Spherical Parametrization of Genus-Zero Meshes using the Lagrange-Newton Method

Ying Li   Zhouwang Yang   Jiansong Deng
Department of Mathematics
University of Science and Technology of China
Hefei 230026, China
liying04@mail.ustc.edu.cn   yangzw|dengjs@ustc.edu.cn

Abstract

This paper addresses the problem of spherical parametrization, i.e., mapping a given polygonal surface of genus-zero onto a unit sphere. We construct an improved algorithm for parametrization of genus-zero meshes and aim to obtain high-quality surfaces fitting with PHT-splines. This parametrization consists of minimizing discrete harmonic energy subject to spherical constraints and solving the constrained optimization by the Lagrange-Newton method. We also present several examples which show that parametric surfaces of PHT-splines can be constructed adaptively and efficiently to fit given meshes associated with our parametrization results.

Keywords. *spherical parametrization, genus-zero meshes, discrete harmonic energy, constrained optimization, the Lagrange-Newton method*

1. Introduction

Parametrization is an important problem in meshing data processing. A parametrization of a polygonal mesh in 3D space can be viewed as a one-to-one mapping from the given mesh to a suitable domain which is also a mesh. Typically, if the mesh is simple, the suitable domain is a connected region on the plane; and if the mesh is with genus-zero, the domain is a unit sphere. Usually, the meshes consist of triangles. Hence the mappings are piecewise linear and we only need to compute the vertex positions of the triangles.

Parametrizations have many applications in various fields, including texture mapping, surface approximation and remeshing, scattered data fitting, repair of CAD models, morphing, reparametrization of spline surfaces, and so on [6]. For a specified application, some parametrization results might behave better than others. Here the choice of different parametrizations depends heavily on the application details. Possibly a parametrization result behaves better for texture mapping, but worse for surface fitting. For a genus-zero mesh, there have existed some methods to parametrize it onto a sphere. But according to our experiences, these results are unfit for surface fitting with PHT splines [1]. In [9], we proposed a method to obtain parametrization results suitable for surface fitting. This method has some disadvantages as well, such as with the local bad shapes of the triangles. Hence in this paper, we introduce a new method to parameterize a genus-zero mesh such that the triangles in the parametrization domain are with better shapes and a surface fitting algorithm in PHT-splines can generate a better result.

1.1. Related works

Now we review some previous works on mesh parametrizations. For a more detailed summary, please refer to [6, 12].

Most of planar parametrization methods establish mappings from a simple mesh to a planar domain by solving linear equations system, such as [4, 5, 13]. Eck et al. [2] introduced the discrete harmonic mapping to parameterize a simple mesh. The methods proposed in the current paper and [9], which are applied to genus-zero meshes, share the same model as in the Eck's method. The Eck's method is a quadratic minimization problem and also can be reduced to a linear system of equations. But our model is for genus-zero meshes.

Spherical parametrization methods build mappings from a genus-zero surface to a sphere. There have been a lot of interesting and novel methods in spherical parametrization. Many of these methods are very similar to those mapping simple meshes onto planar domains, whereas some of the linear methods become non-linear versions. Haker et al. [8] used a method which maps the given genus-zero mesh into the plane and then uses stereographic projection to map to a sphere. Gu and Yau [7] gave an important point that harmonic maps from a closed genus-zero mesh to a unit sphere are conformal, which means harmonic and conformal maps are the same with genus-zero meshes. Later, they proposed an iterative method which approximates a harmonic map without splitting. Praun and Hoppe [10] extended the definition of stretch to consider a spherical parametrization. For any point inside, the Jacobian map provides a local approximation for the mapping. Consequently, they defined the stretch over the triangle with the singular values of the Jacobian map.

In [9], a hierarchical method to parameterize a genus-zero mesh to a unit sphere was presented. In the method, a model of spherical parametrization based on minimizing the discrete harmonic energy [2] was proposed. Then a model solving algorithm based on reset PRP conjugate gradient method was introduced. In the algorithm, a hierarchical idea was applied in order to reduce computing time. All points were fixed layer by layer. Experientially, this method is efficient, practical, and versatile for different surfaces. It is more important that according to our experiences, the PHT-spline surface fitting results with hierarchical parametrization results are better than those with other parametrization results. However, in the result meshes, some triangles between two layers might have bad shapes. And in the solving algorithm, there were many user-specified parameters.

1.2. Our contribution

In this paper, we solve the model of mapping a genus-zero surface to the unit sphere in [9] using the Lagrange-Newton method. At first, the model of spherical parametrization based on minimizing the discrete harmonic energy is reviewed. Then we use the Lagrange-Newton method to introduce a new stable spherical parametrization algorithm which obtains result meshes with good shapes.

Based on our new parametrization results, a surface fitting algorithm with PHT-splines [1] can generate good results, where parametric surfaces can be constructed efficiently and adaptively to fit genus-zero meshes.

The paper is organized as follows. In Section 2, we review the model and the algorithm proposed in [9]. In Section 3, the Lagrange-Newton method is adopted to solving the model. Then, the illustrative results of our solving algorithm are provided in Section 4. Finally, we conclude this paper in Section 5.

2. Review of the model based on discrete harmonic mappings

In its surface fitting, the parametrization of a genus-zero mesh over some standard domain is needed. Usually the triangles in given meshes are with good shape, i.e., that the three edges of the triangle do not change dramatically in their lengthes. In order to obtain good fitting results, we need the triangles in the parametrization domain to be with good shapes as well. Hence, we should propose a mapping from the given mesh to a unit sphere preserving the shapes of the triangles.

Unfortunately, most of the existing spherical parametrization methods do not fulfil this requirement. For a simple mesh, Eck et al. [2] proposed a discrete harmonic method which preserves the aspect ratios of triangles. So in [9], the Eck's model is generalized to spherical parametrization. In the rest of the section, we review the model and the solving algorithm.

2.1. The model

A triangular mesh $M = (V, E)$ is given with a set of vertices $V = \{v_1, \ldots, v_n\}$ and a set of edges $E = \{(v_i, v_j) \mid v_i v_j \text{ is an edge of the mesh } M\}$. Suppose that h is any piecewise linear mapping from M to a unit sphere $S^2 \subset \mathbb{R}^3$ with the restriction conditions

$$\|h(v_i)\|^2 = 1, \ \forall v_i \in V. \tag{1}$$

The mapping h is uniquely determined by its values $h(v_i)$ at the vertices of M. Then the discrete harmonic energy of the mapping h associated with the mesh M is defined as

$$f(h, M) = \frac{1}{2} \sum_{(v_i, v_j) \in E} \kappa_{ij} \, \|h(v_i) - h(v_j)\|^2, \tag{2}$$

where the spring constants κ_{ij} may be computed in many ways. In most cases and the rest of the paper, uniform spring constants are used, i.e., $\kappa_{ij} = 1$, for any i and j.

Let $h(v_i) = X_i \in \mathbb{R}^3$; $\mathbf{x} = ({X_1}^T, {X_2}^T, \ldots, {X_n}^T)^T \in \mathbb{R}^m$, where $m = 3n$; $D(i) = \{j \mid (v_i, v_j) \in E\}$; and $d(i)$ denotes the element number of the set $D(i)$. Then we can setup the parametrization model by minimizing the discrete harmonic energy in (2) with spherical constraints:

$$\begin{aligned} \min \quad & f(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j \in D(i)} \kappa_{ij} \, \|X_i - X_j\|^2, \\ \text{s.t.} \quad & c_i(\mathbf{x}) = \|X_i\|^2 - 1 = 0, \quad i = 1, \ldots, n, \end{aligned} \tag{3}$$

where, $\mathbf{x}$ is call the vector of optimization variables, $f(\mathbf{x})$ the objective function to be minimized, $c(\mathbf{x}) = (c_1(\mathbf{x}), \ldots, c_n(\mathbf{x}))^T$ the vector of equality constraints.

2.2. The algorithm

The size of the constrained nonlinear programming problem (3) is quite large and these constrains cannot easily be eliminated. In [9], a penalty function method is used to solve this optimization problem, which needs to solve a series of unconstrained optimization problem. The reset PRP conjugate gradient algorithm is used to solving the large-scale unconstrained optimization problem. The convergence rate is very slow. In order to reduce computing time, we use a hierarchical idea. All points are fixed layer by layer. After some points are fixed, the spherical constrains may be ignored, and the positions computation the other points is similar to those in planar parametrization. Finally, the points are mapped onto a unit sphere. This procedure leads to a disadvantage that some triangles along or near the boundaries of two layers do not have good shapes. See Figures 1 and 2 for examples.

On the other hand, the Lagrange-Newton algorithm is suitable to solve the equality constraints optimization problem. The object function and all the constraints of the model (3) are quadratic. We can easily give the explicit expressions of the second order information to be computed in the Lagrange-Newton method. Our main work is to pre-treat data to simplify the process. This is what we do in the next section.

3. The Lagrange-Newton method

The Lagrange-Newton method, one of the most efficient numerical methods of solving optimization problems, is developed for problems with equality-type constraints. In the method, the Newton procedure is applied to the first-order optimality system, which has the form of a system of equations. The Lagrange-Newton method is locally quadratically convergent to the solution. This approach has been successfully applied to a class of nonlinear constrained optimization problems.

3.1. Algorithm scheme

At first, we review the general scheme of the Lagrange-Newton method. For an equality-type constrained optimization problem,

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^m} f(\mathbf{x}), \\ &\text{s.t. } c(\mathbf{x}) = 0. \end{aligned} \tag{4}$$

$\mathbf{x}$ is a Kuhn-Tucker point [3] if and only if $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)^T \in \mathbb{R}^n$ exists,

$$\begin{aligned} \nabla f(\mathbf{x}) - \nabla c(\mathbf{x})^T \boldsymbol{\lambda} = 0, \\ -c(\mathbf{x}) = 0. \end{aligned} \tag{5}$$

According to the definition of the Lagrange function

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \boldsymbol{\lambda}^T c(\mathbf{x}), \tag{6}$$

formulas (5) virtually solve the stable points of the Lagrange function (6). So all methods based on formulas (5) are called the Lagrange methods. Given a current iterative point $(\mathbf{x}^k \in \mathbb{R}^m, \boldsymbol{\lambda}^k \in \mathbb{R}^n)$, the Newton step of formulas (5) is $({\delta_{\mathbf{x}}}^k, {\delta_{\boldsymbol{\lambda}}}^k)$ which satisfies,

$$\begin{pmatrix} W(\mathbf{x}^k, \boldsymbol{\lambda}^k) & -A(\mathbf{x}^k) \\ -A^T(\mathbf{x}^k) & 0 \end{pmatrix} \begin{pmatrix} {\delta_{\mathbf{x}}}^k \\ {\delta_{\boldsymbol{\lambda}}}^k \end{pmatrix} = - \begin{pmatrix} \nabla f(\mathbf{x}^k) - A(\mathbf{x}^k)\boldsymbol{\lambda}^k \\ -c(\mathbf{x}^k) \end{pmatrix}, \tag{7}$$

where,

$$\begin{aligned} &A(\mathbf{x}) = \nabla c(\mathbf{x})^T, \\ &W(\mathbf{x}, \boldsymbol{\lambda}) = \nabla^2 f(\mathbf{x}) - \sum_{i=1}^{n} \lambda_i \nabla^2 c_i(\mathbf{x}). \end{aligned} \tag{8}$$

It is well known that the basic Newton iteration only achieves local convergence. A strategy for controlling the step size is required to obtain global convergence. So we define a penalty function,

$$P(\mathbf{x}, \boldsymbol{\lambda}) = \| \nabla f(\mathbf{x}) - A(\mathbf{x})\boldsymbol{\lambda} \|^2 + \| c(\mathbf{x}) \|^2, \tag{9}$$

for determining the step size α_k of the kth iteration, because the correction $\left({\delta_{\mathbf{x}}}^k, {\delta_{\boldsymbol{\lambda}}}^k\right)$ satisfies

$$\left(\left({\delta_{\mathbf{x}}}^k\right)^T, \left({\delta_{\boldsymbol{\lambda}}}^k\right)^T\right) \nabla P(\mathbf{x}^k, \boldsymbol{\lambda}^k) = -2P(\mathbf{x}^k, \boldsymbol{\lambda}^k) \le 0. \tag{10}$$

The direction represents a descent of $P(\mathbf{x}, \boldsymbol{\lambda})$ at $(\mathbf{x}^k, \boldsymbol{\lambda}^k)$. Then the new iteration is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k {\delta_{\mathbf{x}}}^k, \quad \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha_k {\delta_{\boldsymbol{\lambda}}}^k. \tag{11}$$

In the following, the line search Lagrange-Newton algorithm based on formula (7) is given.

Algorithm 1 *(the Lagrange-Newton Algorithm)*

```
1.1 Give the initial value $(\mathbf{x}^0, \boldsymbol{\lambda}^0)$, the tolerance error $\varepsilon \ge 0$, and $\beta = 0.5$. Let $k := 0$;
1.2 Compute $P(\mathbf{x}^k, \boldsymbol{\lambda}^k)$. If $P(\mathbf{x}^k, \boldsymbol{\lambda}^k) \le \varepsilon$, stop the computation; Otherwise, solve formula (7) to obtain ${\delta_{\mathbf{x}}}^k$ and ${\delta_{\boldsymbol{\lambda}}}^k$, and let $\alpha \leftarrow 1$;
1.3 If
        $P(\mathbf{x}^k + \alpha{\delta_{\mathbf{x}}}^k, \boldsymbol{\lambda}^k + \alpha{\delta_{\boldsymbol{\lambda}}}^k) \le (1 - \beta\alpha)P(\mathbf{x}^k, \boldsymbol{\lambda}^k)$,   (12)
    goto Step 4; Otherwise, $\alpha \leftarrow \alpha/4$, and return to Step 3;
1.4 Update
    $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha{\delta_{\mathbf{x}}}^k$ and $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha{\delta_{\boldsymbol{\lambda}}}^k$.
    Set $k \leftarrow k + 1$, and return to Step 2.
```

3.2. Matrix computation of the model

For the given optimization model (3) of spherical parametrization, we can get the expression of all matrices in the Lagrange-Newton method.

Denote the submatrices W and A of the Hessian matrix in the formula (7) as the form of block matrices,

$$\begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1n} \\ W_{21} & W_{22} & \cdots & W_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n1} & W_{n2} & \cdots & W_{nn} \end{pmatrix},$$

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix},$$

where,

$$\begin{aligned} W_{ij} &= \frac{\partial^2 L}{\partial X_i \partial X_j} \\ &= \begin{pmatrix} \frac{\partial^2 L}{\partial X_{i_1} \partial X_{j_1}} & \frac{\partial^2 L}{\partial X_{i_1} \partial X_{j_2}} & \frac{\partial^2 L}{\partial X_{i_1} \partial X_{j_3}} \\ \frac{\partial^2 L}{\partial X_{i_2} \partial X_{j_1}} & \frac{\partial^2 L}{\partial X_{i_2} \partial X_{j_2}} & \frac{\partial^2 L}{\partial X_{i_2} \partial X_{j_3}} \\ \frac{\partial^2 L}{\partial X_{i_3} \partial X_{j_1}} & \frac{\partial^2 L}{\partial X_{i_3} \partial X_{j_2}} & \frac{\partial^2 L}{\partial X_{i_3} \partial X_{j_3}} \end{pmatrix}, \end{aligned}$$

$$A_{ij} = \frac{\partial^2 L}{\partial \lambda_i \partial X_j} = \begin{pmatrix} \frac{\partial^2 L}{\partial \lambda_i \partial X_{j_1}} \\ \frac{\partial^2 L}{\partial \lambda_i \partial X_{j_2}} \\ \frac{\partial^2 L}{\partial \lambda_i \partial X_{j_3}} \end{pmatrix}.$$

Here W_{ij} is a 3×3 matrix, and A_{ij} is a 3×1 vector. Usually, the uniform spring constants $\kappa_{ij} = 1$ are used. From the model (3), we can obtain the Lagrange function as follows:

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}) &= f(\mathbf{x}) - \sum_{i=1}^{n} \lambda_i c_i(\mathbf{x}) \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j \in D(i)} \| X_i - X_j \|^2 - \sum_{i=1}^{n} \lambda_i (\| X_i \|^2 - 1). \end{aligned}$$

And, in term of the formula (8), all the matrices to be computed can be expressed as follows:

$$W_{ij} = \begin{cases} \big(2d(i) - 2\lambda_i\big) I_3, & i = j, \\ -2I_3, & i \neq j,\ j \in D(i), \\ 0, & \text{otherwise.} \end{cases}$$

$$A_{ij} = \begin{cases} 2X_i, & i = j, \\ 0, & \text{otherwise.} \end{cases}$$

The right part of equation (7) is

$$\nabla_{X_i} f = 2 \sum_{j \in D(i)} (X_i - X_j).$$

Now, we have obtained the expression of all the matrices. In the following subsection, how to solve the corresponding large-scale linear system (7) is discussed.

3.3. Solving the linear system

Denote the linear system (7) as

$$Hx = b. \tag{13}$$

It is easy to show that the matrix H is highly sparse and symmetrical. In our experience, the case that the matrix H is singular occurs extremely rarely, and we can change the initial value to avoid the singular matrix. Hence, we can assume that the matrix H is nonsingular.

For a large-scale sparse linear system of equations, if its coefficient matrix is symmetric and positive definite, the Cholesky decomposition is generally applied. This method is efficient and robust. The linear system (13) is equivalent to

$$H^T H x = H^T b.$$

According to the properties of the matrix H, the new coefficient matrix $H^T H$ is highly sparse, symmetrical, and positive definite. Then the Cholesky decomposition can be applied onto $H^T H$. In the implementation, we use a scheme called row-indexed sparse storage mode, which requires storage of only about two times the number of nonzero matrix elements, to store the large coefficient matrix. The row-indexed storage mode is introduced in detail in [11].

In a valid parametrization result, overlapped parts should be eliminated. We use the same method as in [9] to detect overlapping. An overlapping is a region on the parametrization result, where there exists reverse triangles, whose normals are opposite to the normals of their neighbor triangles. We find these reverse triangles by testing the orientation of the sequence vertices along the boundary of each face. It is important that the three vertices are recorded in a clockwise turn. This can be computed by estimating the sign of $\big((v_{j_2} - v_{j_1}) \times (v_{j_3} - v_{j_1}) \cdot v_{j_1}\big)$, where $t_j = (v_{j_1} v_{j_2} v_{j_3})$ is a triangle.

Starting with the given mesh as the initial value, the new mesh can be obtained using Algorithm 1. While overlaps exist in the current mesh, we update the tolerance error by mutiltying the error by a factor $\theta \in (0, 1)$, say $\theta = 0.5$, and use Algorithm 1 to get a new mesh. Repeat this work until there are no overlapped part in the obtained mesh. This mesh is a valid parametrization result.

4. Results and discussion

The statistical data of some spherical parametrization results using the solving algorithm are provided in Table 1.

From Table 1, it is known that, compared with hierarchical method in [9], our new method cannot reduce computing time effectively. But the new method is improved at

Model	Num. vertices	Num. faces	Runtime (seconds)	Num. iterations
Bishop	250	496	0.891	16
Blob	8036	16068	85	10
Venus face	8268	16532	34	3
Venus body	11362	22720	52	4
Gargoyle	10002	20000	50	4
Cow	11610	23218	201	14
Skull	20002	40000	560	24

Table 1. Genus-zero examples

two main points. Firstly, the triangles in new parametrization results are with better shapes. The local comparison will be provided in the next subsection. Secondly, the new parametrization algorithm is more automatically. In the hierarchical algorithm, there are many parameters to be specified by users. And in the new algorithm, only one parameter, which is the global tolerance error, need to be given.

4.1. Local shape comparison

In Figures 1 and 2, the triangle local shape comparison is provided. In all the figures, there is a point labeled with a red square to identify the same vertex in the original meshes, the whole parametrization results with the methods in [9] and in the current paper, and the local zoom-in triangles, respectively.

With the local zoom-in viewports, we can see that the results using hierarchical method usually have some lathy triangles, and the corresponding triangles in the new results are more better.

4.2. Surface fitting

Based on PHT-spline spaces, parametric surfaces can be constructed efficiently and adaptively to fit a genu-zero mesh after its spherical parametrization has been obtained. In [1] a surface fitting algorithm with PHT-splines is proposed. Now we apply this algorithm to fit genus-zero meshes based on their spherical parametrizations. In Figure 3, we can see that the new result, especially parts around eyes, nose, forehead and cheek, is better than that with the hierarchical parametrization results. Figure 3(a) and 3(d) are the whole fitting surfaces based on the different parametrization results; 3(b) and 3(c) are the local details of 3(a), while 3(e) and 3(f) are the local details of the new result 3(d).

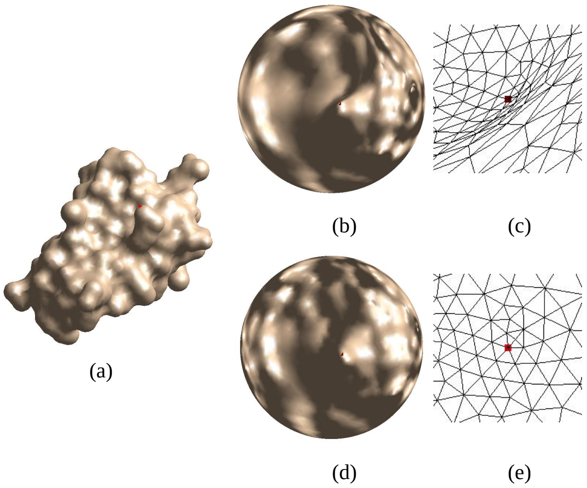

Figure 1. The parametrization results of the model Blob. (a): the original mesh. (b): the parametrization result using the hierarchical method in [9]. (d): the parametrization result using the Lagrange-Newton method in this paper. (c) and (e): the local details comparison of the parametrization results (b) and (d).

5. Conclusions

We have presented a new approach based on the Lagrange-Newton method to solve the spherical parameterization model of a genus-zero mesh proposed in [9]. Compared with the hierarchical algorithm, the new approach has improved the results in two points: the local shapes of the triangles and less user-specified parameters. Specially, the PHT-spline surface fitting results with the new parametrization results are better than those with hierarchical results.

In the future, we will focus on the following works:

- Now, we use the discrete harmonic energy as the objective function, and obtain the results with good shapes. But, the size of triangles are different widely. So a better objective function or a new storage structure may be given.
- In this paper, the genus-zero surfaces can be parameterized efficiently. The parametrization method of higher genus surfaces can be generalized from this method.

Acknowledgement

The authors are support by A National Key Basic Research Project of China (No. 2004CB318000), NSF of

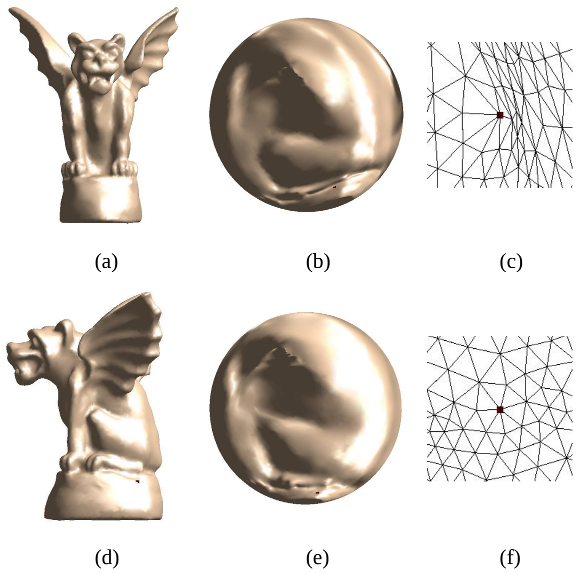

Figure 2. The parametrization of the model Gargoyle. (a) and (d): the diffirent sides of the original mesh. (b): the parametrization result using the hierarchical method in [9]. (e): the parametrization result using the Lagrange-Newton method in this paper. (c) and (f): the local details comparison of the parametrization results (b) and (e).

China (No. 60533060, 60473132 and 10626049), Doctorial Program of MOE of China and the 111 Project (No. b07033).

References

[1] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng. Polynomial splines over hierarchical t-meshes. *Sumbitted to Graphical Models, online version: http://staff.ustc.edu.cn/˜dengjs/files/papers/pht.pdf*, 2006.

[2] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Computer Graphics*, 29(Annual Conference Series):173–182, 1995.

[3] R. Fletcher. *Practial Methods of Optimization*. second edition, 1987.

[4] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(4):231–250, 1997.

[5] M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.

[6] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. *Advances in multiresolution for geometric modelling*, pages 157–186, 2005.

[7] X. Gu and S.-T. Yau. Global conformal parameterization. In *Symposium on Geometry Processing*, pages 127–137, 2003.

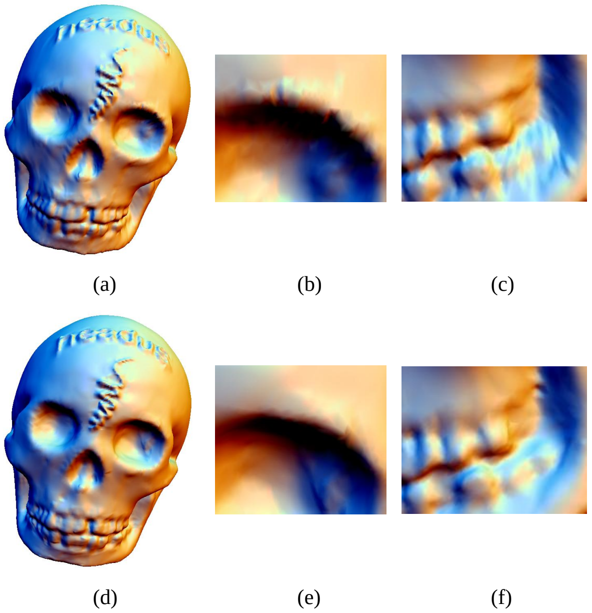

Figure 3. Fitting genus-zero meshes. (a) and (d): the whole fitting surfaces based on the parametrization results using the hierarchical method in [9] and the Lagrange-Newton method in this paper. (b) and (e): the local details on the left eye of the model skull. (c) and (f): the local details on the teeth.

[8] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):181–189, 2000.

[9] Y. Li, Z. Yang, and J. Deng. Spherical parametrization of genus-zero meshes by minimizing discrete harmonic energy. *Journal of Zhejiang University science A*, 7(9):1589–1595, 2006.

[10] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Trans. Graph.*, 22(3):340–349, 2003.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.

[12] A. Sheffer, E. Praun, and K. Rose. *Mesh Parameterization Methods and their applications*. Now Publishers Inc, 2006.

[13] W. T. Tutte. How to draw a graph. *Proceedingsof the London Mathematical Society*, 13:743–768, 1963.