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

Reconstruction from contour lines based on bi-cubic Bézier spline surface*

LI Zhong[†1,2], MA Li-zhuang[2], TAN Wu-zheng[2], ZHAO Ming-xi[2]

([1]*Department of Mathematics and Science, Zhejiang Sci-Tech University, Hangzhou 310018, China*)

([2]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*)

[†]E-mail: lizhongzju@hotmail.com

Received Apr. 6, 2006; revision accepted Apr. 19, 2006

Abstract: A novel reconstruction method from contours lines is provided. First, we use a simple method to get rid of redundant points on every contour, then we interpolate them by using cubic Bézier spline curve. For corresponding points of different contours, we interpolate them by the cubic Bézier spline curve too, so the whole surface can be reconstructed by the bi-cubic Bézier spline surface. The reconstructed surface is smooth because every Bézier surface is patched with G^2 continuity, the reconstruction speed is fast because we can use the forward elimination and backward substitution method to solve the system of tridiagonal equations. We give some reconstruction examples at the end of this paper. Experiments showed that our method is applicable and effective.

Key words: Contour, Surface reconstruction, Bi-cubic Bézier surface, G^2 continuity

doi:10.1631/jzus.2006.A1241 **Document code:** A **CLC number:** TP391

INTRODUCTION

In many research and application areas such as medical science, biomedical engineering, and CAD/CAM, an object is often known by a sequence of 2D cross-sections. With improvements in data acquisition and imaging techniques such as computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound imaging, the cross-sectional images of the object of interest can be obtained with ease. The object in every 2D cross-section can be represented by the contour line. How to reconstruct 3D surface from these contour lines is one of the main researches in Computer Graphics and CAD. This problem has aroused extensive concern (Park and Kim, 1996).

There are some reconstruction researches on contours. Early reconstruction algorithms appeared in the seventies of the 20th century. Currently, popular methods are mesh-based reconstruction (Keppel, 1975; Choi and Park, 1994; Park and Kim, 1995). For the mesh reconstruction algorithm, there are many data points waiting to be processed on different contours, and for the reconstruction of more complex surface of object, these algorithms sometimes are not effective. So many researchers used other reconstruction algorithms, for example, the implicit function method (Jaillet *et al.*, 1997), the particle system method (Bajscy and Solin, 1987), the superquadrics method (Muraki, 1991), etc. Another popular reconstruction algorithm is based on B spline surface, NURBS surface (Bernhard, 1993). These methods can reconstruct the surface smoothly, but in order to get knot vectors, they require the iteration method to solve the large scale system of equations.

In this paper, we present a new reconstruction method from contours based on the bi-cubic Bézier spline surface. First, we use a simple method to get rid

* Project supported by the National Natural Science Foundation of China (Nos. 60373070 and 60573147), Postdoctor Foundation of Shanghai (No. 05R214129), and Zhejiang Education Foundation of China (No. 20050786)

of redundant points on every contour, then we interpolate them by using the cubic Bézier spline curve. For corresponding points in different contours, we interpolate them by the cubic Bézier spline curve, the whole surface can be reconstructed from the bi-cubic Bézier spline surface. The advantage of this method is that the reconstructed surface is smooth because every bi-cubic Bézier surface is patched with G^2 continuity. In the computation process, we can use the forward elimination and backward substitution method to solve the system of tridiagonal equations, the reconstruction speed is fast.

PRELIMINARIES

Cubic Bézier spline curve

There is the form of cubic Bézier spline curve (Farin, 1997). Here, assume that initial control points are d_i (i=0,1,…,n), we construct a cubic Bézier spline curve to fit the control polygon composed of d (i=0,1,…,n). The spline curve should pass through two endpoints of d_0, d_n and it is G^2 continuity inside the curve. Assume that control points of every cubic Bézier curve are B_i (i=0,1,…,3n), which are related on parameter u in [0,1], so they can be rewritten as $B_i(u)$. The first and the last control point are $B_0(u)=d_0$, $B_{3n}(u)=d_n$. Other control points $B_{3i+1}(u)$, $B_{3i+2}(u)$ are in the line segment d_id_{i+1} and satisfy with

$$B_{3i+1}(u)=\left(1-\frac{u}{2}\right)d_i+\frac{u}{2}d_{i+1}, \tag{1}$$

$$B_{3i+2}(u)=\frac{u}{2}d_i+\left(1-\frac{u}{2}\right)d_{i+1}, \tag{2}$$

where i=0, 1, …, n−1.

Two cubic Bézier curves in the immediate neighborhood are G^2 continuity at $B_{3i}(u)$ (i=1, 2,…,n−1), by the computation, we find

$$B_{3i}(u)=\frac{u}{4}d_{i-1}+\left(1-\frac{u}{2}\right)d_i+\frac{u}{4}d_{i+1}. \tag{3}$$

The shape of cubic Bézier spline curve is decided by control points, namely, by parameter u in [0,1]. We can set different u to construct different spline curve with G^2 continuity, initially, u=1/2.

For given u and initial control points d_i, we can get all control points of every cubic Bézier spline curve by Eqs.(1)~(3). Whereas, when we knew parameter u and all interpolate points B_{3i}, we can use the system of equations from Eq.(3) to get all initial control points d_i. We notice that this system of equations is a system of tridiagonal equations in d_i, it can be solved by the forward elimination and backward substitution method rapidly. Then we use Eqs.(1) and (2) to calculate other control points B_{3i+1}, B_{3i+2} of every cubic Bézier spline curve. Here, this process is called the control point decision algorithm for cubic Bézier spline curve.

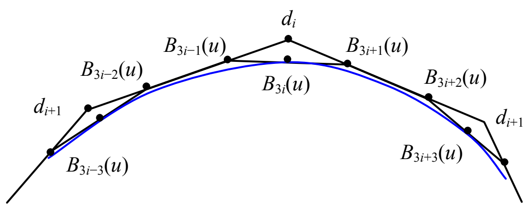

Fig.1 Cubic Bézier spline curve

Bi-cubic Bézier spline surface

The bi-cubic Bézier spline surface is composed of a number of bi-cubic Bézier surfaces which are patched with G^2 continuity. In 3D space, for initial control points d_{ij} (i=1,2,…,m, j=1,2,…,n), we can use m×n bi-cubic Bézier surfaces to generate a bi-cubic Bézier spline surface which interpolates given control points. The bi-cubic Bézier spline surface can be understood as the combination of a series of cubic Bézier spline curve in u direction and a series of cubic Bézier spline curve in v direction. Since these spline curves is G^2 continuity in u, v direction, the bi-cubic Bézier spline surface is smooth for interpolating given points. In Fig.2, there is a smooth bi-cubic Bézier spline surface composed of 4×2 bi-cubic Bézier surfaces for interpolate 8 points. To construct the bi-cubic Bézier spline surface to interpolate given control points, we can use above control point decision algorithm for every cubic Bézier spline curve in u, v direction. And we describe our reconstruction method from contours in the following section in detail.

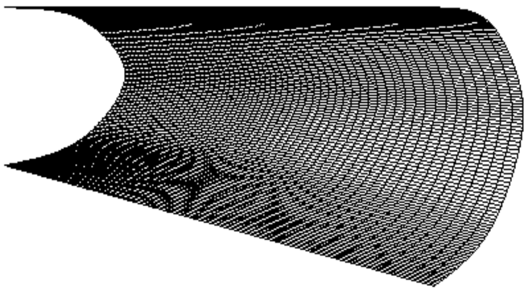

Fig.2 The spline surface composed of 8 bi-cubic Bézier surfaces

RECONSTRUCTION ALGORITHM BASED ON BI-CUBIC BÉZIER SPLINE SURFACE

Optimization for every contour line

When using CT scanner or other scanner techniques to obtain a series of contours, there are too many redundant points. We should re-sample data points on every contour to get rid of redundant points. These re-sampled points should keep the shape of every contour. A popular method is based on uniform sampling. This method is simple and feasible, but it may lose some key points which influence the object shape. Or we use the curvature-based sampling method, but it requires computation of the derivative of order 2 (Chang *et al*., 1991). Here, we give a simple re-sampling method based on angle match.

First, we use line segment to connect all immediate neighboring points in one contour, two neighboring line segments compose an angle α_i ($<\pi/2$) at point d_i. We set a threshold α in advance. When $\alpha \geq \alpha_i$, d_i, d_{i+1}, d_{i+2} can be understood to be in the same line segment closely, so d_{i+1} can be omitted, see Fig.3. The advantage of this method is that we use two endpoints to represent the line segment, other points in the line segment can be omitted. After the above angle match operations for all α_i, we can get a number of re-sampled points which represent well every contour.

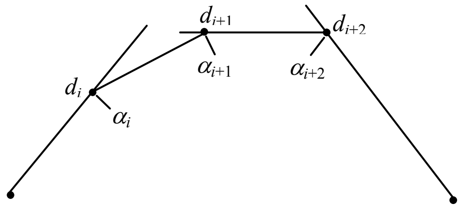

Fig.3 Re-sampling points based on the angle match

Interpolating every contour by cubic Bézier spline curve

We can use the cubic Bézier spline curve to interpolate re-sampled points of every contour. These spline cuvres can be understood as isoparametric curves of bi-cubic Bézier spline surface in u direction. We can use the above control point decision algorithm to get control points of cubic Bézier spline curve. The shape of cubic Bézier spline curve is decided by parameter u. We might as well set $u=1/2$ initially, then adjust value u according to the reconstructed shape.

Interpolating corresponding points in different contours by cubic Bézier spline curve

After interpolating every contour by the cubic Bézier spline curve, we also use it to interpolate corresponding points in different contours. These spline curves can be understood as the isoparametric curves of bi-cubic Bézier spline surface in v direction. At first, we should match all re-sampled points in different contours.

We notice that the number of re-sampled points in different contours may be not the same, so these points cannot be matched directly. We should let the number of re-sampled points in different contours be the same. Assume that the largest number of re-sampled points in one contour is M, in order to reconstruct the original object more accurately, we can complement some points to M for other contours. Here, we give the parameter-based match method. It can make the number of re-sampled points in different contours be the same and all re-sampled points in different contours can be matched one by one correspondence.

Assume that the kth contour has the largest number of re-sampled points, which are M_i (i=1,…, M). In the neighboring (k+1)th contour, the number of re-sampled points is N, these points are N_i (i=1,…,N), $M>N$, see Fig.4.

First, we can find a pair of matching points on two contours by the shortest distance rule. We suppose two matching points are M_1 and N_1. Then, we expand two polygon contours from M_1 and N_1 to two line segments l_1, l_2 in the same direction ($M_{M+1}=M_1$, $N_{N+1}=N_1$), see Fig.5. Inner re-sampled points on two line segments can be represented by the following parameter equation:

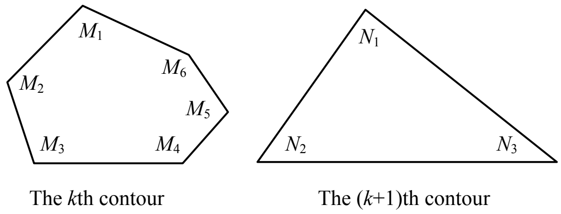

Fig.4 Two contours with different number of re-sampled points (M=6, N=3)

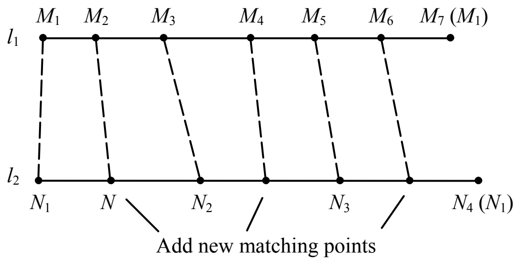

Fig.5 Expand two contours with re-sampled points to two line segments

$$M_i=(1-t_i)M_1+t_iM_7,\ N_j=(1-s_j)N_1+s_jN_4,$$

where $t_i\in(0,1)$, $s_j\in(0,1)$, i=2, …, M, j=2, …, N.

So we can obtain two sequences T and S in increasing t_i and s_j

$$T=\{t_2,\ \ldots,\ t_M\},\ S=\{s_2,\ \ldots,\ s_N\}.$$

Obviously, when we add some elements in S to make sure all t_i and s_i match one by one correspondingly, then map these elements to get corresponding points in the original contours, we can solve the point match problem.

Since $N\leq M$, for different s_j, we can find the different t_i which is closest to s_j as the match element of s_j, so all elements s_j in S can be matched by t_i in T at first. Then there are $M-N$ elements in T waiting to be matched, we should add some desirable elements in S to match these waiting elements. These added elements must be between two neighboring matched elements in S. For example, assume that unmatched point M_2 is between matched point M_1, M_3 in l_1, we know that matching points of M_1, M_3 are N_1, N_2. We can use the parameter equation $M_2=(1-t)M_1+tM_3$ to get the parameter t, then substitute t in the parameter equation $N=(1-t)N_1+tN_2$, to get the new point N in l_2, then map this point to the original polygon contour to get the matching point. After the same operations, all $M-N$ waiting for matching points in kth contour can be matched by new added points in $(k+1)$th contour. By this parameter-based match method, the number of re-sampled points in different contours can be the same and these points in different contours can be matched by one by one correspondence.

Considering special cases, if there are two re-sampled points in the $(k+1)$th contour, we can add $(M-2)$ datapoints with uniform sample in the line segment. If there is only one re-sampled point in the $(k+1)$th contour, this point can be understood as M data points with the same position.

Reconstruction from contours based on bi-cubic Bézier spline surface

As re-sampled points of one contour can be interpolated by the isoparametric curve of bi-cubic Bézier spline surface in u direction, and all corresponding points in different contours can be interpolated by the isoparametric curve of bi-cubic Bézier spline surface in v direction, so all given points can be interpolated by the bi-cubic Bézier spline surface.

Here, we should guarantee that the patch point of cubic Bézier spline curve in u, v direction must be in the same position. For example, in Fig.6a, when we get three cubic Bézier spline curves of the bi-cubic Bézier spline surface in u direction, these curves are G^2 continuity at $B_{i,3j}$, $B_{i+1,3j}$, $B_{i+2,3j}$ respectively. In order to construct three cubic Bézier spline curves with G^2 continuity in v direction that also pass through $B_{i,3j}$, $B_{i+1,3j}$, $B_{i+2,3j}$, we should find corresponding control points $d_{i,j}$, $d_{i+1,j}$, $d_{i+2,j}$ of cubic Bézier spline curve in v direction, see Fig.6b. This problem can also be solved by the above control point decision algorithm of cubic Bézier spline curve. So we can construct

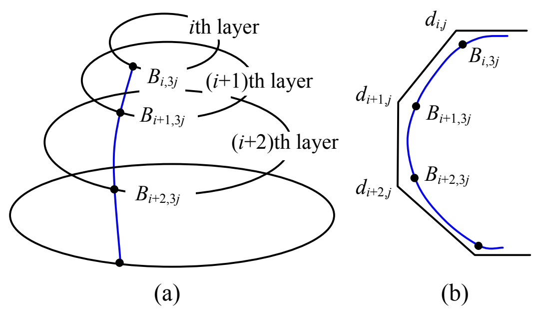

Fig.6 Cubic Bézier spline curve in u (a), v (b) direction which pass through the same positions

cubic Bézier spline curve which passing through $B_{i,3j}$, $B_{i+1,3j}$, $B_{i+2,3j}$ in u, v direction simultaneously.

Special case treatment

For fitting re-sampled points in every contour, the constructed cubic Bézier spline curve may be a closed curve. If we want to get the closed spline curve with G^2 continuity, we need to make some modification. For example, Fig.7a is a polygon composed of $d_0d_1d_2d_3d_4d_5d_6$. If we directly use the above method to fit the polygon from d_0 to d_6, although the curvature at $d_0(d_0)$ in the spline curve is the same (the value is zero), the direction of two tangent vectors is not the same, so the spline curve is not G^2 continuity at this point. In order to solve this problem, we can let d_0' be the midpoint of d_0d_1 and d_6' be the midpoint of d_5d_6. So the original polygon can be divided into two sub-polygons composed of $d_0'd_1d_2d_3d_4d_5d_6'$ and $d_0'd_0d_6'$. Both sub-polygons can be fitted by two cubic Bézier spline curves which are patched at the point of d_0', d_6'. At this time, because the directions of two tangent vectors are the same and the curvature is the same at d_0', d_6', so the closed spline curve is G^2 continuity, see Fig.7b.

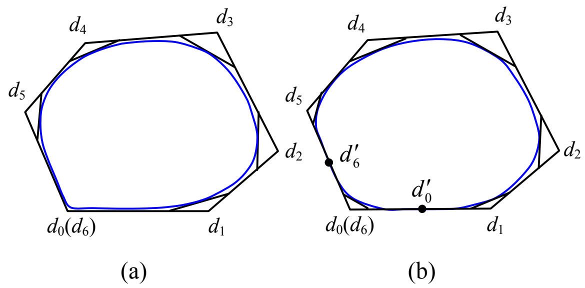

Fig.7 Closed cubic Bézier spline curve with G^2 continuity

EXPERIMENT RESULTS

We use VC6.0 language and OpenGL Graphics tool to realize the reconstruction algorithm from contours. Here, the computer configuration is CPU P4/1.8 G with EMS memory of 256 MB. Fig.8 is the cricoid object reconstruction example (after lighting treatment). There are 20 levels of contour lines and about 1904 data points for the original object, we set $u=v=0.4$ to reconstruct the object. Fig.9 is a tooth reconstruction example. The tooth is partitioned from the whole teeth model. The tooth has 35 levels of contour lines and about 3081 data points. It is reconstructed by the bi-cubic Bézier spline surface. Here, we set $u=v=0.5$. Experiment results showed that the reconstructed object is smooth with G^2 continuity.

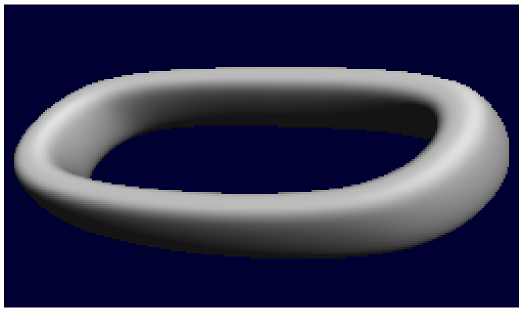

Fig.8 Cricoid object reconstruction

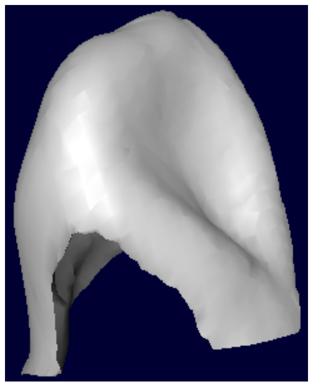

Fig.9 Tooth reconstruction from the teeth model

We compared some different reconstruction algorithms from the contours. Table 1 presents time comparison using different reconstruction methods. Compared to popular triangle mesh methods, these methods can deal with many triangle meshes although the reconstructed surface is not smooth, while our reconstruct algorithm can obtain surface with G^2 continuity. Compared to the implicit function reconstruction algorithm, our method uses popular bi-cubic Bézier surfaces to reconstruct and realize the algorithm easily. Compared to B spline surface and NURBS surface algorithms, in order to get knot vectors, these methods need to solve a large scale system of equations by the iteration method, while our algorithm can use the forward elimination and backward substitution method to solve the system of tridiagonal equations.

Table 1 Time comparison of different reconstruction methods for cricoid object and tooth

Methods	Reconstruction time (s)	
	Cricoid object	One tooth
Mesh algorithm	1.06	2.57
Implicit function method	1.73	2.62
B spline method	0.75	2.46
Our method	0.54	1.31

CONCLUSION AND FUTURE WORK

In this paper, we present a new reconstruction method from contours. The method uses bi-cubic Bézier spline surface to reconstruct surfaces which can be realized easily, the reconstructed surface is smooth with G^2 continuity. We can use the forward elimination and backward substitution method to solve the system of tridiagonal equations, the reconstruction is fast.

For some complex objects, if we directly use bi-cubic Bézier spline surface to reconstruct them, the reconstruction process may be somewhat complicated. We can divide the object into some parts, every part of which can be reconstructed by using the bi-cubic Bézier spline surface. Because two neighboring spline surfaces are patched with the same tangent direction and the same curvature (the value is zero), the whole reconstructed object is smooth with G^2 continuity. This new method can be applied to object reconstruction from contours in the modelling fields of Medicine, Biology, Chemistry, etc.

References

Bajscy, R., Solin, R., 1987. Three-dimensional Object Representation Revisited. Proceedings of the IEEE Conference on Computer Vision, p.231-240.

Bernhard, G., 1993. Three-dimensional Modeling of Human Organs and its Application to Diagnosis and Surgical Planning. Sophia Antipolis, Inria.

Chang, L., Chen, H., Ho, J., 1991. Reconstruction of 3D medical images: a nonlinear interpolation technique for reconstruction of 3D medical images. *CVGIP: Graphical Model and Image Processing*, **53**(4):382-391. [doi:10.1016/1049-9652(91)90041-H]

Choi, Y.K., Park, K.H., 1994. A heuristic triangulation algorithm for multiple planar contours using an extended double branching procedure. *The Visual Computer*, **10**(7):372-387. [doi:10.1007/BF01900663]

Farin, G., 1997. Curves and Surfaces for Computer-aided Geometric Design: A Practical Guide, 4th Ed. Academic Press, San Diego.

Jaillet, F., Shariat, B., Vandorpe, D., 1997. Deformable Volume Object Modeling with a Partical-based System for Medical Applications. Proceedinggs of the WSCG'97, p.192-201.

Keppel, E., 1975. Approximating complex surface by triangulation of contour lines. *IBM Journal of Research and Development*, **19**:2-11.

Muraki, S., 1991. Volumetric shape description of range data using 'blobby Model'. *Computer Graphics*, **25**(4):227-235.

Park, H., Kim, K., 1995. 3-D shape reconstruction from 2-D cross-sections. *Journal of Design and Manufacturing*, **5**:171-185.

Park, H., Kim, K., 1996. Smooth surface approximation to serial cross-sections. *Computer-Aided Design*, **28**(12): 995-1005. [doi:10.1016/0010-4485(96)00019-X]