Hessian Eigenfunctions for Triangular Mesh Parameterization

Daniel Mejia[1], Oscar E. Ruiz[1] and Carlos A. Cadavid[1]

[1]*CAD CAM CAE Laboratory, EAFIT University, Cra 49 No-7 Sur-50, Medellín, Colombia*

{dmejiap,oruiz,ccadavid}@eafit.edu.co

Keywords: Applied DIfferential Geometry, Dimensionality Reduction, Hessian Locally Linear Embedding, Manifold Learning, Mesh Parameterization

Abstract: Hessian Locally Linear Embedding (HLLE) is an algorithm that computes the nullspace of a Hessian functional $\mathcal{H}$ for Dimensionality Reduction (DR) of a sampled manifold M. This article presents a variation of classic HLLE for parameterization of 3D triangular meshes. Contrary to classic HLLE which estimates local Hessian nullspaces, the proposed approach follows intuitive ideas from Differential Geometry where the local Hessian is estimated by quadratic interpolation and a partition of unity is used to join all neighborhoods. In addition, local average triangle normals are used to estimate the tangent plane T_xM at $x \in M$ instead of PCA, resulting in local parameterizations which reflect better the geometry of the surface and perform better when the mesh presents sharp features. A high frequency dataset (*Brain*) is used to test our algorithm resulting in a higher rate of success (96.63%) compared to classic HLLE (76.4%).

1 INTRODUCTION

Dimensionality Reduction (DR) takes a d-manifold $M \subset \mathbb{R}^D$ and computes a map $\mathbf{h} : M \rightarrow \mathbb{R}^d$ such that: 1) $\mathbf{h}$ is bijective and 2) $\mathbf{h}$ and $\mathbf{h}^{-1}$ are continuous. Therefore, $\mathbf{h}$ is an homeomorphism and the image of M under $\mathbf{h}$ is a DR of M.

Mesh Parameterization can be seen as a particular case of DR where $M \subset \mathbb{R}^3$ is a triangular mesh of a 2-manifold (i.e. $D = 3$ and $d = 2$). Triangular meshes are very common data structures in CAD CAM CAE applications and parameterization of such meshes is relevant for areas such as: reverse engineering, tool path planning, feature detection, etc.

A natural way to handle Mesh Parameterization is to attack the problem from the point of view of DR. Classic HLLE (Hessian Locally Linear Embedding) (Donoho and Grimes, 2003) is an algorithm which proposes to compute a DR of M by computing the eigenvectors of a Hessian functional. This article proposes a modification for the classic HLLE which can be applied to triangular meshes. Our proposed approach computes a partition of unity on M and estimates the tangent Hessian on each neighborhood N_i of M by interpolating any function f with second degree polynomials. In addition, local average triangle normals are used to compute the tangent local plane T_xM of M which is more consistent than Principal Component Analysis (PCA) specially for surfaces with sharp features.

The remainder of this article is organized as follows: Section 2 reviews the relevant literature. Section 3 describes the implemented methodology. Section 4 discusses and compares the results of the proposed approach against classic HLLE. Section 5 concludes the paper and introduces what remains for future work.

2 LITERATURE REVIEW

Given a set of points $X = [x_1, x_2, \ldots, x_n] \subset \mathbb{R}^D$ lying on a d-manifold M, DR seeks a homeomorphic function $\mathbf{h} : M \rightarrow \mathbb{R}^d$ such that the set of points $[\mathbf{h}(x_1), \mathbf{h}(x_2), \ldots, \mathbf{h}(x_n)] \subset \mathbb{R}^d$ compose a DR of X. For the rest of the article we assume $D = 2$ and $d = 3$, turning the DR problem into a Mesh Parameterization one.

The most popular algorithm for DR is the Principal Component Analysis (PCA). PCA is a linear algorithm which parameterizes M by projecting X onto a plane, which is only a valid parameterization if $\mathbf{h}$ is linear. However, this assumption limits the algorithm making it useful only for trivial cases.

For nonlinear manifolds, other approaches have been proposed in the literature. For example, Isomap (Tenenbaum et al., 2000) attempts to compute an isometric parameterization of M by computing the geodesic distances in M and reproducing them in the parameter space. Isomap has been succesfully applied in the context of Mesh Parameterization (Sun and Hancock, 2008; Ruiz et al., 2015). Usually a shortest path algorithm such as Dijktra's or Floyd's is used to estimate geodesic distances which fail for non-convex manifolds such as surfaces with holes.

Spectral theory is an important branch of graph theory where several DR algorithms have been derived. Laplacian Eigenmaps (Belkin and Niyogi, 2003) computes the Laplacian matrix which acts over any function defined on the graph of M measuring its curvature. Diffusion Maps (DM) (Lafon and Lee, 2006) computes the Markov matrix which estimates the transition probability between vertices of the graph of M. DR is achieved in both cases by computing the eigenvectors of these matrices respectively. Spectral algorithms preserve topologic properties of the underlying graph keeping adjacent points near in the parameter space. However, these algorithms usually fail to preserve geometric properties which becomes important in Mesh Parameterization applications.

Other DR algorithms focus on a more local approach where each neighborhood is first parameterized locally and then all the neighborhoods are aligned in the parameter space trying to preserve geometric properties. Locally Linear Embedding (LLE) (Roweis and Saul, 2000) expresses each point in M as a linear combination of its neighborhs and then computes the DR attempting to preserve such structure in the parameter space for all the points. Similarly, Local Tangent Space Alignment (LTSA) (Zhang and Zha, 2002) projects each neighborhood onto the tangent plane via PCA and then attempts to align all the neighborhoods in the parameter space using rigid transformations. These algorithms highly preserve geometric properties and perform well for non-convex manifolds. However, they expect that each neighborhood lies on a linear subspace which fails at sharp features of 3D meshes resulting in non-bijective parameterizations. Most Mesh Parameterization algorithms also follow this idea by aligning triangles in the parameter space preserving geometric properties for each triangle (Floater and Hormann, 2005; Hormann et al., 2007).

2.1 Classic Hessian Locally Linear Embedding (HLLE)

Classic Hessian Locally Linear Embedding (Donoho and Grimes, 2003) is a DR algorithm which combines ideas from LLE and LTSA with ideas from discrete differential geometry. HLLE computes a local parameterization of each neighborhood using PCA. But instead of aligning all neighborhoods by a rigid mapping, HLLE estimates a Hessian functional $\mathcal{H}$ on M (similar to Laplacian Eigenmaps which estimates a Laplacian functional) and computes the DR of M by estimating the kernel of $\mathcal{H}$, i.e. $\ker(\mathcal{H}) = \{f|\mathcal{H}f = 0\}$.

In order to compute $\mathcal{H}$, the tangent Hessian $\mathbf{H}_x^{\tan}$ must be defined (Donoho and Grimes, 2003):

$$\mathbf{H}_x^{\tan} f = \begin{bmatrix} \frac{\partial^2 f}{\partial b_1^2} & \frac{\partial^2 f}{\partial b_1 \partial b_2} \\ \frac{\partial^2 f}{\partial b_2 \partial b_1} & \frac{\partial^2 f}{\partial b_2^2} \end{bmatrix}, \tag{1}$$

where $b_1, b_2 \in \mathbb{R}^3$ is an orthonormal basis for the tangent plane T_xM at x. If $\mathbf{f} = \{f_1, f_2, \ldots, f_n\}$ with $f_i = f(x_i)$ is the function f restricted to the set of points X, then the Hessian functional $\mathcal{H}$ is defined as (Donoho and Grimes, 2003):

$$\mathcal{H}f = \int_M \|\mathbf{H}_x^{\tan} f\|_F^2 \mathrm{dA} = \sum_{i=1}^{n} \int_{M_i} \phi_i \|\mathbf{H}_x^{\tan} f\|_F^2 \mathrm{dA} \approx \mathbf{f}^{\mathrm{T}}\mathbf{K}\mathbf{f} \tag{2}$$

where $f \in \mathcal{C}^2(M)$ is a smooth function defined on M, $\|\cdot\|_F$ is the Frobenius norm, dA is a surface differential, $(M_i,\ \phi_i)$ is a partition of unity of M and $\mathbf{K} = (\mathbf{K}_1 + \mathbf{K}_2, \ldots, \mathbf{K}_n)$ is the discrete Hessian estimator computed by adding each local Hessian estimator $\mathbf{K}_i$ at each neighborhood N_i.

Since the matrix $\mathbf{K}$ approximates the Hessian functional on M, the kernel of $\mathcal{H}$ can be estimated by solving the following minimization problem (Donoho and Grimes, 2003):

$$\begin{aligned} &\mathbf{h}_1 = \arg\min_{\mathbf{f}} \mathbf{f}^{\mathrm{T}}\mathbf{K}\mathbf{f}, \qquad \mathbf{h}_2 = \arg\min_{\mathbf{f}} \mathbf{f}^{\mathrm{T}}\mathbf{K}\mathbf{f} \\ &\text{s.t.} \\ &\|\mathbf{h}_i\| = 1, \qquad \mathbf{h}_i \perp \mathbf{1}, \qquad \mathbf{h}_1 \perp \mathbf{h}_2, \\ &i = 1,2 \end{aligned} \tag{3}$$

where $\mathbf{h}_1 = [h_1(x_1), \ldots, h_1(x_n)]^{\mathrm{T}}$ and $\mathbf{h}_2 = [h_2(x_1), \ldots, h_2(x_n)]^{\mathrm{T}}$ are the respective coordinates of X in the parameter space. The constant function $f(x) = c$, $c \in \mathbb{R}$, is in the kernel of $\mathcal{H}$ (the Hessian of any constant function is 0 as per eq. (1)). Therefore, the constraint $\mathbf{h}_i \perp \mathbf{1}$ avoids collapsing all the vertices to a single point. The constraint $\mathbf{h}_1 \perp \mathbf{h}_2$ guarantees linear independence which avoids collapsing the surface into a line. The constraint $\|\mathbf{h}_i\| = 1$ fixes the scale of the solution.

Eq. (3) can be solved by computing the eigenvectors of $\mathbf{K}$ associated to the second and third lowest eigenvalue (the eigenvector associated to the lowest eigenvalue corresponds to the constant function $\mathbf{1}$).

Basically, classic HLLE algorithm consists of: 1) estimate the local Hessian functionals $\mathbf{K}_1, \mathbf{K}_2 \ldots, \mathbf{K}_n$ and the Hessian functional $\mathbf{K} = \mathbf{K}_1 + \mathbf{K}_2 + \cdots + \mathbf{K}_n$ and 2) compute the eigenvectors of $\mathbf{K}$ with the smallest eigenvalue (Donoho and Grimes, 2003).

Like LLE and LTSA, classic HLLE may present problems for datasets with sharp features resulting in non-bijective mappings. In addition, the computation of the matrix $\mathbf{K}_i$ which estimates the local Hessian functional $\mathcal{H}|_i$ is not consistent with the definition in eq. (2) as only Hessian nullspaces are computed.

Conclusions of the literature review. In Mesh Parameterization applications, the preservation of geometric properties is a priority over topology preservation. Algorithms such as Laplacian Eigenmaps and DM present highly distorted parameterizations. Therefore, algorithms that preserve geometric properties such as Isomap, LLE, LTSA and classic HLLE are more effective. However, these algorithms present drawbacks such as the inability to work with convex datasets or high frequency datasets, which are very common in engineering applications. Mesh Parameterization algorithms do not face such problems. However they are only restricted to triangular meshes.

To partially overcome these problems, this article proposes a variation of the classic HLLE algorithm for parameterization of triangular meshes. Classic HLLE algorithm is selected for this purpose since such algorithm has provided better experimental results for Mesh Parameterization than other DR algorithms (Ruiz et al., 2015). Also, since HLLE is a DR algorithm, the proposed approach can be easily extended to meshes composed of non-triangular faces posing a potential advantage over traditional Mesh Parameterization algorithms.

3 METHODOLOGY

In order to parameterize M, we propose to follow the same idea of the classic HLLE which is described in section 2.1 (Donoho and Grimes, 2003): 1) estimate the tangent Hessian $\mathbf{H}_x^{\tan}$ and the Hessian functional $\mathbf{K}_i$ at each N_i, 2) estimate the Hessian functional $\mathcal{H}$ on M as per eq. (2) and 3) estimate the kernel of $\mathcal{H}$ for Mesh Parameterization via eigendecomposition. The algorithm is briefly described below:

1. For each neighborhood estimate the tangent plane T_xM at x_i by computing the local average normal vector $\overline{n}_i$ and compute a local parameterization O_i by projecting N_i onto T_xM.
2. Estimate the tangent Hessian $\mathbf{H}_x^{\tan} f$ and $\|\mathbf{H}_x^{\tan} f\|_{\mathrm{F}}^2$ at x_i by quadratic interpolation.
3. Apply the partition of unity ϕ_i to estimate the local Hessian functional $\int_{M_i} \phi_i \|\mathbf{H}_x^{\tan}\|_{\mathrm{F}}^2 \mathrm{dA} \approx \mathbf{f}^{\mathrm{T}} \mathbf{K}_i \mathbf{f}$.
4. Estimate the global Hessian functional $\mathcal{H} \approx \mathbf{K} = \sum_{i=1}^{n} \mathbf{K}_i$.
5. Compute two orthogonal functions $\mathbf{h}_1$ and $\mathbf{h}_2$ which solve the optimization problem posed in eq. (3) by eigendecomposition of the matrix $\mathbf{K}$.

The steps of the algorithm are detailed below.

3.1 Tangent Plane T_xM and Local Parameterization O_i

In order to estimate the tangent Hessian $\mathbf{H}_x^{\tan}$ at x_i, the tangent plane T_xM at x_i is estimated. Classic HLLE estimates T_xM applying PCA on N_i. However, we propose to estimate T_xM using the information of the triangulation T as follows: let $\{t_{i_1}, t_{i_2}, \ldots\}$ and $\{n_{i_1}, n_{i_2}, \ldots\}$ be the set of triangles adjacent to x_i and their corresponding normal vectors respectively. Set T_xM as the plane with origin x_i and normal $\overline{n}_i$ (where $\overline{n}_i$ is the average of $\{n_{i_1}, n_{i_2}, \ldots\}$). Finally, O_i is computed by projecting N_i onto T_xM.

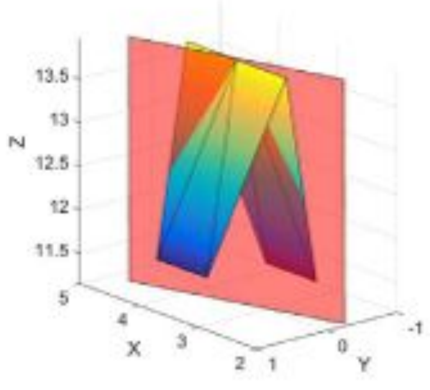
Using the local average adjacent normals to compute T_xM usually results in better approximations of the tangent plane than PCA and if N_i belongs to a sharp region, PCA may fail to recover a bijective parameterization as illustrated in fig. 1 while the local average normals plane results in a bijective parameterization (fig. 2). These local parameterizations affect the resulting global parameterization in the sense that local non-bijectivity results in a folding of surface in the global parameterization.

3.2 Tangent Hessian $\mathbf{H}_x^{\tan} f$ and $\|\mathbf{H}_x^{\tan} f\|_{\mathbf{F}}^2$

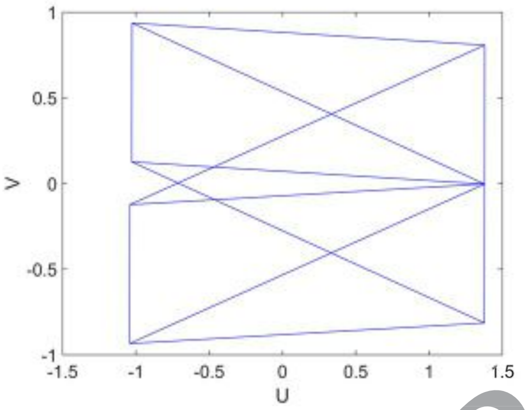
The definition of tangent Hessian in eq. (1) requires a smooth function defined on M. Quadratic interpolation is used in order to estimate such Hessian in a discrete surface. Let $[b_1, b_2]$ be an orthonormal basis of T_xM at x_i. Therefore, any point p on T_xM at x_i can be expressed as $p = ub_1 + vb_2 + x_i$. Let $\{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$ and $\{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ be the corresponding coordinates of O_i in this basis. If $\mathbf{f}_i = \{f_{i_1}, f_{i_2}, \ldots, f_{i_k}\}$ are the values of f restricted to N_i, then f can be interpolated at T_xM by a second order polynomial as follows:

$$f(u,v) = \sum_{j=1}^{k} \mathbb{N}_{i_j}(u,v) f_{i_j}, \tag{4}$$

with $\mathbb{N}_{i_j}(u,v) = \alpha_j u^2 + \beta_j uv + \gamma_j v^2 + \delta_j u + \varepsilon_j v + \zeta_j$. Since $f(u_j, v_j) = f_{i_j}$, the interpolation functions $\mathbb{N}_{i_j}$

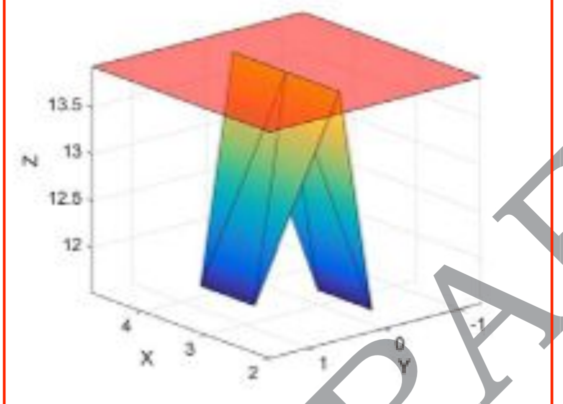


(a) Sharp neighborhood and PCA plane (red).

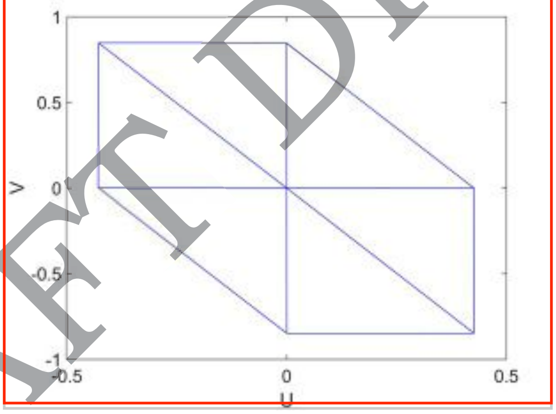


(b) Non-bijective local parameterization O_i.

Figure 1: Estimation of T_xM and local paramaterization via PCA for a sharp feature.



(a) Sharp neighborhood and average normal plane (red).



(b) Bijective local parameterization O_i.

Figure 2: Estimation of T_xM and local paramaterization via local average normal for a sharp feature.

are required to satisfy $\mathbb{N}_{i_j}(u_{i_l}, v_{i_l}) = 1$ if $j = l$ and $\mathbb{N}_{i_j}(u_{i_l}, v_{i_l}) = 0$ if $j \neq l$ (fig. 3). The coefficients of $\mathbb{N}_{i_j}$ are computed by solving the arising linear system of equations in a least squares sense. Afterwards, equation (1) can be approximated as:

$$\mathbf{H}_x^{\tan} f \approx \begin{bmatrix} 2\boldsymbol{\alpha}_i^{\mathrm{T}}\mathbf{f}_i & \boldsymbol{\beta}_i^{\mathrm{T}}\mathbf{f}_i \\ \boldsymbol{\beta}_i^{\mathrm{T}}\mathbf{f}_i & 2\boldsymbol{\gamma}_i^{\mathrm{T}}\mathbf{f}_i \end{bmatrix}, \tag{5}$$

where $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$ and $\boldsymbol{\gamma}_i$ are column vectors with the corresponding coefficients of the quadratic terms in eq. (4). Therefore, the norm of the tangent Hessian can be estimated as $\|\mathbf{H}_x^{\tan} f\|_{\mathrm{F}}^2 \approx \mathbf{f}_i^{\mathrm{T}}\mathbf{C}_i\mathbf{f}_i$, where $\mathbf{C}_i$ is a symmetric matrix defined as:

$$\mathbf{C}_i = 4\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^{\mathrm{T}} + 2\boldsymbol{\beta}_i\boldsymbol{\beta}_i^{\mathrm{T}} + 4\boldsymbol{\gamma}_i\boldsymbol{\gamma}_i^{\mathrm{T}} \tag{6}$$

3.3 Partition of Unity $\boldsymbol{\phi}$ and Local Hessian Functional $\mathbf{K}_i$

Eq. (2) requires a partition of unity (M_i, ϕ_i) defined on M. A partition of unity $\boldsymbol{\phi} = \{\phi_1, \phi_2, \ldots, \phi_n\}$ is a set of functions satisfying the following properties:

1. M_i is an open subset of M.
2. $\bigcup_{i=1}^n M_i = M$.
3. $\phi_i : M \to [0, 1]$.
4. $\phi_i(x) = 0$ if $x \notin M_i$.
5. $\sum_{i=1}^n \phi_i(x) = 1$ for all $x \in M$.

A partition of unity for M can be build as a set of piecewise linear functions such that for N_i, ϕ_i is defined as:

$$\phi_i(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}, \qquad \forall j = 1, 2, \ldots, n. \tag{7}$$

By its definition in eq. (7), ϕ_i vanishes at other neighborhoods. Therefore:

$$\int_M \phi_i \mathrm{dA} = \int_{M_i} \phi_i \mathrm{dA} = \frac{1}{3}\sum_j A_{i_j}, \tag{8}$$

where A_{i_j} is the area of the j-th adjacent triangle of x_i. It is not hard to check that eq. (7) satisfies the

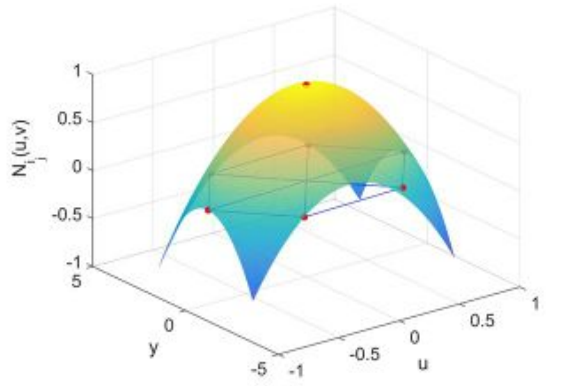


Figure 3: Quadratic interpolation function $\mathbb{N}_{i_j}$ at N_i such that $\mathbb{N}_{i_j}(x_{i_k}) = 1$ if $j = k$, 0 otherwise.

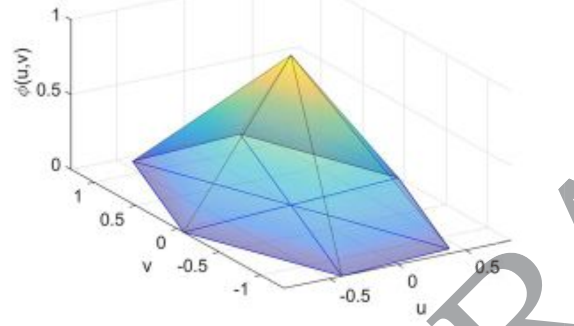


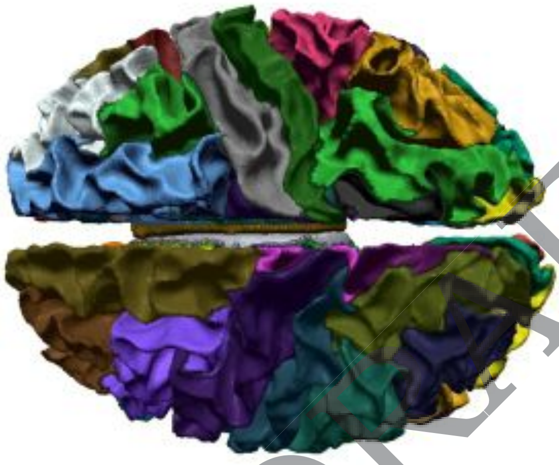
Figure 4: Partition of unity ϕ_i for a selected neighborhood N_i. ϕ_i equals to 1 at x_i and vanishes to 0 at adjacent points.

properties of a partition of unity if M is a triangular mesh.

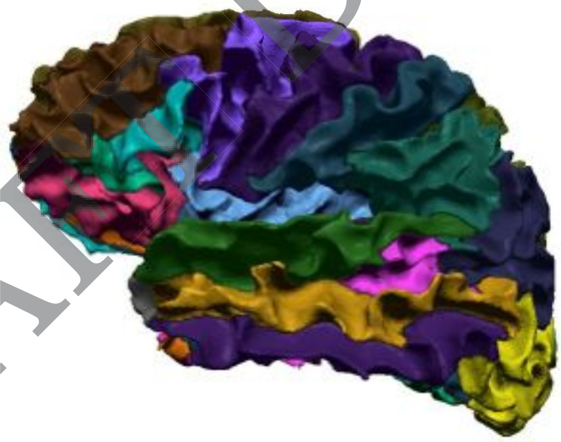
Finally, from eqs. (6) and (8) the local Hessian functional in eq. (2) can be estimated:

$$\int_{M_i} \phi_i \|\mathbf{H}^{\tan}_{\mathbf{x}} f\|_F^2 \mathrm{dA} \approx \left(\int_{M_i} \phi_i \mathrm{dA}\right) \mathbf{f}_i^{\mathrm{T}} \mathbf{C}_i \mathbf{f}_i = \left(\frac{1}{3}\sum_j A_{i_j}\right) \mathbf{f}_i^{\mathrm{T}} \mathbf{C}_i \mathbf{f}_i. \quad (9)$$

The matrix $\left(\frac{1}{3}\sum_j A_{i_j}\right)\mathbf{C}_i$ estimates the local Hessian functional for any $\mathbf{f}_i$. Therefore, the matrix $\mathbf{K}_i$ is built as an $n \times n$ matrix which has the terms of $\left(\frac{1}{3}\sum_j A_{i_j}\right)\mathbf{C}_i$ at the indices dictated by (N_i, N_i), and zeros elsewhere.



(a) *Brain* upper view.



(b) *Brain* side view.

Figure 5: Segmented *Brain* dataset.

3.4 Global Hessian $\mathcal{H}$ and Parameterization of M

The Hessian functional is estimated exactly as described in (Donoho and Grimes, 2003) by adding each local Hessian: $\mathcal{H} \approx \mathbf{K} = \sum_i \mathbf{K}_i$. Finally, the parameterization $\mathbf{h}_1$, $\mathbf{h}_2$ of M is achieved by solving the minimization problem in eq. (3) via eigendecomposition of the matrix $\mathbf{K}$.

4 RESULTS AND DISCUSSION

In this section we present and discuss the parameterization results obtained for the segmented *Brain* dataset (Desikan et al., 2006) (fig. 5) by classic HLLE and our algorithm. The *Brain* dataset presents several challenges in terms of Mesh parameterization given

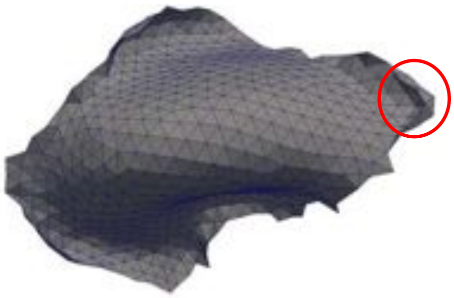


Figure 6: *Left Hemisphere - Frontal Pole* mesh. The red ellipse marks a high frequency zone.

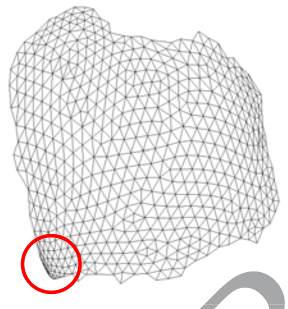
the high curvatures and the low developability of the surface. We remeshed all the sub-meshes and some of them were also partitioned manually prior to parameterization. From the 89 sub-meshes, classic HLLE computed only 68 (76.40%) bijective parameterizations while our algorithm computed 86 (96.63%) bijective mappings.

The *Left Hemisphere - Frontal Pole* sub-mesh (fig. 6) presents a high frequency zone near a corner. Fig. **??** presents the parameterization results obtained by classic HLLE and our algorithm. As described in section 3.1 Classic HLLE parameterization (fig. 7(a)) computes local non-bijective parameterizations at such sharp zone. As a consequence, the parameterized surface folds as detailed in fig. 7(b) resulting in a non-bijective parameterization. On the other hand, our algorithm does not face this problem and correctly unfolds the surface recovering a bijective parameterization (figs. 7(c) and 7(d)).

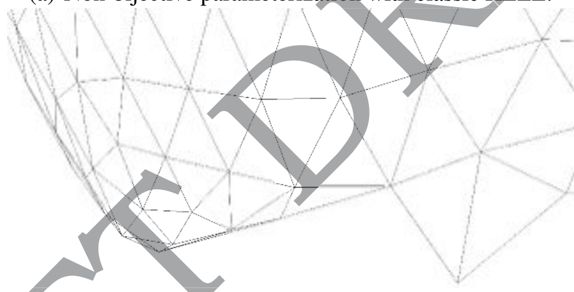
Figs. 8 and 9 present the results for two sub-meshes with several sharp sections (figs. 8(a) and 9(a)) using the classic HLLE algorithm and our algorithm. Classic HLLE fails to adequately results in non-bijective mappings for both meshes as sharp zones are locally non-bijective (figs. 8(b) and 9(b)). Again, our algorithm does not face this problem resulting in bijective mappings for both cases (figs. 8(c). Additionally, less shape distortion can be evidenced in the *Left Hemisphere - Rostral Anterior Cingulate* dataset compared to the classic HLLE algorithm (figs. 8(b) and 8(c)) due to the explicit computation of the local Hessian functional in our algorithm as opposed to classic HLLE.

Results for other sub-meshes of the *Brain* are presented in fig. 10. The texture map of a chessboard pattern illustrates the angular distortion of the respective parameterization where less local distortion is present if the corners of the mapped rectangles are near 90 degrees. All the four parameterizations are bijective despite the high frequencies of the sub-meshes.

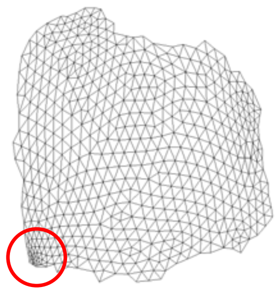
Highly non-developable meshes still pose a prob-



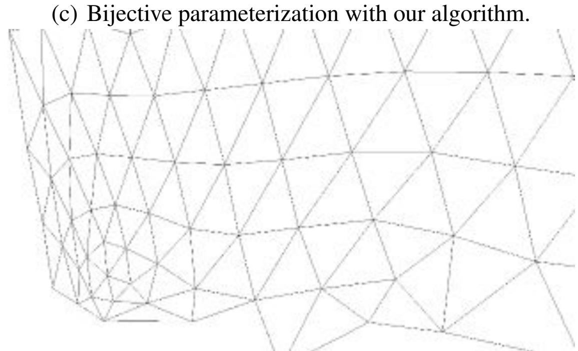
(a) Non-bijective parameterization with classic HLLE.



(b) Zoom into high frequency zone for classic HLLE (non-bijective).

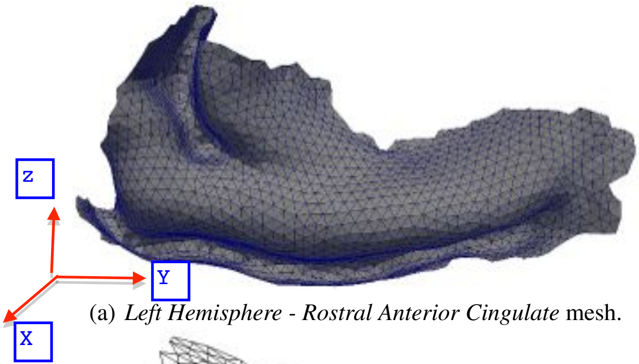


(c) Bijective parameterization with our algorithm.

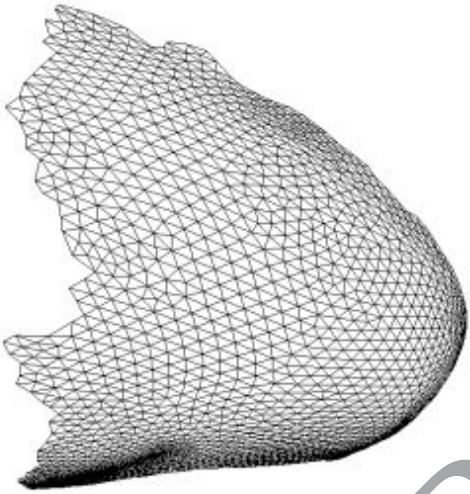


(d) Zoom into high frequency zone for our algorithm (bijective).

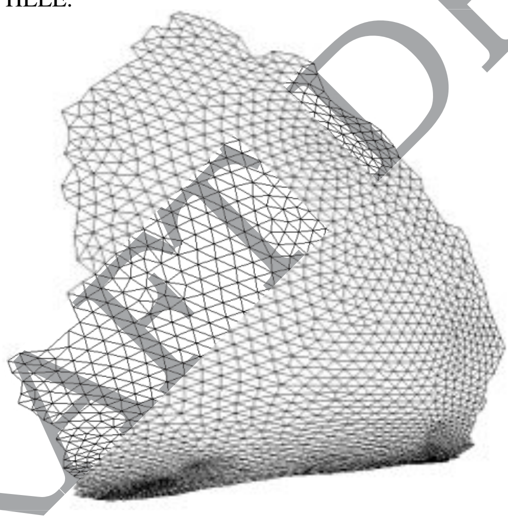
Figure 7: Parameterization results for the *Left Hemisphere - Frontal Pole* mesh. The red ellipse marks the high frequency zone.



(a) *Left Hemisphere - Rostral Anterior Cingulate* mesh.



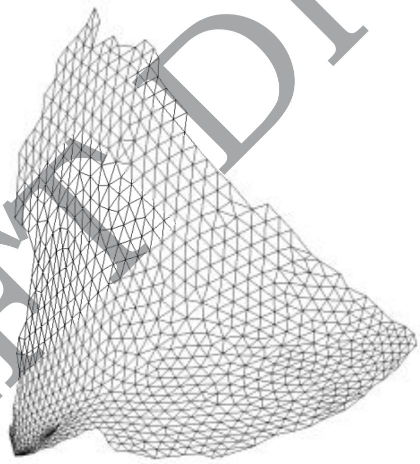
(b) Non-bijective parameterization with classic HLLE.



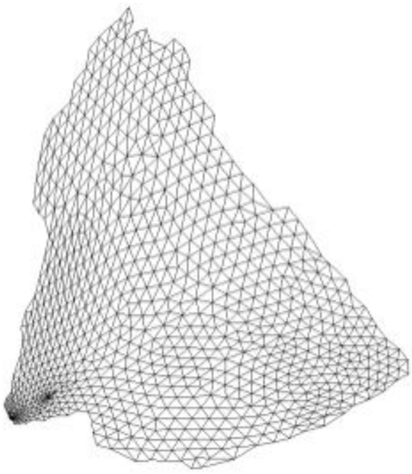
(c) Bijective parameterization with our algorithm.

Figure 8: Parameterization results for the *Left Hemisphere - Rostral Anterior Cingulate* mesh with classic HLLE and our algorithm.

(a) *Right Hemisphere - Temporal Pole* mesh.



(b) Non-bijective parameterization with classic HLLE.



(c) Bijective parameterization with our algorithm.

Figure 9: Parameterization results for the *Right Hemisphere - Temporal Pole* mesh with classic HLLE and our algorithm.

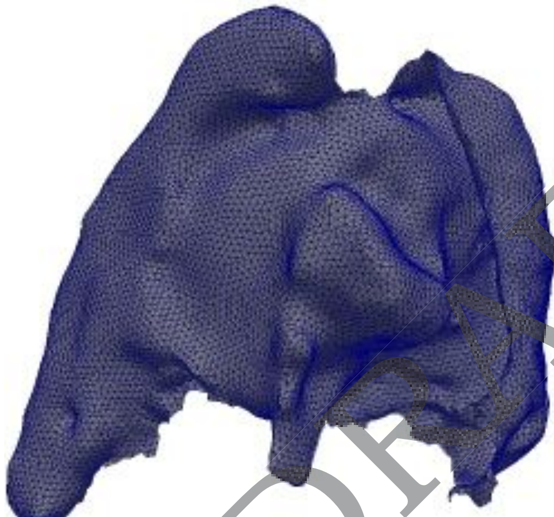
(a) Left Hemisphere - Inferior Parietal texture map.

(b) Left Hemisphere - Rostral Middle Frontal texture map.

(c) Right Hemisphere - Inferior Parietal texture map.

(d) Right Hemisphere - Lateral Occipital bijective texture map.

Figure 10: Texture map for several bijective mappings for the *Brain* dataset.



(a) *Right Hemisphere - Unknown* mesh.

(b) *Right Hemisphere* - ~~*Unknown*~~ non-bijective parameterization with our algorithm.

Figure 11: Failure test for our algorithm.

lem to our algorithm. Fig. 11 presents a case of the *Brain* dataset where the algorithm fails to recover a bijective parameterization. In this case the parameterization degenerates near the boundary (i.e. triangles overlap) in the parameter space due to the high non-developability of such zones in the surface.

5 CONCLUSIONS

This article presents a variation of the classic HLLE algorithm for parameterization of triangular meshes. Classic HLLE was selected for this purpose since it has shown experimentally better results than other DR algorithms for Mesh Parameterization. An intuitive approach from Differential Geometry is followed by estimating locally the tangent Hessian with quadratic interpolation and computing a partition of unity for the triangular mesh M as opposed to classic HLLE approach. In addition, each local parameterization is achieved by projecting onto the local average triangle normals plane instead of the usual PCA, which reflects better the local geometry of the surface specially in cases of sharp features.

The algorithm was tested with the *Brain* dataset which consists of a set of sub-meshes with high curvatures. The resulting parameterizations were compared with the results of classic HLLE. Our algorithm presented a higher rate of success (96.63%) against classic HLLE (76.40%).

5.1 Ongoing Work

Segmentation of complex meshes with high gaussian curvatures into smaller ones increases the probability of finding bijective parameterizations. Therefore, automatic mesh segmentation for this task becomes crucial for parameterization of large and complex datasets.

ACKNOWLEDGEMENTS

This research has been funded by the Research Group and College of Engineering at Universidad EAFIT, Colombia.

REFERENCES

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396.

Desikan, R. S., Sgonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., Buckner, R. L., Dale, A. M., Maguire, R. P., Hyman, B. T., Albert, M. S., and Killiany, R. J. (2006). An automated labeling system for subdividing the human cerebral cortex on {MRI} scans into gyral based regions of interest. *NeuroImage*, 31(3):968 – 980.

Donoho, D. L. and Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596.

Floater, M. and Hormann, K. (2005). Surface parameterization: a tutorial and survey. In Dodgson, N., Floater, M., and Sabin, M., editors, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 157–186. Springer Berlin Heidelberg.

Hormann, K., Lévy, B., and Sheffer, A. (2007). Mesh parameterization: Theory and practice video files associated with this course are available from the citation page. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, New York, NY, USA. ACM.

Lafon, S. and Lee, A. (2006). Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1393–1403.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):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 (IJIDeM)*, pages 1–14.

Sun, X. and Hancock, E. R. (2008). Quasi-isometric parameterization for texture mapping. *Pattern Recognition*, 41(5):1732 – 1743.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319.

Zhang, Z. and Zha, H. (2002). Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Journal of Scientific Computing*, 26:313–338.