

Article

# Quasi-Isometric Mesh Parameterization Using Heat-Based Geodesics and Poisson Surface Fills

Daniel Mejia-Parra <sup>1,2</sup> , Jairo R. Sánchez <sup>2,\*</sup> , Jorge Posada <sup>2</sup> , Oscar Ruiz-Salguero <sup>1</sup>  and Carlos Cadavid <sup>3</sup>

<sup>1</sup> Laboratory of CAD CAM CAE, Universidad EAFIT, Cra 49 no 7-sur-50, 050022 Medellín, Colombia

<sup>2</sup> Vicomtech, Paseo Mikeletegi 57, Parque Científico y Tecnológico de Gipuzcoa, 20009 Donostia/San Sebastián, Spain

<sup>3</sup> Matemáticas y Aplicaciones, Departamento de Ciencias Matemáticas, Universidad EAFIT, Cra 49 no 7-sur-50, 050022 Medellín, Colombia

\* Correspondence: jrsanchez@vicomtech.org

Received: 23 July 2019; Accepted: 13 August 2019; Published: 17 August 2019



**Abstract:** In the context of CAD, CAM, CAE, and reverse engineering, the problem of mesh parameterization is a central process. Mesh parameterization implies the computation of a bijective map  $\phi$  from the original mesh  $M \in \mathbb{R}^3$  to the planar domain  $\phi(M) \in \mathbb{R}^2$ . The mapping may preserve angles, areas, or distances. Distance-preserving parameterizations (i.e., isometries) are obviously attractive. However, geodesic-based isometries present limitations when the mesh has concave or disconnected boundary (i.e., holes). Recent advances in computing geodesic maps using the heat equation in 2-manifolds motivate us to revisit mesh parameterization with geodesic maps. We devise a Poisson surface underlying, extending, and filling the holes of the mesh  $M$ . We compute a near-isometric mapping for quasi-developable meshes by using geodesic maps based on heat propagation. Our method: (1) Precomputes a set of temperature maps (heat kernels) on the mesh; (2) estimates the geodesic distances along the piecewise linear surface by using the temperature maps; and (3) uses multidimensional scaling (MDS) to acquire the 2D coordinates that minimize the difference between geodesic distances on  $M$  and Euclidean distances on  $\mathbb{R}^2$ . This novel heat-geodesic parameterization is successfully tested with several concave and/or punctured surfaces, obtaining bijective low-distortion parameterizations. Failures are registered in nonsegmented, highly nondevelopable meshes (such as seam meshes). These cases are the goal of future endeavors.

**Keywords:** mesh parameterization; geodesic maps; heat transfer analysis; Poisson fills

## 1. Introduction

Mesh parameterization is the process by which a piecewise linear surface (i.e., triangular mesh)  $M$  is mapped with the least possible distortion onto a planar ( $\mathbb{R}^2$ ) region, via a bijective continuous function  $\phi : M \rightarrow \mathbb{R}^2$ . The mesh  $M$  is supposed to be a connected 2-manifold with border (and possibly holes).

Mesh parameterization is central in tool path generation for surface machining, texture mapping, thermoforming of thin layers (metal, leather, plastic, etc.), reverse engineering, finite element remeshing, facial expressions, morphing, etc.

A geodesic curve between two points of a continuous surface is the shortest path *within the surface* joining the two points. The length of such a path is the geodesic (shortest) distance, embedded in the surface, between those two points.

Given any two points  $x_i, x_j \in M$ , ideal parameterizations of such a surface seek to map them to  $\phi(x_i), \phi(x_j) \in \mathbb{R}^2$  so that the geodesic distance between  $x_i$  and  $x_j$  in  $M$  matches the Euclidean 2D distance between  $\phi(x_i)$  and  $\phi(x_j)$  in  $\mathbb{R}^2$ . On the rare occasions that this goal is possible,  $M$  is called a developable surface and  $\phi$  is an isometric map. When the distortion in such distances is small, one qualifies  $M$  as a quasi-developable surface. This case is sufficiently frequent, since large triangular meshes can be segmented with a goal being that the resulting submeshes are quasi-developable or developable.

It is not convenient, when the mesh has holes or concavities in its boundary, to parameterize the mesh via geodesics. The reason is that mesh points being close neighbors in the surface may be far apart via geodesic curves due to mesh gaps or concavities.

Mesh parameterization algorithms can be classified depending on the type of distortion being minimized, as follows: (a) Area-preserving (authalic) algorithms; (b) angle-preserving (conformal) algorithms; and (c) distance-preserving (isometric) algorithms. The remainder of this section discusses a summary of recent mesh parameterization algorithms already present in the literature (Detailed surveys on mesh parameterization algorithms are presented in [1–3]).

### 1.1. Area-Preserving Mesh Parameterization

Area-preserving (authalic) parameterization algorithms rely on the minimization of an area preserving continuous cost function. Zou et al. [4] solved a Lie advection problem on the mesh  $M$ . The gradient of the scalar Lie advection field was then added to an initial parameterization  $\phi_0$  of  $M$ , resulting in an authalic parameterization. Zhao et al. [5] solved an optimal mass transport problem from the mesh  $M$  to its parameterization  $\phi(M)$ . The optimal mass transport poses a partial differential equation in which the parameterization  $\phi(x)$  locally preserves the area at every point  $x \in M$ . Since most optimal transport methods only parameterize meshes with a connected boundary (i.e., without holes), Su et al. [6] introduced additional boundary conditions to allow authalic parameterizations of meshes with more complex topologies.

### 1.2. Angle-Preserving Mesh Parameterization

Angle-preserving (conformal) optimization aims to minimize the parameterization angle distortion. Since this objective can be achieved by collapsing all triangles to a single point, these algorithms rely on constraining the parameterized boundary to a region in  $\mathbb{R}^2$ . Disk geometries are usually used in this context [7,8] however, other geometries such as squared domains have also been proposed [9,10]. The imposed boundary restrictions in these constrained optimization algorithms induce additional distortion in the resulting parameterization.

Free boundary algorithms allow unrestricted boundary parameterizations, producing less distorted maps. Sawhney and Crane [11] presented an algorithm in which the mesh boundary is mapped to  $\mathbb{R}^2$  according to its shape. The parameterized boundary is then used as a constraint to produce a boundary-free parameterization. Starting from a disk parameterization, Bright et al. [12] performed nonlinear optimization while unconstraining the boundary edges of the parameterized mesh. The resulting parameterization allows the (initially mapped to disk) boundary to move freely in the parameter space. Smith and Schaefer [13] presented a mesh parameterization method which introduced a barrier function in its optimization process. The introduced barrier function penalizes nonadjacent triangle overlaps, which guarantees global bijectiveness in the resulting parameterization.

Dimensionality reduction is a superset of mesh parameterization, in which a  $d$ -manifold embedded in a higher dimensional space  $\mathbb{R}^D$ , is parameterized to its corresponding  $\mathbb{R}^d$  domain. As a consequence, these algorithms have been applied successfully in mesh parameterization applications. Such algorithms include Laplacian Eigenmaps [14] and Hessian Locally Linear Embedding [15].

### 1.3. Distance-Preserving Mesh Parameterization

By definition, a distance-preserving (isometric) mapping is a function that simultaneously preserves areas and angles. Mejia et al. [16] presented a nonlinear minimization algorithm for area-angle (isometry) preservation. The minimization function is a linear combination of area and angle distortion, and the weighting parameters for each distortion term are adjusted by the user. The authors pointed out that the algorithm performs better when the angle-preserving term is preponderant over the area-preserving one. Similarly, Yu et al. [17] used polar factorization to introduce area-angle preserving mappings, in which area distortion increases as angle distortion decreases. ARAP (As Rigid As Possible) algorithms divide the parameterization into two optimization steps—local parameterization and global parameterization—performing these steps iteratively one after another until convergence [18–20]. These algorithms produce different bijective parameterizations, but since the weighting parameters are nonoptimized (as they are user-defined), the resulting parameterization is rarely the optimal distance-preserving map.

Ruiz et al. [14] used a dimensionality-reduction geodesic-based algorithm (Isomap) for the computation of isometric parameterization of quasi-developable meshes. However, in addition to the classic nonconvex parameterization problems, such algorithms estimate geodesics using shortest-path graph algorithms which introduce additional distortions in the resulting parameterization. Li et al. [21] presented a geodesic approximation approach in which cutting planes are intersected with the mesh to estimate geodesic paths. This approach solves the problem of nonconvex surfaces and distortion errors induced by mesh graph approximation. However, the method is limited to geodesic curves embedded in  $\mathbb{R}^2$  (i.e., the cutting plane).

### 1.4. Conclusions of the Literature Review

Most of the distance-preserving parameterization algorithms rely on weighting angle vs. area distortion. Such a weighting is defined by the user and drastically changes the resulting parameterization, not providing the optimal isometric mapping. Geodesic-based parameterization algorithms solve this problem by directly minimizing the distance distortion. However, current geodesic-based algorithms rely on shortest-path graph algorithms for geodesics estimation, introducing unnecessary distortion in the resulting parameterization. In addition, estimation of geodesics fails when the surface is nonconvex (such as surfaces with holes and boundary concavities).

To overcome these problems, this manuscript presents a novel heat-geodesic mesh parameterization algorithm. Our algorithm computes a set of temperature maps (heat kernels) on the mesh  $M$ , which are then used to retrieve the set of point-to-point geodesic distances on the discrete mesh. Afterwards, a near-isometric parameterization is obtained by minimizing the difference between the parameterization Euclidean distances and their corresponding geodesics. Since our method relies on finite element mesh discretization to estimate the temperature maps and geodesics, our geodesics estimation is unaffected by mesh-graph topology (as opposed to shortest-path graph algorithms). To overcome the nonconvexity problem, our algorithm uses Poisson surface reconstruction [22], in which the surface holes and boundary concavities are temporarily filled for parameterization. The resulting parameterization for the Poisson reconstructed surface is trimmed with the original boundary of  $M$ , producing a trimmed surface. The implementation and integration of these different techniques provide a novel geodesic-based mesh parameterization algorithm which is (1) unaffected by mesh holes and/or concavities and (2) less sensitive to mesh graph topology.

## 2. Methodology

Consider  $M = (X, \mathcal{T})$  (with point set  $X = \{x_1, x_2, \dots, x_n\}$  and triangle set  $\mathcal{T} = \{t_1, t_2, \dots, t_m\}$ ), a connected 2-manifold with border (and possibly holes) embedded in  $\mathbb{R}^3$ . The problem of mesh parameterization consists of finding a set of points  $\Phi = \{\phi_1, \phi_2, \dots, \phi_n\} \subset \mathbb{R}^2$  such that  $\phi_i$  is the image

of  $x_i \in M$  under the image of a homeomorphism  $\phi : M \rightarrow \mathbb{R}^2$  (i.e.,  $\phi_i = \phi(x_i)$ ). The function  $\phi$  must satisfy the following conditions:

1. **Continuity:** If  $t_i \in \mathcal{T}$  and  $t_j \in \mathcal{T}$  ( $t_i \neq t_j$ ) are adjacent triangles in  $M$ , then  $\phi(t_i)$  and  $\phi(t_j)$  are adjacent in  $\phi(M)$ .
2. **Local bijectiveness:** All mapped triangles  $\phi(M)$  share the same orientation in  $\mathbb{R}^2$ .
3. **Global bijectiveness:** Triangles in  $\phi(M)$  do not overlap each other. This can happen even if all triangles share the same orientation as illustrated in [13].

In addition, define  $g : M \times M \rightarrow \mathbb{R}^+$  as the geodesic distance function in  $M$ . If  $g(x_i, x_j) = \|\phi_i - \phi_j\|$  ( $x_i, x_j \in M$ ), then  $\phi$  is an isometric mapping (i.e.,  $\phi$  preserves geodesic distances) and  $M$  is a developable surface.

As most of the surfaces are not developable in practice, we aim to find the discrete mapping  $\Phi$  that minimizes the difference between these two distances as follows:

$$\begin{aligned} \min_{\Phi} \sum_{i=1}^n \sum_{j=1}^n \|\|\phi_i - \phi_j\| - g(x_i, x_j)\|^2 \\ \text{s.t.} \\ \sum_{i=1}^n \phi_i = \mathbf{0}, \end{aligned} \tag{1}$$

where the restriction  $\sum_{i=1}^n \phi_i = \mathbf{0}$  indicates that the parameterization is mean centered, i.e., the center of mass of the parameterization points is  $\mathbf{0} \in \mathbb{R}^2$ . Such a restriction is introduced to obviate all the possible translations of the same solution.

### 2.1. Algorithm Overview

Our mesh parameterization algorithm aims to retrieve a parameterization  $\phi(M)$  which highly preserves the geodesic distances of  $M$  as Euclidean distances. In order to estimate the geodesic distances  $g$  in  $M$ , the heat-based geodesics algorithm presented in [23] is implemented. Afterwards, we use classic multidimensional scaling to retrieve the 2D coordinates of the parameterization  $\phi$  from the computed geodesic distances. In the case that  $M$  presents holes or concavities, our algorithm applies Poisson surface reconstruction [22] and computes a parameterization on a trimmed surface instead. A summary of the algorithm is presented in Figure 1.

The remainder of this section details the steps to solve Equation (1), and the details of our mesh parameterization algorithm. The algorithm was implemented in MATLAB<sup>®</sup> [24], except for the Poisson Surface Fills which were implemented in C++ using the PCL library [25]. Figures including triangular meshes, scalar fields, vector fields, and 2D parameterizations were produced in MATLAB<sup>®</sup>. Figures including texture maps on the surface were produced using MeshLab [26].

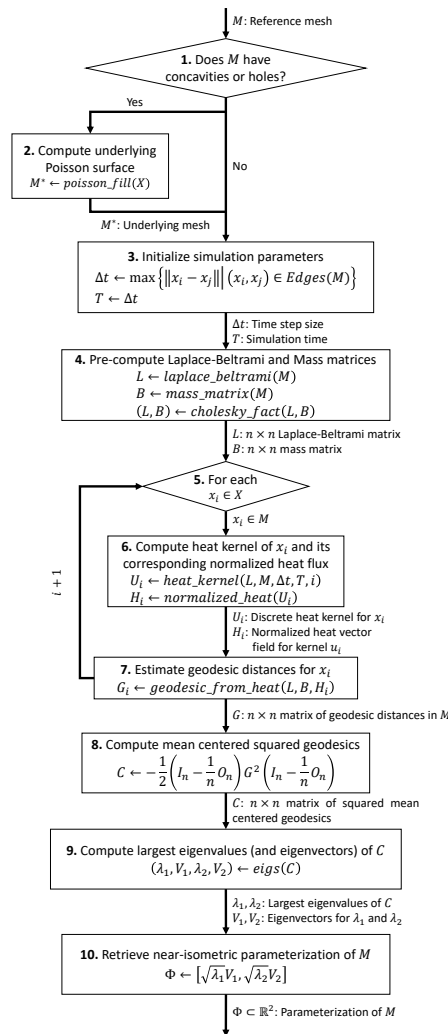


Figure 1. Scheme of our heat-geodesic mesh parameterization algorithm.

### 2.2. Mesh Heat Kernels

A heat kernel of a point  $x_i \in M$  is a function  $u_i : M \times (0, T] \rightarrow \mathbb{R}$  that satisfies the following partial differential equation [27]:

$$\begin{aligned}
 \frac{\partial u_i(x, t)}{\partial t} + \Delta u_i(x, t) &= 0, & x \in M, t \in (0, T], \\
 \frac{\partial u(x, t)}{\partial n} \Big|_{\partial M} &= 0, \\
 u_i(x, 0) &= \delta_{x_i}(x),
 \end{aligned}
 \tag{2}$$

where  $\Delta$  is the Laplace–Beltrami operator,  $u_i$  is the temperature distribution (heat kernel) associated to the source point  $x_i$ ,  $x \in M, t \in (0, T]$  are the spatial and time coordinates, respectively, and  $T > 0$  is the simulation time. The term  $\frac{\partial u(x, t)}{\partial n} \Big|_{\partial M} = 0$  corresponds to the Neumann boundary condition (no boundary heat flux). Finally, the term  $u_i(x, 0) = \delta_{x_i}(x)$  corresponds to Dirac delta initial conditions, i.e.,

$$\begin{aligned}
 \delta_{x_i}(x) &= \begin{cases} \infty & \text{if } x = x_i, \\ 0 & \text{otherwise} \end{cases} \\
 \int_M \delta_{x_i} &= 1.
 \end{aligned}
 \tag{3}$$

The above initial conditions dictate initial infinite temperature at vertex  $x_i$  and 0 everywhere else. After some time  $t > 0$  has passed, heat dissipates from  $x_i$  as illustrated in Figure 2.

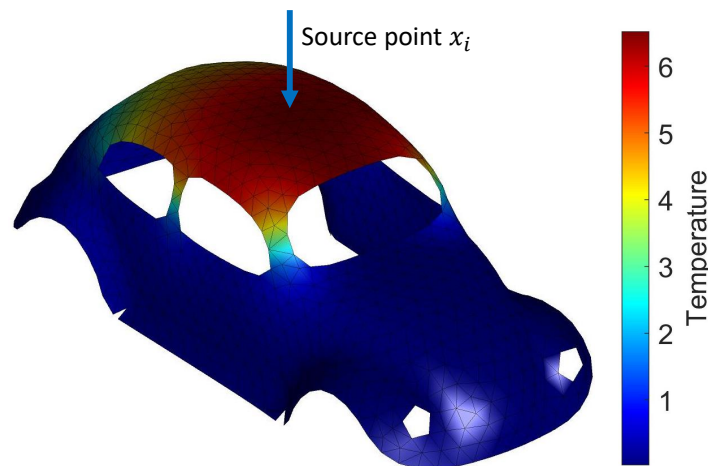


Figure 2. Heat kernel  $u_i(x, t)$  for the vertex source  $x_i$  ( $t > 0$ ). Heat dissipates from  $x_i$ .

Equation (2) can be solved using a Finite Element discretization scheme, as follows:

$$(\Delta t L + B)U_i^{(t+\Delta t)} = BU_i^{(t)}, \tag{4}$$

where  $\Delta t$  is the time step,  $U_i^{(t)} = \{u_{i_1}^{(t)}, u_{i_2}^{(t)}, \dots, u_{i_m}^{(t)}\}$  is the vector of temperatures values ( $u_{ij}^{(t)} = u_i(x_j, t)$ ), and  $L$  and  $B$  are the  $n \times n$  Laplace–Beltrami and mass (sparse and symmetric) matrices, respectively. For a given edge  $e_{ij} = (x_i, x_j)$ , the Laplace–Beltrami matrix  $L$  is defined as follows:

$$L_{ij} = \begin{cases} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2}, & e_{ij} \in Edges(M) \\ -\sum_{k \in E_i^*} L_{ik}, & i = j \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where  $\alpha_{ij}, \beta_{ij} \in (0, \pi)$  are the two opposite angles to the edge  $e_{ij}$ , and  $S_i^* = \{k | e_{ik} \in Edges(M)\}$  is the index set of all incident edges to the vertex  $x_i \in X$ . The entries  $L_{ij}$  of the matrix  $L$  are known as cotangent weights [28].

Similarly, the mass matrix  $B$  is defined as follows:

$$B_{ij} = \begin{cases} \frac{|t_1| + |t_2|}{12}, & t_1, t_2 \in \mathcal{T} \text{ adjacent triangles to } e_{ij} \in Edges(M) \\ \sum_{k \in S_i^*} B_{ik}, & i = j \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where  $t_1, t_2 \in \mathcal{T}$  are the pair of triangles adjacent to the edge  $e_{ij}$  and  $|t_l|$  is the area of the triangle  $t_l$  ( $l = 1, 2$ ).

For each vertex  $x_i \in M$ , Equation (4) is solved for  $U_i$  using a sparse Cholesky linear solver. It is worth noting that for every  $x_i$  and  $t \in (0, T]$ , the matrices  $\Delta t L$  and  $B$  are the same. As a consequence, these matrices are prefactored only once using Cholesky factorization, which speeds up the computation of the heat kernels.

Finally, the simulation parameters  $\Delta t, T$  are chosen according to [23]:

$$\begin{aligned} \Delta t &= \{ \|x_i - x_j\| \mid (x_i, x_j) \in Edges(M) \}, \\ T &= \Delta t, \end{aligned} \tag{7}$$



with  $\Delta t$  computed as the magnitude of the largest edge in  $M$ , and  $T$  is equal to  $\Delta t$ . These values have provided better results in our parameterization experiments than other values.

### 2.3. Heat-Based Geodesic Distance

The vector field  $-\nabla u_i(x, t)$  ( $\nabla$ : gradient operator on  $M$ ) describes the heat flux on  $M$  for the respective heat kernel  $u_i$ . The normalized heat flux vector field  $H_i$  is defined as follows:

$$\vec{H}_i(x, t) = -\frac{\nabla u_i(x, t)}{\|\nabla u_i(x, t)\|}. \tag{8}$$

It is worth noting that the magnitude of the vector field  $\vec{H}_i$  is 1 everywhere ( $\|\vec{H}_i(x, t)\| = 1$ ). In addition, as illustrated in Figure 3, the temperature contours are perpendicular to the geodesic paths from  $x_i \in M$ , and the corresponding vector field points in the same direction as such paths.

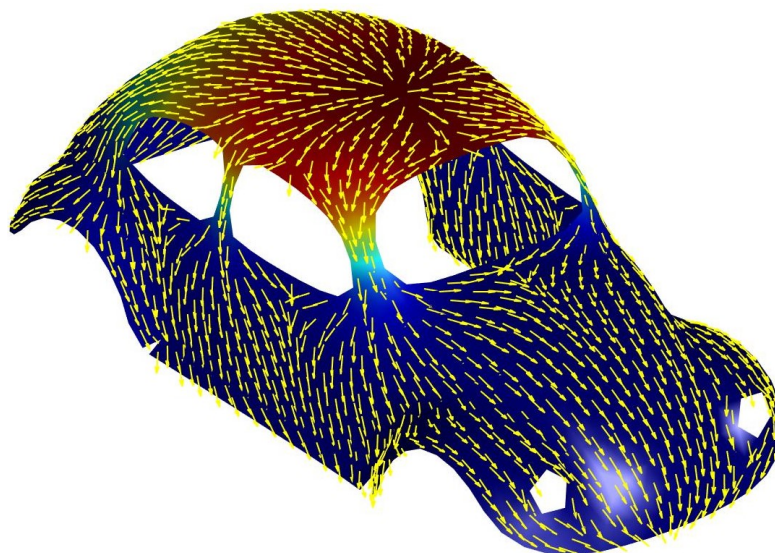
The geodesic field  $g(x_i, x_j)$  satisfies the following differential equation [23]:

$$\Delta g(x_i, x) = \lim_{T \rightarrow 0} \nabla \cdot \vec{H}_i(x, T), \tag{9}$$

where  $\nabla \cdot \vec{H}_i(x, T)$  is the divergence field of the normalized heat flux. Similar to Equation (2), Equation (9) is discretized using the same Finite Element scheme, as follows:

$$LG_i = \lim_{T \rightarrow 0} BK_i^{(T)}, \tag{10}$$

where  $G_i = \{g_{i1}, g_{i2}, \dots, g_{in}\}$  is the vector of geodesic distances  $g_{ij} = g(x_i, x_j)$  and  $K_i^{(T)} = \{k_{i1}^{(T)}, k_{i2}^{(T)}, \dots, k_{in}^{(T)}\}$  is the divergence field of the normalized gradient  $k_{ij}^{(T)} = \nabla \cdot \vec{H}_i(x_j, T)$  [23]. Figure 4 plots the estimated geodesic distance field  $g(x_i, x)$  for the vertex  $x_i$ .



**Figure 3.** Normalized heat flux field  $\vec{H}_i(x, t)$ . The vector field is normalized and points in the direction of the geodesic paths from  $x_i \in M$ .

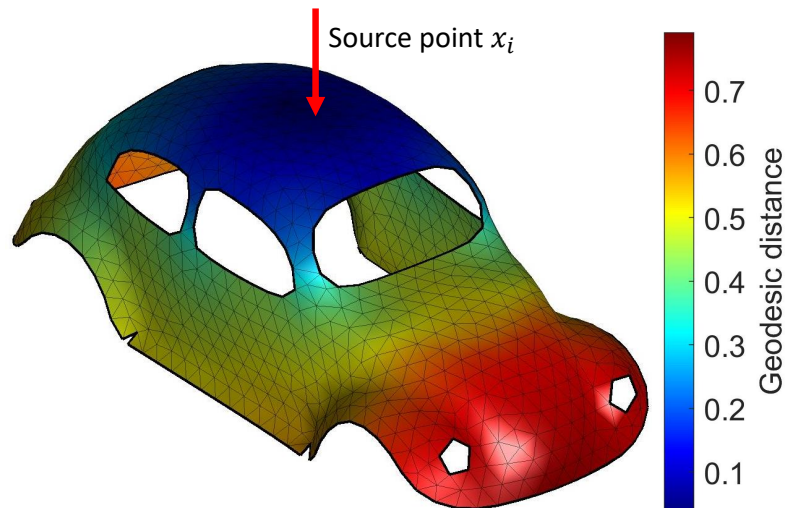


Figure 4. Geodesic field  $g(x_i, x)$  for the vertex  $x_i$ , computed from its respective heat kernel  $u_i(x, T)$ .

### 2.4. Multidimensional Scaling (MDS)

After the geodesic field  $g_{ij} = g(x_i, x_j)$  has been estimated on  $M$ , the minimization problem in Equation (1) can be solved. Classic multidimensional scaling poses an equivalent minimization problem [29]:

$$\min_{\Phi} \sum_{ij} \left[ \frac{1}{n} \left( \sum_{kl} g_{kl}^2 \right) - g_{ij}^2 - 2\phi_i \cdot \phi_j \right]^2. \tag{11}$$

Let  $C$  be the symmetric, semidefinite positive matrix whose entries contain the mean centered squared geodesics (i.e.,  $C_{ij} = -\frac{1}{2}[g_{ij}^2 - \frac{1}{n}(\sum_{kl} g_{kl}^2)]$ ). In matrix form,  $C$  is equivalent to

$$C = -\frac{1}{2} \left( I_n - \frac{1}{n} J_n \right) G^2 \left( I_n - \frac{1}{n} J_n \right), \tag{12}$$

where  $I_n, J_n$  are the  $n \times n$  identity and all-ones matrices, respectively, and  $G^2$  is the  $n \times n$  symmetric matrix whose entries contain the squared geodesic distances in  $M$ , i.e.,  $G_{ij}^2 = g_{ij}^2$ . Then, Equation (11) becomes:

$$\min_{\Phi} \|C - \Phi\Phi^T\|_F^2 \tag{13}$$

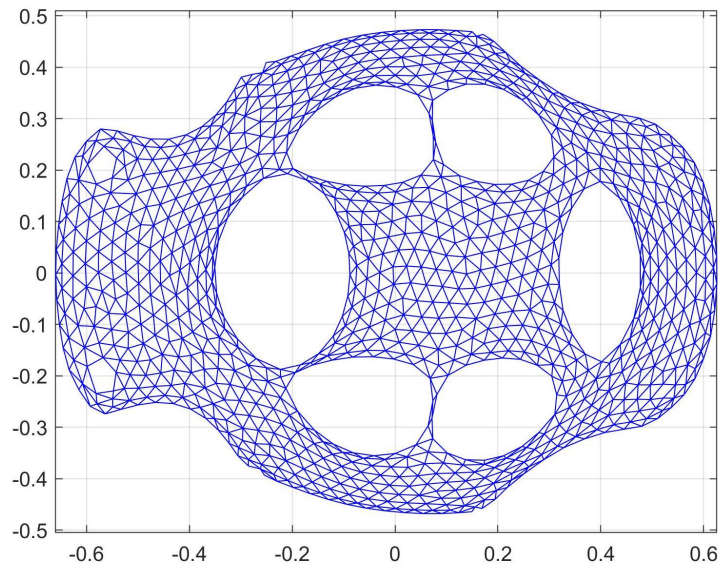
with  $\|A\|_F^2 = \sum_{ij} A_{ij}^2$  as the (squared) Frobenius norm of  $A$ .

Finally, Equation (13) can be solved by an eigendecomposition of  $C$  as follows [29]. Let  $\lambda_1$  and  $\lambda_2$  be the largest positive eigenvalues of  $C$ , with respective  $n \times 1$  eigenvectors  $V_1$  and  $V_2$ . The near-isometric parameterization of  $M$  becomes

$$\Phi = [\sqrt{\lambda_1}V_1, \sqrt{\lambda_2}V_2], \tag{14}$$

where  $\sqrt{\lambda_1}V_1$  are the discrete  $u$ -coordinates and  $\sqrt{\lambda_2}V_2$  are the discrete  $v$ -coordinates of the parameterization  $\Phi = \{\phi_1, \phi_2, \dots, \phi_n\} \subset \mathbb{R}^2$ . Figure 5 plots the resulting parameterization using MDS on the estimated geodesic fields  $G$ .

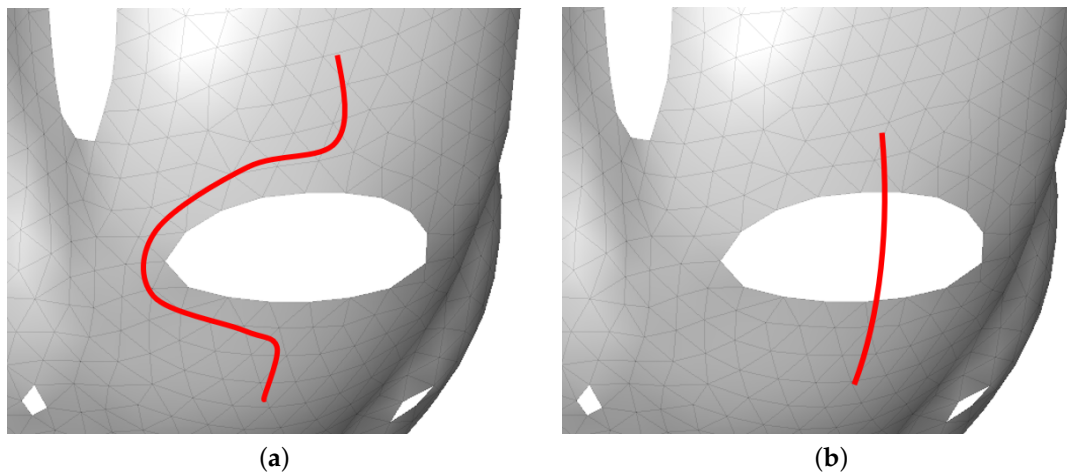




**Figure 5.** Multidimensional scaling (MDS) parameterization  $\Phi = [\sqrt{\lambda_1}V_1, \sqrt{\lambda_2}V_2]$  from the estimated geodesic distances.

2.5. Poisson Mesh Reconstruction

In the case that  $M$  is nonconvex (i.e., it has holes or concavities), we seek to compute an underlying extending surface  $M^*$ . Such a surface contains the points in  $M$ , and fills the holes and concavities by extending  $M$  in such areas ( $M \subset M^*$ ). As an example, a geodesic path in a nonconvex  $M$  circles a given hole (Figure 6a). On the other hand, the same geodesic path in the extended surface  $M^*$  crosses through the hole (Figure 6b).



**Figure 6.** Our algorithm computes an underlying Poisson surface  $M^*$  to fix the geodesic paths on nonconvex mesh  $M$ . (a) Geodesic path on raw mesh  $M$ . (b) Geodesic path on  $M$  with the help of underlying Poisson surface.

To compute the surface  $M^*$ , our parameterization algorithm uses Poisson surface reconstruction [22] from the PCL library [25]. Define  $\chi : \mathbb{R}^3 \rightarrow \mathbb{R}$  as an indicator function such that  $\chi(x) = 1$  if  $x \in \mathbb{R}^3$  is “inside” the solid defined by  $M^*$ ; and  $\chi(x) = 0$  if  $x$  is “outside” such solid. The surface  $M^*$  is composed by the points in between, as follows [22]:

$$M^* = \left\{ p \in \mathbb{R}^3 \mid \chi(p) = \frac{1}{n} \sum_{x \in M} \chi(x) \right\}. \tag{15}$$

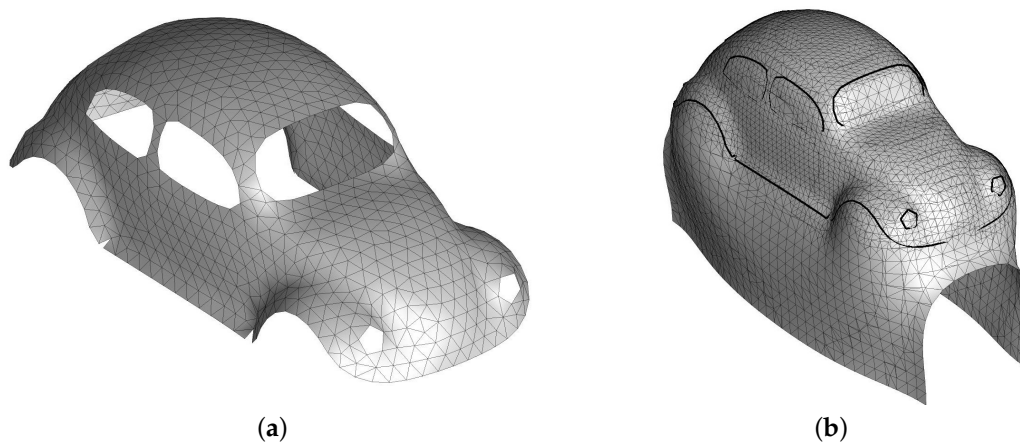
The indicator function  $\chi$  is computed by solving the following partial differential equation in  $\mathbb{R}^3$  [22]:

$$\Delta\chi(x) = \nabla \cdot \vec{N}(x), \tag{16}$$

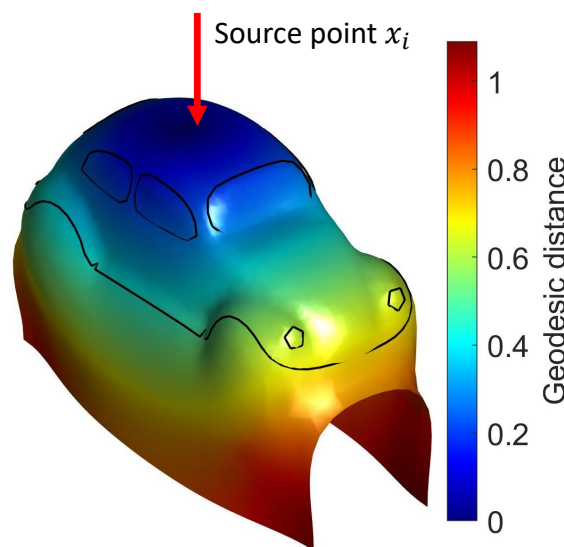
where  $\Delta$  and  $\nabla$  are the  $\mathbb{R}^3$  Euclidean Laplacian and gradient operators, respectively. It is worth noting that these Laplacian and gradient operators are different from the 2-manifold version presented in Sections 2.2 and 2.3.  $\vec{N} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  is a vector field in  $\mathbb{R}^3$  whose value  $\vec{N}(x)$  is the normal vector to the original surface  $M$  if  $x \in M$ , and  $\vec{N}(x) = 0$  everywhere else.

To solve Equation (16), the PCL library uses a hierarchical 3D spatial discretization and a Finite Differences approach [25].

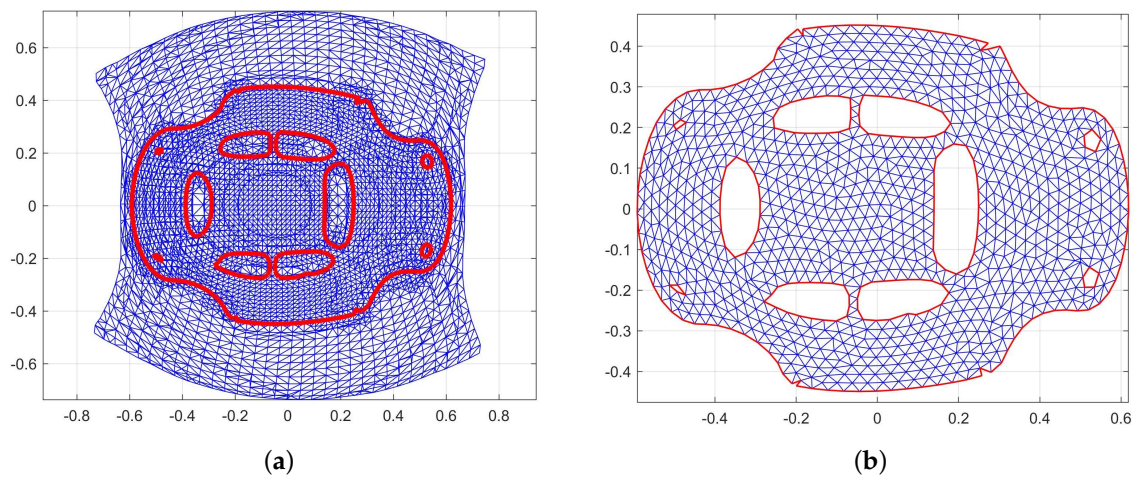
Figure 7 plots the Poisson surface filling  $M^*$  for a given nonconvex mesh  $M$ . The resulting geodesic field (Figure 8) is distributed along the original mesh  $M$  and its extents  $M^* - M$ . The corresponding parameterization of the underlying Poisson surface  $M^*$  (Figure 9a) is finally trimmed in order to retrieve the final parameterization  $\Phi$  of  $M$  (Figure 9b). Figure 10 plots the chessboard texture maps for both parameterization without Poisson filling (Figure 10a) and parameterization with Poisson surface filling (Figure 10b). As illustrated, using Poisson filling reduces parameterization distortions close to mesh holes and boundary concavities.



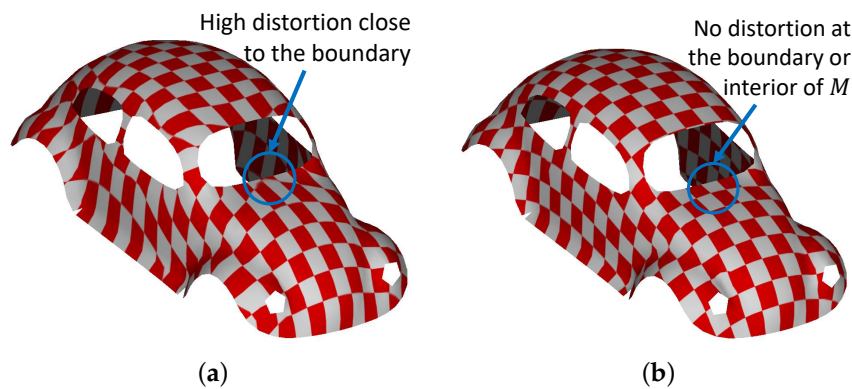
**Figure 7.** Raw mesh  $M$  and its underlying Poisson surface approximation  $M^*$ . (a) Original mesh  $M$ . (b) Poisson surface  $M^*$  underlying the raw mesh  $M^*$ .



**Figure 8.** Geodesic distance estimation on the Poisson surface  $M^*$  approximating raw mesh  $M$ .



**Figure 9.** Parameterization of the Poisson surface  $M^*$  and its corresponding trimmed parameterization  $\Phi$ . (a) Mesh parameterization of the Poisson surface  $M^*$ . (b) Trimmed parameterization  $\Phi$ .



**Figure 10.** Chessboard texture maps from our heat-geodesic based parameterization. (a) Texture on raw mesh  $M$ , distorted at holes and boundary concavities. (b) Texture using underlying Poisson surface  $M^*$ , undistorted.

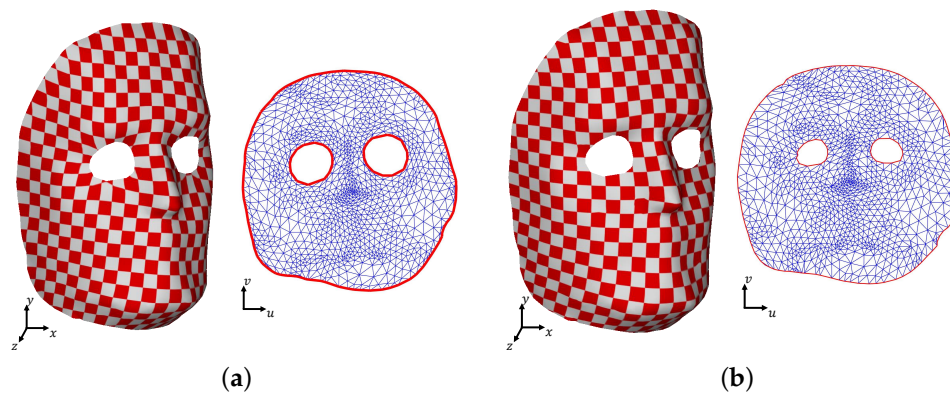
### 3. Results

To test our mesh parameterization algorithm, we run tests with several parameterizable surfaces. Section 3.1 presents a comparison of our mesh parameterization algorithm without Poisson surface filling vs. Poisson surface filling, for quasi-developable nonconvex meshes. Section 3.2 presents parameterization results for some challenging, strongly nondevelopable data sets. Finally, Section 3.3 presents the application of our parameterization algorithm for the reverse engineering of a scanned cow vertebra.

#### 3.1. Nonfilling vs. Poisson Filling Parameterization

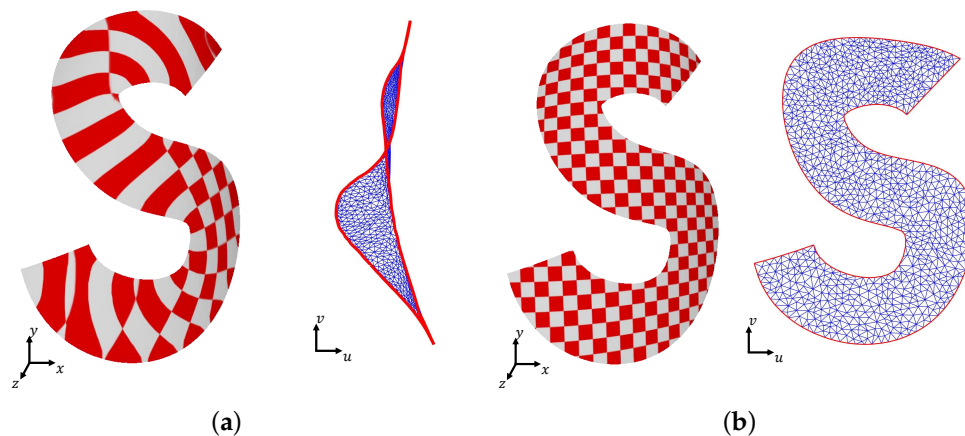
Figure 11 plots the parameterization results (2D  $\Phi$  coordinates and 3D texture map) for the Mask data set. Without using Poisson surface reconstruction, the resulting parameterization is bijective with relatively low distortion, except for the eye holes (Figure 11a). However, such a parameterization is improved by applying the Poisson reconstruction, reducing the distortion close to mesh holes and boundary concavities (Figure 11b).





**Figure 11.** Data set Mask. Hole-distorted and undistorted heat-geodesic parameterizations. (a) Parameterization of  $M$ . 3D texture map (left) and 2D  $\Phi$  coordinates (right). Distorted at holes and concavities. (b) Undistorted parameterization with underlying surface  $M^*$ . 3D texture map (left) and 2D  $\Phi$  coordinates (right).

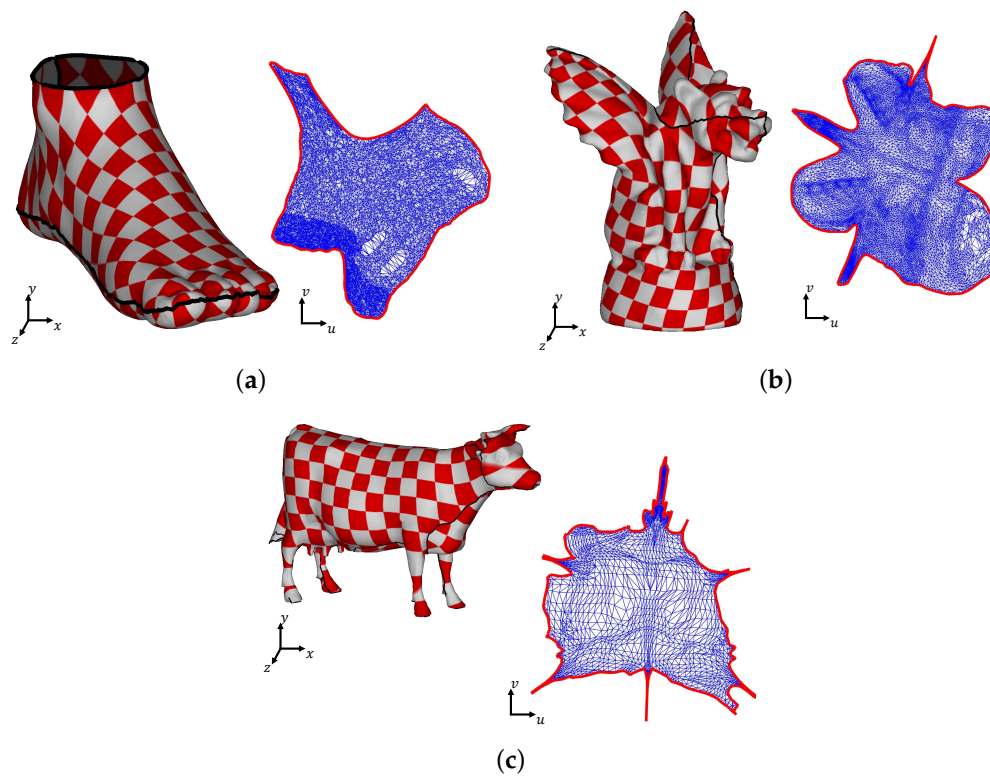
A more extreme case is illustrated in Figure 12, with the S-trimmed-on-cone data set from [14]. Since the S shape is trimmed on a cone,  $M$  is a fully developable surface. Yet, the heat-geodesic parameterization on  $M$  fails to compute a bijective parameterization (Figure 12a). Application of Poisson (extended underlying surface  $M^*$ ) filling (Figure 12b) solves this issue, resulting in a nondistorted bijective parameterization. This case illustrates (a) the vulnerability of geodesic-based parameterizations in the presence of mesh holes or concavities at mesh borders, (b) the capacity of the underlying Poisson extended surface to prevent (a), (c) the natural manner in which geodesic curves isometrically parameterize a developable surface.



**Figure 12.** Data set S-trimmed-on-cone. Nonbijective parameterization using raw mesh  $M$ . Bijective isometric parameterization using underlying Poisson mesh  $M^*$ . (a) Nonbijective parameterization of  $M$ . 3D texture map (left) and 2D  $\Phi$  coordinates (right). (b) Undistorted and bijective parameterization with underlying surface  $M^*$ . 3D texture map (left) and 2D  $\Phi$  coordinates (right).

### 3.2. Parameterization of Strongly Nondevelopable Meshes

In this section, the public data sets Foot, Gargoyle, and Cow are parameterized with our heat-based geodesics algorithm. These benchmark datasets contain an artificially introduced border [30], which allows their parameterization. Figure 13a,b plot the parameterization results for the seam Foot and Gargoyle, respectively. The resulting parameterization is bijective, with some noticeable distortion (e.g., in the Gargoyle head). Figure 13c plots our parameterization results for the seamed Cow, which is not bijective in the head, legs, and tail.

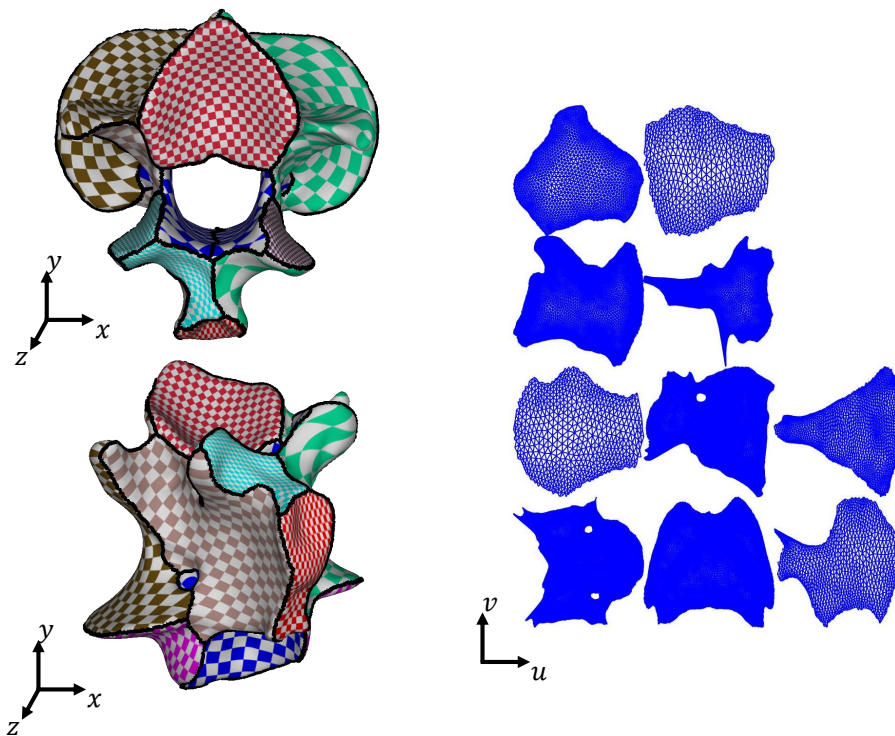


**Figure 13.** Heat-geodesic parameterization of seam meshes [30]. The strong nondevelopability of the meshes produces high parameterization distortions and in some cases nonbijectiveness. (a) Data set Foot. Bijective parameterization. (b) Data set Gargoyle. Bijective parameterization. (c) Data set Cow. Nonbijective parameterization near the legs and tail.

It is worth noting that although parameterizable, these benchmark data sets are strongly nondevelopable without a proper mesh presegmentation. This fact is illustrated in the next section.

### 3.3. Reverse Engineering of Cow Vertebra

For this section, the vertebra of a cow is scanned using a 3D optical scanner. The scanned mesh is closed and therefore accepts no (bijective) parameterization. The closed mesh is segmented into quasi-developable meshes using a heat-based segmentation approach [31]. Figure 14 plots the parameterization results for the segmented mesh. Each submesh bijective parameterization presents low distortion, enabling further reverse engineering operations such as NURBs reparameterization [14], finite element analysis, structural optimization, and/or dimensional inspection [31].



**Figure 14.** Cow Vertebra data set. Undistorted parameterization using heat-based geodesic maps. Segmentation by Mejia et al. [31].

#### 4. Conclusions

This article presents the implementation of a novel application of heat propagation in 2-manifolds used for mesh parameterization. The temperature contours for the heat kernels computed on the mesh are perpendicular at each point to the geodesic curves in the surface. This principle permits determination of geodesic maps in the mesh and in particular vertex-to-vertex geodesic distances. Although Finite Element methods produce better results as the mesh resolution increases (higher polygon count), the geodesics estimation method still produces robust results for low polygon count meshes as each geodesic path traverses across the mesh faces (contrary to graph algorithms which traverse the mesh graph). A quasi-isometric bijective function (i.e., the parameterization) is synthesized to map the 3D mesh to the parameter (2D) space. This parameterization is near isometric in the sense that geodesic distance on the mesh between any two points on the mesh approximates the Euclidean distance between their images in the parametric space. This approach is obviously limited to meshes that are quasi-developable or developable, since mesh developability is necessary for the existence of an isometric parameterization for it.

Our approach circumvents the weakness of geodesic maps in the presence of mesh interruptions (boundary concavities and mesh holes) by devising an underlying continuous Poisson surface that approximates the input mesh but contains no such interruptions. This underlying surface allows for geodesic maps to be computed on it, which are also valid in the input mesh. In this manner, the parameterization computed for the Poisson surface is valid for the input mesh. Finally, the boundaries of the input mesh are explicitly marked on the parameterization to obtain a *trimmed surface* or *FACE* in the Boundary Representation jargon.

Future work is required in these aspects—(a) to eliminate redundant computation that is present in the construction of heat-based geodesic maps and (b) to use failures in the bijectiveness of the computed parameterizations to force mesh segmentation, when the input mesh is strongly nondevelopable.



**Author Contributions:** D.M.-P., and J.R.S. conceived, designed, and implemented the Computational Geometry algorithm. J.P., O.R.-S., and C.C. supervised the Heat Transfer, Computational Geometry, and Applications aspects of this research. All the authors contributed in the writing/reviewing of this manuscript.

**Funding:** This research received no external funding.

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

## Abbreviations

The following abbreviations are used in this manuscript:

- MDS: Multidimensional scaling.
- $M$ : Triangular mesh  $M = (X, \mathcal{T})$  of a connected 2-manifold with border (and possibly holes), embedded in  $\mathbb{R}^3$ .  $M$  is represented as a set of points  $X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^3$  and a set of oriented triangles  $\mathcal{T} = \{t_1, t_2, \dots, t_m\}$ .
- $M^*$ : Triangular mesh  $M^* \subset \mathbb{R}^3$  of a connected 2-manifold with border (but without holes).  $M^*$  is an extension of  $M$  ( $M \subset M^*$ ).
- $\Delta t, T$ : Time step size  $\Delta t$  and total simulation time  $T$  parameters defined for the heat transfer simulations on  $M$ .
- $\phi$ : Continuous and bijective function (homeomorphism)  $\phi : M \rightarrow \mathbb{R}^2$  that maps  $M$  to a planar region in  $\mathbb{R}^2$ . In this manuscript,  $\phi$  is a nearly-isometric map (i.e., highly preserves distances).
- $g$ : Geodesic distance function  $g : M \times M \rightarrow \mathbb{R}^+$  defined on  $M$ .  $g_{ij} = g(x_i, x_j)$ .
- $\delta_{x_i}$ : Dirac delta (temperature) distribution  $\delta_{x_i} : M \rightarrow [0, \infty]$  associated to the source point  $x_i$ , such that the temperature at  $x_i$  is infinite and 0 everywhere else.
- $u_i$ : Heat kernel function  $u_i : M \times (0, T] \rightarrow \mathbb{R}$  associated to vertex  $x_i \in M$ .  $u_i(x, t)$  is the temperature at the point  $x, t$ , due to an initial infinite-heat point-source  $\delta_{x_i}$ .
- $\chi$ : Continuous function  $\chi : \mathbb{R}^3 \rightarrow \mathbb{R}$  that indicates if a point  $p \in \mathbb{R}^3$  is “inside” ( $\chi(p) = 1$ ), “outside” ( $\chi(p) = 0$ ), or “in-between” ( $\chi(p) = \frac{1}{n} \sum_{x \in M} \chi(x)$ ) a solid defined by  $M^*$ .
- $\vec{H}_i$ : Vector field  $\vec{H}_i : M \rightarrow \mathbb{R}^3$  defined on  $M$ .  $\vec{H}_i(x, t) = -\frac{\nabla u_i(x, t)}{\|\nabla u_i(x, t)\|}$  is the normalized heat flux for the heat kernel  $u_i$ .  $\|\vec{H}_i(x, t)\| = 1$ .
- $\vec{N}$ : Vector field  $\vec{N} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  defined on  $\mathbb{R}^3$ . For every  $x \in M$ ,  $\vec{N}(x)$  is the normal vector to the surface  $M$  at  $x$ . For any  $x \notin M$ ,  $\vec{N}(x) = 0$ .
- $\Phi$ : Discrete parameterization  $\Phi = \{\phi_1, \phi_2, \dots, \phi_n\} \subset \mathbb{R}^2$  of  $M$  such that  $\phi_i = \phi(x_i)$ .
- $L, B$ :  $n \times n$  Laplace–Beltrami  $L$  and mass  $B$  matrices, respectively.  $L$  and  $B$  are the sparse and symmetric matrices that approximate the Laplace–Beltrami operator on the triangular mesh  $M$ .
- $U^{(t)}$ :  $n \times n$  matrix with the discrete heat kernels maps associated to each vertex in  $M$ , i.e.,  $U_{ij}^{(t)} = u_i(x_j, t)$ .
- $K^{(t)}$ :  $n \times n$  matrix of the divergence fields of all heat kernels  $U^{(t)}$ .  $K_{ij}^{(t)} = \nabla \cdot \vec{H}_i(x_j, t)$ .
- $G, G^2$ :  $n \times n$  matrices of geodesic and squared geodesic distances, respectively, defined on  $M$ .  $G_{ij} = g(x_i, x_j)$ ,  $G_{ij}^2 = g(x_i, x_j)^2$ .
- $C$ :  $n \times n$  symmetric matrix whose entries contain the mean centered squared geodesic distances in  $M$ , i.e.,  $C_{ij} = -\frac{1}{2}[g_{ij}^2 - \frac{1}{n}(\sum_{kl} g_{kl}^2)]$ .  $C$  is semidefinite positive.
- $I_n, J_n$ :  $n \times n$  identity  $I_n$  and all-ones  $J_n$  matrices.
- $\lambda_1, V_1$ : Largest (positive) eigenvalue  $\lambda_1$  of the semidefinite positive matrix  $C$  and its corresponding  $n \times 1$  eigenvector  $V_1$ .  $\sqrt{\lambda_1} V_1$  are the discrete  $u$ -coordinates of the parameterization  $\Phi \subset \mathbb{R}^2$ .
- $\lambda_2, V_2$ : Second largest (positive) eigenvalue  $\lambda_2$  of the semidefinite positive matrix  $C$  and its corresponding  $n \times 1$  eigenvector  $V_2$ .  $\sqrt{\lambda_2} V_2$  are the discrete  $v$ -coordinates of the parameterization  $\Phi \subset \mathbb{R}^2$ .

## References

- Floater, M.S.; Hormann, K. Surface Parameterization: A Tutorial and Survey. In *Advances in Multiresolution for Geometric Modelling*; Dodgson, N.A., Floater, M.S., Sabin, M.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 157–186.
- Sheffer, A.; Praun, E.; Rose, K. Mesh Parameterization Methods and Their Applications. *Found. Trends Comput. Graph. Vis.* **2006**, *2*, 105–171. doi:10.1561/0600000011. [[CrossRef](#)]

3. Hormann, K.; Polthier, K.; Sheffer, A. Mesh Parameterization: Theory and Practice. In Proceedings of the ACM SIGGRAPH ASIA 2008 Courses, SIGGRAPH Asia'08, Singapore, 10–13 December 2008; ACM: New York, NY, USA, 2008; pp. 12:1–12:87. doi:10.1145/1508044.1508091. [[CrossRef](#)]
4. Zou, G.; Hu, J.; Gu, X.; Hua, J. Authalic Parameterization of General Surfaces Using Lie Advection. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 2005–2014. doi:10.1109/TVCG.2011.171. [[CrossRef](#)] [[PubMed](#)]
5. Zhao, X.; Su, Z.; Gu, X.D.; Kaufman, A.; Sun, J.; Gao, J.; Luo, F. Area-Preservation Mapping Using Optimal Mass Transport. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2838–2847. doi:10.1109/TVCG.2013.135. [[CrossRef](#)] [[PubMed](#)]
6. Su, K.; Cui, L.; Qian, K.; Lei, N.; Zhang, J.; Zhang, M.; Gu, X.D. Area-preserving mesh parameterization for poly-annulus surfaces based on optimal mass transportation. *Comput. Aided Geom. Des.* **2016**, *46*, 76–91. doi:10.1016/j.cagd.2016.05.005. [[CrossRef](#)]
7. Yueh, M.H.; Lin, W.W.; Wu, C.T.; Yau, S.T. An Efficient Energy Minimization for Conformal Parameterizations. *J. Sci. Comput.* **2017**, *73*, 203–227. doi:10.1007/s10915-017-0414-y. [[CrossRef](#)]
8. Wang, Z.; Luo, Z.; Zhang, J.; Saucan, E. A novel local/global approach to spherical parameterization. *J. Comput. Appl. Math.* **2018**, *329*, 294–306; (The International Conference on Information and Computational Science, Dalian, China, 2–6 August 2016.) doi:10.1016/j.cam.2017.05.011. [[CrossRef](#)]
9. Yu, H.; Lee, T.; Yeh, I.; Yang, X.; Li, W.; Zhang, J.J. An RBF-Based Reparameterization Method for Constrained Texture Mapping. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 1115–1124. [[CrossRef](#)] [[PubMed](#)]
10. Liu, C.; Luo, Z.; Shi, X.; Liu, F.; Luo, X. A fast mesh parameterization algorithm based on 4-point interpolatory subdivision. *Appl. Math. Comput.* **2013**, *219*, 5339–5344. [[CrossRef](#)]
11. Sawhney, R.; Crane, K. Boundary First Flattening. *ACM Trans. Graph.* **2017**, *37*, 5:1–5:14. [[CrossRef](#)]
12. Bright, A.; Chien, E.; Weber, O. Harmonic Global Parametrization with Rational Holonomy. *ACM Trans. Graph.* **2017**, *36*, 89:1–89:15. [[CrossRef](#)]
13. Smith, J.; Schaefer, S. Bijective Parameterization with Free Boundaries. *ACM Trans. Graph.* **2015**, *34*, 70:1–70:9. [[CrossRef](#)]
14. Ruiz, O.E.; Mejia, D.; Cadavid, C.A. Triangular mesh parameterization with trimmed surfaces. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2015**, *9*, 303–316. [[CrossRef](#)]
15. Mejia, D.; Ruiz-Salguero, O.; Cadavid, C.A. Hessian Eigenfunctions for Triangular Mesh Parameterization. In Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016), Rome, Italy, 27–29 February 2016; Volume 1; pp. 75–82. [[CrossRef](#)]
16. Mejia, D.; Acosta, D.A.; Ruiz-Salguero, O. Weighted area/angle distortion minimization for Mesh Parameterization. *Eng. Comput.* **2017**, *34*, 1874–1895. [[CrossRef](#)]
17. Yu, X.; Lei, N.; Zheng, X.; Gu, X. Surface parameterization based on polar factorization. *J. Comput. Appl. Math.* **2018**, *329*, 24–36. (The International Conference on Information and Computational Science, Dalian, China, 2–6 August 2016.) [[CrossRef](#)]
18. Liu, L.; Zhang, L.; Xu, Y.; Gotsman, C.; Gortler, S.J. A Local/Global Approach to Mesh Parameterization. *Comput. Graph. Forum* **2008**, *27*, 1495–1504. [[CrossRef](#)]
19. Wang, Z.; Luo, Z.x.; Zhang, J.l.; Saucan, E. ARAP++: An extension of the local/global approach to mesh parameterization. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 501–515. [[CrossRef](#)]
20. Rabinovich, M.; Poranne, R.; Panozzo, D.; Sorkine-Hornung, O. Scalable Locally Injective Mappings. *ACM Trans. Graph.* **2017**, *36*, 37a. [[CrossRef](#)]
21. Li, Z.; Jin, Y.; Jin, X.; Ma, L. Approximate straightest path computation and its application in parameterization. *Vis. Comput.* **2012**, *28*, 63–74. [[CrossRef](#)]
22. Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson Surface Reconstruction. In Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP'06, Cagliari, Sardinia, 26–28 June 2006; Eurographics Association: Aire-la-Ville, Switzerland, 2006; pp. 61–70.
23. Crane, K.; Weischedel, C.; Wardetzky, M. Geodesics in Heat: A New Approach to Computing Distance Based on Heat Flow. *ACM Trans. Graph.* **2013**, *32*, 152:1–152:11. [[CrossRef](#)]
24. MATLAB®. *Version 9.4.0.813654 (R2018a)*; MathWorks®: Natick, MA, USA, 2018.
25. Rusu, R.B.; Cousins, S. 3D is here: Point Cloud Library (PCL). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4. [[CrossRef](#)]

26. Cignoni, P.; Callieri, M.; Corsini, M.; Dellepiane, M.; Ganovelli, F.; Ranzuglia, G. MeshLab: An Open-Source Mesh Processing Tool. In Proceedings of the Eurographics Italian Chapter Conference, Salerno, Italy, 2–4 July 2008; Scarano, V., Chiara, R.D., Erra, U., Eds.; The Eurographics Association: Aire-la-Ville, Switzerland, 2008. [[CrossRef](#)]
27. Sun, J.; Ovsjanikov, M.; Guibas, L. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Comput. Graph. Forum* **2009**, *28*, 1383–1392. [[CrossRef](#)]
28. Reuter, M.; Biasotti, S.; Giorgi, D.; Patanè, G.; Spagnuolo, M. Discrete Laplace-Beltrami operators for shape analysis and segmentation. *Comput. Graph.* **2009**, *33*, 381–390. (IEEE International Conference on Shape Modelling and Applications 2009.) [[CrossRef](#)]
29. Cox, M.A.A.; Cox, T.F. Multidimensional Scaling. In *Handbook of Data Visualization*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 315–347. [[CrossRef](#)]
30. Sheffer, A.; Hart, J.C. Seamster: Inconspicuous low-distortion texture seam layout. In Proceedings of the IEEE Conference on Visualization (VIS 2002), Boston, MA, USA, 27 October–1 November 2002; pp. 291–298. [[CrossRef](#)]
31. Mejia, D.; Ruiz-Salguero, O.; Sánchez, J.R.; Posada, J.; Moreno, A.; Cadavid, C.A. Hybrid geometry/topology based mesh segmentation for reverse engineering. *Comput. Graph.* **2018**, *73*, 47–58. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).