Direct spherical parameterization based on surface curvature

Bogdan Mocanu
Institut Télécom / Télécom SudParis,
ARTEMIS Department
UMR CNRS 8145 MAP5
Evry, France
bogdan.mocanu@it-sudparis.eu

Titus Zaharia
Institut Télécom / Télécom SudParis,
ARTEMIS Department
UMR CNRS 8145 MAP5
Evry, France
titus.zaharia@it-sudparis.eu

***Abstract*—In this paper we propose a novel spherical parameterization for closed, genus-0, two-manifold, 3D triangular meshes. The key point of our method concerns the Gaussian curvature criterion involved, which makes it possible to detect iteratively salient mesh vertices and to locally flatten them, until a sphere-like surface, adapted to a direct spherical parameterization is obtained. The experimental evaluation, carried out on a set of 3D models of various shapes and complexities, shows that the proposed method makes it possible to reduce both angle and area distortions with more than 78% and 40% respectively.**

Keywords-spherical parameterization, genus-zero 3D mesh, Gaussian curvature, barycentric coordinates.

I. INTRODUCTION

Parameterization techniques have been widely used in various computer graphics applications, including texture mapping, remeshing, object recognition, morphing...

The parameterization of a 3D surface $S \subset \mathrm{IR}^3$ is defined as a homeomorphism $\Phi : S \rightarrow D$ which maps the surface S over an appropriate 2D domain $D \subset \mathrm{IR}^2$. In the case of 3D meshes, a parameterization is defined as a piece-wise linear embedding. More precisely, let $M = (V, E, F)$ be a 3D triangular mesh, where V, E and F respectively denote the sets of vertices, edges and triangles. The parameterization of the mesh surface is completely specified by a function $\varphi : V \rightarrow D$, which associates to each vertex p_i of V a point $\varphi_i \overset{not}{=} \varphi(p_i)$ in the 2D domain D.

Let us not that the bijectivity property associated to a homeomorphism is required in order to ensure that the corresponding faces in the parametric domain D do not overlap.

In this paper, we propose a novel spherical parameterization method for genus-0, manifold 3D meshes which exploits the local surface curvature.

The rest of this paper is organized as follow. After a brief recall of the most representative state of the art parameterization methods (Section II), Section III introduces the proposed spherical mapping algorithm. In Section IV, we present and discuss the experimental results obtained for a subset of 3D objects selected from the Princeton and MPEG 7 database. Finally, Section V concludes the paper and opens perspectives of future work.

II. RELATED WORK

A first family of mesh parameterization approaches consider open 3D models that are homeomorphic to a disk. The principle consists of modeling the 3D mesh as a mass-spring system, where edges are springs connected to the mesh vertices. A generic method to embed a 3D mesh with a boundary in the plane was described by Eck *et al.* [1] as a generalization of the basic technique proposed by Tutte [2] for a planar graph.

Since then, the various methods of the literature aim at optimizing the system spring energy while ensuring a shape preserving parameterization. Different distortion measures are considered, such as angle [3] (harmonic/conformal mappings), length or area [4] (authalic mappings) deformations.

In the case of meshes with arbitrary topologies (*i.e.*, different from the unit disk) the issue of finding appropriate parameterizations becomes more complex. In particular, this issue concerns the case of closed, genus 0 meshes with spherical topology.

A first idea is to extend and adapt existing planar parameterization methods, as proposed in [5]. However, such approaches suffer from lack of robustness and efficiency.

Another approach to handle closed meshes consists of creating an artificial boundary by cutting the mesh along an appropriate path in order to obtain a patch (or sets of patches) that can be mapped onto the unit disk (*e.g.* [6], [7]). In this way, the object can be seen as an open mesh and can be parameterized with the planar mapping methods. However, finding an appropriate path that minimizes the parameterization distortion is still an open issue of research.

Other methods [8], [9], [10], [11] propose to directly parameterize closed genus-0 mesh models onto a spherical domain, since this type of objects are topological equivalent to a sphere. In practice, the process proves to be more challenging and much more complex than the planar one.

Kent *et al*. [12] introduce a technique that returns valid spherical embeddings only if the mesh has a convex shape. Unfortunately most available models have complex structure so developing new methods to resolve the overlapping problem, are necessary.

978-0-7695-4413-7/11 $26.00 © 2011 IEEE
DOI 10.1109/DMDCM.2011.61

Starting from the Kent technique, Alexa [13] develop a new vertex relaxation process that iteratively modifies each vertex position by placing it in the barycenter of its neighbors. The relaxation procedure resolves the fold-overs with reduced computational time. However, the Alexa's algorithm can breakdown into a single point and it is necessary to fix several vertices (called anchors) in the parametric domain. Without a sufficient number of adequately selected anchors the embedding may collapse. The drawback of such an approach comes from the fact that anchor vertices cannot be moved and cause triangle overlapping problems. In order to overcome such a limitation, the author propose to re-place the anchors after a number of iterations. However, such a procedure does not guarantee a valid embedding in all possible cases.

A different approach is introduced by Gotsman [14], which extends the barycentric coordinates theory to the spherical case. Unfortunately, the method needs to determine the solution of a non-linear system of equations, which is computationally complex.

In [9], Shefer *et al.* extend their previous method of planar embedding to the spherical case. The principle here is to consider rather triangle angles then vertex positions. Unfortunately, the spherical formulation is numerically much less stable than its planar version. In addition, because of the computational burden, the method can be effectively applied solely for meshes which do not exceed a few hundreds of vertices.

A different spherical parameterization technique, based on a multiresolution representation, is suggested by Praun and Hoppe [15]. First, a mesh simplification procedure is applied to the original mesh until a tetrahedron is obtained. The tetrahedron is then simply mapped onto the sphere. Next, the vertices are successively inserted on the sphere with the help of a progressive mesh sequence by applying vertex split operations. After each vertex insertion, an optimization procedure is applied in order to minimize the stretch metric of the parameterization.

The spherical parameterization method proposed in this paper extends the previous state of the art methods introduced in [12] and [13]. The principle consists of exploiting the Gaussian curvature associated to the mesh vertices. Valid spherical embeddings are obtained by locally flattening the mesh in an iterative manner, starting from vertices with maximal curvature values. This principle makes it possible to define a sequence of flattening operations that transform the initial mesh into a rounded, sphere-like surface that can be mapped onto the unit sphere. To our very best knowledge, the proposed approach is the first one that exploits the Gaussian curvature for parameterization purposes. In addition, the algorithm avoids solving complex, non-linear systems of equations and thus is computationally efficient.

The proposed technique is detailed in the next section.

III. SPHERICAL PARAMETERIZATION BASED ON SURFACE CURVATURE

Curvature measures are well-defined for smooth, C^2-differentiable surfaces [16]. However, when dealing with discrete 3D meshes, such measures need to be re-formulated appropriately. In our case, we have adopted the approximation proposed in [17]. Let p be an arbitrary mesh vertex and $\{p_i\}_{i=1}^{N_p}$ the ordered set (ring) of its adjacent vertices. The Gaussian curvature K^p at point p is then defined as:

$$K^p = \frac{3(2\pi - \sum_i \alpha_i)}{\sum_i A_i(p)}, \tag{1}$$

where α_i, is the angle $\angle(p_i, p, p_{i+1})$ and $A_i(p)$ is the area of the triangle (p_i, p, p_{i+1}).

The following two pre-processing steps are first considered.

Step 1 – Pose/size normalization:

The initial 3D mesh vertices are specified in an arbitrary Cartesian coordinate system. In order to obtain consistent measures for all meshes, we apply a PCA-based normalization [18]. The object is aligned with respect to its principal axes and scaled to the maximum distance between the object's gravity center and the mesh vertices.

Step 2 – Mesh simplification:

Analogously to the method in [15], we apply a mesh simplification procedure, in order to reduce the number of 3D mesh vertices and thus to significantly decrease the computational complexity of the parameterization process. Here, we have adopted the simplification method presented in [19] which generates high quality, simplified approximations of 3D meshes, by minimizing the quadratic error metric. Unlike [15], we stop the mesh simplification process when the Hausdorff distance [20] between the initial mesh and its simplified version exceeds a predefined threshold *Err*. In our experiments, the *Err* threshold has been set to 1-5% of the initial object's bounding box diagonal.

The algorithm proceeds with the following three core steps:

Step 3 – Curvature-driven iterative flattening :

First, we compute the Gaussian curvature K_p for each vertex p of the mesh. Then, we determine the vertex p_{max} with the maximum absolute value of the Gaussian curvature. The barycenter of its neighboring nodes is computed, as described by the following equation:

$$p'_{max} = \frac{\sum_{p_i \in Neigh(p_{max})} p_i}{val(p_{max})} \tag{2}$$

where $Neigh(p_{max})$ denotes the set of vertices adjacent to p_{max} and $val(p_{GMax})$ represents its valence.

If the Euclidian distance between new and initial positions $\| p'_{max} - p_{\max} \|$ is superior to a threshold *dist*, its position is changed to p'_{max}. Otherwise, the considered vertex is not affected, the algorithm selects as a candidate the following highest curvature vertex and repeats the process.

When modifying the position of a vertex, the various measures (triangle areas and angles) involved in the computation of the Gaussian curvatures, need to be re-computed. This is done locally, exclusively for the displaced vertex and for its neighbors, since the other mesh vertices are not affected.

The process is repeated recursively: determine the vertex with maximum Gaussian curvature, compute the barycentric coordinates, displace the vertex and re-compute K only for the affected vertices.

After each iteration a local, flattened version of the 3D mesh model is obtained.

The classical Gaussian curvature in equation (1) privileges the selection of vertices located in densely sampled mesh regions, where the triangle areas tend to zero. This behavior can penalize our algorithm, which can perform long sequences of iterations inside such regions. In order to avoid this problem, we have considered a modified expression of the Gaussian curvature, defined as:

$$K_S = \frac{2\pi - \sum_i \alpha_i}{\chi_s + \sum_i A_i(p)/3}, \tag{3}$$

where χ_s denotes the average triangle area, computed over the entire mesh. The correction factor χ_S makes it possible to reinforce, in the selection process, the influence of the angular defection term $\left(2\pi - \sum_i \alpha_i\right)$ and thus to avoid long loops in densely sampled regions characterized by low values of the triangle areas.

Step 3 is successively repeated for a number It of iterations. At the end of step 3, the PCA normalization (step 1) is re-applied in order to avoid shrinkage problems.

Fig. 1 illustrates a mesh obtained at the end of this stage.

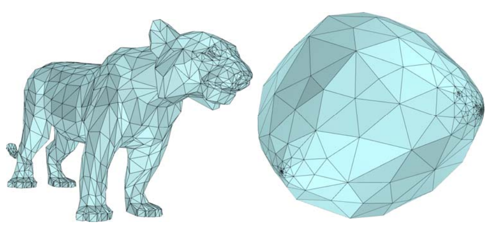


Figure 1. Intial (left) and flattened meshes (end of step 3).

Step 4 – Visibility check and projection onto the sphere:

At this stage, we first check if the mesh obtained at the end of step 3 can be projected onto the sphere. This consists in applying for each mesh vertex a visibility test. More precisely, if all the mesh vertices are visible from the object's gravity center, the parameterization on the unit sphere is simply obtained by vertex projection as defined by the following equation:

$$\forall p_i \in V, \quad \phi_i = p_i / \|p_i\|, \tag{4}$$

where φ_i is the image on the unit surface sphere of the vertex p_i.

The visibility property ensures that the obtained parameterization is bijective. If the visibility condition is not satisfied, then step 3 is re-iterated.

Step 5: The vertices removed at step 2 are iteratively re-inserted on the sphere by constructing a progressive mesh sequence analogously to the method described in [15].

IV. Experiments And Results

In order to evaluate the proposed parameterization algorithm we have considered a subset of 7 objects from the Princeton Shape Benchmark which is freely available over the Internet (http://shape.cs.princeton.edu/benchmark/) and MPEG-7 collection. The selected objects are genus-0 manifold triangular mesh models with various complexities and shapes.

The various parameters involved have been set to It = 1000, dist = 0.0001, Err = 5%.

For all the meshes considered in the test set, the proposed method yielded valid spherical parameterizations.

Fig. 2 presents some results. As it can be observed, the achieved spherical mappings yield valid embeddings which preserve well the models shape.

In order to objectively evaluate the method, we have considered the angle and area distortions (D_A and D_S) [21] defined as follows:

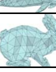
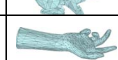
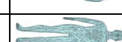
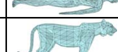
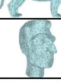
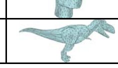
$$D_A = \sum_{i=1}^{3\cdot T}\left(\frac{\alpha_{iM} - \alpha_{iS}}{\alpha_{iS}}\right)^2, \tag{5}$$

$$D_S = \sum_{l=1}^{T}\left(\frac{A_{l,M}}{A_{T,M}} - \frac{A_{l,S}}{A_{T,S}}\right)^2, \tag{6}$$

where T is the number of triangles, α denote the mesh angles, while A represents the triangle areas. Indices M and S respectively indicate original and parameterized models.

Ideally, both distortions should be as close as possible to zero. The results synthesized in Table 1 present a comparative evaluation of our method and the Alexa's technique in [13].

TABLE I. Comparative study concerning area and angle distortions.

Name	Model	Our method		Alexa method	
		D_S	D_A	D_S	D_A
Dolphin		0.00404	0.39536	0.00425	0.43551
Bunny		0.00262	0.30394	0.00617	0.34577
Arm		0.00415	0.46341	Over-lapping	Over-lapping
Man		0.00339	0.51438	0.00356	3.15074
Lion		0.00250	0.31121	0.0069	2.37459
Head		0.00025	0.12398	0.00281	2.29459
Dino		0.00403	0.35241	0.00458	0.86131

The results show that in all cases, the proposed method provides superior performances in terms of both angular and area distortions, with average gains of 78% and 40% respectively when compared to Alexa's method.

V. CONCLUSIONS AND PERSPECTIVES

In this paper, we have proposed a new, curvature-driven spherical mesh parameterization method, applicable to closed, manifold and genus-0 3D meshes.

The proposed technique shows superior performances with respect to the state of the art methods and provides low angle and area distortion rates with a reduction of more than 78% and 40% respectively. In addition, a key factor of our algorithm is given by its complete automatic nature, which does require any human intervention.

Our future work concerns the integration of our method in a more general framework of mesh morphing applications. To achieve this aim, we need to solve the matching step between two topological different models. The parameterization algorithm introduced in this paper can be refined by adding supplementary constraints when we reposition the vertices with the maximum values for the Gaussian curvature in order to further improve the embedding and reduce the shape deformation.

REFERENCES

[1] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle. Multiresolution analysis of arbitrary meshes. Proceedings of SIGGRAPH, pages 173–182, 1995.

[2] W. T. Tutte. How to draw a graph. Proceedings on the London Mathematical Society, vol. 3, pages 743-768, 1963.

[3] M. S. Floater. Parametrization and smooth approximation of surface triangulations. Computer Aided Geometric Design, vol. 14(3), pages 231-250, 1997.

[4] M. Desbrun, M. Meyer and P. Alliez. Intrinsic Parameterizations of Surface Meshes. Computer Graphics Forum, vol. 21, pages 210-218, 2002.

[5] I. Friedel, P. Schröder, M. Desbrun, Unconstrained Spherical Parameterization. ACM SIGGRAPH Technical Sketches, ACM, New York, USA, 2005.

[6] O. Sorkine, D. Cohen-Or, R. Goldenthal and D. Lischinski. Bounded distortion piecewise mesh parametrization. In IEEE Visualization, pages 355–362, 2002.

[7] A. Sheffer and J. Hart. Seamster: Inconspicuous low-distortion texture seam layout. In IEEE Visualization, pages 291–298, 2002.

[8] Y. Wu, Y. He and H. Tian. Relaxation of spherical parameterization meshes. The Visual Computer vol. 21, pages 897-904, 2005.

[9] A. Sheffer, C. Gotsman and N. Dyn. Robust Spherical Parameterization of Triangular Meshes. In Proceedings of 4th Israel-Korea Binational Workshop on Computer Graphics and Geometric Modeling, pages 94-99, 2003.

[10] A. Asirvatham, E. Praun, and H. Hoppe. Consistent spherical parameterization. Proceedings International Conference on Computational Science, pages 265–272, 2005.

[11] Y. Qiu, Y. Wang, X. Liu, and W. Mio. Spherical representations of shape using parameterizations with minimal distortion. Proceedings of the Sixth IEEE international conference on symposium on biomedical imaging: from Nano to Macro, Boston, USA, pages 674-677, 2009.

[12] J. R. Kent, W. E. Carlson and R. E. Parent. Shape transformation for polyhedral objects. Computer Graphics, SIGGRAPH Proceedings 26, vol.2, pp. 47-54, 1992.

[13] M. Alexa. Recent Advances in Mesh Morphing. In Computer Graphics Forum, vol. 21, pages. 173-196, 2002.

[14] C. Gotsman, X. Gu and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. ACM Transactions on Graphics vol. 22, pages 358–363, 2003.

[15] E. Praun, H. Hoppe, Spherical parametrization and remeshing. ACM Transactions on Graphics, vol. 22, no. 3, July 2003.

[16] M. P. do Carmo, Differential Geometry of Curves and Surfaces, Prentice Hall 1976.

[17] Z. Xu and G. Xu. Discrete schemes for Gaussian curvature and their convergence. Computers & Mathematics with Applications, vol. 57, pages 1187-1195, April 2009.

[18] I. T. Jolliffe, Principal component analysis, 2nd edition, Springer, New York, 2002.

[19] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH 97 Proc.*, pages 209–216, 1997.

[20] P. Cignoni, C. Rocchini and R. Scopigno, "Metro: Measuring error on simplified surfaces.", Computer Graphics Forum, pg. 167–174 1998.

[21] H. Lee, Y. Tong and M. Desbrun, "Geodesics-Based One-to-One Parameterization of 3D Triangle Meshes", IEEE Multimedia, vol. 12, no.1 January 2005.

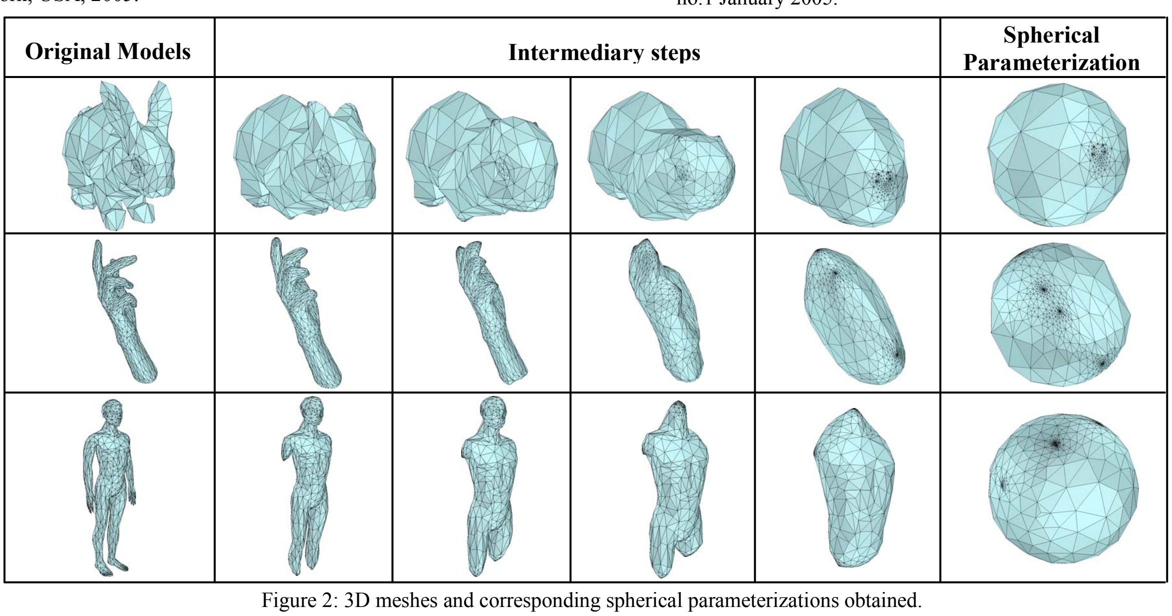


Figure 2: 3D meshes and corresponding spherical parameterizations obtained.