}^*} - x_{\mathbf{q}_2} & z_{\mathbf{p}^*} - z_{\mathbf{q}_2} & y_{21} \\ y_{\mathbf{p}^*} - y_{\mathbf{q}_2} & x_{\mathbf{p}^*} - x_{\mathbf{q}_2} & z_{21} \\ z_{\mathbf{q}_1} - z_{\mathbf{p}^*} & y_{\mathbf{q}_1} - y_{\mathbf{p}^*} & x_{12} \\ x_{\mathbf{q}_1} - x_{\mathbf{p}^*} & z_{\mathbf{q}_1} - z_{\mathbf{p}^*} & y_{12} \\ y_{\mathbf{q}_1} - y_{\mathbf{p}^*} & x_{\mathbf{q}_1} - x_{\mathbf{p}^*} & z_{12} \end{bmatrix} \begin{bmatrix} \left(z_{\mathbf{p}^*} - z_{\mathbf{q}_1}\right) x_{12} - \left(x_{\mathbf{p}^*} - x_{\mathbf{q}_1}\right) z_{12} \\ \left(y_{\mathbf{p}^*} - y_{\mathbf{q}_1}\right) x_{12} - \left(x_{\mathbf{p}^*} - x_{\mathbf{q}_1}\right) y_{12} \\ d_{\mathrm{pe}}^2 \end{bmatrix}. \tag{16}$$

$$\nabla_{[\mathbf{q}_1;\mathbf{q}_2;\mathbf{q}_3]} d_{\mathrm{pf}}^2 = 2h \begin{bmatrix} 0 & z_{32} & y_{23} & z_{\mathbf{q}_2} y_{\mathbf{q}_3} - y_{\mathbf{q}_2} z_{\mathbf{q}_3} \\ z_{23} & 0 & x_{32} & x_{\mathbf{q}_2} z_{\mathbf{q}_3} - z_{\mathbf{q}_2} x_{\mathbf{q}_3} \\ y_{32} & x_{23} & 0 & y_{\mathbf{q}_2} x_{\mathbf{q}_3} - x_{\mathbf{q}_2} y_{\mathbf{q}_3} \\ 0 & z_{13} & y_{31} & z_{\mathbf{q}_3} y_{\mathbf{q}_1} - y_{\mathbf{q}_3} z_{\mathbf{q}_1} \\ z_{31} & 0 & x_{13} & x_{\mathbf{q}_3} z_{\mathbf{q}_1} - z_{\mathbf{q}_3} x_{\mathbf{q}_1} \\ y_{13} & x_{31} & 0 & y_{\mathbf{q}_3} x_{\mathbf{q}_1} - x_{\mathbf{q}_3} y_{\mathbf{q}_1} \\ 0 & z_{21} & y_{12} & z_{\mathbf{q}_1} y_{\mathbf{q}_2} - y_{\mathbf{q}_1} z_{\mathbf{q}_2} \\ z_{12} & 0 & x_{21} & x_{\mathbf{q}_1} z_{\mathbf{q}_2} - z_{\mathbf{q}_1} x_{\mathbf{q}_2} \\ y_{21} & x_{12} & 0 & y_{\mathbf{q}_1} x_{\mathbf{q}_2} - x_{\mathbf{q}_1} y_{\mathbf{q}_2} \end{bmatrix} \begin{bmatrix} x_{\mathbf{p}^*} \\ y_{\mathbf{p}^*} \\ z_{\mathbf{p}^*} \\ 1 \end{bmatrix} + 2h^2 \begin{bmatrix} 0 & z_{23} & y_{32} \\ z_{32} & 0 & x_{23} \\ y_{23} & x_{32} & 0 \\ 0 & z_{31} & y_{13} \\ z_{13} & 0 & x_{31} \\ y_{31} & x_{13} & 0 \\ 0 & z_{12} & y_{21} \\ z_{21} & 0 & x_{12} \\ y_{12} & x_{21} & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{17}$$

$$\text{with } h = \frac{a x_{\mathbf{p}^*} + b y_{\mathbf{p}^*} + c z_{\mathbf{p}^*} + g}{a^2 + b^2 + c^2}$$

4. Enhanced ROI Geometry Images

Region-of-Interest (ROI) is the area that attracts the attention of human visual attention, and the other area is called non-Region-of-Interest (non-ROI). While it is difficult to directly select ROI on the geometry image, the normal vector coordinates from normal images can accurately reflect the original triangular mesh surface's details. Figure 5 shows the geometry image and the normal image of the foot model. When we select the ROI, the normal vector image can be used. The first and second subfigures of Figure 6 illustrate the selection of ROI, where the left subfigure marks the region which contains more information of the model than the region in the middle subfigure.

After the ROI is selected, we transform the ROI of the normal images into the parametrization domain and divide it into five regions according to the right subfigure of Figure 6. We calculate the number of vertices of the parameterized mesh and the number of vertices of the reconstructed mesh within the five regions, respectively. To give a suitable setting of weights of non-ROI vertices, we denote $n_{\mathrm{ideal}} = (n_c/|\mathscr{V}_{\mathscr{M}}|)n^2$ as the ideal number of sampling points in each non-ROI, where n_c is the number of the parameterized mesh vertex and n is the sampling rate. For the four non-ROI, we use the actual number of the reconstructed mesh vertex minus n_{ideal} to obtain four values and arrange them in a descending order. Negative value indicates the insufficient number of sampling points in this area, while positive value indicates the redundant number of sampling points. Finally, the weights are assigned from low to high according to this order. While the weight of ROI is the highest, the weight of each non-ROI region is the same.

Vertex movement must be in its first order domain to ensure that the topology of the final mesh is unchanged. We define $\mathbf{p}_{\mathrm{new}} = (\sum_{i=1}^{6} w_i \|\mathbf{p}_0 - \mathbf{p}_i\| \mathbf{p}_i) / \sum_{i=1}^{6} w_i \|\mathbf{p}_0 - \mathbf{p}_i\|$ to be the new position of the moving point, where w_i is the weight of the region in which the point locates and $\mathbf{p}_i$ are the 2D coordinates of one-ring neighborhood of $\mathbf{p}_0$ (note that the definition of $\mathbf{p}_{\mathrm{new}}$ tends to move each sampling vertex towards the greater-weight direction within its 1-ring neighborhood). The increasing number of sampling points in the ROI is defined by user. After the increase of the sampling points in the ROI, we use the adaptive sampling scheme of Section 3.3 over ROI, where the optimization function is the directed Hausdorff distance from ROI to the original mesh instead of the symmetric Hausdorff distance, so that the distribution of sampling points of ROI is more suitable.

5. Experimental Results

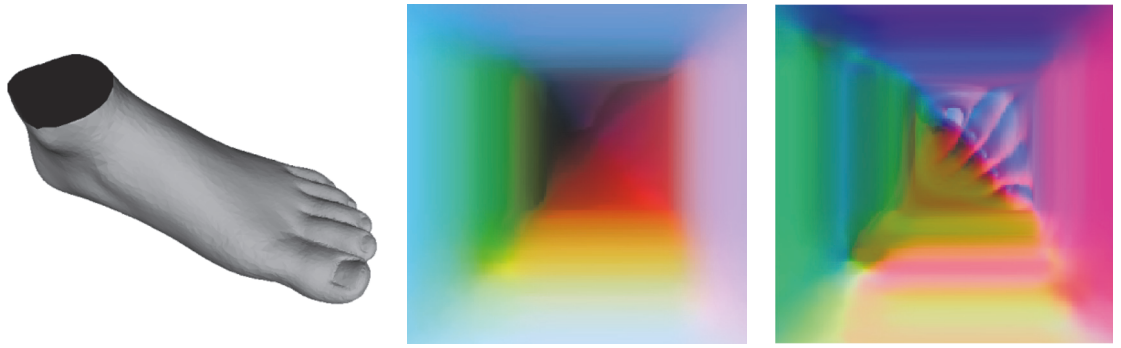
We show experimental results in this section. The first subsection shows the Hausdorff error driven geometry images while the second subsection shows geometry images with enhanced ROI reconstruction. Both qualitative results and quantitative results are compared. The second subsection also shows results with image compression with JPEG2000 codec.

5.1. Optimal Hausdorff Error Driven Geometry Images. We use two kinds of parametrization method: barycentric parametrization [18] and geometric-stretch parametrization [19] and three sampling methods: regular sampling, [4]'s adaptive sampling, and vertex density equalization adaptive sampling. We do three groups of experiments. Qualitative results are shown in Figure 7, where each column from left to

Input. A target point $\mathbf{p}^*$, a target face f^*, a mesh $\mathscr{M}$ and stepsize threshold $K = 100$, $\rho = 1/2$
Output. New locations of the end-vertices of the face f^*

(1) **switch** *Distance Type* **do**
(2) **case** *Point-Point*
(3) Find the closest end-vertex $\mathbf{q}$ of f^* from $\mathbf{p}^*$ by $\mathbf{q} \leftarrow \arg\min_{\mathbf{q}\in\mathscr{V}_{f^*}} \|\mathbf{p}^* - \mathbf{q}\|$;
(4) Compute the gradient $\nabla_{\mathbf{q}} d^2_{\text{pp}}$ using (9);
(5) **for** $k = 0, \ldots, K$ **do**
(6) Update the Hausdorff distance set $\mathscr{D}_1^{\text{pp}}(k)$ using (6) with $\mathbf{q}$ replaced by $\mathbf{q} - \rho^k \nabla_{\mathbf{q}} d^2_{\text{pp}}$;
(7) **if** $\max \mathscr{D}_1^{\text{pp}}(k) < \max \mathscr{D}_1$ **then**
(8) Update $\mathbf{q}$ by $\mathbf{q} \leftarrow \mathbf{q} - \rho^k \nabla_{\mathbf{q}} d^2_{\text{pp}}$;
(9) Break;
(10) **end**
(11) **end**
(12) **if** $\max \mathscr{D}_1^{\text{pp}}(k) \geq \max \mathscr{D}_1, \forall k = 0, \ldots, K$ **then**
End the algorithm;
(13) **case** *Point-Edge*
(14) Find the closest edge of f' from $\mathbf{p}^*$ by $e' \leftarrow \arg\min_{e\in\mathscr{E}_{f^*}} \text{dist}(\mathbf{p}^*, e)$;
(15) Denote $\mathbf{q}_1, \mathbf{q}_2$ to be the end-vertices of e';
(16) Compute the gradient $\nabla_{[\mathbf{q}_1;\mathbf{q}_2]} d^2_{\text{pe}}$ using (16);
(17) **for** $k = 0, \ldots, K$ **do**
(18) Update the Hausdorff distance set $\mathscr{D}_1^{\text{pe}}(k)$ using (6) with $\mathbf{q}_1, \mathbf{q}_2$ replaced by $[\mathbf{q}_1; \mathbf{q}_2] - \rho^k \nabla_{[\mathbf{q}_1;\mathbf{q}_2]} d^2_{\text{pe}}$;
(19) **if** $\max \mathscr{D}_1^{\text{pe}}(k) < \max \mathscr{D}_1$ **then**
(20) Update $\mathbf{q}_1, \mathbf{q}_2$ by $[\mathbf{q}_1; \mathbf{q}_2] \leftarrow [\mathbf{q}_1; \mathbf{q}_2] - \rho^k \nabla_{[\mathbf{q}_1;\mathbf{q}_2]} d^2_{\text{pe}}$;
(21) Break;
(22) **end**
(23) **end**
(24) **if** $\max \mathscr{D}_1^{\text{pe}}(k) \geq \max \mathscr{D}_1, \forall k = 0, \ldots, K$ **then**
End the algorithm;
(25) **case** *Point-Face*
(26) Denote $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ to be the end-vertices of f^*;
(27) Compute the gradient $\nabla_{[\mathbf{q}_1;\mathbf{q}_2;\mathbf{q}_3]} d^2_{\text{pf}}$ using (17);
(28) **for** $k = 0, \ldots, K$ **do**
(29) Update the Hausdorff distance set $\mathscr{D}_1^{\text{pf}}(k)$ using (6) with $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ replaced by $[\mathbf{q}_1; \mathbf{q}_2; \mathbf{q}_3] - \rho^k \nabla_{[\mathbf{q}_1;\mathbf{q}_2;\mathbf{q}_3]} d^2_{\text{pf}}$;
(30) **if** $\max \mathscr{D}_1^{\text{pf}}(k) < \max \mathscr{D}_1$ **then**
(31) Update $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ by $[\mathbf{q}_1; \mathbf{q}_2; \mathbf{q}_3] \leftarrow [\mathbf{q}_1; \mathbf{q}_2; \mathbf{q}_3] - \rho^k \nabla_{[\mathbf{q}_1;\mathbf{q}_2;\mathbf{q}_3]} d^2_{\text{pf}}$;
(32) Break;
(33) **end**
(34) **end**
(35) **if** $\max \mathscr{D}_1^{\text{pf}}(k) \geq \max \mathscr{D}_1, \forall k = 0, \ldots, K$ **then**
End the algorithm;
(36) **endsw**
(37) **endsw**

Algorithm 2: Update the position of end-vertices of target face f^* when $\max \mathscr{D}_1 > \max \mathscr{D}_2$.



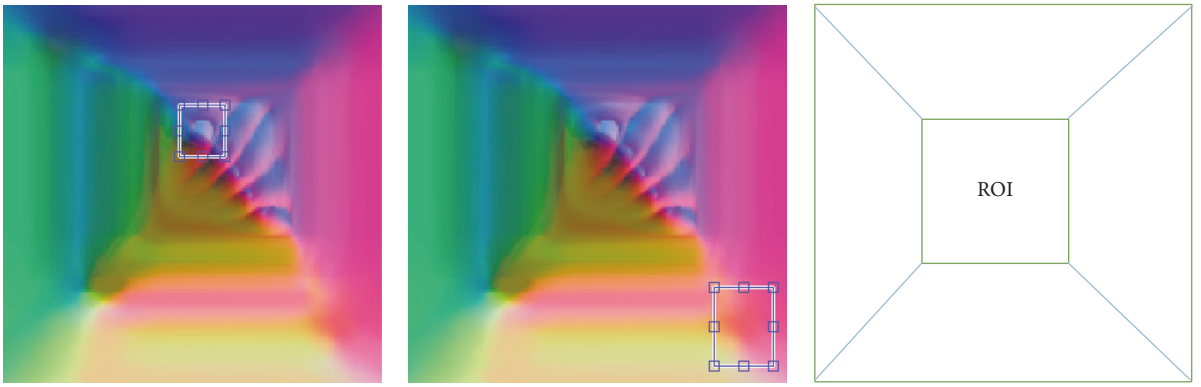
(a) (b) (c)

Figure 5: Geometry image (b) and normal-map image (c) for foot.

Input. A target point $\mathbf{p}^*$, a target face f^*, a mesh $\mathscr{M}$ and stepsize threshold $K = 100$, $\rho = 1/2$
Output. New locations of the end-vertices of the face $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$

(1) **switch** *Distance Type* **do**
(2) **case** *Point-Point*
(3) Compute the gradient $\nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\text{pp}}$ using (10);
(4) **for** $k = 0, \ldots, K$ **do**
(5) Update the Hausdorff distance set $\mathscr{D}_2^{\text{pp}}(k)$ using (6) with $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ replaced by $[\mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3] - \rho^k \nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\text{pp}}$;
(6) **if** $\max \mathscr{D}_2^{\text{pp}}(k) < \max \mathscr{D}_2$ **then**
(7) Update $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ by $[\mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3] \leftarrow [\mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3] - \rho^k \nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\text{pp}}$;
(8) Break;
(9) **end**
(10) **end**
(11) **if** $\max \mathscr{D}_2^{\text{pp}}(k) \geq \max \mathscr{D}_2, \forall k = 0, \ldots, K$ **then**
End the algorithm;
(12) **case** *Point-Edge*
(13) Compute the gradient $\nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\text{pe}}$ using (12);
(14) **for** $k = 0, \ldots, K$ **do**
(15) Update the Hausdorff distance set $\mathscr{D}_2^{\text{pe}}$ using (6) with $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ replaced by $[\mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3] - \rho^k \nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\text{pe}}$;
(16) **if** $\max \mathscr{D}_2^{\text{pe}}(k) < \max \mathscr{D}_2$ **then**
(17) Update $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ by $[\mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3] \leftarrow [\mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3] - \rho^k \nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\text{pe}}$;
(18) Break;
(19) **end**
(20) **end**
(21) **if** $\max \mathscr{D}_2^{\text{pe}}(k) \geq \max \mathscr{D}_2, \forall k = 0, \ldots, K$ **then**
End the algorithm;
(22) **case** *Point-Face*
(23) Compute the gradient $\nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\text{pf}}$ using (15);
(24) **for** $k = 0, \ldots, K$ **do**
(25) Update the Hausdorff distance set $\mathscr{D}_2^{\text{pf}}$ using (6) with $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ replaced by $[\mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3] - \rho^k \nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\text{pf}}$;
(26) **if** $\max \mathscr{D}_2^{\text{pf}}(k) < \max \mathscr{D}_2$ **then**
(27) Update $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ by $[\mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3] \leftarrow [\mathbf{p}_1; \mathbf{p}_2; \mathbf{p}_3] - \rho^k \nabla_{[\mathbf{p}_1;\mathbf{p}_2;\mathbf{p}_3]} d^2_{\text{pf}}$;
(28) Break;
(29) **end**
(30) **end**
(31) **if** $\max \mathscr{D}_2^{\text{pf}}(k) \geq \max \mathscr{D}_2, \forall k = 0, \ldots, K$ **then**
End the algorithm;
(32) **endsw**
(33) **endsw**

Algorithm 3: Update the position of end-vertices of the face containing $\mathbf{p}^*$ when $\max \mathscr{D}_1 \leq \max \mathscr{D}_2$.



(a) (b) (c)

Figure 6: ROI selection for foot. From (a) to (c): ROI selection with more details, ROI selection with less details, and the ROI/non-ROI selection scheme.

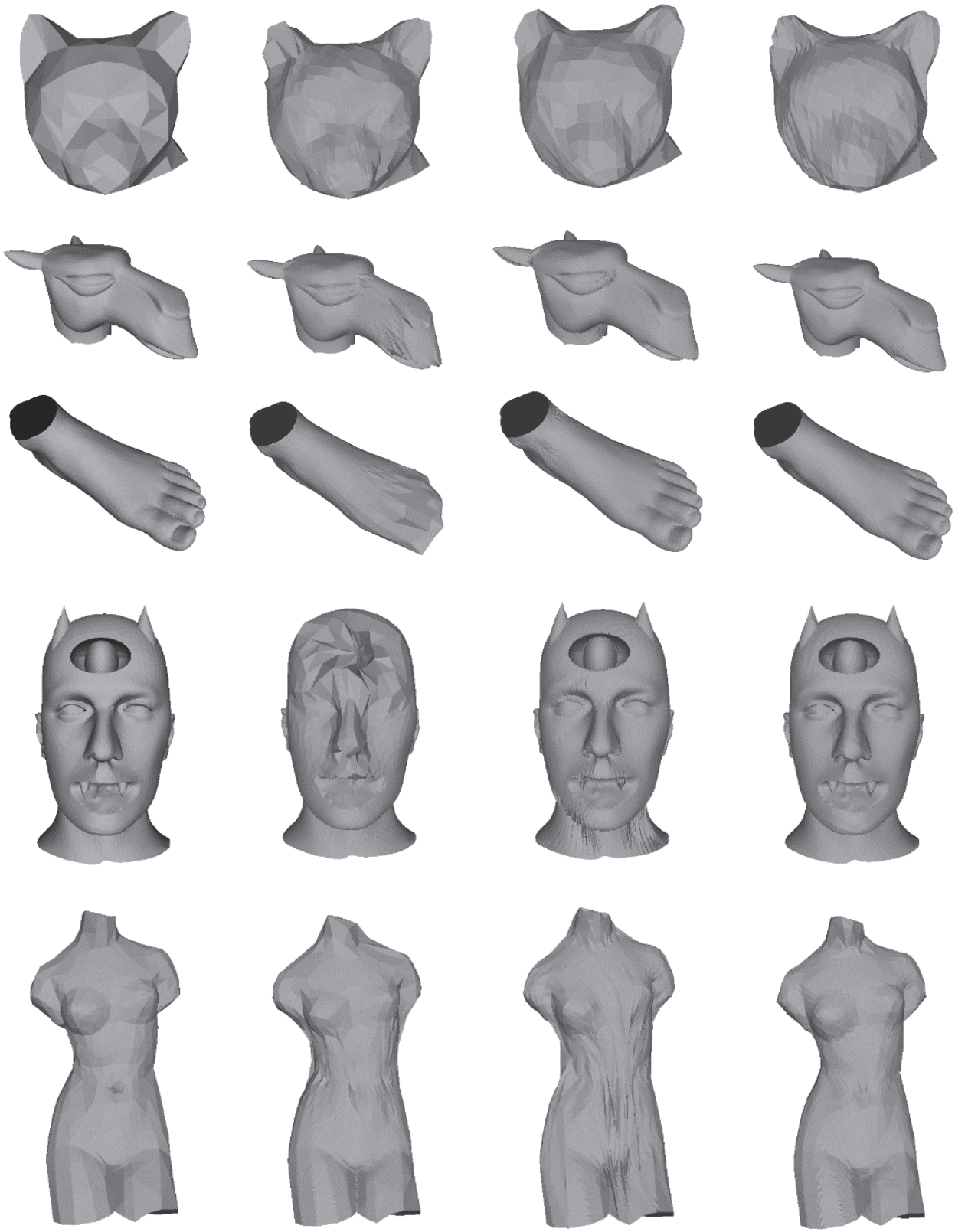


Figure 7: Qualitative results with different models. From (a) to (d): the original model, [4]'s sampling model, the regular sampling model, and our sampling model.

Table 1: Comparative results of PSNR values of reconstructed models.

Model	Face Num	Vertex Num	Sampling Rate	Regular Sampling	PSNR by Zhou et al. [4]	PSNR by Our model	Hausdorff error Before AS	Hausdorff error After AS
Cathead	248	131	45 ∗ 45	47.1377	47.6548	**51.7978**	0.1379	0.0625
Venus	1396	711	70 ∗ 70	50.3154	50.5554	**58.9261**	0.8178	0.0577
Camel	22704	11381	180 ∗ 180	61.4743	56.6976	**71.2418**	0.0102	0.0055
Foot	19974	10010	180 ∗ 180	66.4185	51.8281	**71.3321**	0.0170	0.0129
Mannequin	25888	12977	300 ∗ 300	59.0039	50.2263	**68.4832**	1.6027	0.2300

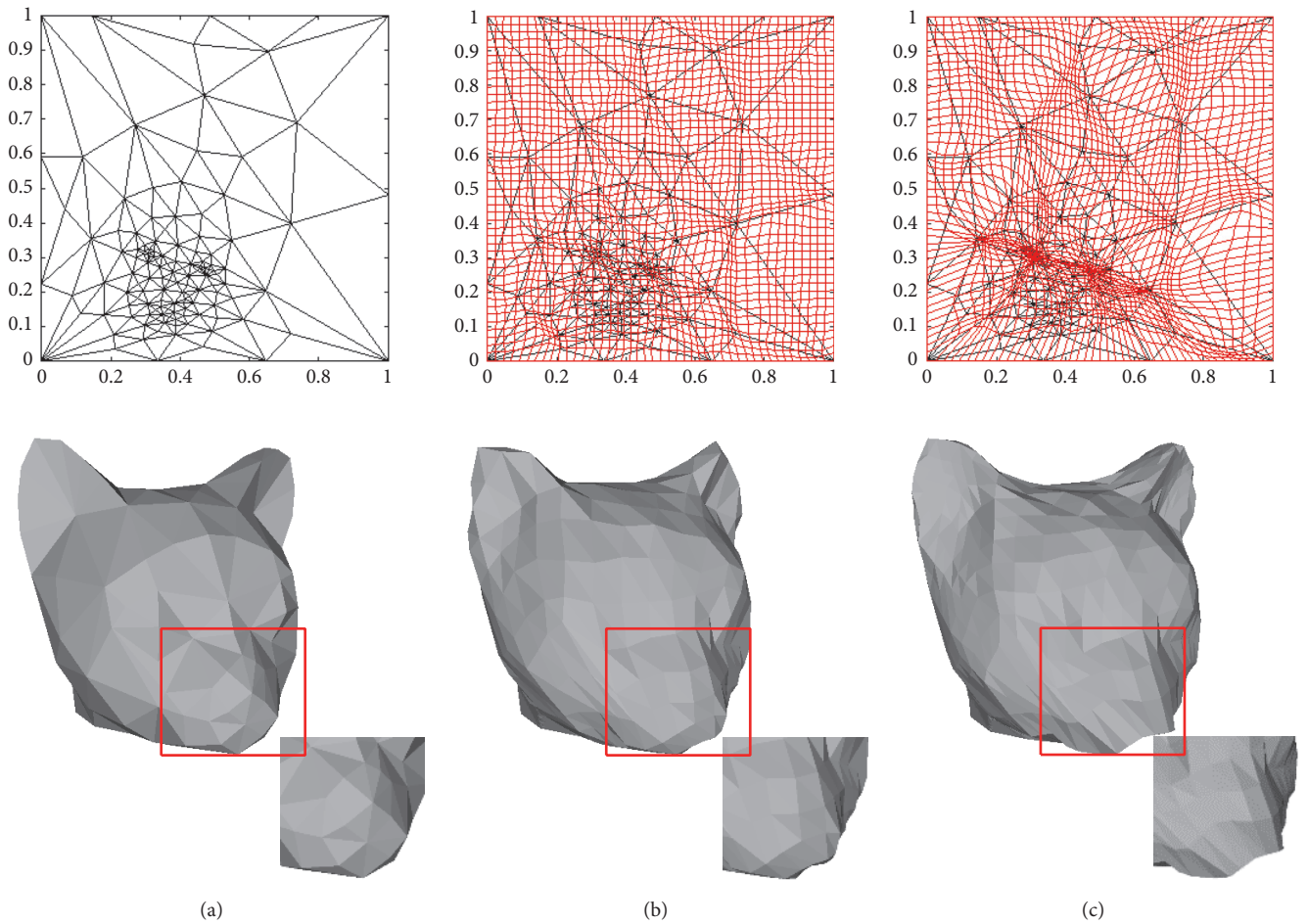
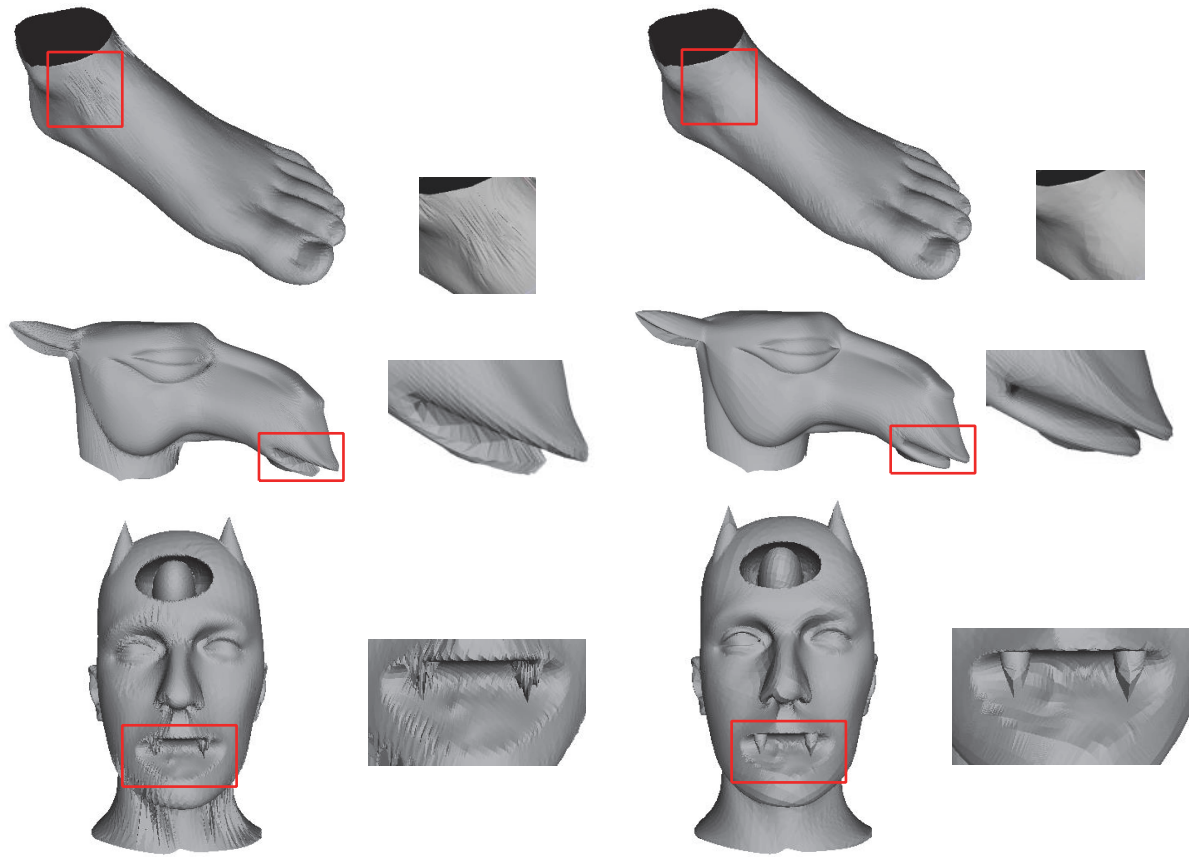


Figure 8: Adaptive sampling for different times of adjustment. From (a) to (c) are original mesh, reconstruction mesh after twice adaptive sampling, and reconstruction mesh after 15 times of adaptive sampling.

right is the original model, [4]'s sampling model, the regular sampling model, and our sampling model.

Table 1 shows details of the five models, such as the number of faces and vertices, as well as the reconstruction PSNRs and Hausdorff errors of models before adaptive sampling and after adaptive sampling. Under the specified sampling rate, we can see that our method achieves the greatest PSNR among all methods. By optimizing the reconstructed mesh, the Hausdorff error is reduced. With the same parametrization method, we see from the comparative results that our sampling method is better than [4]'s sampling method. Because [4]'s method adjusts the sampling points which is based on regular sampling; the effect of the sampling rate is decided by the effect of the sampling rate. Different adjustment times can be obtained for different reconstructed meshes. However, with the increase of adjustment times, the sampling points are distributed within the region of the parametrization domain where more vertices concentrate. In [4], in order to preserve the edge information, detailed regions may not reach the expected reconstruction accuracy. If the detailed regions are up to reconstruction accuracy, the edge information may be lost. Figure 8 shows different reconstructions in terms of



(a) (b)

Figure 9: Qualitative results of regular sampling method (a) and our method (b).

different times of adjustment, where the first row of black mesh is a parametrization mesh and the red mesh is sampling mesh.

Two methods use different parametrization strategies, and the effect of the barycentric mapping method is worse than the geometric-stretch parametrization method. Figure 9 shows comparative results of the regular sampling and our adaptive sampling, where the reconstruction effect within some detailed regions of ours method is better than the regular sampling method.

5.2. Enhanced ROI Geometry Images. Experiments of enhanced ROI geometry images are given by enhancing the number of ROI sampling points before adaptive sampling and after adaptive sampling. We mainly used the barycentric mapping and regular sampling. The sampling rates are $40 * 40$ (cathead), $80 * 80$ (fist), $150 * 150$ (foot), and $300 * 300$ (davidhead).

Figure 10 shows the qualitative results of ROI reconstructions using different sampling schemes, where the second row is the ROI of original meshes, the third row is the ROI reconstructions using regular sampling, and the fourth row is the ROI reconstructions using enhanced ROI adaptive sampling. We can see from the figure that the sampling number of ROI can be increased, which improves the quality of the reconstruction of ROI.

Figure 11 shows the qualitative results of ROI reconstructions using JPEG2000 with non-ROI and ROI codecs, where the second row is the ROI reconstructions using non-ROI JPEG2000 codec, and the third row is the ROI reconstructions using ROI JPEG2000 codec, both of which are under the same compression rate. We can see that the reconstruction meshes using ROI codec keep more detailed information than the reconstruction meshes using non-ROI codec.

Table 2 shows quantitative results of enhanced ROI geometry images and traditional geometry images, which indicates that the Hausdorff error is reduced after our adaptive sampling is applied. Moreover, the more sampling points the ROI has, the higher reconstruction precision of ROI we obtain.

6. Conclusions

We propose two kinds of adaptive geometry images for remeshing triangular meshes. The first scheme, referred to as Hausdorff error driven geometry images, achieves minimum Hasdorff distance between original meshes and reconstructed meshes. The second scheme, referred to as enhanced

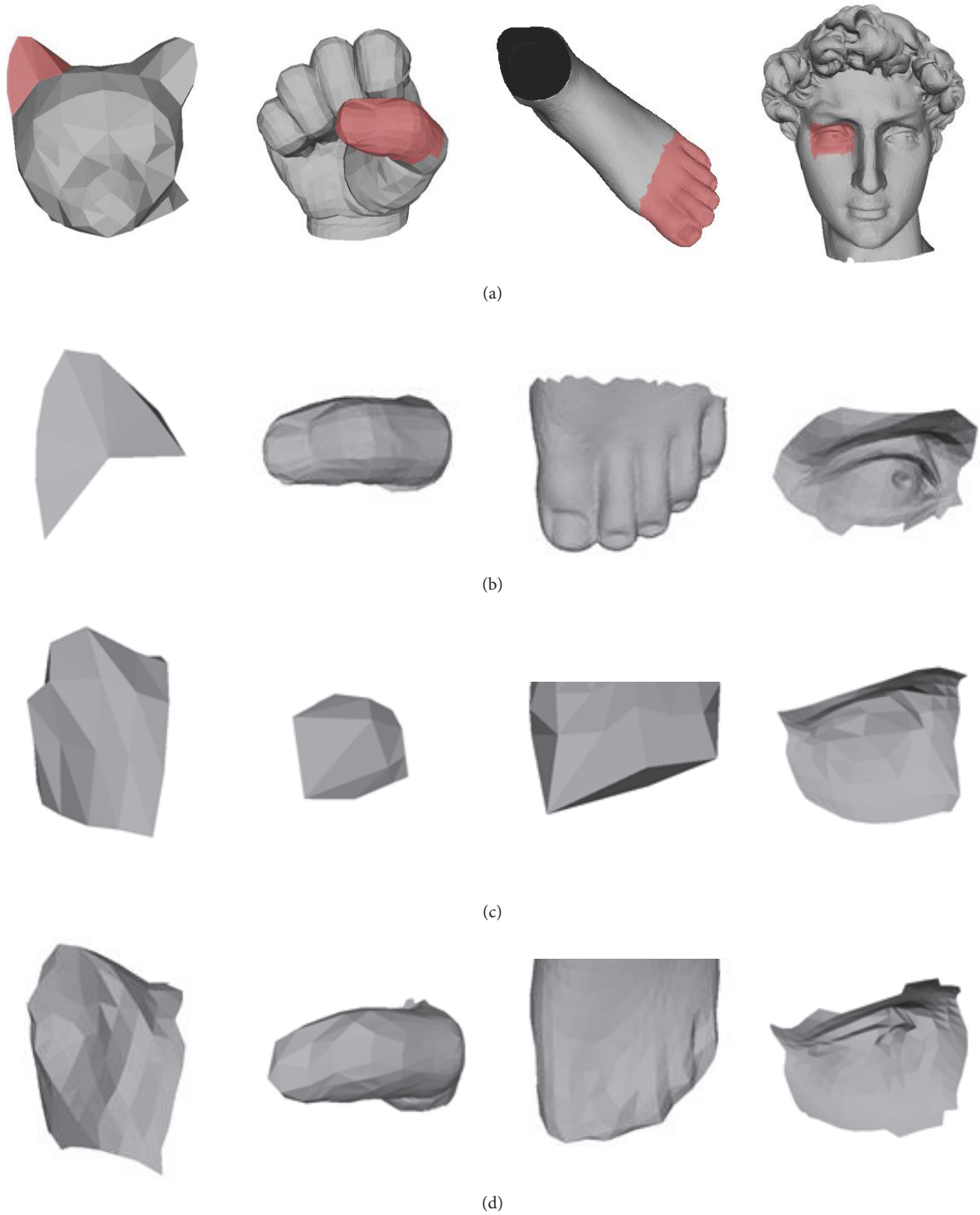


FIGURE 10: Qualitative results of ROI reconstruction. (a) to (d): original meshes, amplified details, reconstructions with traditional sampling rate, and reconstructions with increasing sampling.

ROI geometry images, preserves more details over ROI regions. Experimental results show the effectiveness of our method compared with traditional regular sampling based geometry images. In future work, we shall improve our models by computing Hausdorff distance between meshes using local maximum instead of global one; also we shall consider the parallel implementation of adjusting sample vertices.

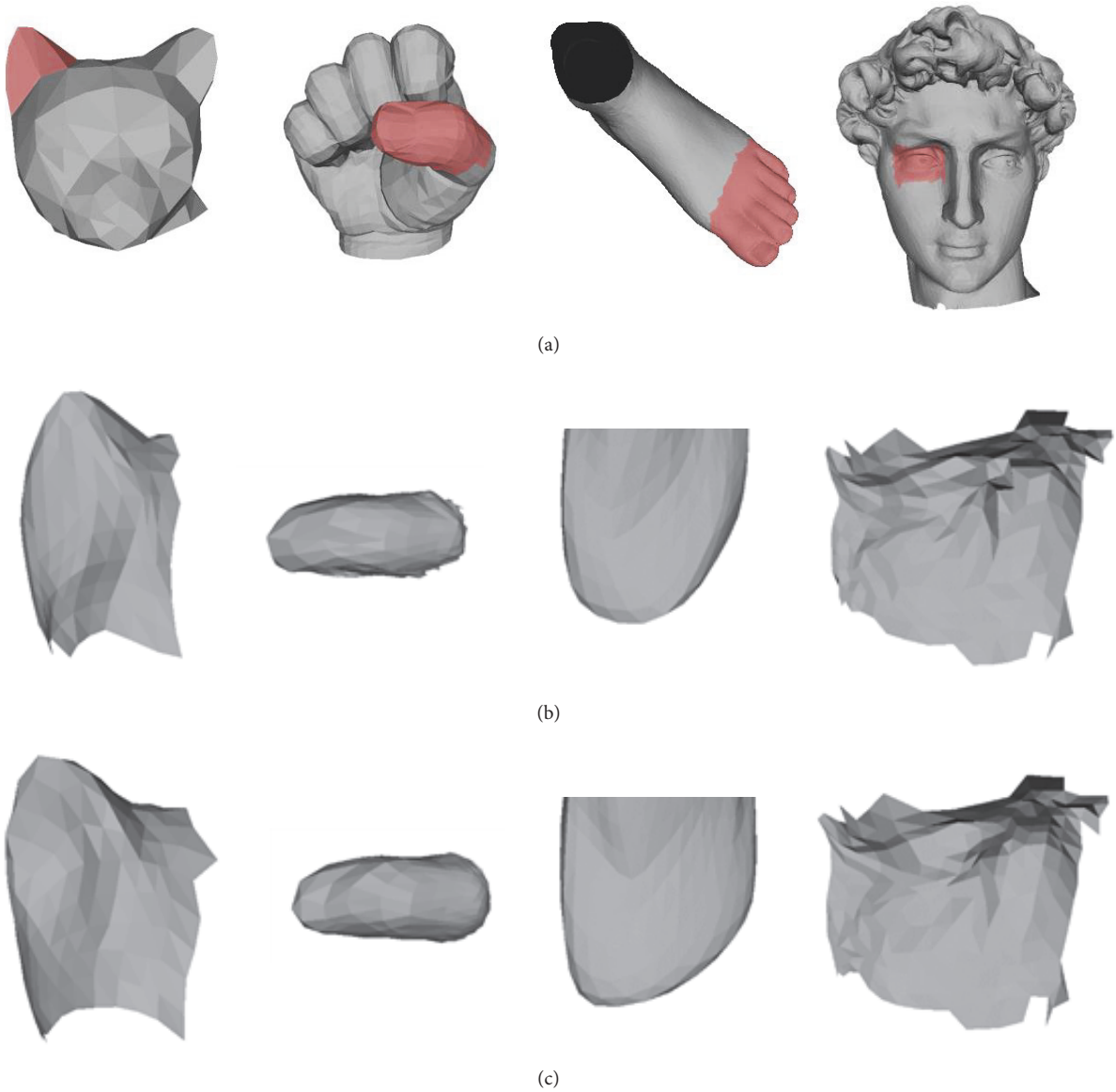


Figure 11: Qualitative results of ROI compression with JPEG2000 codec. (a) to (c): original meshes, reconstructions using non-ROI compression, and reconstructions using ROI compression.

Table 2: Hausdorff error of different models using ROI sampling scheme.

Model	ROI sampling num		Hausdorff error		Rate	Hausdorff error	
	Before AS	After AS	Before AS	After AS		Non-ROI compress	ROI compress
Cathead	30	147	0.0933	0.0657	1.2	1.1077	0.0931
Fist	16	717	0.0963	0.0129	0.7	0.0381	0.0341
Foot	169	1218	0.1557	0.0367	0.1	0.0767	0.0517
Davidhead	110	360	0.0168	0.0103	0.1	0.0252	0.0251

Conflicts of Interest

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

Acknowledgments

This work was supported by the Natural Science Foundation of China under Grants 61370120, 61632006, and

11601151, the Beijing Natural Science Foundation under Grants 4162009 and 4152009, the Beijing Municipal Science and Technology Project under Grants Z171100000517003 and Z151100002115040, the Beijing Transportation Industry Science and Technology Project, the Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality under Grant IDHT20150504, and the Jing-Hua Talents Project of the Beijing University of Technology.

References

[1] X. Gu, S. J. Gortler, and H. Hoppe, "Geometry images," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 355–361, 2002.

[2] E. Praun and H. Hoppe, "Spherical parametrization and remeshing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 340–349, 2003.

[3] M. Gauthier and P. Poulin, "Geometry images of arbitrary genus in the spherical domain," *Computer Graphics Forum*, vol. 28, no. 8, pp. 2201–2215, 2009.

[4] K. Zhou, H. Bao, J. Shi, and Q. Peng, "Geometric signal compression," *Journal of Computer Science and Technology*, vol. 19, no. 5, pp. 596–606, 2004.

[5] M. Gauthier and P. Poulin, "Preserving sharp edges in geometry images," *In Proceedings of Graphics Interface*, pp. 1–6, 2009.

[6] W. Meng, B. Sheng, W. Lv, H. Sun, and E. Wu, "Differential geometry images: remeshing and morphing with local shape preservation," *Visual Computer*, vol. 26, no. 1, pp. 51–62, 2010.

[7] G. Tewari, J. Snyder, P. V. Sander, S. J. Gortler, and H. Hoppe, "Signal-specialized parameterization for piecewise linear reconstruction," in *Proceedings of the the 2004 Eurographics/ACM SIGGRAPH symposium*, p. 55, Nice, France, July 2004.

[8] N. A. Carr, J. Hoberock, K. Crane, and J. Hart, "Rectangular multi-chart geometry images," *Eurographics Symposium on Geometry Processing*, pp. 181–190, 2006.

[9] C. Y. Yao and T. Y. Lee, "Adaptive geometry image," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 948–960, 2008.

[10] W.-W. Feng, B.-U. Kim, Y. Yu, L. Peng, and J. Hart, "Feature-preserving triangular geometry images for level-of-detail representation of static and skinned meshes," *ACM Transactions on Graphics*, vol. 29, no. 2, article no. 11, 2010.

[11] P. Alliez, M. Meyer, and M. Desbrun, "Interactive geometry remeshing," *ACM Transactions on Graphics*, vol. 21, no. 21, pp. 347–354, 2002.

[12] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun, "Anisotropic polygonal remeshing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 485–493, 2003.

[13] S. Dong, S. Kircher, and M. Garland, "Harmonic functions for quadrilateral remeshing of arbitrary manifolds," *Computer Aided Geometric Design*, vol. 22, no. 5, pp. 392–423, 2005.

[14] J. Huang, M. Zhang, J. Ma, X. Liu, L. Kobbelt, and H. Bao, "Spectral quadrangulation with orientation and alignment control," *ACM Transactions on Graphics*, vol. 27, no. 5, article no. 147, 2008.

[15] M. Zhang, J. Huang, X. Liu, and H. Bao, "A wave-based anisotropic quadrangulation method," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 157–166, 2010.

[16] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "MESH: Measuring errors between surfaces using the Hausdorff distance," in *Proceedings of the 2002 IEEE International Conference on Multimedia and Expo, ICME 2002*, pp. 705–708, IEEE, Lausanne, Switzerland, August 2002.

[17] M. Bartoň, I. Hanniel, G. Elber, and M. Kim, "Precise Hausdorff distance computation between polygonal meshes," *Computer Aided Geometric Design*, vol. 27, no. 8, pp. 580–591, 2010.

[18] W. T. Tutte, "How to draw a graph," *Proceedings of the London Mathematical Society. Third Series*, vol. 13, pp. 743–767, 1963.

[19] S. Yoshizawa, A. Belyaev, and H. P. Seidel, "A fast and simple stretch-minimizing mesh parameterization," in *Proceedings of the Shape Modeling Applications*, IEEE, Genova, Italy, June 2004.