Weighted area/angle distortion minimization for Mesh Parameterization

Daniel Mejia
Laboratorio de CAD CAM CAE, Universidad EAFIT, Medellín, Colombia
Diego A. Acosta
Grupo de Desarrollo y Diseño de Procesos, Universidad EAFIT, Medellín, Colombia, and
Oscar Ruiz-Salguero
Laboratorio de CAD CAM CAE, Universidad EAFIT, Medellín, Colombia

Received 8 March 2016
Revised 19 September 2016
14 December 2016
Accepted 19 December 2016

Abstract

Purpose – Mesh Parameterization is central to reverse engineering, tool path planning, etc. This work synthesizes parameterizations with un-constrained borders, overall minimum angle plus area distortion. This study aims to present an assessment of the sensitivity of the minimized distortion with respect to weighed area and angle distortions.

Design/methodology/approach – A Mesh Parameterization which does not constrain borders is implemented by performing: isometry maps for each triangle to the plane $Z = 0$; an affine transform within the plane $Z = 0$ to glue the triangles back together; and a Levenberg–Marquardt minimization algorithm of a nonlinear F penalty function that modifies the parameters of the first two transformations to discourage triangle flips, angle or area distortions. F is a convex weighed combination of area distortion (weight: α with $0 \leq \alpha \leq 1$) and angle distortion (weight: $1 - \alpha$).

Findings – The present study parameterization algorithm has linear complexity [$\mathcal{O}$(n), n = number of mesh vertices]. The sensitivity analysis permits a fine-tuning of the weight parameter which achieves overall bijective parameterizations in the studied cases. No theoretical guarantee is given in this manuscript for the bijectivity. This algorithm has equal or superior performance compared with the ABF, LSCM and ARAP algorithms for the Ball, Cow and Gargoyle data sets. Additional correct results of this algorithm alone are presented for the Foot, Fandisk and Sliced-Glove data sets.

Originality/value – The devised free boundary nonlinear Mesh Parameterization method does not require a valid initial parameterization and produces locally bijective parameterizations in all of our tests. A formal sensitivity analysis shows that the resulting parameterization is more stable, i.e. the UV mapping changes very little when the algorithm tries to preserve angles than when it tries to preserve areas. The algorithm presented in this study belongs to the class that parameterizes meshes with holes. This study presents the results of a complexity analysis comparing the present study algorithm with 12 competing ones.

Keywords Reverse engineering, Sensitivity analysis, Complexity analysis, Levenberg–Marquardt, Mesh Parameterization, Nonlinear optimization

Paper type Research paper

Abbreviations

LM = Levenberg–Marquardt.
I_k = Identity matrix of degree k.
$\mathcal{O}(f(n))$ = Computational *time* complexity of an algorithm being asymptotic to $f(n)$, with n being the measuring unit.

Engineering Computations
Vol. 34 No. 6, 2017
pp. 1874-1895

0264-4401
DOI 10.1108/EC-02-2016-0072

M	= Triangular mesh (with non-empty border) of a two-manifold embedded in $\mathbb{R}^3$, composed by the set of triangles $T = \{t_1, t_2, \ldots, t_m\}$ with vertex set $X = \{x_1, x_2, \ldots, x_n\}$ ($X \subset \mathbb{R}^3$).
$\phi(x)$	= Parameterization of M which is a piecewise affine mapping $\{\phi_1, \phi_2, \ldots, \phi_m\}$ with $\phi_i = \Psi_i \circ \eta_i . \phi : M \to \phi(M) \subset \mathbb{R}^2$ is a homeomorphism.
U	= Set of points $U = \{u_1, u_2, \ldots, u_n\}$ corresponding to the image of X under the mapping ϕ (i.e. $u_i = \phi(x_i)$).
η_i	= Rigid transformation which maps the triangle t_i of M to the plane $Z = 0$ (also called XY plane here) and its center of mass $\bar{x}_i$ to the origin.
R^i	= Image of the vertices $[x_{i_1}, \ x_{i_2}, \ x_{i_3}]$ of the i-th triangle of M under the mapping η_i.
Q^i	= Right pseudo-inverse of R^i (i.e. $R^i Q^i = I_2$).
Ψ_i	= An affine mapping $\mathbb{R}^2 \to \mathbb{R}^2$ which maps R^i to its final place in the parameterization (ϕ). $\Psi_i(\xi) = A^i \xi + c_i$.
A^i	= Jacobian matrix of Ψ_i.
D^i_{area}	= Area distortion of the triangle t_i under the mapping ϕ_i defined as $D^i_{area} = (\det(A^i) - 1)^2$.
D^i_{angle}	= Angle distortion of the triangle t_i under the mapping ϕ_i defined as $D^i_{angle} = (A^2_{11} - A^2_{22})^2 + (A^2_{12} - A^2_{21})^2$.
$F(U)$	= Penalty function $\mathbb{R}^{2n} \to \mathbb{R}$ to be minimized which sums the weighted area and angle distortion of the triangles in M under the mapping $U = \phi(X)$.
F^*	= Value at which the penalty function F is a local minimum.
α	= Parameter $0 \leq \alpha \leq 1$ which weighs area distortion (α) against angle distortion ($1 - \alpha$) in the penalty function F.
∇	= Gradient operator.
$\mathscr{H}$	= Hessian operator.
λ	= Damping parameter of the LM algorithm.
ε	= Tolerance parameter of the LM algorithm.
S^f_p	= Relative sensitivity of a penalty function f with respect to a p parameter.

1. Introduction

In CAD, CAM, CAE, it is usual to have a triangular mesh $M \subset \mathbb{R}^3$ because of the segmentation of a larger triangular mesh. M is a two-manifold with non-empty border and low curvature (i.e. M is near-developable). Therefore, M admits a two-variable parameterization which is a homeomorphism between M and a polygonal region in $\mathbb{R}^2$.

Mesh Parameterization consists of finding a mapping $\phi : M \to \phi(M) \subset \mathbb{R}^2$ such that:

- ϕ and ϕ^{-1} are continuous (i.e. connectivity of the triangles is preserved after the mapping); and
- ϕ is bijective (i.e. triangles do not overlap after the mapping).

ϕ is a homeomorphism and the image of M under ϕ is a parameterization of M. In addition, local preservation of properties (e.g. angle, area, dimensions, etc.) is pursued but rarely achieved in parameterizations of actual engineering M meshes.

Mesh Parameterization is relevant in areas such as reverse engineering, tool path planning, feature detection, re-design, etc. This article proposes an algorithm for computing ϕ by minimizing a penalty function F which discourages triangle flips, angle and area distortions. Different results may be obtained for the same mesh by changing the parameter α (which weighs angle vs area preservation) in F, allowing the user to pick up the best parameterization. Fine-tuning of this parameter allows in some cases to reach globally bijective parameterizations from non-bijective ones.

The remainder of this article is structured as follows: Section 2 reviews the relevant literature. Section 3 describes the implemented methodology. Section 4 presents and discusses the results of the test runs. Section 5 concludes the paper and introduces opportunities for future work.

2. Literature review

Mesh Parameterization is usually achieved by posing an optimization problem where some kind of distortion measure is minimized in the parameter space. Depending on the characteristics of the method, Mesh Parameterization algorithms can be classified into:

- Constrained-Boundary methods;
- Free-Boundary methods; or
- Dimensionality Reduction (DR) methods.

Hormann *et al.* (2007) and Sheffer *et al.* (2006) present a survey of the state of the art in Mesh Parameterization methods. Extending such surveys, this section discusses the relevant literature in the topic.

2.1 Constrained-Boundary Mesh Parameterization

In Constrained-Boundary parameterizations, the border of the mesh is constrained in the resulting parameterization. Such constraint is forced by mapping the vertices of the mesh border to a fixed shape (e.g. a disk) in the parameter space. Barycentric coordinates methods (Floater, 1997) solve the parameterization problem by expressing each vertex in the parameter space as a convex combination of its neighbors. In Yoshizawa *et al.*'s (2004) study, a nonlinear-gradient algorithm which minimizes a stretch measure on a given Floater parameterization is proposed. The Signal-Specialized method (Sander *et al.*, 2002) proposes a non-linear algorithm for mapping the surface to a rectangular domain based on a function defined on the surface (such as a color map). In Pietroni *et al.*'s (2011) study, a local flattening operator is proposed to interactively parameterize rectangular patches on a surface to their corresponding planar domain. In Zou *et al.*'s (2011) study, an area-preserving mapping is sought by simulation of Lie advection (moving mass property change) on the surface, whereas in Zhao *et al.*'s (2013) study, Su *et al.*'s (2016) area-preserving mappings are computed using the optimal mass transport technique.

Constrained-Boundary methods present an additional problem of mapping the boundary to the parameter space in a separate step. This problem leads to highly distorted mappings in most application cases. The Virtual Boundary algorithm (Lee *et al.*, 2002) partially overcomes this problem by introducing an artificial boundary connected to the real boundary. However, the shape of the artificial boundary affects the result and still introduces several distortions.

2.2 Free-Boundary Mesh Parameterization

In contrast to Constrained-Boundary parameterizations, Free-Boundary methods do not require fixing the boundary in the parameter space. The MIPS (Most Isometric Parameterizations) method (Hormann and Greiner, 2000) proposes to minimize a nonlinear-gradient Dirichlet energy without boundary conditions, which produces a free boundary parameterization. Similarly, the Intrinsic Parameterizations algorithm (Desbrun *et al.*, 2002) minimizes an energy functional which is a linear combination of both area and angle distortions. LSCM (Least Squares Conformal Maps) seeks a conformal parameterization by discretizing the Cauchy–Riemann equations on a triangular mesh (Lévy *et al.*, 2002). The One-Step Inverse Forming approach (Li *et al.*, 2010; Zhu *et al.*, 2013) is a Mesh Parameterization algorithm based on the simulation of physical plastic deformation of the surface. In Li *et al.*'s (2012) study, an isometric parameterization is sought by computing geodesics on the surface. The aforementioned Free-

Boundary algorithms present the shortcoming of only accepting surfaces that are homeomorphic to a disk in $\mathbb{R}^2$ (i.e. surfaces without holes).

The ABF (Angle-Based Flattening) algorithm (Sheffer and de Sturler, 2001) is a nonlinear-gradient algorithm which poses an optimization problem in terms of the angles of the mesh triangles to find a conformal parameterization. However, the ABF algorithm is computationally expensive, making it impractical for large data sets. The ABF++ algorithm (Sheffer *et al.*, 2005) and the Linear ABF (Zayer *et al.*, 2007) present variations to the original ABF algorithm which potentially improves the computation time at the cost of global distortion. Angle-based algorithms usually require a post-processing step where the parameterization must be recovered from the computed angles of the mesh. The Circle Patterns (Kharevych *et al.*, 2006), Curvature Prescription (Ben-Chen *et al.*, 2008), Conformal Equivalence (Springborn *et al.*, 2008) and Controlled-distortion (Myles and Zorin, 2013) algorithms seek conformal parameterizations by transferring the Gaussian curvature of the triangular mesh to a selected set of nodes known as *cone singularities*. A similar approach extended to hexagonal meshes is proposed by Nieser *et al.* (2012). Automatic algorithms for placing such *cone singularities* on the mesh have been proposed in the literature (Ben-Chen *et al.*, 2008; Springborn *et al.*, 2008; Myles and Zorin, 2012). However, such placement is an additional pre-processing step to the Mesh Parameterization process that increases the complexity of the algorithm.

In Zayer *et al.*'s (2005) study, a nonlinear-gradient algorithm which uses conformal and quasi-harmonic maps for parameterizing surfaces with holes is proposed. ASAP/ARAP (As Similar As Possible/As Rigid As Possible) and ARAP++ (Liu *et al.*, 2008; Wang *et al.*, 2016) minimize an energy functional which is a linear combination of both area and angle distortions. However, the ASAP/ARAP method requires a post-processing step owing to triangle flips occurring in the resulting parameterization. The Constrained Parameterization on Parallel Platforms algorithm (Athanasiadis *et al.*, 2013) minimizes an energy function similar to the MIPS energy, using GPU resources to improve the computational efficiency. These nonlinear-gradient methods allow to parameterize surfaces with holes as opposed to previous methods. However, nonlinear-gradient methods require computing an initial valid parameterization (i.e. with no triangle flips) which increases the computational cost of the algorithm and the implementation complexity.

The problem of globally bijective parameterizations cannot be guaranteed in most cases (especially for Free-Boundary methods). The recently proposed Bijective Parameterizations with Free Boundaries algorithm (Smith and Schaefer, 2015) overcomes this problem by posing a nonlinear optimization problem with barrier functions which do not allow boundary overlapping. However, this algorithm becomes highly expensive when evaluating boundary overlaps. In addition, as most nonlinear-gradient parameterization algorithms, it requires an initial valid parameterization.

2.3 Dimensionality Reduction for Mesh Parameterization

Mesh Parameterization is an application of DR. DR techniques perform a parameterization of a *k*-manifold M ($k \in \mathbb{N}$), using the information about a proximity graph of the manifold. DR does not, in general, take advantage of the explicit mesh triangulation structure. Algorithms such as Isomap (Tenenbaum *et al.*, 2000), Laplacian Eigenmaps (Belkin and Niyogi, 2003), LTSA (Local Tangent Space Alignment) (Zhang and Zha, 2002), LLE (Locally Linear Embedding) (Roweis and Saul, 2000) and HLLE (Hessian Locally Linear Embedding) (Donoho and Grimes, 2003) are popular DR algorithms which have been relevant in the Mesh Parameterization literature. In Sun and Hancock's (2008) study, a method that combines Isomap and barycentric coordinates is proposed to produce an isometric parameterization. However, such method relies on estimation of geodesics which is inappropriate for non-convex manifolds. In addition, geodesics are computationally expensive to estimate. Ruiz *et al.* (2015) (using Laplacian Eigenmaps) and Mejia *et al.* (2016) (using a modified HLLE) avoid the computation of geodesics. This modification largely offsets the problems related to geodesic distance estimation. However, they cannot guarantee preservation of angle or areas resulting in high distortions. The Optimal Local Flattening algorithm (Chen *et al.*, 2007) is a Mesh Parameterization algorithm based on the LTSA DR technique which presents low distortions. However, triangle flips (local non-bijectivity) may arise during the mapping.

2.4 Conclusions of the literature review

As discussed earlier, current Mesh Parameterization algorithms may present one or more of the following disadvantages: constrained boundary, local overlaps (triangle flips), requirement of an initial valid parameterization to avoid local minima (for the penalty function) and non-bijective mappings, etc. In this article, we propose a Free-Boundary Mesh Parameterization algorithm where each triangle is mapped individually to the plane $Z = 0$ by a rigid transformation and then a penalty function F measuring distortion in the global parameterization is minimized. In contrast to most non-linear gradient algorithms, our algorithm does not require an initial valid parameterization. We observed no triangle flips in the cases processed by our algorithm, although we have no theoretical guarantee for such a behavior. A $0 \leq \alpha \leq 1$ parameter weighs area distortion against angle distortion weighed by $1 - \alpha$. The weighting scheme proposed in this manuscript restricts the α parameter by a convex combination which potentially avoids numerical instabilities that may arise as opposed to unbounded weighting parameters as in studies by Desbrun *et al.* (2002), Degener *et al.* (2003) and Liu *et al.* (2008).

To minimize F, we implement the Levenberg–Marquardt (LM) algorithm. LM is a gradient descent method with high convergence rate allowing to evade the computation of an initial valid parameterization which is a requirement in most nonlinear-gradient algorithms. The test runs consider both surfaces with and without holes, showing that no triangle flips occur in the resulting parameterization. A fine-tuning of the α parameter results in globally bijective parameterizations. We present a complexity analysis of our algorithm and a sensitivity analysis for the minimized F^* with respect to α. As per our Literature Review, a sensitivity analysis for weighting parameters has not yet been presented in the Mesh Parameterization.

3. Methodology

Consider $M = (X, T)$, a connected two-manifold in $\mathbb{R}^3$ with border which is homeomorphic to a polygonal region in $\mathbb{R}^2$, possibly with holes. Our goal is to find a set of points $U = \{u_1, u_2, \ldots, u_n\} \subset \mathbb{R}^2$ such that u_i is the image of $x_i \in X$ under a homeomorphism $\phi : M \rightarrow \mathbb{R}^2$.

We build ϕ as the composition $\Psi \circ \eta$, as follows:

- η: $M \rightarrow XY \subset \mathbb{R}^3$ maps in arbitrary rigid manner each triangle t_i of M onto the plane $Z = 0$; and
- an affine mapping Ψ in $\mathbb{R}^2$ glues the 2D (parametric space) triangles as they originally were in M.

The function ϕ presents a compromise between angle vs area preservation. The extent of such a compromise is part of the findings of this manuscript.

The algorithm for finding the parameterization $U = \phi\ (X)$ is described in Figure 1:

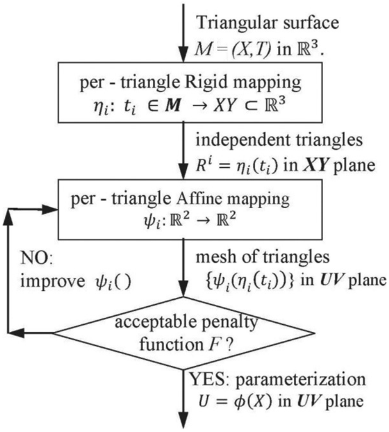


Figure 1.
Implemented Mesh Parameterization algorithm

- *Rigid mapping* η_i: $M \rightarrow XY \subset \mathbb{R}^3$: Find the rigid transformation η_i: $M \rightarrow XY \subset \mathbb{R}^3$ that maps the triangle t_i to the XY plane and its center of mass $\bar{x}_i$ to the origin. The matrix $R^i = \eta([x_{i1}, x_{i2}, x_{i3}])$ corresponds therefore to the image of the vertices $[x_{i1}, x_{i2}, x_{i3}]$ of the triangle t_i under such mapping.
- *Affine mapping* Ψ_i: $\mathbb{R}^2 \rightarrow \mathbb{R}^2$: Because each triangle has been mapped individually to $\mathbb{R}^2$, an affine mapping $\Psi_i(\xi) = A^i\xi + c_i$ which maps each R^i to the final parameterization ϕ is constructed. The Jacobian matrix A^i can be computed in terms of R^i and the vertices $[u_{i_1}, u_{i_2}, u_{i_3}]$ of the triangle $\phi(t_i)$. From this construction, $\phi_i = \Psi_i \circ \eta_i$ is an affine mapping and $\phi = \{\phi_1, \phi_2, \ldots, \phi_m\}$ is a piecewise affine mapping which parameterizes M. The continuity of ϕ is implied in $\Psi = \{\Psi_1, \Psi_2, \ldots, \Psi_m\}$ from the connectivity of M, i.e. if t_i and t_j share the edge (x_k, x_l) then Ψ_i and Ψ_j overlap in the edge (u_k, u_l).
- *Weighted penalty function F(U)*: A penalty function F: $\mathbb{R}^{2n} \rightarrow \mathbb{R}$ which penalizes the weighted area and shape distortion of each triangle is constructed in this step. Because $\phi_i = \Psi_i \circ \eta_i$ is an affine mapping and η_i is rigid, all the distortion of ϕ_i can be extracted from A^i. An area (D^i_{area}) and angle (D^i_{angle}) distortion is built for each triangle in terms of A^i, and a weighted sum of these terms over all the triangles compose the penalty function F.
- *Parameterization* $U = \phi(X)$: Because ϕ has a minimum distortion, U is estimated by minimizing F. Because ∇F is nonlinear, we implement the LM algorithm for this optimization process.

A detailed discussion follows.

3.1 *Rigid mapping* η_i: $M \rightarrow XY \subset \mathbb{R}^3$

To build the function F, we propose to individually map each triangle t_i of M to the XY plane first. One way to do this is to compute the center of mass $\bar{x}_i$ and the normal vector $\vec{n}_i$ of the triangle t_i. If $B_i = \left[\vec{v}^{\,i}_1, \ \vec{v}^{\,i}_2\right]$ is an orthonormal basis of the plane with normal $\vec{n}_i$, then η_i: $\mathbb{R}^3 \rightarrow XY \subset \mathbb{R}^3$ defined as:

$$\eta_i(x) = B_i^T(x - \bar{x}_i) \tag{1}$$

is a projection which isometrically maps the triangle t_i to the plane tangent to t_i (Figure 2). Therefore, the matrix R_i corresponds to the image of the current triangle vertices under the map η_i, i.e.:

$$R^i = \eta_i([x_{i_1},\ x_{i_2},\ x_{i_3}]) \tag{2}$$

where x_{i_j} is the j-th vertex of t_i.

3.2 Affine mapping Ψ_i: $\mathbb{R}^2 \rightarrow \mathbb{R}^2$

ϕ_i is an affine transformation such that $\phi_i = \Psi_i \circ \eta_i$, where Ψ_i: $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ is an affine transformation that maps R^i to the global parameterization U. Therefore:

$$\psi_i(\xi) = A^i\xi + c_i, \tag{3}$$

where A^i is a 2 × 2 linear transformation and $c_i = \frac{1}{3}\sum_{j=1}^{3} u_{i_j}$ is a translation term corresponding to the center of mass of the triangle ϕ (t_i).

Because η_i is isometric to t_i, the matrix A^i contains all the information about the distortion of the triangle t_i under the mapping ϕ_i. Recalling that ϕ (t_i) = Ψ_i (R^i), we solve equation (3) for A^i:

$$A^i = [u_{i_1} - c_i,\ u_{i_2} - c_i, u_{i_3} - c_i]Q^i, \tag{4}$$

where Q^i is the right pseudoinverse of R^i (i.e., $R^iQ^i = I_2$). The preservation of the connectivity of M under ϕ is implied in equation (4). Specifically, the set of matrices $\boldsymbol{A} = \{A^1, A^2, \ldots, A^m\}$ is correlated in the sense that if t_i and t_j share an edge (x_k, x_l), the matrices A^i and A^j share the terms u_k, u_l.

3.3 Weighted penalty function F(U)

Calculating A^i in terms of the parameterization coordinates U allows to evaluate the local authalic (area) and conformal (angle) distortion for each triangle under the parameterization ϕ. A transformation is authalic if and only if its Jacobian determinant is ±1. A consistent orientation is important to avoid local overlaps (triangle flips) which may result in a non-

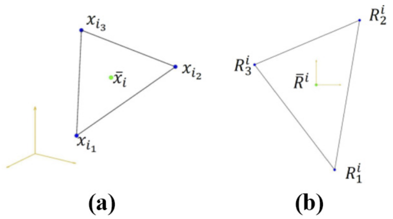


Figure 2.
Mapping of a triangle on M to $\mathbb{R}^2$ by projecting it onto the tangent plane

Notes: (a) A triangle (t_i) on the surface M; (b) mapping of the triangle t_i to the plane XY. The mapped triangle R_i is isometric to *ti* and it is mean-centered bijective mapping; therefore, the minus sign is discarded. Thus, we measure the area distortion D^i_{area} on each triangle by setting:

$$D^i_{area} = \left(\det\left(A^i\right) - 1\right)^2 \quad (5)$$

On the other hand, a mapping is conformal if its Jacobian matrix is k times a rotation matrix. Similar to Lévy *et al.* (2002), we construct the angle distortion measure D^i_{angle} of each triangle as follows:

$$D^i_{angle} = \left(A^i_{11} - A^i_{22}\right)^2 + \left(A^i_{12} + A^i_{21}\right)^2 \quad (6)$$

If we define the area and shape distortion of the mapping as the weighted sum of all D^i_{area} and all D^i_{angle} respectively, we can measure the global distortion as a convex combination of the global area and angle distortion:

$$F = \sum_{i=1}^{m} \alpha D^i_{area} + (1-\alpha) D^i_{angle} \quad (7)$$

with $0 \leq \alpha \leq 1$ being a weighting parameter such that F measures only angle distortion if $\alpha \rightarrow 0$ and F measures only area distortion if $\alpha \rightarrow 1$. F is only dependent on $\boldsymbol{Q} = \{Q^1, Q^2, \ldots, Q^m\}$ and $U = \{u_1, u_2, \ldots, u_n\}$ which are required to compute A^i as described in equation (4), and the weighting parameter α is introduced to control the resulting parameterization. The parameter α produces a compromise between area-preserving vs angle-preserving parameterizations. This compromise is central in the cases were both criteria cannot be satisfied. Simultaneous area and angle preservation is only possible with isometric parameterizations, which are feasible only for the special case of developable surfaces (e.g. with null Gaussian curvature at each surface point).

3.4 Parameterization $U = \phi$ (X)

To find the global parameterization $U = \phi$ (X), it is necessary to minimize the F defined in equation (7). Therefore, our global parameterization is given by the value of U that solves the following unconstrained problem:

$$\min_{U} \; F = \left\{ \sum_{i=1}^{m} \alpha D^i_{area} + (1-\alpha) D^i_{angle} \right\} \quad (8)$$

The function F from equation (7) is continuous and has $2n$ degrees of freedom (two degrees for each coordinate value u_i). The nonlinear nature of the gradient of F requires a nonlinear method for finding a solution to equation (8). Therefore, we choose the LM algorithm for such a purpose, which is described below.

3.5 The LM algorithm

We use a LM algorithm to update the parameterization coordinates according to the following scheme (Ravindran *et al.*, 2007):

$$U^{k+1} = U^k - \left(\mathcal{H}[F(U^k)] + \lambda^k I_{2n}\right)^{-1} \nabla F(U^k) \tag{9}$$

where $k+1$ is the current iteration, λ^k is the LM damping parameter which is updated iteratively according to the current solution and $\mathscr{H}[F(U^k)]$ is the Hessian matrix of F defined as: $\mathcal{H}_{ij}[F] = \frac{\partial^2 F}{\partial u_i \partial u_j}$ (Papadimitriou and Steiglitz, 1982).

We coded an implementation of the LM algorithm in MATLAB for the solution of equation (8). The advantages of our implementation are described below:

- *Initial parameterization*: Our implementation allows an initial random parameterization U^0 for iterating equation (9), providing consistent parameterizations in all our test cases. This is superior to most nonlinear-gradient algorithms which require an initial valid parameterization (estimated by a linear parameterization algorithm) to proceed, such as in Athanasiadis *et al.* (2013), Liu *et al.* (2008) and Smith and Schaefer (2015).
- *Hessian estimator*: The LM algorithm proposes to estimate the Hessian matrix as $\mathscr{H}[F] \approx \nabla F \cdot \nabla F^T$. This approach leads to a dense Hessian matrix. Our implementation computes the Hessian of each triangle distortion ($\alpha\mathcal{H}[D^i_{area}] + (1-\alpha)\mathcal{H}[D^i_{angle}]$) individually, and then adds up such terms to the global Hessian matrix $\mathscr{H}[F]$. Therefore, our estimated Hessian $\mathscr{H}[F]$ ends up being a $2n \times 2n$ (with n = number of mesh nodes) symmetric sparse matrix where only adjacent points in M have respective nonzero elements. The sparsity of our Hessian matrix has positive effects (Section 4.5) on the computing expenses of our algorithm.

The iterative procedure is applied on equation (9) until certain criteria are met:

- the norm of the gradient ∇F is lower than a fixed tolerance ε ($\varepsilon \in \mathbb{R}$); or
- a certain number of iterations has occurred.

3.6 Sensitivity analysis

We perform a relative sensitivity analysis of the minimized penalty function F^* with respect to the weighing parameter α to compare the resulting parameterization and the global distortion for a chosen value for α numerically (Edgar and Himmelblau, 2001):

$$S^F_\alpha = \frac{\partial \ln F^*}{\partial \ln \alpha} = \frac{\alpha}{F^*}\frac{\partial F^*}{\partial \alpha} \approx \frac{\overline{\alpha}}{\overline{F^*}}\frac{\Delta F^*}{\Delta \alpha} \tag{10}$$

Equation (10) provides an idea of how small changes in the α parameter affect the minimized penalty function F^* given mesh M.

3.7 Computer experiment set up

The tests run to assess our algorithm performance include several data sets and comparison with competitor algorithms. Table I discusses the data sets used. Table II lists the competitor algorithms tested along with ours. We do not intend to run a full benchmark test because we cannot guarantee even conditions to run all algorithms.

4. Results and discussion

In this section, two case studies from the literature are presented and analyzed thoroughly. Section 4.1 presents the first case corresponding to the *Beetle* data set [Figure 3(a)]. We show

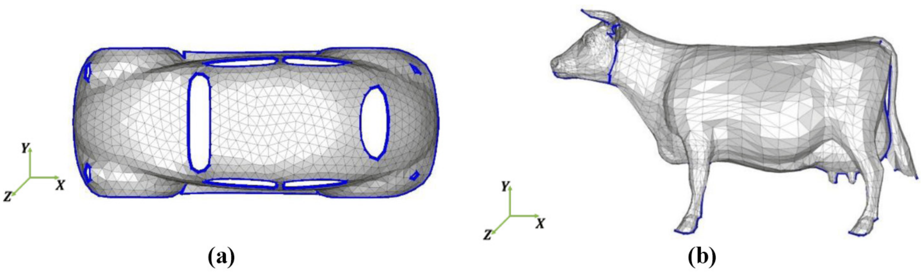
Data set	Natural border	Added boundary	Origin
Tests run in competition with ARAP, ABF, LSCM			
Balls	Yes	None	ARAP site
Beetle	Yes	None	ARAP site
Cow	No (closed)	Yes (Sheffer and Hart, 2002)	ALICE project-team (2008)
Gargoyle	No (closed)	Yes (Sheffer and Hart, 2002)	ARAP site
Tests of our algorithm alone			
Bull	No (closed)	Yes (Sheffer and Hart, 2002)	ALICE project-team (2008)
Foot	Yes	Yes (Sheffer and Hart, 2002)	ALICE project-team (2008)
Fandisk	No (closed)	Yes (Sheffer and Hart, 2002)	ALICE project-team (2008)
Sliced-Glove	Yes	Yes (open cut to use one hemisphere)	ALICE project-team (2008)

Note: ARAP site: www.math.zju.edu.cn/ligangliu/cagd/Projects/ARAPPara/default.htm

Table I. Data sets used for the algorithm appraisal

Algorithm	Requires a valid initial parameterization	Generality of tests	Reference	Code
ARAP	Yes	Restricted for data sets *Balls, Beetle, Gargoyle, Cow*	(Liu *et al.*, 2008)	Interpreted (MATLAB)
ABF	No	Not restricted	(Sheffer and de Sturler, 2001)	Compiled (C)
LSCM	No	Not restricted	(Lévy *et al.*, 2002)	Compiled (C)
Our algorithm	No	Not restricted	This article	Interpreted (MATLAB)

Table II. Conditions of competitor algorithms considered



Notes: (a) *Beetle* data set; (b) *Cow* data set

Figure 3. Case studies data sets

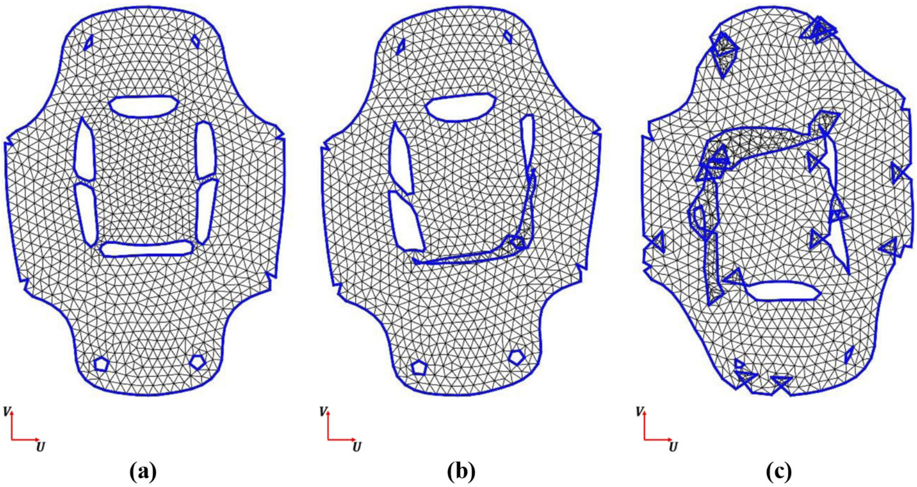
that by tuning adequately the α parameter, a valid parameterization can be achieved. Section 4.2 presents the second case study, namely, the *Cow* data set [Figure 3(b)]. This case study has presented several problems and though a nearly valid parameterization is achieved with our algorithm, global overlaps cannot be helped (Sheffer and de Sturler, 2001; Smith and Schaefer, 2015). Section 4.3 presents and discusses a summary of the results of our parameterization algorithm applied to other data sets, and Section 4.4 compares our parameterization results with ABF, LSCM and ARAP. Finally, Section 4.5 presents the results of a complexity analysis comparing our algorithm with 12 competing ones.

4.1 Beetle data set results

Figure 4 presents the resulting parameterization U for the *Beetle* data set with different values of α. Setting $\alpha = 0.1$ results in a valid parameterization with low shape distortion as seen in Figure 4(a). This is not the case for $\alpha = 0.5$ [Figure 4(b)], where global overlaps occur as some area preservation is demanded to the algorithm (triangle flips do not happen). A highly authalic mapping ($\alpha = 0.9$) results in a parameterization with higher shape distortion and low area distortion [Figure 4(c)]. Despite no triangle flips occurring, the boundary and non-adjacent triangles overlap, resulting in a non-bijective parameterization. The mapped texture in Figure 5 shows how the shape is highly preserved as squares attain its form through the bijective mapping for $\alpha = 0.1$. Similar results for this data set have been presented by other authors (Liu *et al.*, 2008; Sun and Hancock, 2008).

Recalling that the implemented algorithm converges to the same solution despite the initial (possibly non-valid) parameterization, Figure 6 presents the initial, intermediate and final stages for $\alpha = 0.1$ of the LM for different initial parameterizations:

- an initial parameterization U^0_{Isomap} computed by the Isomap algorithm (Tenenbaum *et al.*, 2000) [Figure 6(a)];
- an initial parameterization U^0_{LapEig} computed by the Laplacian Eigenmaps algorithm (Belkin and Niyogi, 2003) [Figure 6(b)]; and
- a randomly generated (non-bijective) initial parameterization U^0_{Rand} [Figure 6(c)].



Notes: (a) $\alpha = 0.1$ (quasi-conformal); (b) $\alpha = 0.5$ (conformal and authalic); (c) $\alpha = 0.9$ (quasi-authalic)

Figure 4.
Resulting parameterization for the *Beetle* data set for different α values

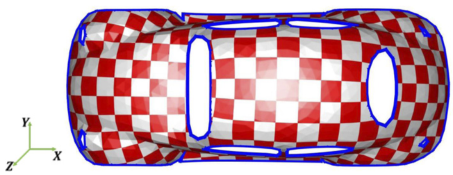


Figure 5.
Texture map for the *Beetle* data set ($\alpha = 0.1$)

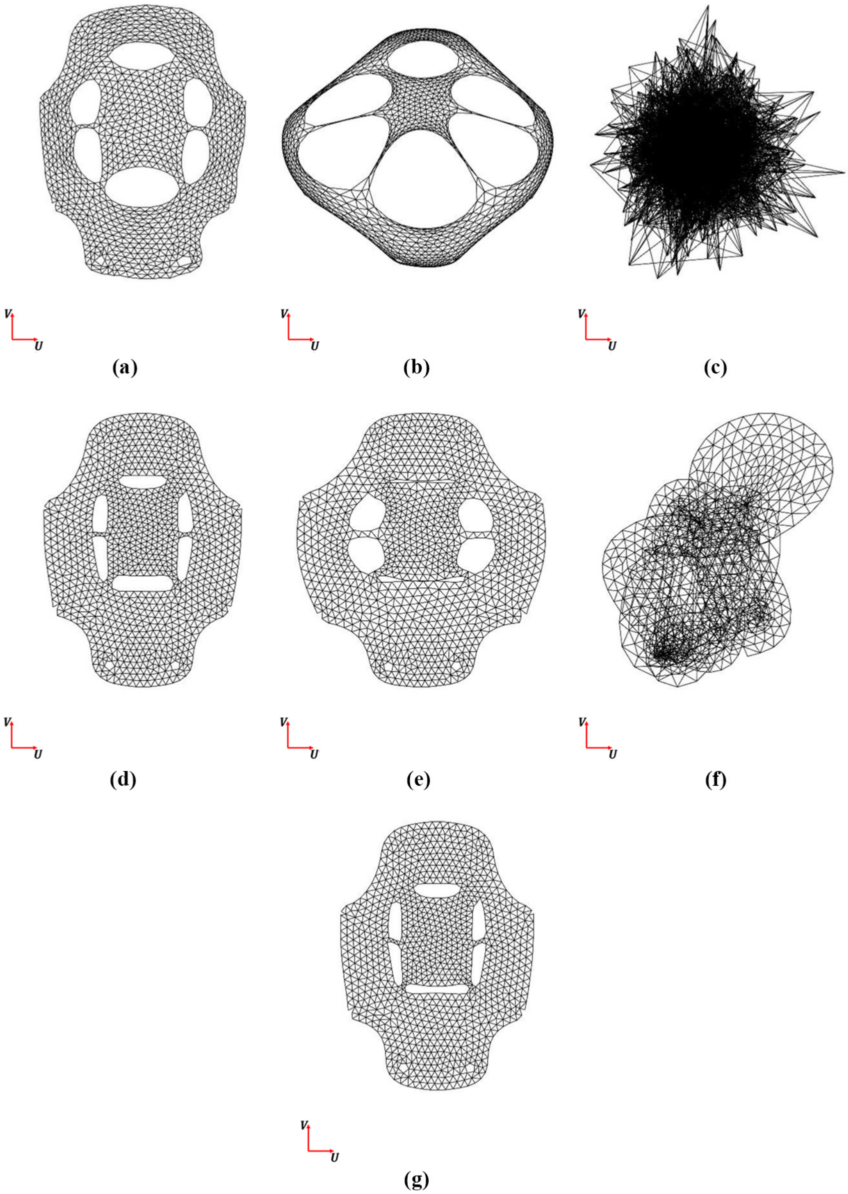


Figure 6.
Initial, intermediate and final stages of LM optimization for the *Beetle* data set using different initial parameterizations

Notes: (a) Isomap initial parameterization U^0_{Isomap}; (b) Laplacian Eigenmaps initial parameterization U^0_{LapEig}; (c) random (non-bijective) initial parameterization U^0_{Rand}; (d) Intermediate solution for U^0_{Isomap}; (e) intermediate solution for U^0_{LapEig}; (f) intermediate solution for U^0_{Rand}; and (g) final LM parameterization for: i) U^0_{Isomap}, ii) U^0_{LapEig} and iii) U^0_{Rand} for (i) isomap U^0_{Isomap}, (ii) Laplacian eigenmaps U^0_{LapEig} and (iii) random parameterization U^0_{Rand}; α is set to 0.1.

We coded an implementation of both DR algorithms (Isomap and Laplacian Eigenmaps) in MATLAB, whereas the random parameterization is generated by the MATLAB *rand()* routine. The respective intermediate steps [Figures 6(d), 6(e) and 6(f)] show how the surface is unfolded in each case, and Figure 6(g) presents the resulting parameterization for all the cases illustrating the consistency of the algorithm (even for the random non-valid initial parameterization U^0_{Rand}) as discussed in Section 3.5.

Using the random initial solution of Figure 6(c), Figure 7(a) presents the evolution of the penalty function F for different α values. Higher values of α take more iterations before converging. Also, though F^* reaches a lower value for $\alpha = 1$ than for $\alpha = 0.5$, this is not a guarantee of a better result as seen in Figure 4(b) and (c). Figure 7(b) plots the relative sensitivity of F^* with respect to α as per equation (12). F^* becomes highly sensitive to α for values greater than 0.6. In this particular case, the resulting parameterization becomes non-valid for higher values of α as seen in Figure 4(a) and (b).

4.2 Cow data set results

For a random initial parameterization U^0_{Rand}, Figure 8 presents the resulting parameterization U for the *Cow* data set with different α values. Setting $\alpha = 0.1$ results in a non-valid parameterization where the head of the *Cow* overlaps its body [Figure 8(a)]. For $\alpha = 0.01$, the head no longer overlaps the body in the resulting parameterization [Figure 8(b)]. However, the resulting parameterization is still non-bijective. Attempting a purely conformal parameterization ($\alpha = 0$) results in a high distorted mapping where the head and the legs present a high area distortion [Figures 8(c) and 9]. Zooming into the head and tail of the *Cow*, it is clear that the boundary self-intersects and the parameterization is not bijective [Figure 8(d) and (e)]. It is important to emphasize that none of the discussed results above presents triangle flips despite the map not being bijective. Our results are in concordance with other authors (Ben-Chen *et al.*, 2008; Liu *et al.*, 2008; Sheffer *et al.*, 2005; Zayer *et al.*, 2007) where non-bijective parameterizations (e.g. global overlaps) have been also reported for the *Cow* data set. Finally, higher values of α were tested resulting in worse parameterizations.

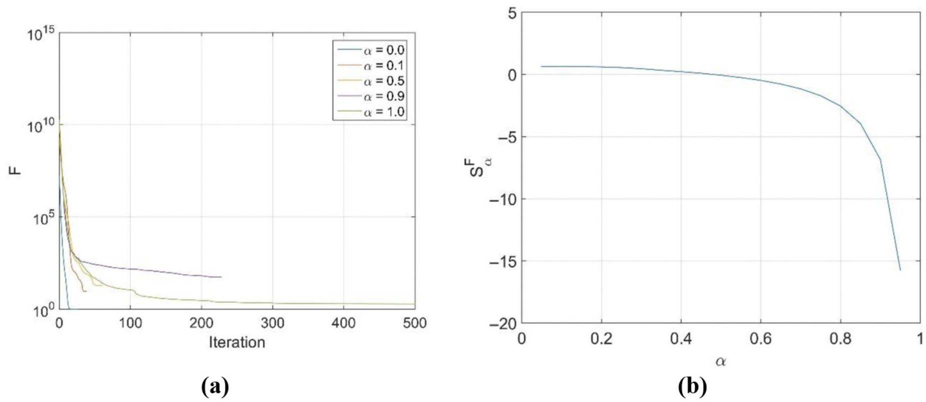
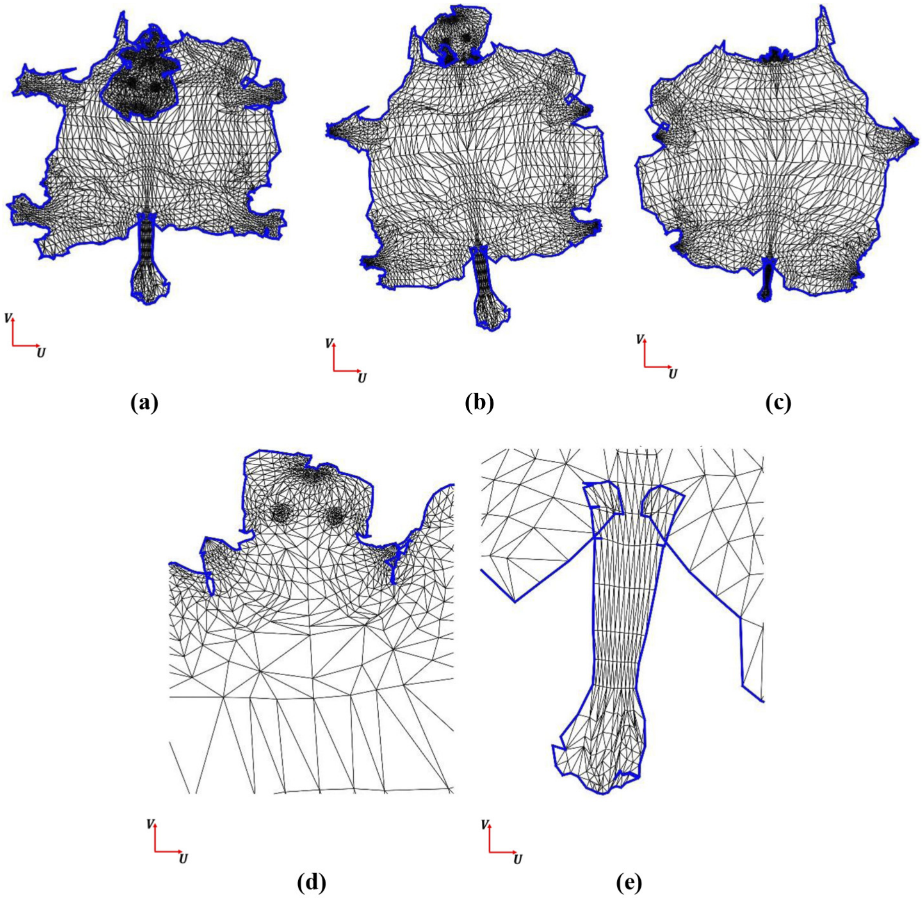


Figure 7.
Beetle data set. Sensitivity analysis of F with respect to α

Notes: For area-preserving parameterizations, (a) Evolution of F for different α values: the algorithm evidences slower convergence for higher values of α (b) Relative sensitivity S^F_α: the penalty function F is highly sensitive in the area-preserving ($\alpha \to 1$) side



Notes: (a) *Cow* parameterization ($\alpha = 0.1$); (b) *Cow* parameterization ($\alpha = 0.01$); (c) *Cow* parameterization ($\alpha = 0$); (d) *Cow* parameterization ($\alpha = 0$). Zoom into head; (e) *Cow* parameterization ($\alpha = 0$). Zoom into tail

Figure 8. Parameterization results for the *Cow* data set

Figure 10 presents a sensitivity analysis of F with respect to α for the *Cow* data set. As before, higher values of α require more iterations to converge to a minimum of F [Figure 10(a)]. Similar to our *Beetle* sensitivity results, the relative sensitivity of F with respect to α [Figure 10(b)] shows how F becomes highly sensitive for higher values of α ($\alpha > 0$). Again, the sensitivity analysis must be complemented by the user criteria as different values for α produce different (and not necessarily valid) parameterizations (Figure 8).

4.3 Results for other data sets

Figure 11 presents the parameterization results of the proposed algorithm for the *Sliced-Glove*, *Fandisk*, *Foot* and *Bull* data sets. With an $\alpha = 0.5$, the algorithm converged in all the presented cases to valid parameterizations (except for the *Bull* parameterization which is not

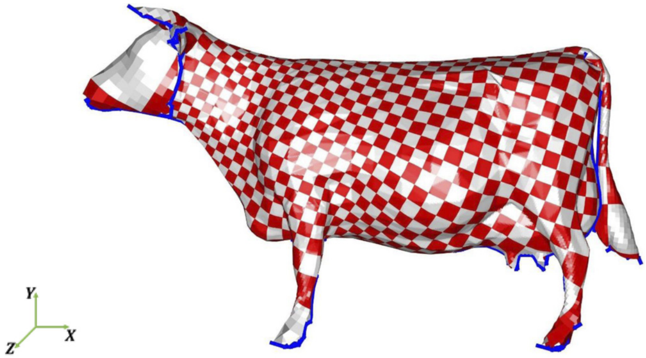


Figure 9.
Texture map for the *Cow* data set ($\alpha = 0$)

Note: Area distortion is more noticeable in the head, legs and tail

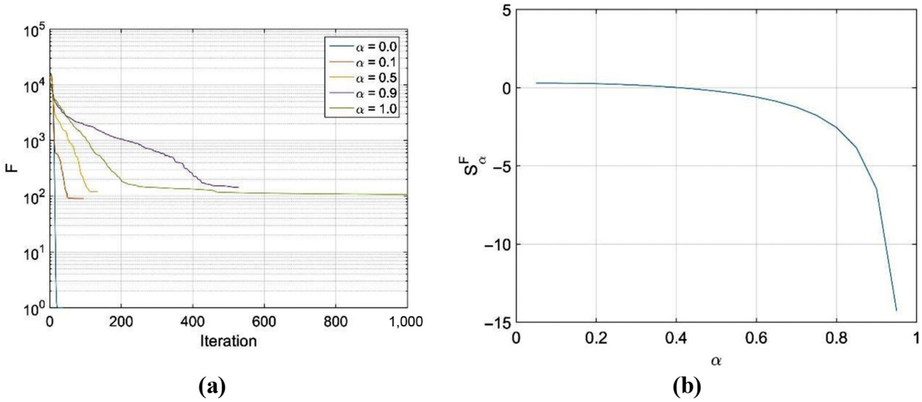


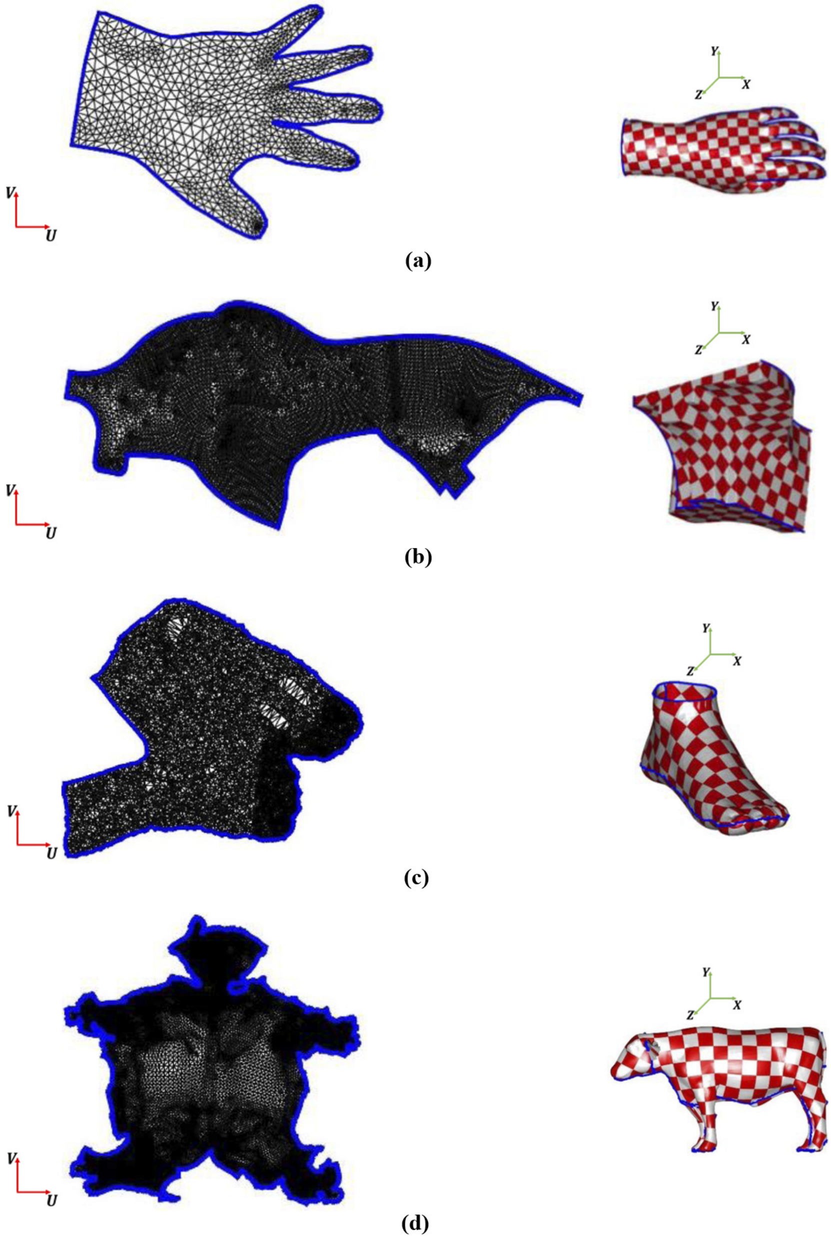
Figure 10.
Cow data set. Sensitivity analysis of F with respect to α

Notes: For area-preserving parameterizations, (a) Evolution of F for different α values: the algorithm evidences slower convergence for higher values of α; (b) Relative sensitivity S_α^F: the penalty function F is highly sensitive in the area-preserving ($\alpha \rightarrow 1$) side

bijective). Table III presents a quantification of the performance of our algorithm for the case studies. A random initial parameterization U^0_{Rand} is used and in all the test cases, $\|\nabla F^*\| < 10^{-8}$ and the lowest eigenvalue of the Hessian is nonnegative, indicating that a local minimum has been reached and the algorithm has converged. All the resulting parameterizations are locally bijective (no triangle flips).

4.4 Comparison with competitor algorithms

To compare our algorithm with other Mesh Parameterization algorithms, the GraphiteLE software (https://gforge.inria.fr/frs/?group_id=1465, accessed 1 August 2016) is used to run the ABF and the LSCM algorithms, while the MATLAB version of ARAP (see Table II) status of August 01, 2016, is used for computing such parameterization. The algorithms are run with default parameters and no modifications to the routines have been made.



Notes: All meshes are two-manifolds with border; (a) *Sliced-Glove* bijective parameterization and texture map; (b) *Fandisk* bijective parameterization and texture map; (c) *Foot* bijective parameterization and texture map; (d) *Bull* non-bijective parameterization and texture map

Figure 11. Parameterization results for several data sets and their corresponding texture map

Furthermore, ARAP is run with a valid initial parameterization (ABF by default), while our algorithm is run with a random initial guess.

Figure 12 presents the parameterization results obtained by our algorithm vs ABF, LSCM and ARAP for the *Balls*, *Beetle*, *Cow* and *Gargoyle* data sets. We do not measure execution times because:

- the algorithms are written in different programming languages (C and MATLAB);
- ARAP has its own data which requires pre-processing; and
- in general, we cannot provide an even field for algorithm comparison.

Table IV and Figure 12 present the results for the different algorithms. For the four data sets, ABF and our algorithm present no triangle flips, while LSCM and ARAP invert some triangles through the process. Our algorithm results in globally bijective parameterizations for the *Balls*, *Beetle* and *Gargoyle* data sets. None of the algorithms reaches a globally bijective parameterization for the *Cow* data set.

4.5 Complexity analysis

The time complexity of our algorithm is discussed in this section. In Ueda and Yamashita's (2012) study, a complexity analysis of the LM algorithm has been presented which shows that LM iterates $\mathcal{O}(\ln \varepsilon^{-1})$ times. In our algorithm, for a mesh with m triangles and n nodes, each LM iteration must perform the following operations:

- Compute $F(U^k)$, $\nabla F(U^k)$ and $\mathcal{H}[F(U^k)]$.
- Solve the linear system $(\mathcal{H}[F(U^k)] + \lambda^k I_{2n})^{-1} \nabla F(U^k)$ as per equation (9).

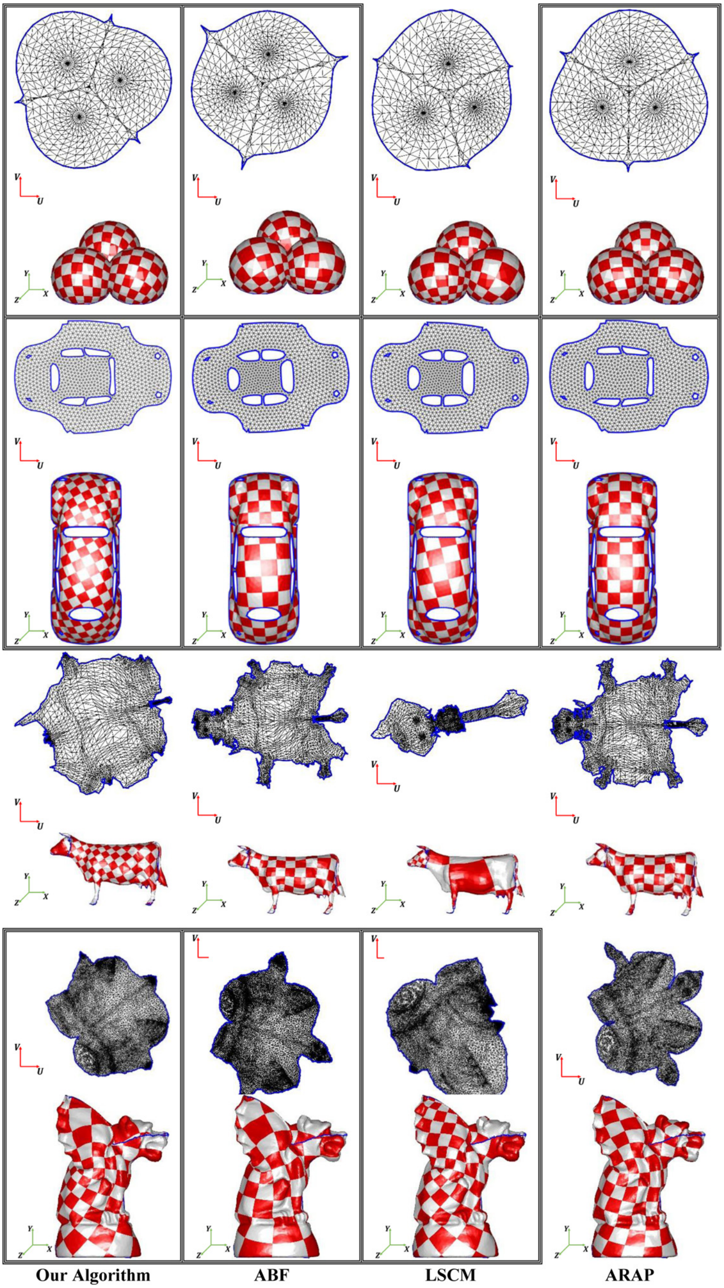
In the first step of the LM iteration, $F(U^k)$, $\nabla F(U^k)$ and $\mathcal{H}[F(U^k)]$ are computed by adding the distortion D^i_{area} and D^i_{angle} and their corresponding derivatives for each individual triangle. The cost of this operation is $\mathcal{O}(m)$. For the linear system in the second step, the matrix $\mathcal{H}[F(U^k)] + \lambda^k I_{2n}$ is a $2n \times 2n$ symmetric sparse matrix whose nonzero elements correspond to adjacent nodes in the mesh as discussed in Section 3.5. The solution of this linear system costs $\mathcal{O}(nz)$ (where nz is the number of nonzeros in the matrix $\mathcal{H}[F(U^k)] + \lambda^k I_{2n}$). Owing to the Euler characteristic of triangular meshes, $nz \approx 28n$. Hence, the complexity order for the linear system solution becomes $\mathcal{O}(n)$ (Botsch *et al.*, 2005).

In summary, the order of our algorithm is of the same order of the LM algorithm times the internal loop, i.e. $\mathcal{O}(\ln \varepsilon^{-1})$ $(\mathcal{O}(m) + \mathcal{O}(n))$. However, for a fixed ε and assuming (by the Euler characteristic) that $\mathcal{O}(m) = \mathcal{O}(n)$, our algorithm becomes of the order $\mathcal{O}(c \cdot n)$ (where c is

Table III. Quantification of our algorithm performance for the cases studied

Data set	n	α	$\|\nabla F\|$	Result
Beetle	988	0.1	2×10^{-12}	Figure 5
Cow	3195	0	4×10^{-11}	Figure 9
Sliced-Glove	985	0.5	7×10^{-11}	Figure 11(a)
Fandisk	6699	0.5	4×10^{-11}	Figure 11(b)
Foot	10211	0.5	6×10^{-11}	Figure 11(c)
Bull	17918	0.1	4×10^{-9}	Figure 11(d)

Notes: n = number of points; α = weight of area-preserving criterion; $\|\nabla \boldsymbol{F}\|$ = gradient of objective function



Note: Legal parameterizations are enclosed with a framed cell

Figure 12.
Parameterization results for the *Balls*, *Beetle*, *Cow* and the *Gargoyle* data sets using our algorithm and several state-of-the-art mesh parameterization algorithms

the number of iterations of the LM algorithm before convergence). Table V presents a comparison of computational complexities for several Mesh Parameterization algorithms.

5. Conclusions

This article presents an algorithm for parameterizing a triangular mesh M of a two-manifold with non-empty border embedded in $\mathbb{R}^3$. The proposed algorithm consists of mapping each triangle individually to the plane $Z = 0$ by a rigid transformation η and then mapping it to the global parameterization ϕ by an affine mapping Ψ. The parameterization $U = \phi(X)$ is obtained by minimizing the weighted area and angle distortion of Ψ (which also penalizes triangle flips) with the LM algorithm. The complexity analysis of our algorithm shows asymptotic linear behavior $\mathcal{O}(c \cdot n)$ in the number of vertices which makes our method comparable to most Mesh Parameterization methods that are also asymptotically linear in time as illustrated in Table V. Our algorithm presents the advantage over other nonlinear-gradient parameterization algorithms of not requiring an initial valid parameterization.

A weighting parameter α is introduced in the penalty distortion function which allows tuning by the user to favor area against angle preservation turning a non-bijective parameterization into a bijective one in specific cases. Our sensitivity analysis shows that our penalty function is very sensitive in the domain of area preservation ($\alpha \rightarrow 1$). Our experiments show that global overlaps are more frequent when preserving areas than when

Table IV. Appraisal of the parameterization results for our algorithm, ABF, LSCM and ARAP

	Our algorithm		ABF		LSCM		ARAP	
Data set	Triangle flips	Global overlaps	Triangle flips	Global overlaps	Triangle flips	Global overlaps	Triangle flips	Global overlaps
Balls	No	No	No	No	Yes	Yes	No	No
Beetle	No	No	No	No	No	No	No	No
Cow	No	Yes	No	Yes	Yes	Yes	Yes	Yes
Gargoyle	No	No	No	No	No	No	Yes	Yes

Note: A necessary condition for a valid parameterization is the absence of triangle flips and global overlaps

Table V. Computational time complexity of several mesh parameterization algorithms

Reference	Algorithm	Complexity
N/A	Our LM Mesh Parameterization	$\mathcal{O}(c \cdot n)$
(Donoho and Grimes, 2003)	Classic HLLE	$\mathcal{O}(c \cdot n)$
(Floater, 1997)	Floater Parameterization	$\mathcal{O}(c \cdot n)$
(Yoshizawa *et al.*, 2004)	Stretch Minimizing Parameterization	$\mathcal{O}(c_1 \cdot c_2 \cdot n)$
(Desbrun *et al.*, 2002)	Intrinsic Parameterization	$\mathcal{O}(c \cdot n)$
(Lévy *et al.*, 2002)	LSCM	$\mathcal{O}(c \cdot n)$
(Lee *et al.*, 2002)	Virtual Boundary Parameterization	$\mathcal{O}(c_1 \cdot n + c_2 \cdot \lvert\partial M\rvert)$
(Zayer *et al.*, 2007)	Linear ABF	$\mathcal{O}(c \cdot n)$
(Liu *et al.*, 2008)	ASAP	$\mathcal{O}(c \cdot n)$
(Liu *et al.*, 2008)	ARAP	$\mathcal{O}(c_1 \cdot c_2 \cdot n + c_3 \cdot \lvert\partial M\rvert)$
(Sheffer and de Sturler, 2001)	ABF	$\mathcal{O}(c_1 \cdot c_2 \cdot n)$
(Kharevych *et al.*, 2006)	Discrete Conformal Mappings	$\mathcal{O}(c_1 \cdot c_2 \cdot n)$
(Smith and Schaefer, 2015)	Free Boundary Parameterization	$\mathcal{O}(c(n + \lvert\partial M\rvert))$

Notes: c denotes the number of iterations before convergence for that particular algorithm; while $\lvert\partial M\rvert$ denotes the number of vertices at the boundary of M

preserving angles, encouraging angle preservation. It must be remarked that a sensitivity analysis assesses the influence of the parameters on the penalty function $\boldsymbol{F}$ and not the goodness of $\boldsymbol{F}$ for the problem at hand.

Compared to other Mesh Parameterization algorithms in general, our algorithm converged, presenting correct results across the data sets, rendering low distortion, non-overlapping and valid parameterizations except for highly non-developable data sets (i.e. *Cow* and *Bull*).

5.1 Ongoing work

Segmentation of large meshes into smaller ones increases the probability of finding bijective individual parameterizations for the smaller ones. Therefore, it is of interest to explore mesh segmentation as a necessary step for Mesh Parameterization.

Bijectivity of the resulting parameterization relates to:

- local overlaps (triangle flips); and
- global overlaps.

Although not proven, all of our experiments present no triangle flips, partially addressing the problem of local overlaps. However, global bijectivity is a non-trivial constraint which increases the computational complexity of the algorithm as shown by Smith and Schaefer (2015). Therefore, further work is required on this aspect.

References

ALICE project-team (2008), "Unwrapped meshes", available at: http://alice.loria.fr/index.php/software/7-data/37-unwrapped-meshes.html (accessed 10 February 2016).

Athanasiadis, T., Zioupos, G. and Fudos, I. (2013), "Efficient computation of constrained parameterizations on parallel platforms", *Computers & Graphics*, Vol. 37 No. 6, pp. 596-607.

Belkin, M. and Niyogi, P. (2003), "Laplacian eigenmaps for dimensionality reduction and data representation", *Neural Computation*, Vol. 15 No. 6, pp. 1373-1396.

Ben-Chen, M., Gotsman, C. and Bunin, G. (2008), "Conformal flattening by curvature prescription and metric scaling", *Computer Graphics Forum*, Vol. 27 No. 2, pp. 449-458.

Botsch, M., Bommes, D. and Kobbelt, L. (2005), "Efficient linear system solvers for mesh processing", in Martin, R., Bez, H. and Sabin, M. (Eds), *Proceedings of the 11th IMA International Conference Mathematics of Surfaces*, Loughborough, 5-7 September, *Springer Berlin Heidelberg*, Berlin, pp. 62-83.

Chen, Z., Liu, L., Zhang, Z. and Wang, G. (2007), "Surface parameterization via aligning optimal local flattening", *Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling (SPM '07)*, *ACM*, Beijing, pp. 291-296.

Degener, P., Meseth, J. and Klein, R. (2003), "An adaptable surface parameterization method", *Proceedings of the 12th International Meshing Roundtable*ô, New Mexico, pp. 201-213.

Desbrun, M., Meyer, M. and Alliez, P. (2002), "Intrinsic parameterizations of surface meshes", *Computer Graphics Forum*, Vol. 21 No. 3, pp. 209-218.

Donoho, D.L. and Grimes, C. (2003), "Hessian eigenmaps: locally linear embedding techniques for high-dimensional data", *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 100 No. 10, pp. 5591-5596.

Edgar, T.F. and Himmelblau, D.M. (2001), *Optimization of Chemical Processes*, 2nd ed., McGraw-Hill, New York, NY.

Floater, M.S. (1997), "Parameterization and smooth approximation of surface triangulations", *Computer Aided Geometric Design*, Vol. 14 No. 3, pp. 231-250.

Hormann, K. and Greiner, G. (2000), "MIPS: an efficient global parametrization method", in Laurent, P. J., Sablonniere, P. and Schumaker, L.L. (Eds), *Curve and Surface Design: Saint-Malo 1999*, Vanderbilt University Press, Nashville, pp. 153-162.

Hormann, K., Lévy, B. and Sheffer, A. (2007), "Mesh parameterization: theory and practice", *SIGGRAPH 2007 Course Notes*, ACM Press, San Diego, CA, pp. 6-115.

Kharevych, L., Springborn, B. and Schröder, P. (2006), "Discrete conformal mappings via circle patterns", *ACM Transactions on Graphics*, Vol. 25 No. 2, pp. 412-438.

Lee, Y., Kim, H.S. and Lee, S. (2002), "Mesh parameterization with a virtual boundary", *Computers & Graphics*, Vol. 26 No. 5, pp. 677-686.

Lévy, B., Petitjean, S., Ray, N. and Maillot, J. (2002), "Least squares conformal maps for automatic texture atlas generation", *ACM Transactions on Graphics*, Vol. 21 No. 3, pp. 362-371.

Li, B., Zhang, X., Zhou, P. and Hu, P. (2010), "Mesh parameterization based on one-step inverse forming", *Computer-Aided Design*, Vol. 42 No. 7, pp. 633-640.

Li, Z., Jin, Y., Jin, X. and Ma, L. (2012), "Approximate straightest path computation and its application in parameterization", *The Visual Computer*, Vol. 28 No. 1, pp. 63-74.

Liu, L., Zhang, L., Xu, Y., Gotsman, C. and Gortler, S.J. (2008), "A local/global approach to mesh parameterization", *Proceedings of the Symposium on Geometry Processing*, *Eurographics Association*, Aire-la-Ville, pp. 1495-1504.

Mejia, D., Ruiz-Salguero, O. and Cadavid, C.A. (2016), "Hessian eigenfunctions for triangular mesh parameterization", *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Rome, pp. 75-82.

Myles, A. and Zorin, D. (2012), "Global parametrization by incremental flattening", *ACM Transactions on Graphics*, Vol. 31 No. 4, pp. 109:1-109:11.

Myles, A. and Zorin, D. (2013), "Controlled-distortion constrained global parametrization", *ACM Transactions on Graphics*, Vol. 32 No. 4, pp. 105:1-105:14.

Nieser, M., Palacios, J., Polthier, K. and Zhang, E. (2012), "Hexagonal global parameterization of arbitrary surfaces", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 18 No. 6, pp. 865-878.

Papadimitriou, C.H. and Steiglitz, K. (1982), *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Upper Saddle River, NJ.

Pietroni, N., Massimiliano, C., Cignoni, P. and Scopigno, R. (2011), "An interactive local flattening operator to support digital investigations on artwork surfaces", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17 No. 12, pp. 1989-1996.

Ravindran, A., Ragsdell, K.M. and Reklaitis, G.V. (2007), "Functions of several variables", in *Engineering Optimization: Methods and Applications*, John Wiley & Sons, Hoboken, NJ, pp. 78-148.

Roweis, S.T. and Saul, L.K. (2000), "Nonlinear dimensionality reduction by locally linear embedding", *Science*, Vol. 290 No. 5500, pp. 2323-2326.

Ruiz, O.E., Mejia, D. and Cadavid, C.A. (2015), "Triangular mesh parameterization with trimmed surfaces", *International Journal on Interactive Design and Manufacturing*, Vol. 9 No. 4, pp. 303-316.

Sander, P.V., Gortler, S.J., Snyder, J. and Hoppe, H. (2002), "Signal-specialized parametrization", *Proceedings of the 13th Eurographics Workshop on Rendering*, *Eurographics Association*, Aire-la-Ville, pp. 87-98.

Sheffer, A. and de Sturler, E. (2001), "Parameterization of faceted surfaces for meshing using angle-based flattening", *Engineering with Computers*, Vol. 17 No. 3, pp. 326-337.

Sheffer, A. and Hart, J.C. (2002), "Seamster: inconspicuous low-distortion texture seam layout", *Proceedings of the Conference on Visualization '02*, *IEEE Computer Society*, Washington, DC, pp. 291-298.

Sheffer, A., Praun, E. and Rose, K. (2006), "Mesh parameterization methods and their applications", *Foundations and Trends® in Computer Graphics and Vision*, Vol. 2 No. 2, pp. 105-171.

Sheffer, A., Lévy, B., Mogilnitsky, M. and Bogomyakov, A. (2005), "ABF++: fast and robust angle based flattening", *ACM Transactions on Graphics*, Vol. 24 No. 2, pp. 311-330.

Smith, J. and Schaefer, S. (2015), "Bijective parameterization with free boundaries", *ACM Transactions on Graphics*, Vol. 34 No. 4, pp. 70:1-70:9.

Springborn, B., Schröder, P. and Pinkall, U. (2008), "Conformal equivalence of triangle meshes", *ACM Transactions on Graphics*, Vol. 27 No. 3, pp. 77:1-77:11.

Su, K., Cui, L., Qian, K., Lei, N., Zhang, J., Zhang, M. and Gu, X.D. (2016), "Area-preserving mesh parameterization for poly-annulus surfaces based on optimal mass transportation", *Computer Aided Geometric Design*, Vol. 46, pp. 76-91, available at: www.sciencedirect.com/science/article/pii/S0167839616300528

Sun, X. and Hancock, E.R. (2008), "Quasi-isometric parameterization for texture mapping", *Pattern Recognition*, Vol. 41 No. 5, pp. 1732-1743.

Tenenbaum, J.B., de Silva, V. and Langford, J.C. (2000), "A global geometric framework for nonlinear dimensionality reduction", *Science*, Vol. 290 No. 5500, pp. 2319-2323.

Ueda, K. and Yamashita, N. (2012), "Global complexity bound analysis of the Levenberg-Marquardt method for nonsmooth equations and its application to the nonlinear complementarity problem", *Journal of Optimization Theory and Applications*, Vol. 152 No. 2, pp. 450-467.

Wang, Z., Luo, Z.X., Zhang, J.L. and Saucan, E. (2016), "ARAP++: an extension of the local/global approach to mesh parameterization", *Frontiers of Information Technology & Electronic Engineering*, Vol. 17 No. 6, pp. 501-515.

Yoshizawa, S., Belyaev, A. and Seidel, H.P. (2004), "A fast and simple stretch-minimizing mesh parameterization", *Proceedings of the Shape Modeling International 2004*, *IEEE Computer Society*, Washington, DC, pp. 200-208.

Zayer, R., Lévy, B. and Seidel, H.P. (2007), "Linear angle based parameterization", *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, *Eurographics Association*, Aire-la-Ville, pp. 135-141.

Zayer, R., Rössl, C. and Seidel, H.P. (2005), "Setting the boundary free: a composite approach to surface parameterization", *Proceedings of the Third Eurographics Symposium on Geometry Processing*, *Eurographics Association*, Aire-la-Ville.

Zhang, Z. and Zha, H. (2002), "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment ", *SIAM Journal on Scientific Computing*, Vol. 26 No. 1, pp. 313-338.

Zhao, X., Su, Z., Gu, X.D., Kaufman, A., Sun, J., Gao, J. and Luo, F. (2013), "Area-preservation mapping using optimal mass transport", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 19 No. 12, pp. 2838-2847.

Zhu, X.F., Hu, P., Ma, Z.D., Zhang, X., Li, W., Bao, J. and Liu, M. (2013), "A new surface parameterization method based on one-step inverse forming for isogeometric analysis-suited geometry", *The International Journal of Advanced Manufacturing Technology*, Vol. 65 No. 9, pp. 1215-1227.

Zou, G., Hu, J., Gu, X. and Hua, J. (2011), "Authalic parameterization of general surfaces using lie advection", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17 No. 12, pp. 2005-2014.

Corresponding author
Diego A. Acosta can be contacted at: dacostam@eafit.edu.co