Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Spherical parametrization of genus-zero meshes by minimizing discrete harmonic energy*

LI Ying, YANG Zhou-wang†, DENG Jian-song

(*Department of Mathematics, University of Science and Technology of China, Hefei 230026, China*)

†E-mail: yangzw@ustc.edu.cn

Received Apr. 20, 2006; revision accepted May 21, 2006

Abstract: The problem of spherical parametrization is that of mapping a genus-zero mesh onto a spherical surface. For a given mesh, different parametrizations can be obtained by different methods. And for a certain application, some parametrization results might behave better than others. In this paper, we will propose a method to parametrize a genus-zero mesh so that a surface fitting algorithm with PHT-splines can generate good result. Here the parametrization results are obtained by minimizing discrete harmonic energy subject to spherical constraints. Then some applications are given to illustrate the advantages of our results. Based on PHT-splines, parametric surfaces can be constructed efficiently and adaptively to fit genus-zero meshes after their spherical parametrization has been obtained.

Key words: Genus-zero meshes, Spherical parametrization, Discrete harmonic energy, Constrained optimization
doi:10.1631/jzus.2006.A1589 **Document code:** A **CLC number:** O24

INTRODUCTION

Parametrization is an important problem in Computer Graphics. 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 used domain is a connected region on the plane (Desbrun *et al*., 2002; Eck *et al*., 1995; Floater, 1997; 2003; Sheffer and Sturler, 2000); and if the mesh is with genus-zero, the used domain is a unit sphere (Gotsman *et al*., 2003; Haker *et al*., 2000; Praun and Hoppe, 2003; Sheffer *et al*., 2004). Usually, the meshes consist of triangles, so the mappings are piecewise linear and we only need to compute the positions of the vertices. Parametrizations have many applications in various fields, including texture mapping, scattered data fitting, surface approximation and remeshing, reparametrization of spline surfaces, repair of CAD models, morphing, and so on. And for a certain 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 are some methods to parametrize it onto a sphere. But according to our experiences, these results are unfit for surface fitting with splines. Hence in this paper, we will propose a method to parametrize a genus-zero mesh so that a surface fitting algorithm in PHT-splines can generate a good result.

Related works

In this subsection we review some previous works on mesh parametrizations. For a more detailed summary, please refer to (Floater and Hormann, 2002; 2005).

1. Planar parametrization

* Project supported by the Outstanding Youth Grant of Natural Science Foundation of China (No. 60225002), the National Basic Research Program (973) of China (No. 2004CB318000), the National Natural Science Foundation of China (Nos. 60533060 and 60473132)

A class of methods is known as convex combination maps (Tutte, 1963). These methods map the mesh boundary to some convex polygon and define each interior vertex as a convex combination of its neighbors. Tutte (1963) placed each vertex at the centroid of its neighbors. This method only considers the topological structure of the mesh. Floater (1997) used a specific weight to improve the effect of the mapping, and this method is shape-preserving in the sense that planar convex mesh parametrization is affine invariant. Furthermore, Floater (2003) gave a method based on mean value coordinates which obtains the resulting parametrization depending smoothly on the vertices of the mesh.

Maillot *et al*.(1993) gave a mathematical formulation for the distortion of the mapped image and viewed the mapping as an energy-minimization process by considering a general optimization function. Here the energy is an approximation of the integral of the Green-Lagrange deformation tensor.

Sheffer and Sturler (2000) proposed a method which minimizes the relative distortion of the planar angles with respect to their counterparts in 3D space.

Sander *et al*.(2001) developed a method which minimizes texture stretch to balance sampling rates over all locations and directions on the surface and texture deviation to obtain accurate textured mesh approximations.

Eck *et al*.(1995) introduced the discrete harmonic map to parametrize a simple mesh. The method proposed in the current paper is hinted by this work. Eck's method is a quadratic minimization problem and can be reduced to a linear system of equations.

2. Spherical parametrization

In surface fitting a genus-zero mesh, its parametrization over a sphere is needed. The problem of mesh spherical parametrization is that of mapping a piecewise linear surface with a discrete representation onto a spherical surface. The mapping is represented by the parametric locations of vertices of the surface. There are many interesting methods in spherical parametrization. Many of these methods are very similar to those of mapping simple meshes onto planar domains, whereas some of the linear methods become non-linear versions.

Haker *et al*.(2000) used a method which maps the given genus-zero mesh into the plane and then uses stereographic projection to map it to a sphere. Unfortunately, they did not describe how the surface is split to allow for mapping into the plane. For some models, Haker's method fails to converge.

Gu and Yau (2003) gave an important point that harmonic maps from a closed genus-zero mesh to the 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. In the discrete case, piecewise linear mappings h: $M\to S^2$ are considered with the property that $h(P)$ lies on the unit sphere S^2 for every vertex P of the mesh M.

Sheffer *et al*.(2004) proposed a method which ensures a valid embedding. They formulated a set of necessary and sufficient conditions for the spherical angles of a triangulation to form a valid spherical triangulation. But the numerical procedure to solve the system is quite slow and not practical for meshes containing more than a few hundred vertices.

Praun and Hoppe (2003) extended the definition of stretch to consider a spherical parametrization h: $M\to S^2$. They analyzed the map h from a triangle T to the sphere. For any point P inside T, the Jacobian map J_h provides a local approximation for h. Consequently, distances around $h(P)$ get stretched through the map by a factor between $1/\gamma$ and $1/\Gamma$ with γ and Γ the singular values of J_h. They defined the stretch over the triangle T as:

$$L^2(T)=\sqrt{\frac{1}{A_{M_T}}\iint\left(\frac{1}{\gamma}+\frac{1}{\Gamma}\right)\mathrm{d}A_{M_T}(s,t)}, \qquad (1)$$

where $\mathrm{d}A_{M_T}(s,t)=\mathrm{d}s\mathrm{d}t$ is the differential mesh triangle area. Then, they minimized the stretch by adding a regularization term $\varepsilon(A_{M_T}/4\pi)^{p/2+1}(\Gamma)^p$ to avoid oversampling when $\Gamma>>\gamma$.

Our contribution

In this paper, we solve the problem of mapping a genus-zero surface to the unit sphere. First, the model of spherical parametrization based on minimizing the discrete harmonic energy is given, but the computation is much more complicated. Then we introduce a stable hierarchical spherical parametrization algorithm which reduces much time. Based on our parametrization results, a surface fitting algorithm with

PHT-splines (Deng *et al*., 2006) can generate good results, where parametric surfaces can be constructed efficiently and adaptively to fit genus-zero meshes.

The rest of the paper is organized as follows. In Section 2, we present our model and describe algorithm in detail. In Section 3, we give the results of our algorithm and compare them with other parametrization results. An application is given in Section 4. Finally, in Section 5, we conclude the paper.

MODEL AND ALGORITHM

The triangles in the mesh are usually with good shapes, i.e., the three edge lengths do not change dramatically. In surface fitting a mesh, its parametrization over some standard domain is needed. Hence for a triangle in the mesh, we need its corresponding triangle in the parametrization domain to be with good shape as well.

Unfortunately, the existing spherical parametrization methods do not satisfy these properties. On the other hand, for a simple mesh, Eck *et al*.(1995) proposed a discrete harmonic method which satisfies this property. Thus we will generalize Eck's model to spherical parametrization, so that the triangles in the sphere have good shape.

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

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

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

$$f(h,M)=\frac{1}{2}\sum_{(v_i,v_j)\in E}\kappa_{ij}\,\|h(v_i)-h(v_j)\|^2, \tag{3}$$

where the spring constants κ_{ij} are computed in many ways. In most cases and the rest of the paper, uniform spring constants are used.

The model

Denote

$$h(v_i)=\boldsymbol{X}_i\in\mathbb{R}^3,\ \boldsymbol{X}=(\boldsymbol{X}_1^{\mathrm{T}},\boldsymbol{X}_2^{\mathrm{T}},\ \ldots,\boldsymbol{X}_n^{\mathrm{T}}),$$
$$\hat{E}=\{(i,j)\,|\,(v_i,v_j)\in E\}.$$

Then we can setup the parametrization model by minimizing discrete harmonic energy in Eq.(3) with spherical constraints:

$$\min f(\boldsymbol{X})=\frac{1}{2}\sum_{(i,j)\in\hat{E}}\|\boldsymbol{X}_i-\boldsymbol{X}_j\|^2, \tag{4}$$
$$\text{s.t. } c_i(\boldsymbol{X})=\|\boldsymbol{X}_i\|^2-1=0,\ i=1,\ \ldots,\ n,$$

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

The algorithm

When solving constrained nonlinear programming problem Eq.(4), in which the constraints cannot easily be eliminated, it is necessary to balance the aims of educing the objective function and staying inside or close to the feasible region, in order to induce global convergence. This inevitably leads to the idea of a penalty function which enables f to be minimized whilst controlling constraint violations by penalizing them. The penalty function is some combination of f and c, and is smooth so as to use efficient techniques for smooth unconstrained optimization. For the equality problem, the penalty function is

$$P(\boldsymbol{X},\sigma)=\frac{1}{2}\sum_{(i,j)\in\hat{E}}\|\boldsymbol{X}_i-\boldsymbol{X}_j\|^2+\frac{1}{2}\sigma\sum_{i=1}^{n}(\|\boldsymbol{X}_i\|^2-1)^2. \tag{5}$$

The penalty is formed from a sum of squares of constraint violations and the parameter σ determines the amount of the penalty. Thus the technique of solving a sequence of minimization problems is suggested.

The objective function in Eq.(5) is with degree four, and it is hard to obtain its solution in one step. Hence we propose the following strategy to deal with this large-scale optimization problem. At first, two definitions are introduced to simplify the following description.

Definition 1 Given a mesh $M=(V,E)$, for any v_i, define $Star^m(v_i)$ in a recursive fashion:

$$Star^0(v_i)=\{v_i\},$$
$$Star^1(v_i)=\{v_i\}\cup\{v_j\,|\,(v_i,v_j)\in E\},$$
$$Star^m(v_i)=\bigcup_{v_j\in Star^1(v_i)}\{Star^{m-1}(v_j)\}.$$

Definition 2 Given a mesh $M=(V,E)$, for a triangle $t_j=(v_{j_1}v_{j_2}v_{j_3})$ in M, define:

$$Star^m(t_j)=Star^m(v_{j_1})\cup Star^m(v_{j_2})\cup Star^m(v_{j_3}).$$

1. Hierarchical strategy

First, we map the mesh M to a unit sphere using a simple map, such as a central projection. Starting with the current solution, the nearest local minimum of the objective function can be found. But there possibly exist some parts of parametrization mesh on the sphere which are overlapped. Among each part, there exists at least a triangle, whose normal is opposite to the normal of one of its neighbour triangles. Those triangles are defined as reverse ones. Then the following work is to find all the reverse triangles, and determine the corresponding set $Star^m(t_j)$ for every reverse triangle t_j (see the next strategy for details). All vertices not being included in this set are fixed after normalization, and we only need to deal with the other vertices by minimizing the discrete harmonic energy. In this way, more and more vertices are fixed. Finally, the algorithm is terminated on condition that all vertices are fixed. This is a hierarchical process, and the computing speed is fast.

2. Overlapping detection

In the parametrization result, overlapped parts should be eliminated. We find the 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 $((v_{j_2}-v_{j_1})\times(v_{j_3}-v_{j_1})\cdot v_{j_1})$, where $t_j=(v_{j_1}v_{j_2}v_{j_3})$ is a triangle. If the considered mesh has overlapped parts, we can find many triangles with the reverse orientation. We should expand these triangles $\{t_j\}$ to get the set $\bigcup_{t_j}\{Star^m(t_j)\}$. For different models, the parameter m is different.

Now, we give in detail our algorithm.

Algorithm 1 (Hierarchical algorithm):

Input a mesh M, the penalty factor $\sigma>0$ and the number m.

Output h: $M\to S^2$, where h is a valid parametrization.

Step 1: Given an initial map $h^{(0)}$: $M\to S^2$. Let the index set of the vertices $\hat{I}=\{1, \ldots, n\}$ and $l:=0$.

Step 2: Solve the unconstrained optimization problem

$$\min P(\boldsymbol{X})=\frac{1}{2}\sum_{i\in\hat{I},(i,j)\in\hat{E}}\|\boldsymbol{X}_i-\boldsymbol{X}_j\|^2+\frac{1}{2}\sigma\sum_{i\in\hat{I}}(\|\boldsymbol{X}_i\|^2-1)^2,$$

using Sub-Algorithm 2.

Step 3: Determine the set $T=\{t_j|t_j=(\boldsymbol{X}_{j_1}\boldsymbol{X}_{j_2}\boldsymbol{X}_{j_3})$ is a reverse triangle$\}$ in the current solution, and set $\boldsymbol{X}_i\leftarrow\frac{\boldsymbol{X}_i}{\|\boldsymbol{X}_i\|}$, $i\notin\hat{I}$, where $\hat{I}=\{i|\boldsymbol{X}_i\in\bigcup_{t_j\in T}Star^m(t_j)\}$. Then $\boldsymbol{X}=(\ldots,\boldsymbol{X}_i^{\mathrm{T}},\ldots)_{i\in\hat{I}}^{\mathrm{T}}$ is the variable in the next iteration.

Step 4: If $\hat{I}=\varnothing$, output $h=h^{(l)}$; otherwise, let $l:=l+1$, $\sigma=0$, and return to Step 2.

In the first step, the initial map $h^{(0)}$ is given as follows. Transfer all vertices to make the barycenter of the mesh M be the origin, scale the coordinate of all vertex to make the minimal box containing the whole mesh be a cube, and normalize the position vector of the vertex. In this way, the changed position of a vertex v_i is the value $h^{(0)}(v_i)$. In our implementation, we set $\sigma=0.5$ experientially.

In the second step, we use reset PRP conjugate gradient method (Fletcher, 1987) to solve this optimization problem. There are two conditions. In the first iteration, the object function is fourth order while in the later iterations, it becomes only second order because the penalty factor is zero. In the following, the algorithms are described in detail.

Sub-Algorithm 2 (Reset PRP conjugate gradient algorithm):

Input the initial vector $\boldsymbol{X}^{(0)}$, the allowable error ε.

Output the solution vector $\boldsymbol{X}^*$.

Step 1: Compute $\boldsymbol{g}^{(0)}=\boldsymbol{g}(\boldsymbol{X}^{(0)})$, $\boldsymbol{d}^{(0)}=-\boldsymbol{g}^{(0)}$, $\gamma_0=(\boldsymbol{g}^{(0)})^{\mathrm{T}}\boldsymbol{g}^{(0)}$. Let $k:=0$.

Step 2: Find the step size $\alpha_k=\arg\min_{\alpha>0}P(\boldsymbol{X}^{(k)}+\alpha\boldsymbol{d}^{(k)}\}$ through exact line search.

Step 3: Update $\boldsymbol{X}^{(k+1)}=\boldsymbol{X}^{(k)}+\alpha_k\boldsymbol{d}^{(k)}$. Compute $\boldsymbol{g}^{(k+1)}=\boldsymbol{g}(\boldsymbol{X}^{(k+1)})$, $\gamma_1=(\boldsymbol{g}^{(k+1)})^{\mathrm{T}}\boldsymbol{g}^{(k+1)}$, $\gamma_2=(\boldsymbol{g}^{(k)})^{\mathrm{T}}\boldsymbol{g}^{(k+1)}$.

Step 4: If $\gamma_1<\varepsilon$, stop and output the solution vector $\boldsymbol{X}^*=\boldsymbol{X}^{(k+1)}$.

Step 5: If $|\gamma_2|<0.5\gamma_1$, $\beta=(\gamma_1-\gamma_2)/\gamma_0$; otherwise, $\beta=0$. Compute $\boldsymbol{d}^{(k+1)}=-\boldsymbol{g}^{(k+1)}+\beta\boldsymbol{d}^{(k)}$. Set $\gamma_0=\gamma_1$, $k:=k+1$, and return to Step 2.

In Sub-Algorithm 2,

$$\boldsymbol{g}(\boldsymbol{X})=\nabla_{\boldsymbol{X}}P=(\ldots,\ (\nabla_{X_i}P)^{\mathrm{T}},\ \ldots)^{\mathrm{T}},\ i\in\hat{I}.$$

Our objective function is simple, and the element of $\boldsymbol{g}(\boldsymbol{X})$ can be computed as

$$\nabla_{X_i}P=\sum_{(i,j)\in\hat{E}}(\boldsymbol{X}_i-\boldsymbol{X}_j)+2\sigma(\|\boldsymbol{X}_i\|^2-1)\boldsymbol{X}_i.$$

In Step 2 of Sub-Algorithm 2, the exact step size α_k can be obtained by solving $\varphi'(\alpha)=0$. In the first iteration process, the function $\varphi'(\alpha)$ is third order. From the second iteration on, $\varphi'(\alpha)$ has become one order and the exact step size is got directly by the formulae

$$\alpha_k=\frac{-(\boldsymbol{g}^{(k)})^{\mathrm{T}}\boldsymbol{d}^{(k)}}{(\boldsymbol{g}^{(k+1)}-\boldsymbol{g}^{(k)})^{\mathrm{T}}\boldsymbol{d}^{(k)}}.$$

RESULTS AND DISCUSSION

Our main work is computing the surface spherical parametrization. We have implemented our method, and obtained some spherical parametrization results which prove that the algorithm is fast and efficient. The statistical data are listed in Table 1. In our algorithm, there are two parameters that must be rectified, which is the allowable errors and the number m of the set $\bigcup_{t_i}\{Star^m(t_i)\}$.

Table 1 Genus-zero examples

Model	Vertices	Faces	Runtime (s)	Iterations
Santa Claus	46048	92092	108	7
Venus	50002	100000	214	5
Bunny	34817	69630	131	11
Cow	11610	23216	118	6
Horse	48476	96948	465	13
Tyra	100002	200000	1866	19
Gargoyle	100002	200000	1357	6

Fig.1 demonstrates the process of Santa Clause spherical parametrization. Fig.1a is the mesh of Santa Clause, and Fig.1b is the initial mapping which has many reverse triangles. After the first iteration, some vertices are fixed and the others are put into the origin as the initial value for the next iteration as we can see in Fig.1c. Fig.1d shows the final parametrization result. The number m=8. We set the allowable error to be 0.6 in the first iteration, and 10^{-5} in the next iteration. Since then, the tolerance is multiplied by 0.1 after each iteration.

In Figs.2 and 3, comparison of our spherical parametrization results with those of Praun's (Praun and Hoppe, 2003) is shown. We can see that our triangles are usually with good shapes.

SURFACE FITTING

Based on PHT-spline spaces, parametric surfaces can be constructed efficiently and adaptively to fit a genus-zero mesh after its spherical parametrization has been obtained.

In (Deng *et al*., 2006) spline spaces are defined over T-meshes, which have enough flexibility in adaptive surface fitting. Based on the theory in (Deng *et al*., 2006), a set of basis functions of polynomial spline spaces over hierarchical T-meshes are defined, and a surface fitting algorithm is proposed recently. Now we apply this algorithm to fit the given genus-zero mesh based on its spherical parametrization and Praun's result (Praun and Hoppe, 2003). In Fig.4, we can see that our results are better than those with Praun's parametrization results.

CONCLUSION

We have presented an approach to parametrize a genus-zero mesh to a unit sphere and demonstrated our spherical parametrization results on a collection of challenging models. The algorithm is efficient, intrinsic, practical, and versatile for different surfaces. Specially, the surface fitting results with these parametrization results are better than those with other parametrization results.

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

(1) Use a better objective function to obtain better solutions.

(2) Solve parametrization problem of higher genus surfaces.

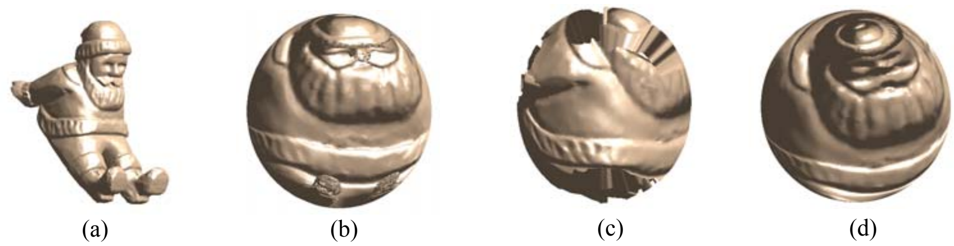

Fig.1 The parametrization process of Santa Claus. (a) The model; (b) The initial map; (c) The result after one step computing; (d) The final spherical parametrization

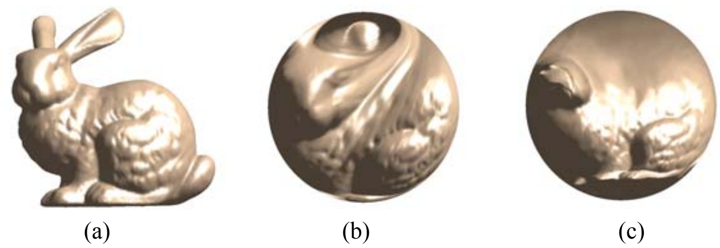

Fig.2 The parametrization of bunny. (a) The model of bunny; (b) Praun's result; (c) Our result

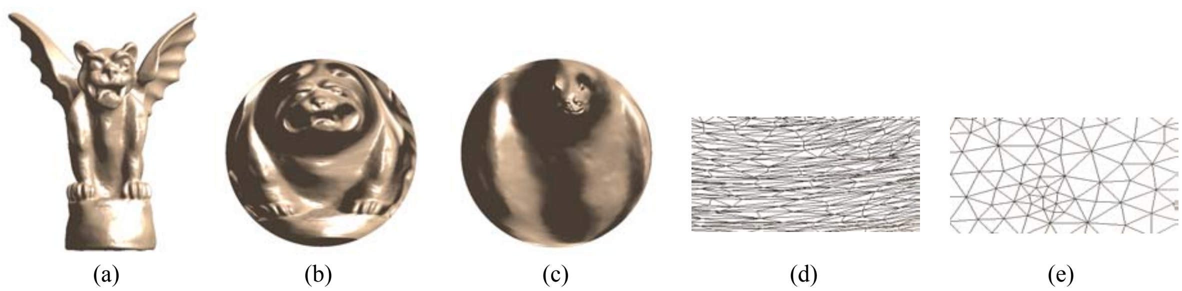

Fig.3 The parametrization of gargoyle. (a) The model of gargoyle; (b) Praun's result; (c) Our result; (d) Local details of Praun's result; (e) Local details of our result

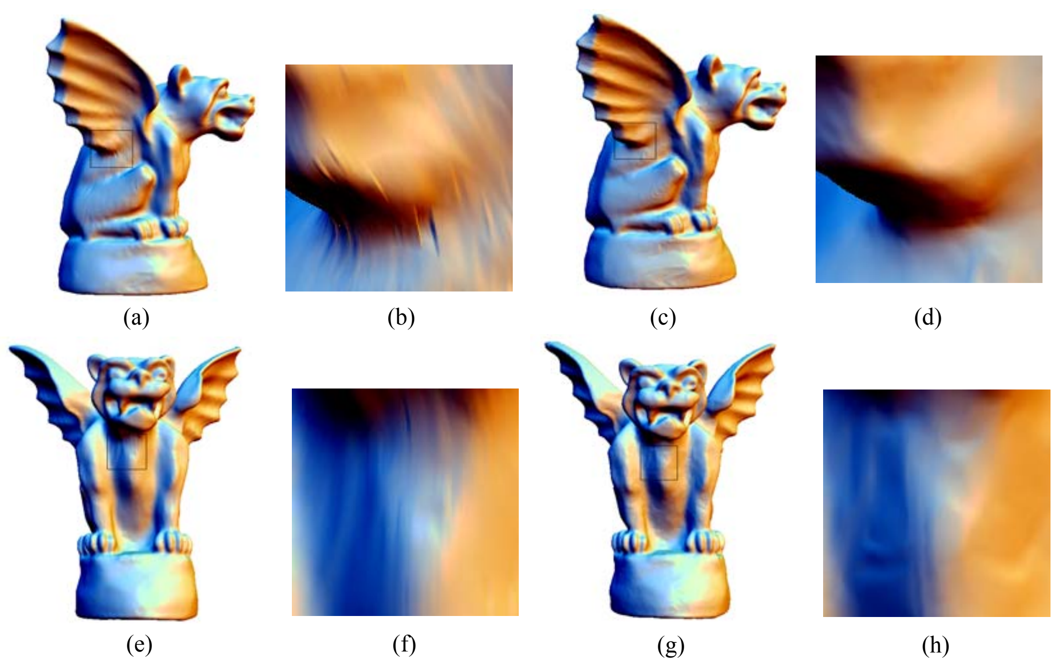

Fig.4 Fitting genus-zero meshes. (a) and (e): The fitting surfaces using Praun's parametrization result; (c) and (g): The fitting surfaces using our parametrization result; (b), (d), (f), (h): Local details

References

Deng, J., Chen, F., Feng, Y., 2006. Dimensions of spline spaces over T-meshes. *Journal of Computational and Applied Mathematics*, **194**(2):267-283. [doi:10.1016/j.cam.2005.07.009]

Desbrun, M., Meyer, M., Alliez, P., 2002. Intrinsic parameterizations of surface meshes. *Comput. Graph. Forum*, **21**(3):209-218. [doi:10.1111/1467-8659.00580]

Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., Stuetzle, W., 1995. Multiresolution analysis of arbitrary meshes. *Computer Graphics*, **29**:173-182.

Fletcher, R., 1987. Practial Methods of Optimization (Second Ed.). John Wiley & Sons Ltd.

Floater, M.S., 1997. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, **14**(3):231-250. [doi:10.1016/S0167-8396(96)00031-3]

Floater, M.S., 2003. Mean value coordinates. *Computer Aided Geometric Design*, **20**(1):19-27. [doi:10.1016/S0167-8396(03)00002-5]

Floater, M.S., Hormann, K., 2002. Parameterization of Triangulations and Unorganized Points. *In*: Iske, A., Quak, E., Floater, M.S. (Eds.), Tutorials on Multiresolution in Geometric Modelling. Springer-Verlag, Heidelberg, p.287-316.

Floater, M.S., Hormann, K., 2005. Surface Parameterization: A Tutorial and Survey. *In*: Dodgson, N.A., Floater, M.S., Sabin, M.A. (Eds.), Advances in Multiresolution for Geometric Modelling. Springer-Verlag, Heidelberg, p.157-186. [doi:10.1007/3-540-26808-1_9]

Gotsman, C., Gu, X., Sheffer, A., 2003. Fundamentals of spherical parameterization for 3D meshes. *ACM Trans. Graph.*, **22**(3):358-363. [doi:10.1145/882262.882276]

Gu, X., Yau, S.T., 2003. Global conformal surface parameterization. *In*: Kobbelt, L., Schröder, P., Hoppe, H. (Eds.), Proceedings of the 2003 Eurographics Symposium on Geometry Processing. Eurographics Association, p.127-137.

Haker, S., Angenent, S., Tannenbaum, A., Kikinis, R., Sapiro, G., Halle, M., 2000. Conformal surface parameterization for texture mapping. *IEEE Trans. on Visualization and Computer Graphics*, **6**(2):181-189. [doi:10.1109/2945.856998]

Maillot, J., Yahia, H., Verroust, A., 1993. Interactive Texture Mapping. Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH'93, p.27-34. [doi:10.1145/166117.166120]

Praun, E., Hoppe, H., 2003. Spherical parametrization and remeshing. *ACM Trans. Graph.*, **22**(3):340-349. [doi:10.1145/882262.882274]

Sander, P.V., Snyder, J., Gortler, S.J., Hoppe, H., 2001. Texture Mapping Progressive Meshes. SIGGRAPH 2001, p.409-416.

Sheffer, A., Sturler, E.D., 2000. Surface Parameterization for Meshing by Triangulation Flattening. Proceedings of the 9th International Meshing Roundtable, Sandia National Laboratories, p.161-172.

Sheffer, A., Gotsman, C., Dyn, N., 2004. Robust spherical parameterization of triangular meshes. *Computing*, **72**(1-2):185-193. [doi:10.1007/s00607-004-0056-9]

Tutte, W.T., 1963. How to Draw a Graph. Proceedings of London Mathematical Society, **13**:743-768.