Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Graphics 29 (2005) 972–979
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

Technical Section

A novel constrained texture mapping method based on harmonic map ☆

Yanwen Guo[a,b,*], Jin Wang[a], Hanqiu Sun[c], Xiufen Cui[a], Qunsheng Peng[a,b]

[a] State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310027, China
[b] Department of Mathematics, Zhejiang University, Hangzhou 310027, China
[c] Department of Computer Science and Engineering, CUHK, Hong Kong

Abstract

In this paper, we present a novel constrained texture mapping method based on the harmonic map. We first project the surface of a 3D model on a planar domain by an angle-based-flattening technique and perform a parametrization. The user then specifies interactively the constraints between the selected feature points on the parametric domain of the 3D model and the corresponding pixels on the texture image; the texture coordinates of other sample points on the 3D model are determined based on harmonic mapping between the parametric domain of the model and the texture image; finally we apply an adaptive local mapping refinement to improve the rendering result in real-time. Compared with other interactive methods, our method provides an analytically accurate solution to the problem, and the energy minimization characteristic of the harmonic map reduces the potential distortion that may result in the constrained texture mapping. Experimental data demonstrate good rendering effects generated by the presented algorithm.

Keywords: Texture mapping; Parametrization; Harmonic map

1. Introduction

Texture mapping has been widely applied in computer graphics and virtual reality. Without it, modeling and rendering the details of complex models would be a time-consuming task. The fundamental issue of texture mapping is to construct a one-to-one mapping between each point on the specified surface area of a 3D object and that on the texture plane. To facilitate such a mapping, a parametrization of the specified surface area is commonly performed. Furthermore, it is frequently necessary to ensure a rigid correspondence between the feature points of the 3D object and the respective pixels on the texture plane. For example, consider the mapping shown in Fig. 1, where an image of Andrey Hepburn is to be mapped onto the 3D head model. In this case we must ensure the precise mapping of the eyes, nose, etc. from the image and the model. This problem is commonly referred to as constrained texture mapping, for which the texture coordinates of some given feature points on the model must first be specified by the user, whereas the rest need be solved by some mechanism. These given feature points are usually called constrained points.

The constrained texture mapping of a 3D mesh is more tedious since no intrinsic parametrization of the 3D mesh satisfying such constraints is available.

☆ Project supported by National 973 program (No. 2002CB312101) and NSFC Grant (No. 60033010) and (No. 60403038), and CUHK Direct Grant (No. 4450002).
*Corresponding author. Tel.: +86 571 882 06681(503); fax: +86 571 882 06680.
E-mail address: ywguo@cad.zju.edu.cn (Y. Guo).

0097-8493/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2005.09.013

Previous methods of constrained texture mapping of 3D meshes are mainly based on iterative optimization which provide only an approximate solution. In this paper we present a novel analytical solution to this problem based on harmonic map. Our new method is not only accurate but also more efficient than previous methods. Experiments show good results of this method (Fig. 2).

1.1. Previous work

An important step of texture mapping onto a 3D mesh is the parametrization of the mesh model. This process generates a 2D planar mesh in the parametric space which maintains the same topology as the original 3D mesh, meanwhile minimizing the distortion of metric structures. A valid parametrization must ensure that the parameterized triangles have no foldovers or tears. In the past several years, many methods of 3D mesh parametrization have been proposed [1–12]. For example: Floater presented a shape preserving method [2], Eck et al. based their parametrization method on a harmonic model [1], and the methods in [9,12] try to approximate a conformal map. Some of these methods need to bind the boundary of a 3D mesh to a convex polygon [1–3,10], while others [4,9,12] do not have such constraints and the boundary of the embedded triangles is computed during the parametrization.

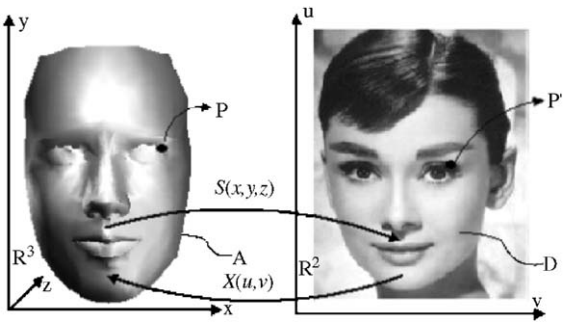

Fig. 1. Constrained texture mapping. $S(x, y, z)$ defines a one-to-one correspondence between surface A of R^3 and a subset D of R^2 with some user-defined constraints. For example, the point P on the canthus of the 3D model must correspond with the pixel P| on the canthus of Hepburn. $X(\,)$ is the inverse of $S(\,)$.

While the above approaches allow an image to be mapped onto a 3D mesh, they make no guarantee of matching features between the model and the texture. Continuous efforts were therefore made concerning constrained texture mapping. Most of the proposed methods [3,13–15] deal with the mesh model with disc topology as the parametrization of sample points within a disc domain and can easily be converted to a problem of solving a set of linear equations [2,12]. On the other hand, a complicated mesh model can always be decomposed into several "meaningful" disc topological patches [16].

Levy et al. proposed a method capable of dealing with iso-parametric curves [3]. Subsequently Levy solved the problem of feature mapping by respecting an arbitrary set of constrained features [13]. Levy's method works well for a small number of constraints but can lead to invalid parametrization when dealing with a large set of constraints. Tang et al. presented a fast analytic method of constrained texture mapping based on radial basis function (RBF) interpolation [14]. However their method relies heavily on the choice of the basis function of RBF and the result would be unsatisfactory if the basis functions employed are improper.

In a later work, Kraevoy et al. brought forward a method based on iterative optimization and obtained fine results [15]. This method first parameterizes the 3D meshes onto the planar region; then it triangulates the texture plane with respect to the specified constraints and divides the planar mesh consistently to construct the correspondence between the texture and the planar mesh. Secondly, it conducts a refinement operation to adjust the texture coordinates of the mesh points. This method needs to add Steiner vertices [15] and the number of Steiner vertices varies from several tens to thousands according to the complexity and density of the mesh model. Consequently, the efficiency of their method is low and the solution is approximate.

Fig. 2. A cow head mapped with the image of a leopard, using the algorithm presented in this paper.

1.2. Overview of our new approach

Our proposed method consists also of three steps. It first parameterizes a 3D mesh model using the angle-based-flattening (ABF) method [9]; it then specifies the corresponding feature points on the 3D model and the texture plane by user interaction and calculates the texture coordinates of all other sample points on the 3D mesh by harmonic mapping; finally it performs an "adaptive local mapping refinement" to adjust the texture coordinates of some points within the mismatched area. Compared with previous methods, our method has the following advantages:

- *Distortion minimization.* Due to the energy minimization property of the harmonic mapping, the metric distortion of our constrained texture mapping is small, as is testified by the results.
- *Analytical solution.* It is an analytically accurate solution to the constrained texture mapping problem, while other methods [13,15] are based on iterative optimization and provide only approximate solutions.
- *High performance.* Our method can deal with a large amount of data. Since it involves solving only sparse linear symmetric equations, our algorithm is much more efficient and much faster than others.
- *Real-time refinement.* With our "adaptive local mapping refinement" technique, the result of the mapping can be refined in real-time.
- *Concise process.* Our method need only specify a few correspondence constraints then solves the texture coordinates using a harmonic map while the method provided by Kraevoy et al. needs to segment the mesh and texture into several patches with Steiner vertices [15].

The remainder of this paper is organized as follows. Section 2 gives a brief review of ABF parametrization. Section 3 introduces the concept of harmonic map and its application in constrained texture mapping. Section 4 describes the details of an adaptive approach for local texture coordinate refinement. Experiment data are given in Section 5. In the last section, we summarize the proposed method and highlight the future work.

2. ABF parametrization

Our texture mapping method first parameterizes the 3D mesh onto the planar region. The chosen parametrization method should minimize the angular or metric distortion of the embedded 2D mesh, while still guaranteeing its validity. The ABF method [9] introduced by Sheffer et al. satisfies these conditions. This method defines the flattening problem entirely in terms of angles and minimizes the shape distortion of the embedded triangles by maintaining a consistent angular distribution of incident edges around each vertex on the parametric plane. After the angular distributions have been computed by solving an energy equation, the flat mesh can be fully determined once we fix the position of an initial interior vertex and the length and direction of an initial edge incident to that vertex.

As the ABF method can ensure the validity of parametrization and minimize the angular distortion, it is employed in our approach to parameterize the 3D mesh. Fig. 3 shows the parametrization results of two 3D mesh models.

3. The harmonic map

We apply the harmonic map technique to establish the correspondence between the parameterized 2D mesh and the texture plane. Harmonic maps have been studied intensively by many researchers over a long period of time. It is an extreme of the energy density that satisfies the corresponding Euler–Lagrange

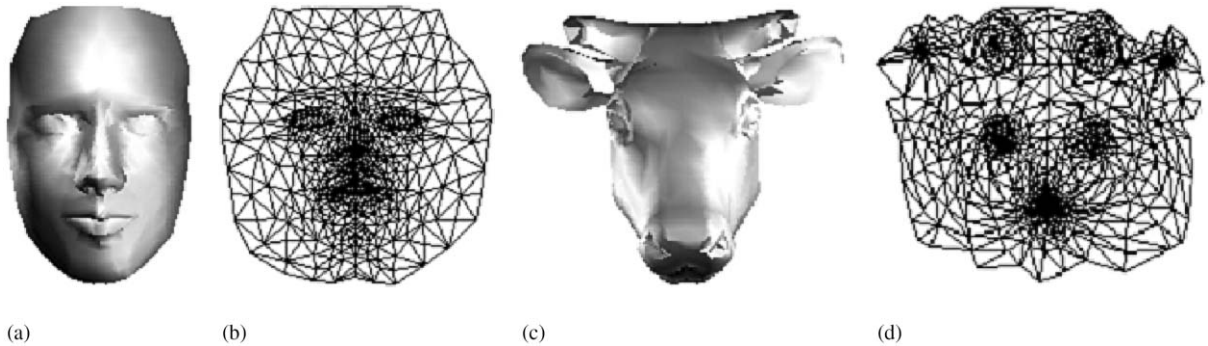

Fig. 3. 3D models and corresponding results of ABF parametrization. (a), (c) the face model (1344 triangles and 690 vertices) and the cow head model (1896 triangles, 972 vertices); (b), (d) the parametrization results of (a) and (c).

equation. Owing to its important properties of energy minimization, harmonic maps have wide applications to problems such as mesh parametrization [1] and surface matching [17].

In the following, we consider a special 2D harmonic map that maps between two simple manifolds with disc topology.

3.1. Two-dimensional harmonic map

Suppose: D and Ω are two parametric domains of disc topology on the 2D plane, $(\varepsilon,\eta)\in D$, $(u,v)\in\Omega$. ϕ represents a mapping between D and Ω, i.e., $\phi(\varepsilon,\eta)=(u,v)$ with $u=\phi_u(\varepsilon,\eta)$ and $v=\phi_v(\varepsilon,\eta)$. ϕ is harmonic if ϕ_u, ϕ_v satisfy the following two equations:

$$\Delta u=\partial^2u/\partial\varepsilon^2+\partial^2u/\partial\eta^2=0\in\Omega, \tag{1}$$

$$\Delta v=\partial^2v/\partial\varepsilon^2+\partial^2v/\partial\eta^2=0\in\Omega. \tag{2}$$

To get the analytical form of a harmonic map, let:

$$I(u)=\frac{1}{2}\int\int_{\Omega}[(\partial u/\partial\varepsilon)^2+(\partial u/\partial\eta)^2]\,\mathrm{d}\varepsilon\,\mathrm{d}\eta \tag{3}$$

$$I(v)=\frac{1}{2}\int\int_{\Omega}[(\partial v/\partial\varepsilon)^2+(\partial v/\partial\eta)^2]\,\mathrm{d}\varepsilon\,\mathrm{d}\eta. \tag{4}$$

It is easy to show that the solution of $\min I(u)$ and $\min I(v)$ yields a 2D harmonic mapping. In fact $I(u)$ and $I(v)$ stand for the energy function which estimates the distortions of a 2D mesh when it is mapped to the texture plane, the harmonic map minimizes such distortions. In the following we describe in detail how to find the harmonic map.

3.2. Applying harmonic map to texture mapping

Suppose the source region D of the harmonic map is a planar embedding of the 3D mesh resulting from parametrization: $D\approx\cup\Delta'$, where Δ' represents a triangle of D with $\Delta'=\Delta(p'_i,p'_j,p'_k)$, $p'_i=(\varepsilon_i,\eta_i)$ is the embedding point of a vertex on the 3D mesh. The solution region Ω is the triangular mesh on the texture plane with the same topology as D:

$$\Omega\approx\cup\Delta,\quad \Delta=\Delta(p_i,p_j,p_k),\quad p_i=(u_i,v_i),$$

where p_i refers to the texture coordinate of p'.

Assume that a point $p'=(\varepsilon,\eta)$ lies in Δ' and B_i, B_j and B_k are the barycentric coordinates of p' in Δ', let a point $p=(u,v)$ in Δ with the same barycentric coordinates as p', then in each triangle Δ we have:

$$p=B_ip_i+B_jp_j+B_kp_k,$$

i.e.,

$$u=B_iu_i+B_ju_j+B_ku_k, \tag{5}$$

$$v=B_iv_i+B_jv_j+B_kv_k. \tag{6}$$

Since B_i, B_j and B_k are computed as linear expressions of ε and η, the above equations define a mapping between ε, η and u, v. Assume the mapping is ϕ, i.e., $\phi(p')=p$, what we need to do is to solve for the minimum of Eqs. (3) and (4).

Note that in the discrete case, (3) and (4) are approximated by:

$$I(u)=\tfrac{1}{2}\sum[(\partial u/\partial\varepsilon)^2+(\partial u/\partial\eta)^2], \tag{7}$$

$$I(v)=\tfrac{1}{2}\sum[(\partial v/\partial\varepsilon)^2+(\partial v/\partial\eta)^2]. \tag{8}$$

Substituting (5), (6) on each triangle in (7), (8) and minimizing them, respectively, we obtain two sets of linear algebraic equations:

$$\mathbf{A}_{n\times n}\overrightarrow{\mathbf{u}}=\overrightarrow{\mathbf{k}}_{\mathbf{u}n\times1}, \tag{9}$$

$$\mathbf{A}_{n\times n}\overrightarrow{\mathbf{v}}=\overrightarrow{\mathbf{k}}_{\mathbf{v}n\times1}, \tag{10}$$

where $\mathbf{A}$ is a coefficient matrix and $\overrightarrow{\mathbf{k}}_{\mathbf{u}}$, $\overrightarrow{\mathbf{k}}_{\mathbf{v}}$ are two known vectors related to the specified texture coordinates of constrained points, which can be easily obtained through the minimization process. $\overrightarrow{\mathbf{u}}=(u_1,u_2,\ldots,u_n)$ and $\overrightarrow{\mathbf{v}}=(v_1,v_2,\ldots,v_n)$ are the unknown vectors to be solved. They represent the texture coordinates of the unconstrained sample points. Due to the uniform coefficients about u and v in (5) and (6), the coefficient matrices about $\overrightarrow{\mathbf{u}}$ and $\overrightarrow{\mathbf{v}}$ are the same.

It is easy to see that the matrix $\mathbf{A}$ is symmetric, positive definite and sparse, the conjugate gradient method can be used to solve the equations effectively [18].

4. Adaptive local mapping refinement

Constrained texture mapping normally involves specifying a large number of accurate feature points on both the mesh model and the texture plane. If the feature points on the 3D model are not matched precisely with the respective pixels in the texture, the texture mapping result will be poor (see Fig. 5d). We then need to adjust interactively the constrained texture pixels with respect to the selected feature points on the 3D model in order to generate a satisfactory rendering effect. Unfortunately, most previous methods need to re-calculate texture coordinates of the entire vertex set of the 3D model when adjusting the texture coordinate of even one vertex. This significantly decreases the efficiency of constrained texture mapping. In fact, the effect imposed by adjusting the texture coordinate of a vertex is mainly exerted in a local area (see *Appendix*), so we need only refine the texture coordinates of the vertices which are near to the concerned vertex. We call this approach "adaptive local mapping refinement".

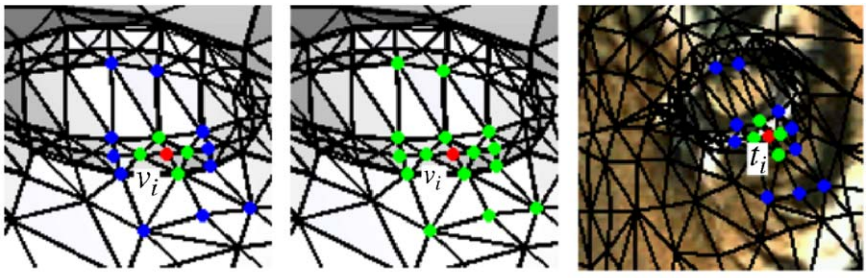

Fig. 4. (a) On the 3D mesh model, the green vertices belong to 1-*ring* of v_i; the blue are 2-*ring* of v_i. (b) The green vertices are 2-*ring region* of v_i. (c) On the texture plane, t_i is the texture point corresponding to v_i. Δ_1, Δ_2 of v_i is the sum of displacement of the green points, blue points, respectively, during each adjustment of t_i on the texture plane.

Assume that v_i, p_i are a pair of constrained feature points on the 3D mesh and the texture plane that need to be adjusted. Normally, we adjust the positions of p_i on the texture plane to fit v_i.

For description convenience, we give the following notations for a 3D mesh model (see Fig. 4):

- *topological distance between* v_i *and* v_j: the number of edges on the shortest path connecting v_i with v_j.
- *k-ring of* v_i: those vertices whose topological distance to v_i is k.
- *k-ring region of* v_i: those vertices whose topological distance from v_i is not greater than k.
- Δ_k *of* v_i: total texture coordinates' variance of the vertices within the *k-ring* of v_i.

After the user adjusts p_i, our algorithm first re-calculates the texture coordinates of vertices within the 1-*ring* of v_i by using the harmonic model described in Section 3 and solving a set of equations whose rank is equal to the vertex number of the 1-*ring*, while the texture coordinates of vertices outside the 1-*ring* remain unchanged; if Δ_1 *of* v_i is smaller than a given threshold, the algorithm terminates; otherwise the algorithm continues to adjust the texture coordinates of the 2-*ring region of* v_i which involves solving equations of a higher rank. The above process is iteratively conducted until the final variance is smaller than the threshold or the iterative count reaches a maximum constant. This approach can be described briefly as follows.

Algorithm: Adaptive local mapping refinement

step1 Given a threshold ε and a positive constant K.
step2 Set $k = 1$ and $\Delta = +\infty$.
step3 while $\triangle > \varepsilon$
begin
Use harmonic map (as in Section 3.2) to adjust those texture coordinates of the vertices included in *k-ring region* of v_i;
Compute Δ_k of v_i; $\Delta = \Delta_k$;
$k = k + 1$;
if $k > K$ **return**;
end
end

Since our approach involves solving low rank sparse equations, it is efficient and can be performed in real-time. If several pairs of feature points on the 3D model and the texture plane need to be adjusted, the above approach can be performed iteratively. Fig. 5 shows an example of real-time adjustment.

5. Experimental results

We employed several typical models and textures to test our algorithm on an Intel Pentium IV 1.6 GHz PC with 256 MB main memory under the Windows XP operating system (In Figs. 2, 6–8, the red dots are the constraints). Table 1 lists the performance statistics of our algorithm, including the number of triangles and vertices of the mesh model (#Tris/Vers), the number of constraints specified (#Cons), the rank of Eqs. (3) and (4) (#Eqas), the number of iterations for solving the equations (#Ites) and the computation time (Running time).

In Fig. 6, the image of a tiger (a) and a leopard (c) are mapped onto a face model with 23 and 24 constraints, respectively. Fig. 6(b) and (d) show two rendering results.

Fig. 7 shows the mapping of an image of girl shirt onto a body model composed of 3016 triangles, it takes 0.472 s to get the fine rendering result (Fig. 7(c)).

Fig. 8 shows the result of mapping a tiger image to the cow head model (the same one as Fig. 3(c)), which has

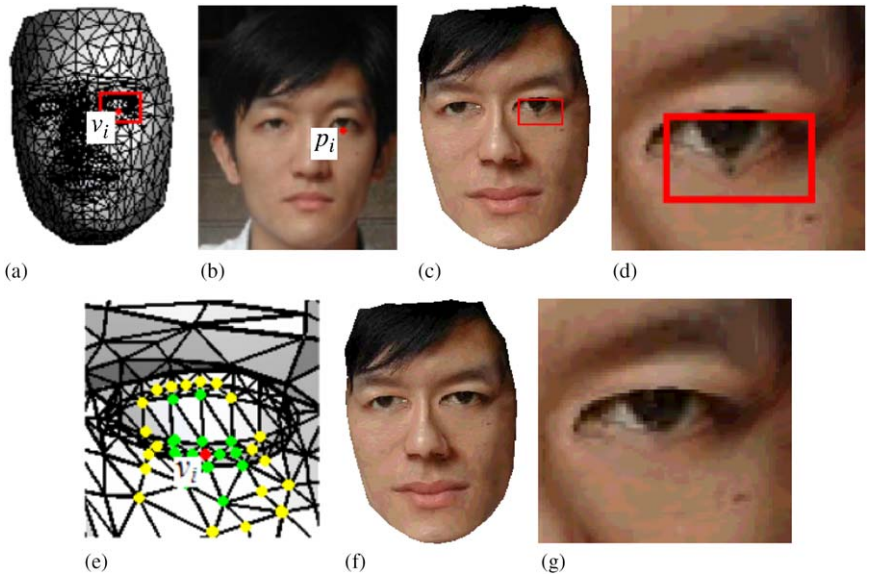

Fig. 5. Mapping a facial image (b) to a 3D face model (a), we get (c). However, the quality of the eyelid (region in red rectangle of (c)) is less satisfactory and needs be refined. (d) is a close-up of the eyelid. (f) is the result of "adaptive local mapping refinement" with two iterations after adjusting p_i and (g) is a close-up of the refined eyelid. (e) is the zoomed region surrounded by the red rectangle in (a); the vertices within the 2-*ring region* of v_i are showed in green; while the vertices included in the 3-*ring* of v_i are shown in yellow. As the 2-*ring region* of v_i contains rare vertices, solving the local mapping is real-time.

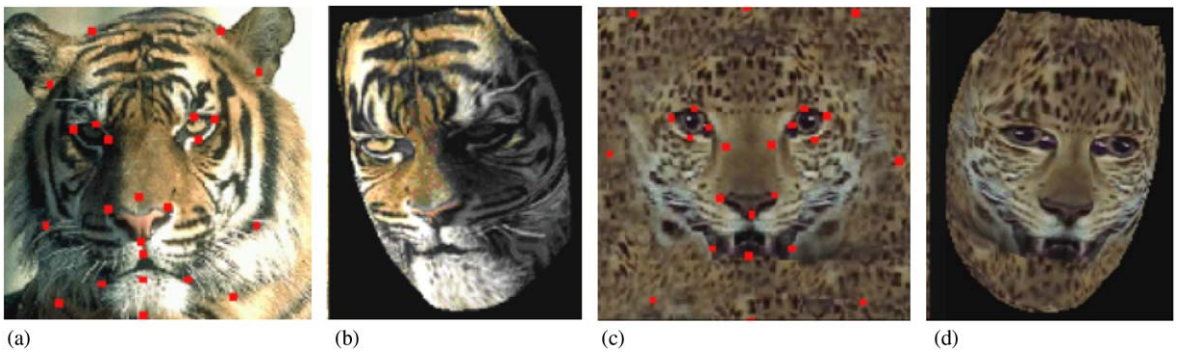

Fig. 6. A face model is mapped with different textures: (a), (c) are original textures with constrained feature points; (b), (d) are textured results.

1896 triangles. 28 constraints were imposed. The equations were solved by 118 iterations in 0.0638 s. In Fig. 2, a leopard face is mapped onto the cow head model.

6. Conclusions and future work

In this paper a novel constrained texture mapping method based on harmonic mapping is presented; it provides an analytically accurate solution to the problem of constrained texture mapping. Good results are achieved, and the result can be refined in real-time with an "adaptive local mapping refinement" technique. Experiments show that our algorithm is efficient and robust.

An interesting problem for future research is the automatic specification of constraints, the technology of biometrics and computer vision can help with the placement of these constraints. Besides, currently a complicated model, (e.g., one with several genera) must be cut into several parts for texture mapping leading to discontinuity of the rendering result. How to resolve this problem is another piece of future work.

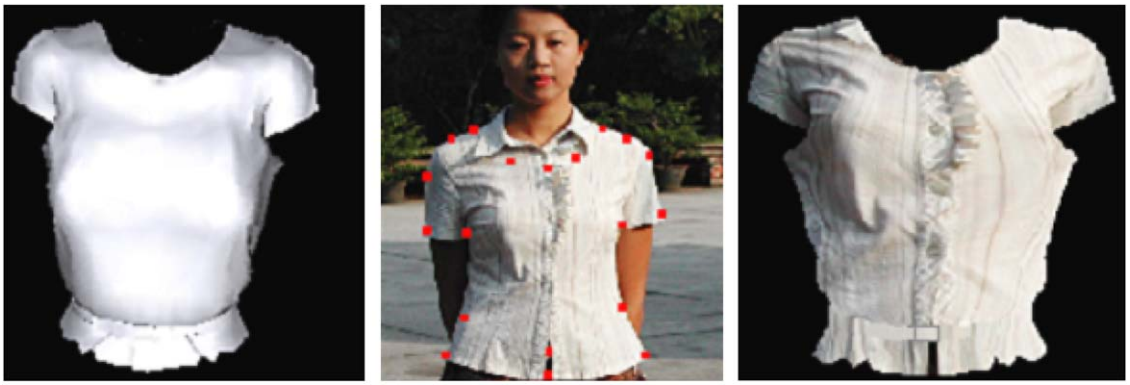

Fig. 7. Texturing of a body model: (a) 3D body model; (b) the texture image with constrained feature points (red dots); (c) textured model.

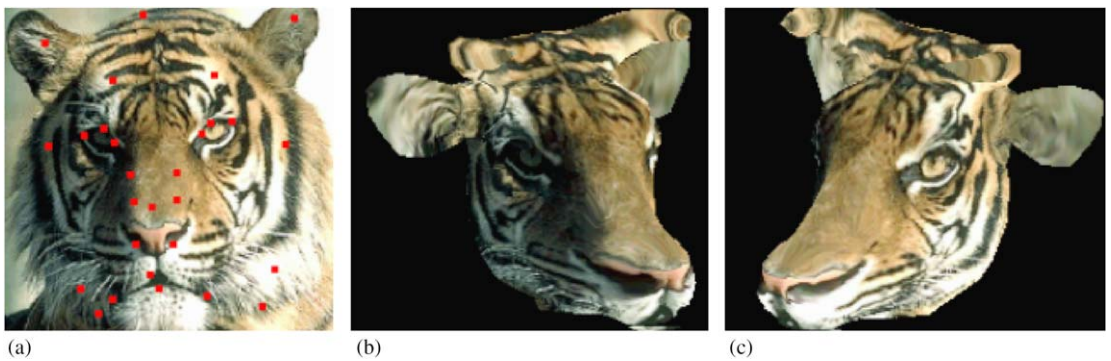

Fig. 8. Texturing of a cow head: (a) the texture image with constrained feature points (red dots); (b), (c) textured model at different views.

Table 1
Performance results

Model/Texture	#Tris/Vers	#Cons	#Eqas	#Ites	Running time (s)
Cow/Leopard	1896/972	21	951	122	0.064
Rachel/Sadam	1344/690	24	666	81	0.045
Rachel/Tiger	1344/690	23	667	86	0.047
Rachel/Leopard	1344/690	24	666	83	0.046
Body/Shirt	3016/1650	19	1631	436	0.472
Cow/Tiger	1896/972	28	944	118	0.0638

Appendix

Without losing generality assume that v_i, p_i are a pair of constrained feature points on the 3D mesh and texture image, we show that the "adaptive local mapping refinement" with one iteration after adjusting the texture coordinate of one constrained feature vertex on the 3D mesh produces an approximation to the accurate solution. In this case, only the texture coordinates of the vertices within 1-*ring* of v_i are re-calculated.

The equations deduced from the harmonic map to solve initial texture coordinates in Subsection 3 is the formula (5). Here, we only consider $\mathbf{A}\vec{\mathbf{u}} = \vec{\mathbf{k}}_{\mathbf{u}}$, where $\mathbf{A} = (a_{ij})_{n\times n}$ is a symmetric matrix relating to the 3D model, $\vec{\mathbf{u}} = (u_1, u_2, \ldots, u_n)^{\mathrm{T}}$ is the initial texture coordinates vector and $\vec{\mathbf{k}}_{\mathbf{u}} = (k_{u1}, k_{u2}, \ldots, k_{un})^{\mathrm{T}}$ is the vector resulting from constrained feature points.

Assume: 1-*ring* of $v_i = \{v_1, \ldots, v_l\}$, 2-*ring* $= \{v_{l+1}, \ldots, v_{l+m}\}$, $\overrightarrow{\mathbf{u_l}} = (u_1, \ldots, u_l)^{\mathrm{T}}$, $\overrightarrow{\mathbf{u_m}} = (u_{l+1}, \ldots, u_{l+m})^{\mathrm{T}}$, $\overrightarrow{\mathbf{u_n}} = (u_{l+m+1}, \ldots, u_n)^{\mathrm{T}}$, $\overrightarrow{\mathbf{k_{ul}}} = (k_{u1}, \ldots, k_{ul})^{\mathrm{T}}$, $\overrightarrow{\mathbf{k_{um}}} = (k_{u(l+1)}, \ldots, k_{u(l+m)})^{\mathrm{T}}$, $\overrightarrow{\mathbf{k_{un}}} = (k_{u(l+m+1)}, \ldots, k_{un})^{\mathrm{T}}$, then $\mathbf{A}\overrightarrow{\mathbf{u}} = \overrightarrow{\mathbf{k_u}}$ can be written as:

$$\begin{pmatrix} A_{l\times l} & B_{l\times m} & 0_{l\times(n-l-m)} \\ B_{m\times l} & D_{m\times m} & E_{m\times(n-l-m)} \\ 0_{(n-l-m)\times l} & E_{(n-l-m)\times m} & G_{(n-l-m)\times(n-l-m)} \end{pmatrix} \cdot \begin{pmatrix} \overrightarrow{\mathbf{u_l}} \\ \overrightarrow{\mathbf{u_m}} \\ \overrightarrow{\mathbf{u_n}} \end{pmatrix} = \begin{pmatrix} \overrightarrow{\mathbf{k_{ul}}} \\ \overrightarrow{\mathbf{k_{um}}} \\ \overrightarrow{\mathbf{k_{un}}} \end{pmatrix}, \tag{11}$$

in particular, the upper-right and lower-left submatrices of $\mathbf{A}$ are zero matrices reflecting that there are no edges that directly link 1-*ring* vertices of v_i with the vertices out of 2-*ring*.

After adjusting p_i, $k_{u1}, \ldots, k_{ul}$ are changed to $k_{u1} + \Delta_{u1}, \ldots, k_{ul} + \Delta_{ul}$. Let $\overrightarrow{\Delta_l} = (\Delta_{u1}, \ldots, \Delta_{ul})^{\mathrm{T}}$, thus the objective equation is changed to:

$$\begin{pmatrix} A_{l\times l} & B_{l\times m} & 0_{l\times(n-l-m)} \\ B_{m\times l} & D_{m\times m} & E_{m\times(n-l-m)} \\ 0_{(n-l-m)\times l} & E_{(n-l-m)\times m} & G_{(n-l-m)\times(n-l-m)} \end{pmatrix} \cdot \begin{pmatrix} \overrightarrow{\mathbf{u_l^1}} \\ \overrightarrow{\mathbf{u_m^1}} \\ \overrightarrow{\mathbf{u_n^1}} \end{pmatrix} = \begin{pmatrix} \overrightarrow{\mathbf{k_{ul}}} + \overrightarrow{\Delta_l} \\ \overrightarrow{\mathbf{k_{um}}} \\ \overrightarrow{\mathbf{k_{un}}} \end{pmatrix}, \tag{12}$$

where $\overrightarrow{\mathbf{u^1}} = (\overrightarrow{\mathbf{u_l^1}}, \overrightarrow{\mathbf{u_m^1}}, \overrightarrow{\mathbf{u_n^1}})$ is the accurate solution deduced from the harmonic map.

However the texture coordinates vector derived after "adaptive local mapping refinement" is $\overrightarrow{\mathbf{u'}} = (\overrightarrow{\mathbf{u_l^1}}, \overrightarrow{\mathbf{u_m}}, \overrightarrow{\mathbf{u_n}})$, in which $\overrightarrow{\mathbf{u_l^1}}$, i.e. the texture coordinates of 1-*ring* are computed by solving:

$$A_{l\times l} \cdot \overrightarrow{\mathbf{u_l^1}} + B_{l\times m} \cdot \overrightarrow{\mathbf{u_m}} = \overrightarrow{\mathbf{k_{ul}}} + \overrightarrow{\Delta_l}.$$

It is easy to see that $\overrightarrow{\mathbf{u_l^1}}$ and $\overrightarrow{\mathbf{u_n}}$ satisfy (7) by substituting $\overrightarrow{\mathbf{u_m^1}}$ with $\overrightarrow{\mathbf{u_m}}$, while $\overrightarrow{\mathbf{u_m}}$ dose not satisfy:

$$B_{m\times l} \cdot \overrightarrow{\mathbf{u_l^1}} + D_{m\times m} \cdot \overrightarrow{\mathbf{u_m}} + E_{m\times(n-l-m)} \cdot \overrightarrow{\mathbf{u_n}} = \overrightarrow{\mathbf{k_{um}}}.$$

However, it follows from (6) that:

$$B_{m\times l} \cdot \overrightarrow{\mathbf{u_l^1}} + D_{m\times m} \cdot \overrightarrow{\mathbf{u_m}} + E_{m\times(n-l-m)} \cdot \overrightarrow{\mathbf{u_n}} = \overrightarrow{\mathbf{k_{um}}} + B_{m\times l} \cdot (\overrightarrow{\mathbf{u_l^1}} - \overrightarrow{\mathbf{u_l}}),$$

and therefore,

$$\overrightarrow{\mathbf{u'}} - \overrightarrow{\mathbf{u^1}} = \mathbf{A}^{-1} \cdot \Delta\overrightarrow{\mathbf{k}_u} \quad \text{with } \Delta\overrightarrow{\mathbf{k}_u} = (0, \ldots, 0, (B_{m\times l} \cdot (\overrightarrow{\mathbf{u_l^1}} - \overrightarrow{\mathbf{u_l}}))^{\mathrm{T}}, 0, \ldots, 0).$$

In general, the total displacement of texture coordinates of 1-*ring* is relatively small. Hence if $\|(\overrightarrow{\mathbf{u_l^1}} - \overrightarrow{\mathbf{u_l}})\| < \varepsilon$, then $\|\overrightarrow{\mathbf{u'}} - \overrightarrow{\mathbf{u^1}}\| \leqslant \varepsilon \cdot \|\mathbf{A}^{-1}\| \cdot \|B_{m\times l}\|$.

This shows that the texture coordinates derived by "adaptive local mapping refinement" is an acceptable approximation to the accurate solution computed from harmonic map.

References

[1] Eck M, DeRose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution analysis of arbitrary meshes. In: Proceedings of SIGGRAPH'95, Los Angeles, CA; 1995. p. 173–82.

[2] Floater MS. Parametrization and smooth approximation of surface triangulations. Computer Aided Geometric Design 1997;14(3):231–50.

[3] Levy B, Mallet JL. Non-distortion texture mapping for sheared triangulated meshes. In: Proceedings of SIGGRAPH'98, Orlando, FL; 1998. p. 343–52.

[4] Hormann K, Greiner G. MIPS-An efficient global parametrization method. In: Curve and surface design conference proceedings, Saint, Malo; 1999. p. 153–62.

[5] McCarteny J, Hinds BK, Seow BL. The flattening of triangulated surfaces incorporating darts and gussets. Computer Aided Design 1999;31(4):249–60.

[6] Marcum DL, Gaiter JA. Unstructured surface grid generation using global mapping and physical space approximation. In: Eighth international meshing roundtable, South Lake Tahoe, CA; 1999. p. 397–406.

[7] Haker S, Angenent S, Tannenbaum A, Kikinis R, Sapiro G, Halle M. Conformal surface parametrization for texture mapping. IEEE Transactions on Visualization and Computer Graphics 2000;6(2):181–9.

[8] Eckstein L, Surazhsky V, Gotsman C. Texture mapping with hard constraints. Computer Graphics Forum 2001;20(3):95–104.

[9] Sheffer A, Sturler ED. Parametrization of faceted surfaces for meshing using angle-based flattening. Engineering with Computers 2001;17(3):326–37.

[10] Sander PV, Snyder J, Gortler S, Hoppe H. Texture mapping progressive meshes. In: Proceedings of ACM SIGGRAPH'01, Los Angeles CA; 2001. p. 409–16.

[11] Desbrun M, Meyer M, Alliez P. Intrinsic parametrizations of surfaces meshes. Computer Graphics Forum 2002;21(3):210–8.

[12] Levy B, Petiejean S, Ray N, Maillot J. Least squares conformal maps for automatic texture atlas generation. ACM Transactions on Graphics 2002;21(3):362–71.

[13] Levy B. Constrained texture mapping for polygonal meshes. In: Proceedings of SIGGRAPH'01, Los Angeles CA; 2001. p. 417–24.

[14] Tang Y, Wang J, Bao HJ, Peng QS. RBF-based constrained texture mapping. Computers and Graphics 2003;27(3):415–22.

[15] Kraevoy V, Sheffer A, Gotsman C. Matchmaker: Constructing constrained texture maps. ACM Transactions on Graphics 2003;22(3):326–33.

[16] Zuckerberger E, Tal A, Shlafman S. Polyhedral surface decomposition with applications. Computers and Graphics 2002;26(5):733–43.

[17] Zhang D, Hebert M. Harmonic maps and their applications in surface matching. In: IEEE proceedings of computer vision and pattern recognition, Colorado; 1999. p. 524–30.

[18] Cheng YP. Matrix theory. Xian, China: Northwestern Polytechnical University Press; 2001.