3D Triangular Mesh Parametrization Using Locally Linear Embedding

Xianfang Sun[1,2] and Edwin R. Hancock[2]

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, P.R. China
[2] Department of Computer Science, The University of York, York YO10 5DD, UK
{xfsun, erh}@cs.york.ac.uk

Abstract. In this paper we describe a new mesh parametrization method which combines the mean value coordinates and the Locally Linear Embedding (LLE) method. The mean value method is extended to compute the linearly reconstructing weights of both the interior and the boundary vertices of a 3D triangular mesh, and the weights are further used in the LLE algorithm to compute the vertex coordinates of a 2D planar triangular mesh parametrization. Examples are provided to show the effectiveness of this parametrization method.

1 Introduction

Triangular mesh parametrization aims to determine a 2D triangular mesh with its vertices, edges, and triangles corresponding to that of the original 3D triangular mesh, satisfying an optimality criterion. The technique has been applied in a wide range of problems in computer graphics and image processing, including texture mapping [9], morphing [7], and remeshing [4]. Extensive research has been undertaken into the theoretical issues underpinning the method and its practical application. For a tutorial and survey, the reader is referred to [3].

A well-known parametrization method is that proposed by Floater [1]. It is a generalization of the basic procedure originally proposed by Tutte [8] which was used to draw planar graphs. The basic idea underpinning this method is to use the vertex coordinates of the original 3D triangular mesh to compute reconstructing weights of each interior vertex with respect to its neighbour vertices.These weights are subsequently used together with the boundary vertex coordinates on a plane to compute the interior vertex coordinates of a 2D triangular mesh. A drawback of Floater's parametrization method is that the boundary vertex coordinates must be determined manually beforehand.

There are many methods for computing the reconstructing weights. The simplest one is Tutte's barycentric coordinates [8]. Floater provided a method of computing the weights in his first paper about parametrization [1], which has a so-called shape-preserving property. More recently, Floater computes mean value coordinates as the reconstructing weights[2]. These mean value coordinates perform better than the earlier shape-preserving weights of [1].

A. Gagalowicz and W. Philips (Eds.): CAIP 2005, LNCS 3691, pp. 96–103, 2005.
© Springer-Verlag Berlin Heidelberg 2005

The Linearly Local Embedding (LLE) [5] is a method of mapping high dimensional data to a low dimensional Euclidean space. The idea underlying the method is to use the high dimensional data to compute locally linear reconstructing weights for each data point. These weights are then used to compute the point coordinates in a low dimensional data space. It can be used in a natural way to map 3D coordinates to 2D coordinates. Thus it can be used as a parametrization method. However, because for 3D triangular meshes, the dimension of the data (here it is 3) is usually less than the number of neighbours of any data point, the original algorithm does not compute optimal weights. Hence, LLE is not a good parametrization method.

In this paper, we combine the advantages of both the mean value coordinates and LLE to develop a new parametrization method. The paper is organised as follows. Section 2 introduces the basic problem and provides a overview of the proposed algorithm. Sections 3 and 4 describes the mean value coordinates and the LLE method, and their adaptation for use in our algorithm. Section 5 provides some experimental examples of the method. Finally, Section 6 draws some conclusions.

2 Problem and Algorithm Overview

Consider a triangular mesh $T = T(V, E, F, X)$ with vertex set $V = \{i : i = 1, 2, ..., N\}$ and corresponding coordinate set $X = \{x_i : \ x_i \in R^d, \ i \in V\}$ ($d = 2$ or 3), edge set $E = \{(i, j) : (i, j) \in V \times V\}$, and triangular face set $F = \{(i, j, k) : (i, j), (i, k), (j, k) \in E\}$. Here an edge (i, j) is represented by a straight line segment between vertices i and j, and a triangular face (i, j, k) is a triangular facet bounded by three edges (i, j), (i, k) and (j, k). When $d = 2$, T is drawn on a plane and represents a planar triangular mesh, while $d = 3$, T is drawn in a 3-dimensional space and represents a 3D triangular mesh. A triangular mesh is called valid if the only intersections between edges are at common end points (vertices) and the only intersections between triangular faces are on the common edges. Hereafter, when a triangular mesh is referred without qualification, it implies that the triangular mesh is valid.

The parametrization is made on a valid 3D triangular mesh. A parametrization of a valid 3D triangular mesh $T = T(V, E, F, X)$ is any valid planar triangular mesh $T_p = T_p(V, E, F, Y)$ with $Y = \{y_i : \ y_i \in R^2, \ i \in V\}$ being the corresponding coordinates of V.

The parametrization algorithm proposed here combines the mean co-ordinates of Floater and the LLE method. It consists of the following three steps.

- Based on the algorithm proposed by Floater [2], the mean value coordinates (or the reconstructing weights) are computed for each vertex using the vertex coordinates X.
- Using the LLE algorithm [5], the weights obtained above are used to recover the vertex coordinates Y of the planar triangular mesh.

– If $T_p(V, E, F, Y)$ is not valid, then the coordinates of the boundary vertices are fixed, and the coordinates of the interior vertices are computed using Floater's algorithm [1].

Note that in Step 1 of our algorithm, the reconstructing weights of both the interior and boundary vertices are computed, while only the weights of interior vertices are computed in Floater's algorithm [1,2].

3 Mean Value Coordinates

Given a 3D triangular mesh $T = T(V, E, F, X)$, where the vertex set V is divided into disjoint boundary and interior vertex subsets, i.e. $V = V_I \cup V_B$, where V_I is the interior vertex set, V_B is the boundary vertex set and $V_I \cap V_B = \emptyset$. The parametrization method proposed by Floater [1] is a generalization of Tutte's method of drawing a planar graph [8], which consists of the following steps.

– For each interior vertex $i \in V_I$, assign a non-negative weight $W_{i,j}$ to each of its incident edges $(i, j) \in E$ such that $\sum_{(i,j)\in E} W_{i,j} = 1$, and $W_{i,j} = 0$ for all $(i, j) \notin E$.
– For each boundary vertex $i \in V_B$, determine a coordinate $y_i \in R^2$ in the plane such that the order of the boundary vertices in the plane remains the same as that of the original ones, and they form a closed convex polygon.
– Solve the following linear system for the coordinates of the interior vertices

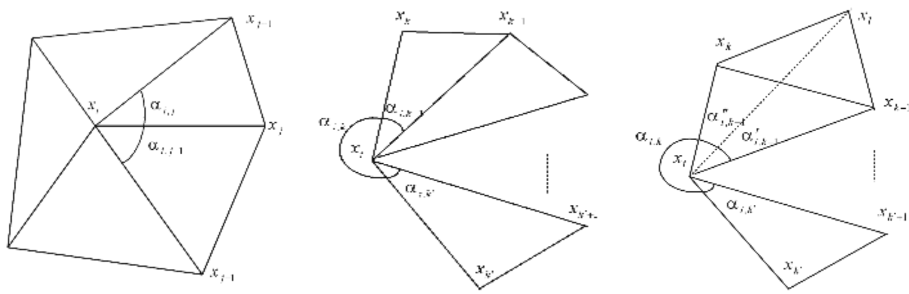
$$y_i = \sum_{(i,j)\in E} W_{i,j} y_j, \; i \in V_I \; . \tag{1}$$

There are some important features of this algorithm that deserve further comment. In the first step, although Tutte's barycentric coordinates [8] and Floater's early shape-preserving weights [1] can be used here as the reconstructing weights, a better choice are the mean value coordinates recently proposed by Floater [2]. The mean value coordinates are computed using the formula

$$W_{i,j} = \frac{\lambda_{i,j}}{\sum_{(i,k)\in E} \lambda_{i,k}}, \; \lambda_{i,j} = \frac{\tan(\alpha_{i,j-1}/2) + tan(\alpha_{i,j}/2)}{\|x_j - x_i\|} \; , \tag{2}$$

where $\alpha_{i,j-1}$ and $\alpha_{i,j}$ are the angles between the edge (i, j) and its two neighbouring edges $(i, j-1)$ and $(i, j+1)$ (see Fig. 1(a)).

Next, we consider the second step. The boundary vertex coordinates are determined manually. This a drawback since if the boundary vertex coordinates are selected inappropriately, the resulting parametrization is poor. In our method, the boundary vertex coordinates are determined by the LLE method. Hence, although we dispense with this step, the weights of boundary vertices must still be computed.



(a) (b) (c)

Fig. 1. Elements for the computation of mean value coordinates: (a) interior vertex, (b) general boundary vertex, (c) boundary vertex in case with additional neighbour vertex

Here, we still use the mean value coordinates as the weights of the boundary vertices. For the boundary vertex i, the weights $\lambda_{i,k}$ and $\lambda_{i,k'}$ of its neigbouring boundary vertices k and k' are computed using the formula (refer to Fig. 1(b))

$$\lambda_{i,k} = \frac{\tan(\alpha_{i,k-1}/2) + tan(\alpha_{i,k}/2)}{\|x_k - x_i\|}, \ \lambda_{i,k'} = \frac{\tan(\alpha_{i,k}/2) + tan(\alpha_{i,k'}/2)}{\|x_{k'} - x_i\|} , \quad (3)$$

Because $\alpha_{i,k}$ is not strictly less than π, two problems arise when (2) and (3) are used to compute $W_{i,j}$. The first problem occurs when $\alpha_{i,k} = \pi$, i.e. $\tan(\alpha_{i,k}/2) = \infty$, which causes a computational overflow. The solution of this problem is to simply set $\lambda_{i,k} = 1/\|x_k - x_i\|$, $\lambda_{i,k'} = 1/\|x_{k'} - x_i\|$, and $\lambda_{i,j} = 0$ for all other $j \in V$. The second problem occurs when $\sum_{(i,j)\in E} \lambda_{i,j} = 0$, which causes a divide-by-zero error when $W_{i,j}$ is computed. In this case, an additional vertex l, which is originally not the neighbour vertex of i, but that of one of i's neighbouring vertices, is now taken as the neighbour vertex of i in computing the mean value coordinates (see Fig. 1(c)).

After computing the reconstructing weights, we depart from Floater's version of Tutte's algorithm. Our idea here is to borrow ideas from LLE algorithm to compute the co-ordinates of the interior vertices.

4 Locally Linear Embedding

In this paper, we only exploit a component part of the LLE method for mesh parametrization. However, for completeness and further analysis of this method, the complete LLE algorithm is described. The LLE algorithm consists of following three steps:

- For each data point x_i, find the K nearest neighbours $\{x_{i1}, \cdots, x_{iK}\}$.
- Compute the weights $W_{i,j}$ that best linearly reconstruct x_i from its neighours through minimizing the cost function

$$E(W) = \sum_i \|x_i - \sum_j W_{i,j} x_j\|^2 \qquad (4)$$

with the additional constraints $\sum_j W_{i,j} = 1$, and $W_{i,j} = 0$ if x_j is not the K nearest neighbour of x_i.

– Compute the low dimensional embedding vector y_i that is best reconstructed by $W_{i,j}$ by minimizing the embedding cost function

$$\Phi(Y) = \sum_i \|y_i - \sum_j W_{i,j} y_j\|^2 \qquad (5)$$

with additional constraints $\sum_i y_i = 0$ and $\frac{1}{N}\sum_i y_i y_i^T = I$.

When the LLE method is directly used for parametrization of a 3D triangular mesh, the first step (i.e. the selection of the K nearest neighbours) may seem superfluous, since the 3D triangular mesh has its own natural neighbourhood. However, the number of neighbours significantly affects the performance of the algorithm. If the number of neighbours is too small, then the reconstructed embedding will be poor. Unfortunately, the numbers of natural neighbours in a triangular mesh are usually small. Hence, better results can be obtained by choosing a suitable value of K.

The second step attempts to locate the best weights that minimize (4). However, because for most of the interior vertices, the valency (the number of neighbour vertices) is greater than 3, the solution of (4) has been conditioned in the original LLE algorithm [5], thus the weights are in fact not optimal. In this paper, the mean value coordinates described in Section 3 are used as alternative weights to obtain a better result.

After the weights are obtained, y_i in Step 3 can be easily obtained by using the eigenvectors of the matrix $M = (I - W)^T(I - W)$.

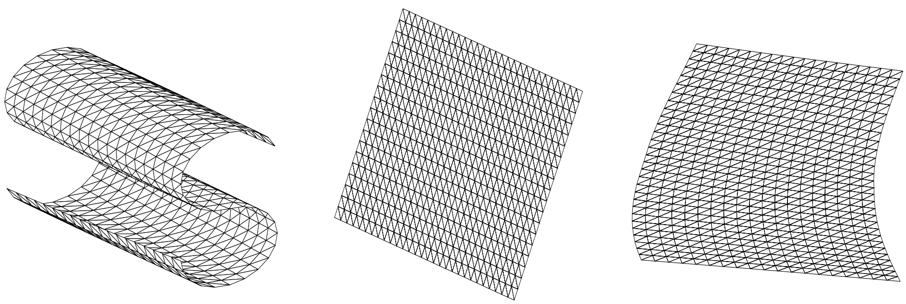
Let $\Lambda = diag(\lambda_1, \lambda_2, \lambda_3, ...)$ be the matrix with the ordered eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 ...$ as diagonal elements, and let $\Phi = (\phi_1|\phi_2|\phi_2|...)$ be the matrix with the corresponding eigevalues as columns. The eigendecomposition of the matrix M is $M = \Phi\Lambda\Phi^T$. The eigenvector of this matrix corresponding to eigenvalue $\lambda_1 = 0$ is the unit vector with equal components, and is discarded. The eigenvectors ϕ_2 and ϕ_3 give us the 2D coordinates Y, and $y_i = (\phi_2(i), \phi_3(i))^T$.

Now that Y has been obtained, then $T_p(V, E, F, Y)$ gives us a parametrization of $T(V, E, F, X)$. In most of the cases, $T_p(V, E, F, Y)$ is a valid planar triangular mesh. However, when the original 3D mesh has too high curvatures on some points, the above resulting planar triangular mesh may fold over. In this case, we only need to select the 2D coordinates of the boundary vertices from Y and adjust their order if necessary. Fixing the coordinates of the boundary vertices, the coordinates of the interior vertices can then be computed by solving equation system (1), and finally, a valid planar triangular mesh $T_p(V, E, F, Y)$ is obtained.

5 Examples

In this section, two examples are provided to illustrate some of the properties of the algorithm proposed in this paper. In the first example, we consider an

S-shaped manifold [6]. It is an intrinsically two dimensional manifold. Figure 2(a) shows a regular sample of $N = 600$ data points and its triangulation in the 3D space. Figure 2(b) shows the parametrization using the algorithm proposed here, and Fig. 2(c) shows the result using LLE with $K = 12$ neighbours per data point. It is evident that the algorithm of this paper performs better than the LLE algorithm. In particular, the current algorithm results in a parametrization with an appearance which is closer to the original one than that obtained by the LLE algorithm.



(a) (b) (c)

Fig. 2. Parametrization of S-shape manifold: (a) regular triangulation, (b) parametrization using our algorithm, (c) parametrization using LLE algorithm with K=12

Figure 3(a) shows a random sample of $N = 600$ data points on the same S-shape manifold. Figure 3(b) shows the parametrization using the algorithm proposed here, and Fig. 3(c)˜(e) shows the result obtained using LLE with $K = 6,\ 12,\ 24$ neighbours per data point, respectively. Again, it can be seen that the proposed algorithm results in a parametrization of better appearance. Moreover, the performance of the LLE algorithm is highly dependent on a suitable choice of the parameter K. When K is too small (here, $K = 6$) or too large ($K = 24$), the resulting planar triangular mesh is invalid because some of the interior or boundary edges cross each other (see Fig. 3(c) and 3(e)). The second example uses the *peaks* function of Matlab. Figure 4(a) shows an irregular triangulation of this function. Figure 4(b) shows the result using only the first two steps of the proposed algorithm and Fig. 4(c) shows a local zoom-in part of Fig. 4(b). It can be seen that some triangles are folded over, and the resulting parametrization is invalid. From the other point, however, we can see that the boundary vertices have been self-adjusted on the plane, and thus we can use these coordinates of the boundary vertices and perform Step 3 of the proposed algorithm to obtain a valid parametrization of the original 3D triangular mesh.

We have also scaled down the z-coordinates by 1/3 and directly obtained a valid parametrization using just the first two steps of the proposed algorithm. The result is shown in Fig. 4(d). Figure 4(e) is the result of texture mapping using the parametrization of Fig. 4(d). It can be seen that the resulting parametrization is suitable for the texture mapping application.

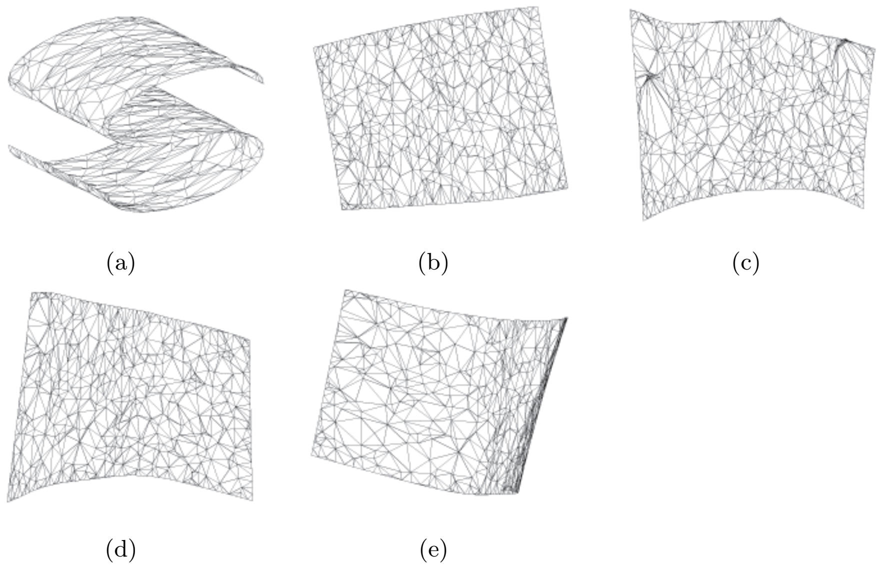


Fig. 3. Parametrization of S-shape manifold with random sample: (a) triangulation, (b) parametrization using our algorithm, (c)˜(e) parametrization using LLE algorithm with K=6,12,24, respectively

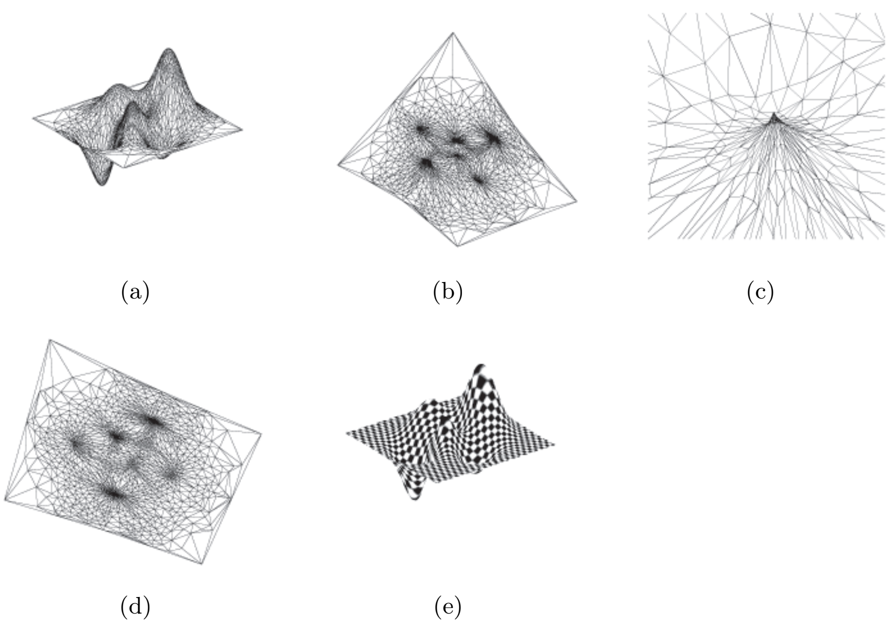


Fig. 4. Parametrization of peaks function: (a) an irregular triangulation, (b) parametrization with only first two steps, (c) zoom-in part of (b), (d) parametrization for z-axis being scaled down by 1/3, (e) texture mapping

6 Conclusion

In this paper, we have combined the mean value coordinates and the LLE method to construct a new parametrization method. Although the parametrization method using mean value coordinates has a drawback of requiring manually determined boundary vertex coordinates and the LLE method has the drawback that reconstructing weights are not optimal, a combination of these two methods have been proved to be reasonable. We have used examples to show that the proposed parametrization method can automatically find good boundary vertex coordinates and it is practically useful.

References

1. Floater, M. S.: Parametrization and smooth approximation of surface triangulations. Comp. Aided. Geom. Design. **14** (1997) 231–250
2. Floater, M. S.: Mean value coordinates. Comp. Aided. Geom. Design. **20** (2003) 19–27
3. Floater, M. S., Hormann, K.: Surface Parameterization: a Tutorial and Survey. in Advances in Multiresolution for Geometric Modelling, N. A. Dodgson, M. S. Floater, and M. A. Sabin (eds.), Springer-Verlag, Heidelberg, 2004, 157–186
4. Praun, E., Hoppe, H.: Spherical parametrization and remeshing. ACM Trans. Graphics. **22** (2003) 340–349
5. Roweis, S. T., Saul, L. K.: Nonlinear dimensionality reduction by locally linear embedding. Science. **290** (2000) 2323–2326
6. Saul, L. K., Roweis, S. T.: An introduction to locally linear embedding. Report at AT & T Labs-Research, 2000.
7. Surazhsky, V., Gotsman, C.: Intrinsic morphing of compatible triangulations. Int. J. Shape Modelling.**9** (2003) 191–201
8. Tutte, W. T.: How to draw a graph. Proc. London Math. Soc. **13** (1963) 743–768
9. Zigelman, G., Kimmel, R., Kiryati N.: Texture mapping using surface flattening via multi-dimensional scaling. IEEE Trans. Visualization Comp. Graphics. **8** (2002) 198–207