Title:	RBF-based Constrained Texture Mapping

Affiliations: State Key Lab of CAD&CG, Zhejiang University, China

Correspondence author: Hujun Bao
Address: State Key Lab of CAD&CG, Zhejiang University,
Hangzhou, 310027, China
Email address: bao@cad.zju.edu.cn
Phone No.: 086 571 87951045
Fax: 086 571 87951780

Abstract There are many methods based on optimization technique for resolving the problem of distortion-minimization in texture mapping. Recently, a new optimization-based method for parameterizing polygonal meshes with minimum deformation has been developed to specifically address the problem of feature matching in texture mapping. However, these optimization-based methods achieve the result of high quality at the expense of long computation time.

In this paper we present a fast analytic texture mapping method based on RBF (Radial Basis Function) interpolation to solve the problem of constrained texture mapping. The users control the mapping process by interactively defining and editing a set of constraints consisting of 3D points picked on the surface and the corresponding 2D points of the texture. RBF is invoked to interpolate the user-defined constraints to provide an analytic parameterization of the surface. The energy-minimization characteristic of RBF also ensures that the mapping function smoothly interpolates the constraints with satisfying non-deformation properties. This method has been applied to several data sets and excellent results have been produced. Our method is much faster than the optimization-based method in texture mapping with the same good effect achieved.

Keywords: Texture Mapping, Radial Basis Function, Optimization, Parameterization

RBF-based Constrained Texture Mapping

Ying Tang, Jin Wang, Hujun Bao*, Qunsheng Peng

State Key Lab of CAD&CG, Zhejiang University, China

(Figure 1 is here)

Figure 1: Using the algorithm presented in our paper, a cow mapped with the image of a tiger.

Abstract

There are many methods based on optimization technique for resolving the problem of distortion-minimization in texture mapping. Recently, a new optimization-based method for parameterizing polygonal meshes with minimum deformation has been developed to specifically address the problem of feature matching in texture mapping. However, these optimization-based methods achieve the result of high quality at the expense of long computation time.

In this paper we present a fast analytic texture mapping method based on RBF (Radial Basis Function) interpolation to solve the problem of constrained texture mapping. The users control the mapping process by interactively defining and editing a set of constraints consisting of 3D points picked on the surface and the corresponding 2D points of the texture. RBF is invoked to interpolate the user-defined constraints to provide an analytic parameterization of the surface. The energy-minimization characteristic of RBF also ensures that the mapping function smoothly interpolates the constraints with satisfying non-deformation properties. This method has been applied to several data sets and excellent results have been produced. Our method is much faster than the optimization-based method in texture mapping with the same good effect achieved.

Keywords: Texture Mapping, Radial Basis Function, Optimization, Parameterization

1 Introduction

Computer graphics professionals dream of reproducing the visual richness and complexity of the real world. Textures are very useful in this field since texture mapping technique replaces the nearly impossible work of modeling every detail at 3D geometric level. With texture mapping, we can not only describe a scene realistically, but also produce some exaggerated effects, such as putting an image of a tiger's head on a cow's head as shown in Figure 1.

Texture mapping puts each object in correspondence with a planar image by assigning a pair of coordinates (u, v) referring to a pixel of the planar image to each point of the object. The notion of parameterization provides a mathematical formalism for studying this problem [4][2]. A parameterization is a function putting the 3D surface to be textured in one-to-one correspondence with a subset of R^2, called the parameter space. When such a parameterization is associated with

*Correspondence author: Tel: 086 571 87951045; Fax: 086 571 87951780
Email Address: bao@cad.zju.edu.cn

a surface, it is possible to texture-map the surface by painting its parameter space with an image. It's obviously very simple for a parametric surface to be mapped for the parametric representation of the surface can be used to build the map. For instance, Catmull [7] has proposed to apply this technique to bi-cubic splines. These methods do not provide any means of controlling the nature and distribution of the deformations, so high distortion is very likely to occur if the mapped object has arbitrary geometry.

Different algorithms have been developed to minimize the distortions in texture mapping. The first attempts are to project the texture onto the surface using an intermediate surface with simple shape for which texture mapping is trivial [6][19]. Another idea is to consider that assigning texture coordinates to any surface is equivalent to flattening it [21][5]. Later many methods use optimization technique to minimize the defined energy function to get the mapping results with minimal distortions [12] [15][20][22] . An approximation of harmonic maps [14] has been used by Eck in [16] to parameterize triangulated surfaces previously decomposed into topological disks. Similarly, the notion of barycentric application [24] has been used in [17] for the same purpose. The problem of minimizing the distortions is addressed there by defining appropriate weights for the convex combinations involved in the barycentric map.

The different approaches mentioned above make it possible to map an image onto an object with satisfying non-distortion properties in a reasonable time. However, it is difficult to take into account of user-specified information to distribute the distortions with the above methods. In some applications, we need *constrained* texture mapping to satisfy our needs. Levy proposes a new global optimization method in [4] which enables the mapping to be customized. By tuning the perpendicularity and homogeneous spacing of iso-parametric curves all over the surface, this method specifies the surface zones where distortions should be minimized in order of preference. It is also possible to make the mapping respect a set of user specified iso-parametric curves. However, this method is not convenient for user to define the desired constraints. Furthermore, it can not be applied in general cases, such as feature matching, since it is not always possible to constrain the features with iso-parametric curves. For instance, it is impossible to map a closed curve (such as the border of an eye) to an iso-parametric curve.

The problem of matching features between the model and the texture has been addressed previously by Levy in [3]. Similar to the morphing technique based on feature constraints [1][23], he considers that the texture can be warped on the model according to user-specified constraints after parameterization. Actually, Levy introduces in [3] a new optimization-based method for parameterizing polygonal meshes with minimum deformations, while enabling the user to interactively define and edit a set of constraints. Each user-defined constraint consists of a relation linking a 3D point picked on the surface and a 2D point in the texture image. By respecting an arbitrary set of constrained features, this method first settles the problem of feature matching in texture mapping.

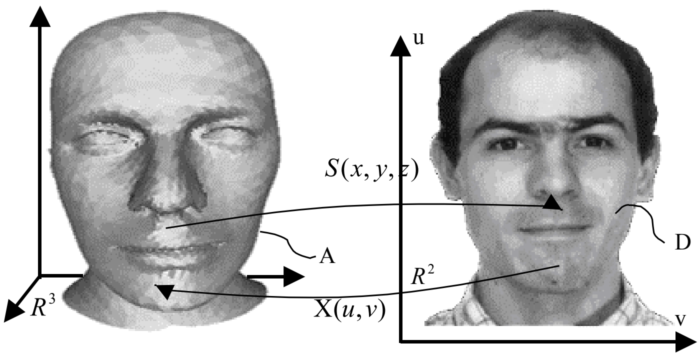
In this paper, we present a new simple method based on RBF (Radial Basis Function) interpolation technique to solve the problem of constrained texture mapping. In our method, the user also needs specifically assign the corresponding 2D points in the texture image to several picked 3D points on the surface. Then we interpolate these data sets as scattered data to get the texture attributes of all the other points on the surface. Since RBF is a popular interpolation tool for its well-known interpolation and energy-minimization properties, we employ it to construct the smooth parameterization of surfaces. Compared with the similar work that Levy [3] has done, our work has the same good qualities as the following:

1. An arbitrary set of constrained features can be respected in our method, whereas other methods can only constrain iso-parametric curves(e.g. [2][4]);
2. The RBF we selected are well suited to extrapolation and interpolation of irregular, non-uniformly sampled data, which means that it is not necessary to constrain the border.

Furthermore, our method has advantages over Levy's [3] method: We find the analytic solution to the mapping function for the polygonal surfaces and thus our mapping procedure is much faster than his. Usually our method requires several seconds to get the mapping results while his method requires several minutes. The reason lies in the different approaches we choose. We solve the linear system to directly get the analytic parameterization, which runs more quickly than the iteration and optimization procedure to approximate the final parameterization.

The paper is organized as follows. In section 2, we first introduce the notion of parameterization for an open surface and explain how to express the constrained parameterization problems in terms of the interpolation technique. In section 3, we detailedly describe the RBF we choose as the interpolant and how we use it in the constrained texture mapping. We show the implementation and additional mapping control that our method can achieve in section 4. Section 5 compares and analyzes the strength of our algorithm. Several examples are shown at the end of the paper.

2 Constrained Texture Mapping



(Figure 2 is here)

Figure 2: A mapping $S()$ puts the surface A of R^3 into one-to-one correspondence with a subset D of R^2. $X()$ is the inverse of $S()$ and is named the parameterization of A

At the beginning of this section, we review the definition of the texture mapping.

As shown in Figure 2, given an open surface *A* of R^3, we define mapping S to be a one-to-one transformation that maps the surface *A* to a subset *D* of R^2.

$$(x,y,z) \in A \rightarrow S(x,y,z) = \begin{bmatrix} u(x,y,z) \\ v(x,y,z) \end{bmatrix}$$

Regarding a mapping, the following definitions can be given:

- The set *D* is called the (u,v) parameter space;
- As S is defined as a one-to-one function, it has an inverse function $X = S^{-1}$ called a *parameterization* of the surface:

$$(u,v) \in D \rightarrow X(u,v) = S^{-1}(u,v) = \begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix}$$

In this paper, we consider the constrained texture mapping. In this case, the texture needs to be mapped on the surface according to the user-specified information. For instance, when considering a face, the user may want to specify the correspondence between the eyes on the model and the eyes on the texture. Each constraint is a pair of points (M_j, U_j) defined by the user; where $M_j \in R^3$ is a point belonging to the surface, and U_j is the corresponding texture coordinate of M_j.

From the above definitions, we know that the crucial work of texture mapping is to find a proper smooth parameterization function X . As for constrained texture mapping, the parameterization function should pass through a set of N data points M_j on the surface associated with the parameter-space points U_j, which means that $X(U_j) = M_j$. Similar to the warping technique in [18], we also use the interpolation theory to construct a mapping that is determined by a small number of points whose mapping is predetermined. Besides the interpolation condition, we also want to ensure a smooth transition between the constraints specified by the user after the parameterization, which implies that the energy of the parameterization should be minimized.

Based on the above requirements, we find that Radial Basis Function meets our needs quite well with its interpolation and energy-minimization properties. In the following section, we will describe the particular Radial Basis Function we choose in our implementation.

3 Radial Basis Function Interpolation

RBF(Radial Basis Function) is a very useful tool in the interpolation field and has wide application in many areas of computer graphics. Turk uses this function to combine the traditional two steps in shape transformation into a single one in [10]. He creates a transformation between two *N*-dimensional objects by casting this as a scattered data interpolation problem in *N+1* dimensions. RBF is also used in the reconstruction and representation of 3D objects in [13] by Carr. An object's surface is defined implicitly as the zero set of a RBF fitted to the given surface data. Fast methods for fitting and evaluating RBFs introduced in their paper allow the users to model large data sets, consisting of millions of surface points by a single RBF. Moreover, Arad [18] uses RBF in the application of facial expressions in image warping. With 2D images and a small number of anchor points which are similar to the constraints in our algorithm, the method of RBF provides a powerful mechanism for processing facial expressions. Our method presented here is like a 3D-2D extension to their method in 2D-2D spaces.

3.1 Radial Basis Function

Radial functions have proven to be an effective tool in multivariate interpolation problems of scattering data. Given the set of values $f = (f_1, f_2, \cdots, f_N)$ at the distinct points $M = \{M_1, M_2, \cdots, M_N\} \subset R^d$, we want to approximate the real valued function $f(M)$ by an interpolant $S(M)$, such that

$$S(M_i) = f_i, \qquad i = 1, \cdots, N \tag{1}$$

We choose $S(M)$ to be a *radial basis function* (RBF) of the form

$$S(M) = p(M) + \sum_{i=1}^{N} \lambda_i \Phi(|M - M_i|), \qquad M \in R^d \tag{2}$$

where p is a polynomial of low degree and the basic function Φ is a real valued function on $[0, \infty\}$, usually unbounded and of non-compact support [8]. λ_i is a real-valued weight, $|\cdot|$ denotes the Euclidean norm and $|M - M_i|$ is simply a distance which shows how far M is from M_i. Actually, a RBF is a weighted sum of translations of a radially symmetric basic function augmented by a polynomial term.

Popular choices for the basic function Φ include the thin-plate spline $\Phi(r) = r^2 \log(r)$ (for fitting smooth functions of two variables), the Guassian $\Phi(r) = \exp(-cr^2)$ (mainly for neural networks), and the multiquatric $\Phi(r) = \sqrt{r^2 + c^2}$ (for various applications, in particular fitting to topographical data). For fitting functions of three variables, good choices include the biharmonic

($\Phi(r) = r$) and triharmonic ($\Phi(r) = r^3$) splines. These polyharmonic splines (which include the thin-plate spline) minimize certain energy semi-norms and are therefore the ``smoothest'' interpolators.

RBFs are popular for interpolating scattered data as the associated system of linear equations is guaranteed to be invertible under very mild conditions on the locations of the data points. For example, the thin-plate spline only requires that the points are not co-linear while the Gaussian and multiquadric place no restrictions on the locations of the points. In particular, RBFs do not require that the data lie on any sort of regular grid.

3.2 RBF-based Texture Mapping

In our paper, the constrained texture mapping can be clearly stated as: Given a set of distinct nodes $M_i\{x_i, y_i, z_i\}_{i=1}^N \subset R^3$ and a set of corresponding 2D texture coordinates $U_i\{u_i, v_i\}_{i=1}^N \subset R^2$, find a mapping function $S(S_u, S_v)$, where

$$\begin{cases} S_u(x_i, y_i, z_i) = u_i \\ S_v(x_i, y_i, z_i) = v_i \end{cases} \quad i = 1, \cdots, N \tag{3}$$

The mapping function S is actually an interpolant. The number of the constraints is N. In our case, it is easier to characterize the mapping function rather than the parameterization X. The computation procedures of S_u and S_v are the same and parallel, so in the following paragraph, we just use S to represent both of them. Note that we use the notation $M = (x, y, z)$ for points $M \in R^3$.

We will choose the interpolant S from $BL^{(2)}(R^3)$, the Beppo-Levi space of distribution on R^3 with square integrabel second derivatives. The space $BL^{(2)}(R^3)$ is equipped with the rotation invariant semi-norm defined by

$$\begin{aligned} \|S\|^2 = \int_{R^3} &\left(\frac{\partial^2 S(M)}{\partial x^2}\right)^2 + \left(\frac{\partial^2 S(M)}{\partial y^2}\right)^2 + \left(\frac{\partial^2 S(M)}{\partial z^2}\right)^2 \\ &+ 2\left(\frac{\partial^2 S(M)}{\partial x \partial y}\right)^2 + 2\left(\frac{\partial^2 S(M)}{\partial y \partial z}\right)^2 + 2\left(\frac{\partial^2 S(M)}{\partial x \partial z}\right)^2 dM \end{aligned} \tag{4}$$

This semi-norm is a measure of the energy or "smoothness" of function: functions with small semi-norm are smoother than those with large semi-norm. Duchon [9] showed that the smoothest interpolation function, which has the minimum semi-norm, has the simple form

$$S^*(M) = p(M) + \sum_{i=1}^{N} \lambda_i |M - M_i| \tag{5}$$

where p is a linear polynomial, which has the form $p(M) = c_1 + c_2 x + c_3 y + c_4 z$. The coefficients λ_i are real numbers and $|\cdot|$ is the Euclidean norm on R^3. This function is a particular example of a RBF. Since triharmonic splines is also the good choice for fitting functions of three variables, we replace biharmonic splines with triharmonic splines in Equation (5) and our interpolant has the following form

$$S^*(M) = c_1 + c_2 x + c_3 y + c_4 z + \sum_{i=1}^{N} \lambda_i |M - M_i|^3 \tag{6}$$

An arbitrary choice of coefficients λ_i in Equation (6) will yield a function S^* that is not a member of $BL^{(2)}(R^3)$. The requirement that $S^* \in BL^{(2)}(R^3)$ implies the orthogonality or side conditions

$$\sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} \lambda_i x_i = \sum_{i=1}^{N} \lambda_i y_i = \sum_{i=1}^{N} \lambda_i z_i = 0 \tag{7}$$

These side conditions along with the interpolation conditions of Equation (3) lead to a linear system to solve the coefficients that specify the RBF. The linear system has the following form

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1N} & 1 & x_1 & y_1 & z_1 \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2N} & 1 & x_2 & y_2 & z_2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi_{N1} & \Phi_{N2} & \cdots & \Phi_{NN} & 1 & x_N & y_N & z_N \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ x_1 & x_2 & \cdots & x_N & 0 & 0 & 0 & 0 \\ y_1 & y_2 & \cdots & y_N & 0 & 0 & 0 & 0 \\ z_1 & z_2 & \cdots & z_N & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

where $\Phi_{i,j} = \left|M_i - M_j\right|^3, \quad i, \mathrm{j} = 1, \cdots, \mathrm{N}$,

Solving the linear system determines $\lambda_i (i = 1, \cdots, N)$ and $c_i (i = 1, \cdots, 4)$, and hence $S^*(M)$. Note the above linear system only resolves the mapping function S_u. Substituting $u_i (i = 1, \cdots, N)$ with $v_i (i = 1, \cdots, N)$ will give the linear system that determines S_v. Finally, S_u and S_v have the following form

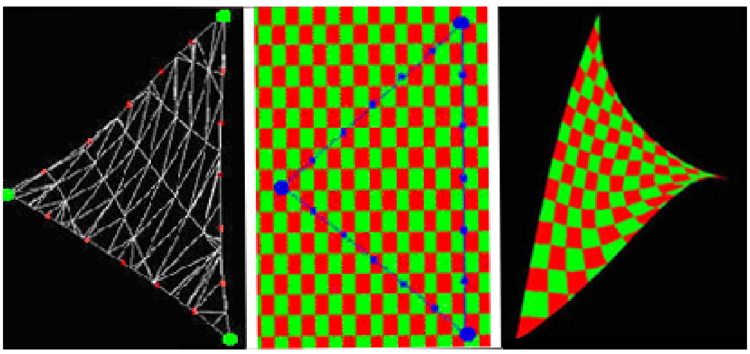
$$\begin{cases} S_u(M) = \sum_{i=1}^{N} \lambda_{iu} \left|M - M_i\right|^3 + c_{1u} + c_{2u}x + c_{3u}y + c_{4u}z \\ S_v(M) = \sum_{i=1}^{N} \lambda_{iv} \left|M - M_i\right|^3 + c_{1v} + c_{2v}x + c_{3v}y + c_{4v}z \end{cases} \tag{9}$$

As soon as S_u and S_v are decided, we get the mapping of each 3D point M of the surface by putting the coordinates of M in Equation (9) to get the according texture coordinates.

In our method, the number of the predetermined constrained points N is relatively small, usually between fifteen and thirty, so the size of the linear system is small enough for us to use the standard numerical method to solve it [11].

We can see that the interpolation nature of RBF well satisfies our requirements of constrained texture mapping, and the energy minimization properties ensure that the mapping is smooth and continuous. Furthermore, we find that the non-compactly supported basic functions are better suited to extrapolation and interpolation of irregular, non-uniform sampled data, which means that we do not need to require the border of the surface to be fixed which can cause deformations.

4 Implementation and Additional Control of Texture Mapping process

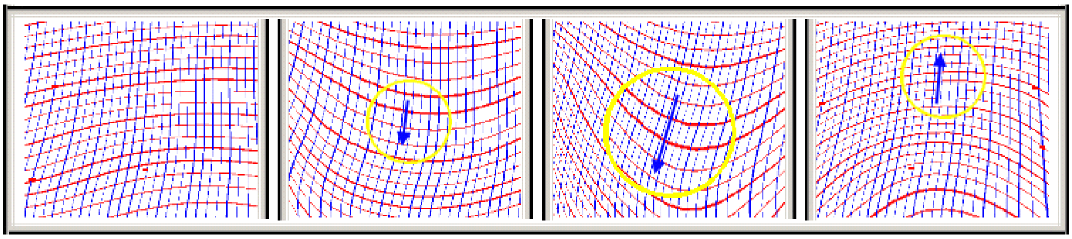


(Figure 3 is here)

Figure 3: (a) The equidistantly sampled points at the borders of the patch in 3D space. (b) The corresponding sampled points in texture space.

In the application of this method, the user interactively selects the feature points in 3D space and their corresponding points in texture space separately. Such user-defined constraints form the data sets to be interpolated. After the user defines a set of constraints, RBF is employed to give us the parameterization result. With each point of polygonal model having been assigned a corresponding texture coordinates after parameterization, we use the support of OpenGL to draw the last mapped result.

Besides the feature-based constraints such as in Levy [3], we can also constrain the border of model that is to be textured in some applications. Here the data sets to be interpolated are the points on the border of the model instead of those located at the features of the model. In this case, the algorithm samples equidistantly the boundaries of the surface to get the 3D points. Meanwhile, we sample equidistantly the 2D points in texture space accordingly. The number of sample points is determined by the length of the border. We show in Figure 3 the sampled points in the object space and texture space separately. The green points in the object space are the corners which define the borders and the red points are the equidistantly sampled points along the border. In Figure 3(b), we see their corresponding points in the texture space. Such pairs of sampled points form the border constraints and we can use RBF to construct the interpolation parameterization. The mapped result is shown in Figure 3(c). If needed, the feature-based constraints and border constraints can be combined to generate satisfying results. For example, when we get the mapping results with only the boundary constraints, we may find that the features does not match well. We can solve it by adding the feature-based constraints within the boundary.



(Figure 4 is here)

Figure 4: Editing the gradients of a parameterizaton. (a): Initial configuration; (b): Changing the direction of the gradients; (c): Changing the magnitude of the gradients; (d): Inverse the direction of the gradients

Controlling the gradients of the parameterization provides an additional way to interact with the texture mapping (See Figure 4). The gradient control, actually the line-based control, can be transformed into point control in our method. Through adding points in the desired gradient direction, we can successfully achieving the effect of directly changing gradients. Figure 4(a) shows the initial configuration of the parameterization. In Figure 4(b), we add a vector by appending a point at the arrowhead and the gradients of the parameterization change in the appointed direction. We change the magnitude of the added vector in Figure 4(c) and the gradients change more. As we inverse the vector's direction in Figure 4(d), the direction of the gradients changes accordingly.

5 Results and Discussion

We implement our algorithm on polygonal models using a 600 Mhz Pentium III PC. The mapping results are shown at the end of this paper and we have indicated explicitly each one's time elapse.

In our method, most of the computation time is devoted to solving the linear system. As we have stated before, the size of the linear system is the number of the constraints plus four, which means that the computation time is determined by the number of the constraints. As for the model of face, we usually specify 15~25 constraints, and for the model of cow's head, we specify 30 points.

Compared with Levy's constrained texture mapping based on optimization [3], our method is much faster. For example, he used five minutes to map a gelada image onto the face model while we only use 7.9 seconds to map it to our head model. Usually he needs several minutes in mapping whereas we just need several seconds. The reason for this is that he minimizes the criterion to get the final result by iteration which costs much more time than our analytic parameterization solution using BRF interpolation.

We also notice that the number of the selected feature points needs not be large, just some necessary points can achieve good effect. The reason is that the mapping function we use is continuous in second derivatives and the mapping results are very smooth. So this ensures that the size of the linear system will not be large. We also provide the users with the ability to control the directions and magnitudes of the gradients of the parameterization, as shown in Figure 4. This supplies an additional way to interact with the texture mapping process.

6 Conclusions

The main contribution of this paper is that we find a simple fast solution to constrained texture mapping which put the details of the texture in correspondence with features of the models. The users can interactively define and edit a set of constraints which are links between 3D points of the model and pixels in texture space. RBF, a powerful tool in multivariate interpolation problems of scattered data, is employed to interpolate those user-defined constraints to give the analytic mapping function. The energy-minimization property of RBF also ensures that our constrained texture mapping is smooth and continuous with minimum deformation.

Our algorithm has a great potential application in visual simulation and surface modeling. For example, our approach can be extended to 3D-3D spaces to act as a 3D morphing tool by interpolating the corresponding 3D points on source and destination models. In the future work, we will incorporate this method in the animation simulation system, freeing the artist from the technicalities encountered in existing texture package.

7 Acknowledgement

The project is partly sponsored by National Natural Science Foundations of China for Distinguished Young Scholars (No.69925204) and for Innovative Research Groups (No. 60021201)

References

[1] A.Tal, G.Elber. Image Morphing with Feature Preserving Texture. *Proc. of Eurographics'99*, 18(3):339-348, 1999.

[2] A.W.F.Lee, W.Sweldens, P.Schroder, L.Cowsar, and D.Dobkin. Maps: Multiresolution adaptive parameterization of surfaces. *Proc.of SIGGRAPH'98*, 95-104, 1998

[3] B.Levy. Constrained Texture Mapping for Polygonal Meshes. *Proc.of SIGGRAPH'2001*, 417-424, 2001

[4] B.Levy, J.L.Mallet. Non-Distortion Texture Mapping for Sheared Triangulated Meshes. *Proc.of SIGGRAPH'98*, 343-352, 1998.

[5] C.Bennis, J.M.Vezien, G.Iglesias. Piecewise Flattening for Non-distorted Texture Mapping. *Computer Graphics*, 25(4):237-246, 1991.

[6] E.Bier, K.Sloan. Two-part Texture Mapping. *IEEE Computer Graphics and Applications*, 6(9):40-53, 1986.

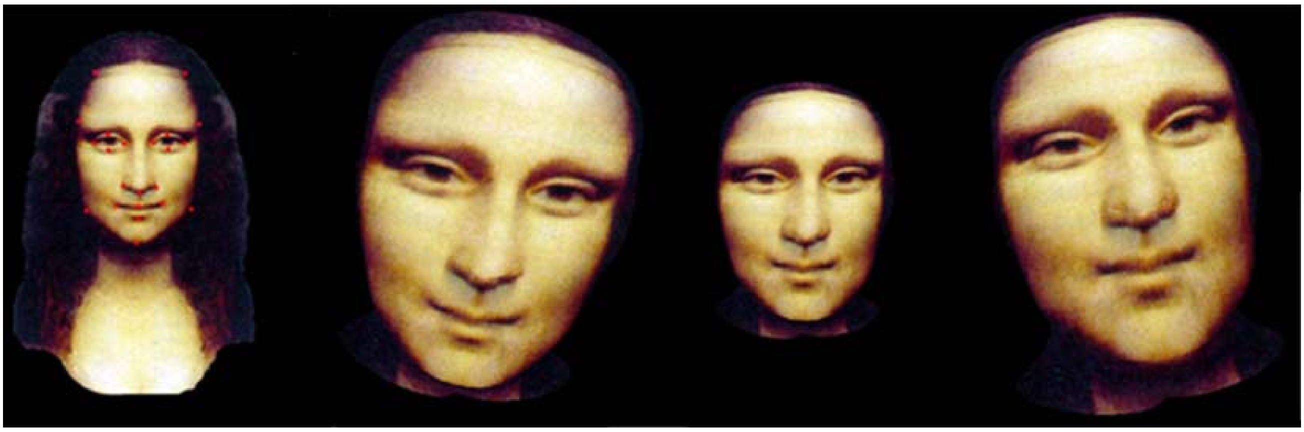
[7] E.Catmull. A Subdivision Algorithm for Computer Display of Curved Surfaces. *PhD thesis*, Dept. of Computer Science of Utah, Dec., 1974.

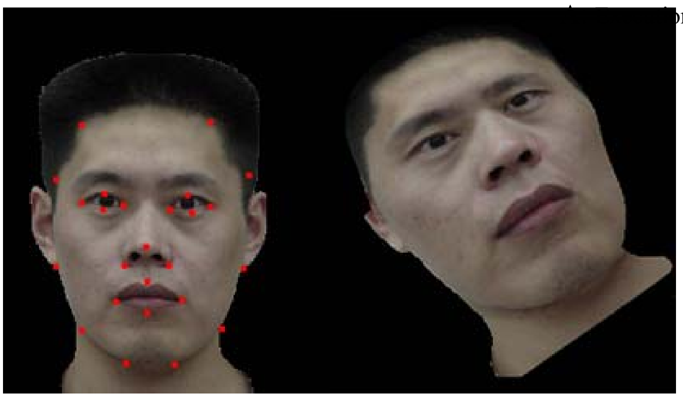
[8] E.W.Cheney, W.A.Light. *A Course in Approximation Theory*. Brooks Cole, Pacific Grove, 1999

[9] J.Duchon. Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces," in *Constructive Theory of Functions of Several Variables*, Lecture Notes in Mathematics, edited by A.Dolb and B.Eckman, Springer-Verlag, 1977, pp. 85-100.

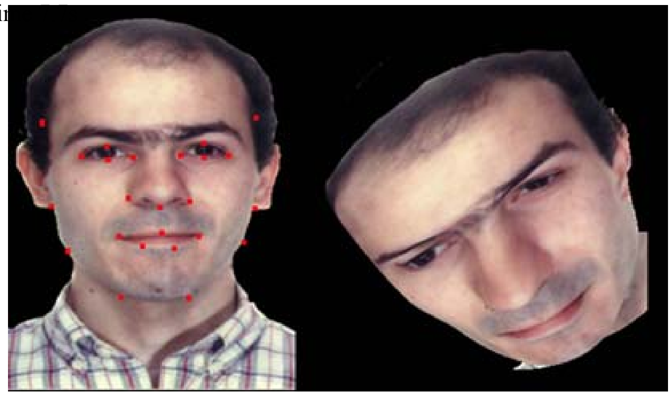
[10] G.Turk, J.F.O'Brien. Shape Transformation Using Variational Implicit Functions. *Proc.of SIGGRAPH'99*,

335-342, 1999
[11] Guan Zhi, Chang Jingliang. *Numerical Calculation*, Tsinghua University Press, 1990.
[12] H.K.Pedersen. Decorating Implicit Surfaces. *Proc.of SIGGRAPH'95*, 291-300, 1995.
[13] J.C.Carr, R.K.Beatson, J.B.Cherrie, T.J.Mitchell, W.T.Fright, B.C.McCallum, T.R.Evans. Reconstruction and Representation of 3D Objects with Radial Basis Functions. *Proc.of SIGGRAPH'200*1, 67-76, 2001
[14] J. Eells and J.H.Sampson. Harmonic mapping of Riemannian manifolds. *Amer. J. Math.*, 86:109-160, 1964
[15] J.Maillot, H.Yahia, A.Verroust. Interactive Texture Mapping. *Computer Graphics*, 27(4):27-34, 1993.
[16] M.Eck, T.DeRose, T.Duchamp, H.Hoppe, M.Lounsbery, and W.Stuetzle. Multiresolution Analysis of Arbitrary Meshes. *Proc.of SIGGRAPH'95*, 173-182, 1995.
[17] M.S.Floater, Parameterization and Smooth Interpolation in Geometric Modeling. *ACM Trans. Comput. Graphics*, 8(2):121-144, 1997
[18] N.Arad, N.Dyn, D.Reifeld and Y.Yeshurun. Image Warping by Radial Basis Functions: Application to Facial Expressions. *CGVIP*, 56(2), 1994
[19] Peachy, R.Darwyn. Solid Texturing of Complex Surfaces. In *SIGGRAPH 85 Conference Proceedings*, volume 19, pages 287-296. ACM, July 1985.
[20] P.Litwinowicz, G.Miller, Efficient Technique for Interactive Texture Placement. *Proc.of SIGGRAPH'94*, 119-122, 1994
[21] Samek, Marcel, C.Slean, and H.Weghorst. Texture Mapping and Distortions in Digital Graphics. *The Visual Computer*, 2(5):313-320, September 1986
[22] S.D.Ma, H.Lin, Optimal Texture Mapping. *Proc. of Eurographics'88*, 421-428, 1988
[23] S.Y.Lee, K.Y.Chwa, S.Y.Shin. Image Metamorphosis Using Snakes and Free-Form Deformations. *Proc. of SIGGRAPH'95*, 439-447, 1995
[24] W.T.Tutte. Convex Representation of Graphs. In *Proc. London Math. Soc.*, Volume 10, 1960





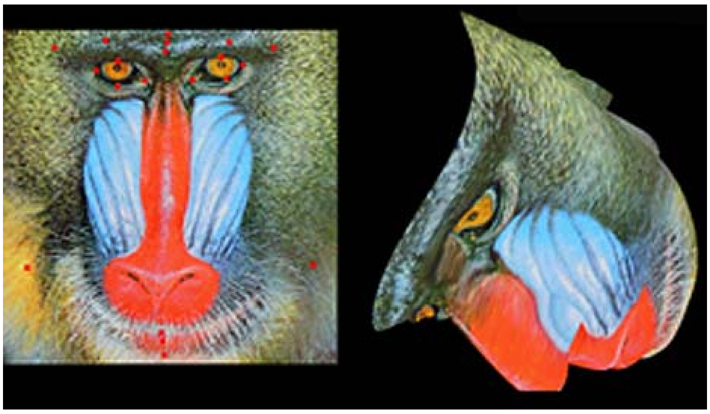
B: Execution time 8.2s



C: Execution time 8.1s

D: Execution time 8.2s

E: Execution time 8.8s



(Figure 5 is here)

F: Execution time 7.9s

Figure 5: Mapping different image sets on the face data set. Each execution time is indicated above. As we can see, the mapping results are smooth and with high precision.