Mean-Value Laplacian Coordinates for Triangular Meshes

H.Y. Wu, Chunhong PAN, Qing Yang, Jia PAN, Songde MA
Institute of Automation, Chinese Academy of Sciences
Email: hywu@nlpr.ia.ac.cn

Abstract

This paper presents an effective approach for triangular mesh editing, based on mean-value laplacian coordinates for triangular meshes. We discretize the Laplace operator using mean value weights instead of uniform weights for fine approximation qualities. The results are obtained by solving a quadratic optimization problem, which can be efficiently minimized by solving a sparse linear system. Moreover, the quadratic energy function is assigned to each triangle rather than each vertex, which is more convenient to add control items. The result shows that our method is effective enough for common applications.

Keywords--- Mean value coordinates, Laplacian mesh editing

1. Introduction

Interactive mesh editing is becoming a prominent field in geometric modeling and character animation. Mesh editing allows the user to move a few vertices on the 3D surface while preserving the surface's local appearance under some global constraints or boundary conditions.

In computer graphics, surfaces can be represented in global or local coordinate system. Global coordinate system explicitly specifies the absolute Euclidean coordinates of the geometric data that define a certain shape. In contrast, local coordinate system encodes the intrinsic geometry of the surface. Global coordinate system is convenient for transformation, texturing, collision detection and rendering, while local coordinate system is suitable for preserving surface details during mesh operations.

In this paper, we present mean-value Laplacian coordinates for triangular meshes, which provide an intrinsic representation of the surface mesh to encode geometric details with differential coordinates. We parameterize the surface with mean value coordinates [4], which are motivated by the mean value theorem for harmonic functions. Furthermore, our approach assigns quadratic error functions to each individual triangle (i.e.

vertices in the same triangle share the same affine transformation) rather than each vertex. This can bring convenience while manipulating individual triangles.

2. Background

With the popularity of 3D scanning technique, artists can manipulate the existing surfaces from 3D model libraries without starting from scratch. With the significant increase of the number of 3D models, it is of critical importance to reuse them. Many successful techniques have been developed to help artists to manipulate and edit existing models.

The skeleton technique [5][6][11] binds mesh vertices to the bones with the vertex weights, and then deforms the mesh with the skeleton bones. Skeleton contains enough pose information and provides intuitive controls, which is also suitable for large deformations. But defining and manipulating a skeleton structure for a 3D model, which is usually represented by triangular mesh, is a not trivial task [19]. Furthermore, many 3D objects do not have obvious skeleton structures or their metamorphoses cannot be described in terms of a skeleton. In this case, we can use FFD or other mesh editing methods.

Free-form deformation (FFD) [10][17] embeds the object to a regular enclosed domain. Then the object is deformed according to the deformation of the domain. A main drawback of skeleton and FFD is that they deform the space surrounding the objects, rather than the objects themselves.

Mesh editing is an active research area in computer graphics, and many methods have been proposed, such as multiresolution approaches [3][9][22], pyramid coordinates [14], Laplacian editing [15][21], Poisson editing [20], and linear rotation-invariant coordinates for meshes [8]. While editing and modifying a surface, the preservation of the geometric details is a key point. [16] gives a detailed review of the state-of-the-art.

3. Notations and Equations

3.1. Laplacian differential coordinates

Let $M = (G, P)$ be a 2-manifold triangular mesh; $G = (V, E, F)$ is a graph where V denotes the set of vertices, E denotes the set of edges and F denotes the set of faces; and P is the geometry associated with each vertex in V . The Laplacian differential coordinates [15] are represented by the difference between v_i and the average of its neighbors (see Figure 1):

$$\delta_i = (\delta_i^{(x)}, \delta_i^{(y)}, \delta_i^{(z)}) = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j \quad (1)$$

where $N(i) = \{j \mid (i, j) \in E\}$ are the edge neighbors, $d_i = |N(i)|$ is the valence of a vertex, i.e. the number of edges which emanate from this vertex.

3.2. Mean value parameterization

Laplacian differential coordinates capture the local shape of the surface by encoding each vertex relative to the centroid of its topological neighbors. The modes of discretizing the Laplacian operator are relevant to the parameterization of a surface which can be viewed as a one-to-one mapping from the surface M to a suitable domain Ω (with the domain boundary $\partial\Omega$).

Given a star-shaped polygon with v_i in its kernel, our aim is to obtain sets of weights $\lambda_{i1}, \dots, \lambda_{id_i}$ such that v_i can be represented by the 1-ring neighbors:

$$\sum_{j=1}^{d_i} \lambda_{ij}(v) v_j = v_i \quad (2)$$

$$\sum_{j=1}^{d_i} \lambda_{ij}(v) = 1 \quad (3)$$

The weights should take geometric properties (angle, orientation, length, etc.) of the embedded mesh into account, which is needed for parameterization and deformation of triangulations.

The special case in which the weights λ_{ij} are uniform, i.e., for each interior vertex v_i they are equal to $1/d_i$, is called a barycentric mapping [23].

Another set of weights, called harmonic weights [12][24], can be expressed as

$$\lambda_{ij} = w_{ij} / \sum_{k=1}^{d_i} w_{ik}, \quad w_{ij} = \cot \theta_{ij} + \cot \gamma_{ij} \quad (4)$$

where θ_{ij} and γ_{ij} are the angles opposite $e_{v_i v_j}$, shown in Figure 1. Harmonic weights arise from the standard piecewise linear finite element approximation to the Laplace equation which minimizes the Dirichlet

energy. One drawback of harmonic weights is that the weight λ_{ij} is negative if $\theta_{ij} + \gamma_{ij} > \pi$.

Here, we discretize the Laplace operator using mean value weights. Mean value coordinate [4], a generalization of barycentric coordinate, is derived from an application of the mean value theorem for harmonic functions. We define mean-value Laplacian coordinates as:

$$\delta_i = v_i - \frac{1}{\sum_{(i,j) \in E} w_{ij}} \sum_{(i,j) \in E} w_{ij} v_j \quad (5)$$

$$w_{ij} = \frac{\tan(\alpha_{ij}/2) + \tan(\beta_{ij}/2)}{\|v_j - v_i\|} \quad (6)$$

where α_{ij} and β_{ij} are the angles shown in Figure 1.

The weights can be guaranteed to be positive. Taken together with a convex embedding of the boundary of M into $\partial\Omega$, this property guarantees an injective ψ , which implies that no triangle is flipped under the parameterization. Moreover, the mean value coordinates depend smoothly on v_i and $v_{N(i)}$.

Figure 1 Laplacian differential coordinates and angles used in the definition of the parameterization weights.

4. Mean-Value Laplacian Coordinates for Triangular Meshes

Laplacian coordinates [2][7][25] intrinsically represent the mesh by encoding each vertex position with its relationship to its neighbors, and allow efficient converting between absolute and intrinsic representations by solving a sparse linear system. Sorkine [15] discretized the Laplace operator using uniform weights, and assigned quadratic error function to each vertex. We present mean-value Laplacian coordinates for triangular meshes, assigning quadratic error functions to each triangle (i.e. vertices in the same triangle share the same affine transformation, see Figure 2). Because the deformation of a triangle (with the fourth vertex in the direction perpendicular to the triangle [18]) can fully determine an affine transformation, while that of a vertex cannot, it can bring more flexibility when manipulating individual triangles.

Figure 2 Vertices in the same triangle share the same affine transformation.

We represent the mesh deformation as a collection of affine transformations tabulated for each triangle. Let v_i and $\tilde{v}_i$ ($i=1,2,3,4$) be the vertices of the original triangle and the deformed triangle, respectively. In [18], v_4 was defined as:

$$v_4 = v_1 + (v_2 - v_1) \times (v_3 - v_1) / \sqrt{|(v_2 - v_1) \times (v_3 - v_1)|} \quad (7)$$

In fact, the v_4 can be any point which is outside of the triangle plane. Note that v_4 cannot be very close to the triangle plane. In order to avoid the square root computing, we also can define v_4 as:

$$v_4 = c^r + (v_1 - c^r) \times (v_2 - c^r) \quad (8)$$

where c^r is the centroid of the triangle.

Then, we define the 3×3 matrix Q in terms of the triangles' vertices, which indicates the affine transformation from the original triangle to the deformed triangle:

$$Q = [\tilde{v}_2 - \tilde{v}_1, \tilde{v}_3 - \tilde{v}_1, \tilde{v}_4 - \tilde{v}_1][v_2 - v_1, v_3 - v_1, v_4 - v_1]^{-1} \quad (9)$$

In order to control a deformation, the user can firstly mark the desired region of interest (ROI) on the mesh that is to be edited. The rest of the mesh will remain unchanged. Next, a few vertices inside the ROI are selected by the user to serve as the handle. Then the user manipulates the handle and applies translation, rotation and scaling transformations to the mesh. The deformed positions of the mesh vertices V' are obtained by solving the following quadric minimization problem:

$$E(V') = \sum_{k=1}^{|T|} w_k \left(\sum_{i=1}^3 \|Q_k \delta_i - \xi(v'_i)\|^2 \right) + w_m \sum_{i=1}^m \|v'_i - u_i\|^2 + \sum_{s=1}^{|T|} w_s \left(\sum_{j \in adj(s)} \|Q_s - Q_j\|_F^2 \right) \quad (10)$$

where Q is the transformation matrix for each triangle; δ coordinates of v_i are the mean-value Laplacian coordinates; $\xi(v'_i)$ is the mean-value Laplacian coordinate of vertex v'_i . u_i is the vertex whose spatial location is known, and the matrix norm $\|\cdot\|_F$ is the Frobenius norm. w_k , w_m and w_s are the weights of each item, respectively.

The first term of E indicates the details of the shape after transformation are preserved. The second term specifies the spatial constraints. The third term indicates that the change in transformations for adjacent triangles should be smooth.

In order to preserve those geometric details after rotation and isotropic scaling transformation, Q_k should be constrained to rotation matrix. Sorkine [15] provided a locally linearized representation to guarantee these constraints. Here, we explicitly formulate these constraints in terms of vertex positions. Thus, rather than solving for the entries of the affine transformations, we solve directly for the deformed vertex positions:

$$Q_k = \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix} = [\tilde{v}_2 - \tilde{v}_1, \tilde{v}_3 - \tilde{v}_1, \tilde{v}_4 - \tilde{v}_1][v_2 - v_1, v_3 - v_1, v_4 - v_1]^{-1} \quad (11)$$

Subject to: $\begin{cases} q_{11} = q_{22} = q_{33} \\ q_{12} + q_{21} = 0 \\ q_{13} + q_{31} = 0 \\ q_{23} + q_{32} = 0 \end{cases}$

The elements of Q_k are linear combinations of the coordinates of the unknown deformed vertices. So the optimization formulation (10) can be minimized by solving a sparse linear system. Setting the gradient of the objective function to zero gives the normal equations:

$$A^T A x = A^T b \quad (12)$$

The computation is numerically efficient with a sparse LU solver [26]. We firstly compute the factorization of the normal equations and then find the solution by back-substitution.

Figure 3 demonstrates some results to validate the method of mean-value Laplacian differential coordinates for triangular meshes. We obtained satisfactory results when editing the different parts of the lion model, such as dragging the legs, curling the tail and opening the mouth. More results are depicted in Figure 4 and Figure 5.

(a)

Figure 3 Editing the lion model. (a) The original lion model. (b)(c)(d) The results after editing operation.

Because the deformation of one triangle can fully determine an affine transformation, more powerful control items can be directly imposed upon the triangle to improve the deformation effect. That is, we can directly specify whether a triangle is rigid and assign its rigidity. In general, the more rigid the triangle is, the larger the weight w_k of the triangle should be, and the smaller the weight w_s should be.

Furthermore, using singular value decomposition (SVD) or QR decomposition method, we can extract the rotation part R_γ and the shearing-scaling part S from the affine matrix Q of the triangle [13]:

$$Q = R_\alpha D R_\beta = R_\alpha (R_\beta R_\beta^T) D R_\beta = (R_\alpha R_\beta) (R_\beta^T D R_\beta) = R_\gamma S \quad (13)$$

Given this formulation, one can obtain as-rigid-as-possible deformation by minimizing $\|S - I\|$, which is like the linear mapping from an original triangle to a deformed triangle in [1].

Additionally, we also provide a painting interface in which user can freely assign the weights to control rigidity of the triangles.

Conclusions

In this paper, we present mean-value Laplacian intrinsic geometry representation for triangular mesh editing. Our approach is numerically efficient, as the solution to the optimization problem can be obtained by fast solving a sparse linear system. For example, ROI sizes of 5K vertices require 0.3 seconds for factorization and 0.06 seconds for back-substitution on an Intel P4/3.0 GHz. Experimental results show that with the proposed technique the user can effectively edit the 3D mesh's shape while reserving the geometric details.

Figure 4 Editing the feline model. (a) The original feline model. (b) The result after editing operation.

Figure 5 Editing the triceratops model. (a) The original triceratops model. (b) The result after editing operation.

References

- [1] M. Alexa, D. Cohen-Or, D. Levin. As-rigid-as-possible shape interpolation. In *Proceedings of ACM SIGGRAPH* (2000), pp. 157–164.
- [2] M. Alexa. Differential coordinates for local mesh morphing and deformation. *The Visual Computer* 19,2(2003), pp. 105–114.
- [3] M. Botsch, L. Kobbelt. An intuitive framework for real-time free form modeling. *Proceedings of ACM SIGGRAPH 2004 / ACM Transactions on Graphics* 23, 3 (Aug. 2004), 630–634.
- [4] M.S. Floater. Mean value coordinates. *Computer Aided Geometric Design* 20, 1 (2003), 19–27.
- [5] P.G. Kry, D.L. James, D.K. Pai. Eigen-Skin: Real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), pp. 153–160.
- [6] J.P. Lewis, M. Cordner, N. Fong. Pose space deformations: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of SIGGRAPH* (2000), pp. 165–172.
- [7] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rössl, H.-P. Seidel. Differential coordinates for interactive mesh editing. *SMI 2004*, pp. 181–190.
- [8] Y. Lipman, O. Sorkine, D. Levin, D. Cohen-Or. Linear rotation-invariant coordinates for meshes. In *Proceedings of SIGGRAPH* (2005), pp. 479–487.
- [9] A.W.F. Lee, W. Sweldens, P. Schröder, L. Cowsar, D. Dobkin. MAPS: Multiresolution adaptive parameterization of surfaces. In *Proceedings of SIGGRAPH 98* (1998), pp. 95–104.
- [10] R. Maccracken, K.I. Joy. Free-form deformations with lattices of arbitrary topology. In *SIGGRAPH* (1996), pp. 181–188.
- [11] N. Magnenat-Thalmann, R. Laperrière, D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Graphics Interface'88* (June 1988), pp. 26–33.
- [12] M. Meyer, M. Desbrun, P. Schröder, A.H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics III*, pp. 35–57.
- [13] K. Shoemake, T. Duff. Matrix animation and polar decomposition. In *Proceedings of Graphics Interface'92* (1992), pp. 258–264.
- [14] A. Sheffer, V. Kraevoy. Pyramid coordinates for morphing and deformation. In *3DPVT* (2004), pp. 68–75.
- [15] O. Sorkine, Y. Lipman, D. Cohen-Or, M. Alexa, C. Rössl, H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2004), pp. 179–188.
- [16] O. Sorkine. Laplacian mesh processing. In *Proceedings of Eurographics STAR* (2005), pp. 53–70.
- [17] T.W. Sederberg, S.R. Parry. Free-form deformation of solid geometric models. In *SIGGRAPH* (1986), pp. 151–160.
- [18] R.W. Sumner, J. Popović. Deformation transfer for triangle meshes. *Proceedings of ACM SIGGRAPH 2004 / ACM Transactions on Graphics* 23, 3 (2004), 399–405.
- [19] S. Yoshizawa, A.G. Belyaev, H.-P. Seidel. Free-form skeleton-driven mesh deformations. In *Symposium on Solid Modeling and Applications* (2003), pp. 247–253.
- [20] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, H.-Y. Shum. Mesh editing with poisson-based gradient field manipulation. *Proceedings of ACM SIGGRAPH 2004 / ACM Transactions on Graphics* 23, 3 (2004), 644–651.
- [21] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, H.-Y. Shum. Large mesh deformation using the volumetric graph laplacian. *Proceedings of ACM SIGGRAPH 2005 / ACMTrans. Graph* 24, 3 (2005), 496–503.
- [22] D. Zorin, P. Schröder, W. Sweldens. Interactive multiresolution mesh editing. In *SIGGRAPH* (1997), pp. 259–268.
- [23] W.T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society* 13(1963), pp. 743–768.
- [24] U. Pinkall, K. Polthier. Computing discrete minimal surfaces. *Experimental Mathematics*, 2(1993), pp. 15–36.
- [25] Z. Karni, C. Gotsman. Spectral compression of mesh geometry. In *Proceedings of ACM SIGGRAPH 2000*, pp. 279–286.
- [26] T.A. Davis. Umfpack version 4.1 user guide. Tech. rep. 2003, University of Florida. TR-03-008.