Send Orders for Reprints to reprints@benthamscience.net

The Open Automation and Control Systems Journal, **2013,** *5,* **15-20**

Open Access

Construction of B-spline Surface via Interpolating to Boundary Information and Approximating Inner Points

Guowang Mu[1,*], Ting Zang[1] and Shijie Dai[2]

[1]*School of Science, Hebei University of Technology,Tianjin, 300401,China*

[2]*School of Mechanical Engineering, Hebei University of Technology, Tianjin, 300131, China*

Abstract: This paper discusses how to construct a B-spline surface from its boundary information and inner points. First, an algorithm is presented which can construct a B-spline surface that interpolates four given boundary curves and simultaneously approximates given inner points. Then, the approach is extended to interpolate the cross-boundary derivatives as well. The main idea of this method is that an initial surface interpolating four given boundary curves and the cross-boundary derivatives is constructed at first. Then, the inner control vertices of the surface are repositioned through energy optimization while the boundary control vertices of the initial surface remain unchanged. Examples are given which prove that the algorithm is practicable and effective.

Keywords: B-spline, Surface Reconstruction, Interpolation, Approximation, Optimization.

1. INTRODUCTION

B-spline surface reconstruction is a critical issue in the field of computer aided design. It also plays a key role in reverse engineering of CAD models. In general, there are two ways of constructing a B-spline surface. One way is to construct a B-spline surface from some discrete geometric data such as points, derivatives, normal vectors and even curvatures. This method is usually called fitting. It can be further classified into two categories: interpolation and approximation. In interpolation, a surface is constructed to precisely satisfy the given data. For approximation, the constructed surface does not need to satisfy the given data precisely and it only needs to approximately pass through them[1]. There have been a huge number of papers on surface fitting and many surface fitting methods have been presented. The most commonly used surface fitting methods can be found in [1-3]. Another way is to construct a B-spline surface from given curves. Many well-known surface construction methods fall into this category, such as skinning, swung, swept, Coons and Gordon surfaces. In skinning, we construct a surface which either interpolates or approximates the specified section curves. Swung surface is constructed via profile curve and trajectory curve, and swept surface is constructed via section curve and trajectory curve [1]. Coons developed a technique for construction of a surface that interpolates its four boundary curves by the Boolean Sum. Gordon called this type of interpolation transfinite interpolation, and extended it to construct surfaces by interpolating bidirectional curve networks [4]. Gregory proposed a method to fill the holes bounded with bi-cubic Bezier patches [5]. Recently, construction of an N-sided surface or an N-sided region has received a lot of attention in the field of CAD. Lazhu Wang *et al*. [6] proposed the Coons Type Blended B-spline Surface and gave the algorithm to convert it to a NURBS surface. Ling Ma *et al*. [7] suggested a method for constructing an N-sided surface with PDE. Dejun Song *et al*. [8] propounded to construct an N-sided surface with the energy minimization method. Guiqing Li *et al*. [9] recommended an approach to blend parametric patches with subdivision surfaces. Some of the latest results can also be found in [10-14].

In surface reconstruction related to reverse engineering, the problem of smooth connection, i.e., how to guarantee that there is no gap or that some degree of continuity is satisfied between neighboring B-spline surfaces, is very important. One solution is as follows: the boundary curves of each B-spline surface are first constructed from boundary points. If the C^1 continuity is required, the cross-boundary derivatives are also estimated from the measured data. Then, a B-spline surface is constructed to interpolate the boundary curves (or even to the cross-boundary derivatives if C^1 continuity is required). If all the surfaces are constructed with this method, it is obvious that C^0 or C^1 continuity condition can be satisfied between adjacent surfaces. Because usually a large number of inner points are also measured from each surface, we also need each surface to approximate the inner points simultaneously.

Although many methods for filling holes or constructing a surface to interpolate boundary curves have been presented, as far as we know, how to construct a B-spline surface to interpolate the four boundary curves and simultaneously approximate the inner points has not been discussed in literatures.

In this paper, we consider the construction of B-spline surfaces by interpolating its boundary curves, or even the cross-boundary derivatives, and approximating the inner

*Address correspondence to this author at the School of Science, Hebei University of Technology, Tianjin, 300401, P.R.China; Tel: +86-022-6043 5638; E-mail: muguow@hebut.edu.cn

1874-4443/13

points simultaneously. The remainder of this paper is organized as follows: the method for the construction of a B-spline surface by interpolating four boundary curves and approximating the inner scattered data points is described in section 2, and the method is extended to interpolate the four boundary curves and the cross-derivatives and approximate the inner scattered data points in section 3. Some examples are given to validate the algorithm in section 4, and finally the paper is summarized in section 5.

2. INTERPOLATING FOUR BOUNDARY CURVES AND APPROXIMATING INNER POINTS

The problem can be described as follows:

Given four B-spline curves $C_{u,0}(u)$, $C_{u,1}(u)$, $C_{0,v}(v)$ and $C_{1,v}(v)$, $u\in[0,1]$, $v\in[0,1]$, and some points Q_k (k=0,1,2,…,N), assuming that the curves satisfy the C^0 compatibility conditions, i.e.,

$C_{u,0}(u=0)= C_{0,v}(v=0)=S_{0,0}$

$C_{u,0}(u=1)= C_{1,v}(v=0)=S_{1,0}$

$C_{u,1}(u=0)= C_{0,v}(v=1)=S_{0,1}$

$C_{u,1}(u=1)= C_{1,v}(v=1)=S_{1,1}$

Construct a surface $S(u,v)$ such that it satisfies:

1) $S(u,v)$ has the four curves as its boundaries, i.e.,

$S(u,0)=C_{u,0}(u)$, $S(u,1)=C_{u,1}(u)$

$S(0,v)=C_{0,v}(v)$, $S(1,v)=C_{1,v}(v)$

2) $S(u,v)$ approximates Q_k (k=0,1,2,…,N) within a given tolerance τ.

The basic idea of our method is: first, we construct an initial B-spline surface $S(u,v)$ which has the four curves $C_{u,0}(u)$, $C_{u,1}(u)$, $C_{0,v}(v)$ and $C_{1,v}(v)$ as its boundaries; then, we reposition the inner control vertices of $S(u,v)$ while the boundary control vertices remain unchanged such that the surface $S(u,v)$ can both interpolate the four boundary curves and approximate the inner points Q_k (k=0,1,2,…,N).

2.1. Construction of the Initial Surface

For simplicity, we assume that $C_{u,0}(u)$ and $C_{u,1}(u)$ are endpoint-interpolating cubic B-spline curves defined on a common knot vector U=[0=u_0= ⋯ = u_p, u_{p+1} ⋯ u_m, u_{m+1}= ⋯ = u_{m+p+1}=1], $C_{0,v}(v)$ and $C_{1,v}(v)$ are endpoint-interpolating cubic B-spline curves defined on a common knot vector V=[0=v_0=⋯= v_q, v_{q+1} ⋯ v_n, v_{n+1}=⋯= v_{n+q+1}=1], here, p=q=3. Therefore, $C_{u,0}(u)$ and $C_{u,1}$(u) can be formulated as

$$C_{u,r}(u)=\sum_{i=0}^{m}P_{r,i}N_{i,p}(u)\ ,\ r=0,1,u\in[0,1] \tag{1}$$

where, $\{P_{r,i}\}$ are the control vertices of boundary curve $C_{u,r}(u)$; $N_{i,p}(u)$, i=0,1,…,m are the B-spline basis functions defined on the knot vector U. Similarly, $C_{0,v}(v)$ and $C_{1,v}(v)$ can be formulated as

$$C_{s,v}(v)=\sum_{j=0}^{n}Q_{s,j}N_{j,q}(v)\ ,\ s=0,1,v\in[0,1] \tag{2}$$

where, $\{Q_{s,j}\}$ are the control vertices of boundary curve $C_{s,v}(v)$; $N_{j,q}(v)$, j=0,1,…,n are the B-spline basis functions defined on the knot vector V. We construct the initial B-spline surface $S(u,v)$ with the knot vector U and V (i.e, the same as $C_{u,r}(u)$ and $C_{s,v}(v)$ respectively), and we formulate $S(u,v)$ as

$$S(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}V_{i,j}N_{i,p}(u)N_{j,q}(v) \tag{3}$$

where, $\{V_{i,j}\}$ are control vertices of the surface. In order to interpolate the four boundary curves, we set

$$V_{i,0}=P_{0,i},\ V_{i,n}=P_{1,i},\ i=0,1,\ldots,m$$
$$V_{0,j}=Q_{0,j},\ V_{mj}=Q_{1,j},\ j=0,1,\ldots,n \tag{4}$$

Due to the form of the knot vectors and the properties of B-spline curves and surfaces [1, 2], if the boundary control vertices are set as above, then, $S(u,0)=C_{u,0}(u)$, $S(u,1)=C_{u,1}(u)$, $S(0,v)=C_{0,v}(v)$ and $S(1,v)=C_{1,v}(v)$ are always true, no matter how the other control vertices (inner control vertices) are positioned. It also shows that the bicubic B-spline surface interpolating its four cubic B-spline curves is not unique.

To determine the inner control vertices $V_{i,j}$ (i=1,…, m-1; j=1,…,n-1) uniquely, we adopt the minimum energy method, that is, we seek $V_{i,j}$ (i=1,…,m-1; j= 1,…,n-1) such that the strain energy of the resulted surface $S(u,v)$ is minimized subject to Eq.(4). In this paper, we adopt

$$E=\iint(S_{uu}^2+2S_{uv}^2+S_{vv}^2)dudv \tag{5}$$

as the energy function of the surface $S(u,v)$, where S_{uu}, S_{uv}, and S_{vv} are the second order derivatives of $S(u,v)$ [15]. Substituting Eq.(3) into Eq.(5), we obtain

$$E=\int_0^1\int_0^1\sum_{i=0}^{m}\sum_{j=0}^{n}V_{i,j}\cdot\sum_{k=0}^{m}\sum_{l=0}^{n}V_{k,l}\begin{bmatrix}N''_{i,p}(u)N_{j,q}(v)N''_{k,p}(u)N_{l,q}(v)\\+2N'_{i,p}(u)N'_{j,q}(v)N'_{k,p}(u)N'_{l,q}(v)\\+N_{i,p}(u)N''_{j,q}(v)N_{k,p}(u)N''_{l,q}(v)\end{bmatrix}dudv \tag{6}$$

Eq.(6) can be rewritten into the matrix form

$$E=V^TMV \tag{7}$$

where, V=[v_0,v_1,…,v_{mn+m+n}] is the vector of the control vertices, M=[$m_{r,c}$]$_{N,N}$ is an $N\times N$ symmetric matrix, N =(m+1)(n+1). According to Eq.(6), the elements of vector V and matrix M can be determined as follows:

for i=0,…, m; j=0,…,n; k=0,…, m; l=0,…,n

let r=i(n+1)+j, c=k(n+1)+l, v_r=$V_{i,j}$, v_c=$V_{k,l}$

$$m_{r,c}=\int_0^1\int_0^1\begin{bmatrix}N''_{i,p}(u)N_{j,q}(v)N''_{k,p}(u)N_{l,q}(v)\\+2N'_{i,p}(u)N'_{j,q}(v)N'_{k,p}(u)N'_{l,q}(v)\\+N_{i,p}(u)N''_{j,q}(v)N_{k,p}(u)N''_{l,q}(v)\end{bmatrix}dudv \tag{8}$$

Finally, the control vertices of the initial surface $S(u,v)$ can be solved through the following optimization procedure:

s.t. $V_{i,0}=P_{0,i}$, $V_{i,n}=P_{1,i}$, i=0,1,…,m

$V_{0,j} = Q_{0,j}$, $V_{m,j} = Q_{1,j}$, $j=0,1,\ldots,n$ (9)

2.2. Parameterization of the Data Points

We used the method proposed by Weiyin Ma [16] to determine the parameters of the given points $Q_k(k=0,1,\ldots,N)$, i.e., we project the point Q_k onto the initial surface $S(u,v)$, and then set the parameter (u_k,v_k) of the projection point as the parameter of Q_k. This also can be expressed by an optimization problem:

$$(u_k, v_k) = \arg\min_{(u,v)\in[0,1]\times[0,1]} dist(Q_k, S(u,v)) \quad (10)$$

where, $dist(Q_k, S(u,v))$ denotes the distance from Q_k to $S(u,v)$.

2.3. Reposition of the Inner Control Vertices

Now, we have obtained the initial surface which interpolates the four boundary curves. Next, we reposition the inner control vertices $V_{i,j}$ ($i=1,\ldots, m-1$; $j=1,\ldots,n-1$) to approximate the inner points $Q_k(k=0,1,2,\ldots,N)$. Again, we convert it into an optimization problem: seek $V_{i,j}(i=1,\ldots,m-1; j=1,\ldots, n-1)$, such that $\sum_{k=0}^{N} d_k^2$ is minimized, where, $d_k=dist(Q_k, S(u,v))$ is the distance from Q_k to the surface by repositioning the inner control vertices. For simplicity, we use $d_k= dist(Q_k, S(u_k,v_k))$ to substitute for $d_k= dist(Q_k, S(u,v))$, where (u_k,v_k) is calculated by the method described in section 2.2.

However, the above optimization problem can be numerically unstable. Considering the stability of solution and the smoothness of the surface, we can add a smooth term (or regularization term) in the objective function. So, we adjust the inner vertices $V_{i,j}$ ($i=1,\ldots, m-1$; $j=1,\ldots,n-1$) with following optimization problem:

$$\min\ J = \sum_{k=1}^{N} w_k \left(S(u_k, v_k) - Q_k\right)^2 + \lambda V^T M V$$

$$\text{s.t. } V_{i,0} = P_{0,i},\ V_{i,n} = P_{1,i},\ i=0,1,\ldots,m;$$

$$V_{0,j} = Q_{0,j},\ V_{m,j} = Q_{1,j}, j=0,1,\ldots,n \quad (11)$$

where, w_k ($k=0,1,2,\ldots, N$) are the weights of the given points, λ is a smooth parameter. When w_k ($k=0,1,2,\ldots, N$) are fixed, λ controls the proportion of the smooth term in the objective. If λ is larger, the resulted surface is smoother, but the distances from the resulted surface to the given points are larger; however, if λ is too small or zero, the solution is either not stable or the surface is not fair. Usually, we set $w_k=1$ ($k=0,1,2,\ldots, N$) and choose a λ in the range of [0.0001,0.1] initially, and then adjust w_k ($k=0,1,2,\ldots, N$) according to the distances from the resulted surface to the given points.

2.4. Description of the Algorithm

Our algorithm for interpolating the four boundary curves and simultaneously approximating the inner points can be described as follows:

Algorithm:

Step 1: Assign a value to the smooth parameter λ, and let $w_k=1$ ($k=0,1,2,\ldots, N$), $iter=1$.

Step 2: (Construct the Initial Surface): Construct the initial B-spline surface $S(u,v)$ which has the four curves $C_{u,0}(u)$, $C_{u,1}(u)$, $C_{0,v}(v)$, and $C_{1,v}(v)$ as its boundaries using the method described in section 2.1.

Step 3: (Parameterizing the Points): Compute the parameters of the given points Q_k ($k=0,1,2,\ldots,N$) using the method described in section 2.2.

Step 4: (Repositioning the Inner Control Vertices): Reposition the inner control vertices of $S(u,v)$ while keeping the boundary control vertices unchanged by solving the optimization problem in Eq.(11).

Step 5: (Adjusting the Weights) Let t=0; For each inner point Q_k, calculate the distance $d_k= dist(Q_k, S(u,v))$; if $dist(Q_k, S(u,v))>\tau$, let $t=t+1$, $w_k=2w_k$.

Step6: If $t= =0$ or $iter= =$MAX_ITER_NUM, go to Step7; Else let $iter=iter+1$, and increase the number of control vertices of the surface by inserting middle knots in each u span and v span, go to step 4.

Step 7: end

3. EXTENSION OF THE ALGORITHM

In this section, we extend our method to construct B-spline surfaces by interpolating the boundary curves and cross-boundary derivatives meanwhile approximating the inner points. The problem can be stated as follows:

Given the four B-spline curves $C_{u,0}(u)$, $C_{u,1}(u)$, $C_{0,v}(v)$,$C_{1,v}(v)$, the cross derivatives $D_{u,0}(u)$, $D_{u,1}(u)$, $D_{0,v}(v)$ and $D_{1,v}(v)$, $u\in[0,1]$, $v\in[0,1]$, suppose that compatibility conditions

$$C_{0v}(0) = C_{u0}(0) = S_{00}, C_{u0}(1) = C_{1v}(0) = S_{10}$$

$$C_{1v}(1) = C_{u1}(1) = S_{11}, C_{u1}(0) = C_{0v}(1) = S_{01}$$

$$\frac{dC_{0v}}{dv}(0) = D_{u0}(0),\ \frac{dC_{u0}}{du}(0) = D_{0v}(0)$$

$$\frac{dC_{u0}}{du}(1) = D_{1v}(0), \frac{dC_{1v}}{dv}(0) = D_{u0}(1)$$

$$\frac{dC_{1v}}{dv}(1) = D_{u1}(1), \frac{dC_{u1}}{du}(1) = D_{1v}(1)$$

$$\frac{dC_{u1}}{du}(0) = D_{0v}(1), \frac{dC_{0v}}{dv}(1) = D_{u1}(0)$$

$$\frac{dD_{0v}}{dv}(0) = \frac{dD_{u0}}{du}(0),\ \frac{dD_{u0}}{du}(1) = \frac{dD_{1v}}{dv}(0)$$

$$\frac{dD_{1v}}{dv}(1) = \frac{dD_{u1}}{du}(1), \frac{dD_{u1}}{du}(0) = \frac{dD_{0v}}{dv}(1)$$

are satisfied, also, given N inner points $Q_k(k=0,1,2,\cdots N)$, we want to construct a B-spline surface $S(u,v)$ such that it satisfies:

1) $S(u,v)$ has the four curves $C_{u,0}(u)$, $C_{u,1}(u)$, $C_{0,v}(v)$ and $C_{1,v}(v)$ as its boundaries;

2) $S(u,v)$ has cross-boundary derivatives $D_{u,0}(u)$, $D_{u,1}(u)$, $D_{0,v}(v)$ and $D_{1,v}(v)$ along the four boundary curves, i.e.,

$$\frac{\partial S}{\partial v}(u,0) = D_{u0}(u), \frac{\partial S}{\partial v}(u,1) = D_{u1}(u)$$

$$\frac{\partial S}{\partial u}(0,v) = D_{0v}(v), \frac{\partial S}{\partial u}(1,v) = D_{1v}(v) \quad (12)$$

3) $S(u,v)$ approximates Q_k (k=0,1,2,…,N) within a given tolerance τ .

We adopt a similar method used in section 2: first, we construct an initial B-spline surface $S(u,v)$ which interpolates the four boundary curves and the cross-boundary derivatives; then, we reposition the inner control vertices of $S(u,v)$ while the boundary control vertices remains unchanged such that the surface $S(u,v)$ interpolates the four boundary curves and the cross-boundary derivatives meanwhile approximates the inner points Q_k (k=0,1,2,…,N) .

3.1. Generation of the Initial Surface

Suppose that the four boundary curves $C_{u,0}(u)$, $C_{u,1}(u)$, $C_{0,v}(v)$ and $C_{1,v}(v)$ are all the same as in the section 2, $D_{u,0}(u)$ and $D_{u,1}(u)$ are endpoint-interpolating cubic B-spline curves defined over the knot vector U, which is same as the knot vector of $C_{u,0}(u)$ and $C_{u,1}(u)$. $D_{0,v}(v)$ and $D_{1,v}(v)$ are endpoint-interpolating cubic B-spline curves defined on the same knot vector V, which is same as the knot vector of $C_{0,v}(v)$ and $C_{1,v}(v)$, p=q=3. Therefore, $D_{u,0}(u)$ and $D_{u,1}(u)$ can be formulated as

$$D_{u,r}(u) = \sum_{i=0}^{m} P'_{r,i} N_{i,p}(u) \ , \ r = 0,1, u \in [0,1] \quad (13)$$

where, $\{P'_{r,i}\}$ are the control vertices of derivative curve $D_{u,r}(u)$. Similarly, $D_{0,v}(v)$ and $D_{1,v}(v)$ can be formulated as

$$D_{s,v}(v) = \sum_{j=0}^{n} Q'_{s,j} N_{j,q}(v) \ , \ s = 0,1, v \in [0,1] \quad (14)$$

where, $\{Q'_{s,j}\}$ are the control vertices of derivative curve $D_{s,v}(v)$. We construct the initial B-spline surface $S(u,v)$ with the knot vector U and V respectively, also, we formulate $S(u,v)$ as in Eq.(3).

In order to make $S(u,v)$ interpolate the four boundary curves, we set the outermost control vertices as in Eq.(4).

To make $S(u,v)$ interpolate the given cross-boundary derivatives on the four boundary curves, we further set

$$V_{i,1} = V_{i,0} + \frac{v_4}{3} P'_{0,i}, \ V_{i,n-1} = V_{i,n} - \frac{1-v_n}{3} P'_{1,i}, \ i = 0,1,2 \cdots m$$

$$V_{1,j} = V_{0,j} + \frac{u_4}{3} Q'_{0,j}, V_{m-1,j} = V_{m,j} - \frac{1-u_m}{3} Q'_{1,j}, j = 0,1,2 \cdots n \quad (15)$$

To determine the inner control vertices $V_{i,j}$ (i=2,…, m-2; j=2,…,n-2), we adopt the energy method similar as in section 2.1 to seek $V_{i,j}$ (i=2,…, m-2; j=2,…,n-2) such that the strain energy of the resulted surface $S(u,v)$ is minimized subject to Eq.(4) and Eq.(15). The form of the energy function is same as that in section 2.1.

3.2. Adjust the Inner Control Vertices

Now, we have obtained an initial surface which interpolates the four boundary curves and the cross-boundary derivatives on the boundaries. Next, we reposition the inner control vertices $V_{i,j}$ (i=2,…, m-2; j=2,…,n-2) to approximate the inner points Q_k (k=0,1,2,…,N). Similarly as in section 2, we seek $V_{i,j}$ (i=2,…, m-2; j=2,…,n-2) with an optimization problem

$$\min \ J = \sum_{k=1}^{N} w_k \left(S(u_k, v_k) - Q_k \right)^2 + \lambda V^T M V$$

s.t. $V_{i,0} = P_{0,i}$, $V_{i,n} = P_{1,i}$, $i = 0,1,\cdots,m$,

$V_{0,j} = Q_{0,j}$, $V_{m,j} = Q_{1,j}$, $j = 0,1,\cdots,n$

$$V_{i,1} = V_{i,0} + \frac{v_4}{3} P'_{0,i}, \quad V_{i,n-1} = V_{i,n} - \frac{1-v_n}{3} P'_{1,i}, \quad i = 0,1,\cdots,m;$$

$$V_{1,j} = V_{0,j} + \frac{u_4}{3} Q'_{0,j}, V_{m-1,j} = V_{m,j} - \frac{1-u_m}{3} Q'_{1,j}, \quad j = 0,1,\cdots,n \quad (16)$$

where, $w_k (k = 0,1,2,\cdots N)$, λ are same as in section 2, (u_k,v_k) are the parameter of points Q_k (k=0,1,2,…,N), they are determined using the same method as in section 2.

Fig. (1). The boundary curves and the inner points.

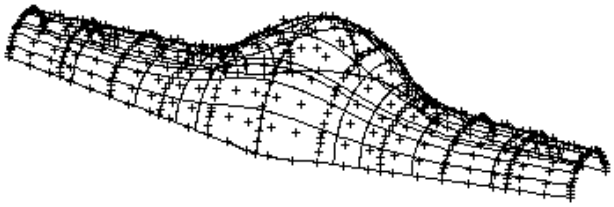

Fig. (2). The surface that interpolates the boundary curves and approximates the inner points.

4. EXAMPLES

In this section, we give two examples of B-spline surface reconstruction with our method. The first is to generate a B-spline surface by interpolating four given boundary curves and approximating the inner scattered points, the second is to generate a B-spline surface by interpolating given boundary curves and cross-boundary derivatives on the boundaries and approximating the inner points.

Example 1: Four boundary curves and some inner points are shown in Fig. (**1**), and the B-spline surface interpolating the four boundary curves and approximating the inner points is reconstructed with the algorithm described in section 2.4, and the final surface and the original data are depicted in Fig. (**2**). In our experiment, we set smooth parameter λ=0.0001.

Example 2: Fig. (**3**) through Fig. (**5**) show an example of B-spline surface reconstruction in reverse engineering. Suppose we have the measured data points on a surface, and we

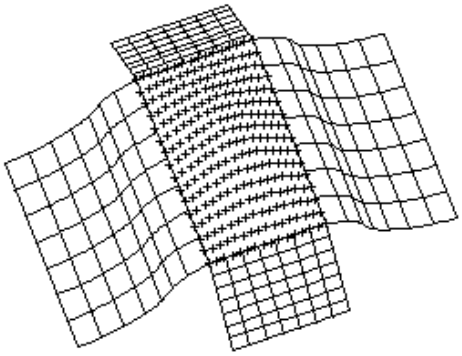

Fig. (3). The measured data points of a surface and four surrounding B-spline surfaces.

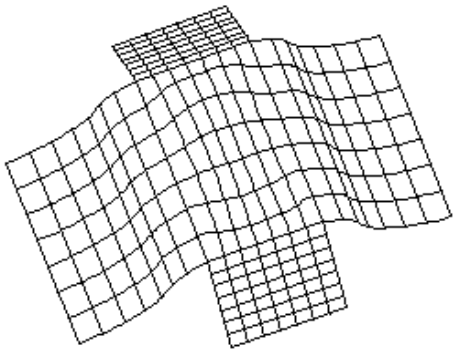

Fig. (4). The surface interpolating the four boudary curves and the cross-boundary derivatives and its surrounding surfaces.

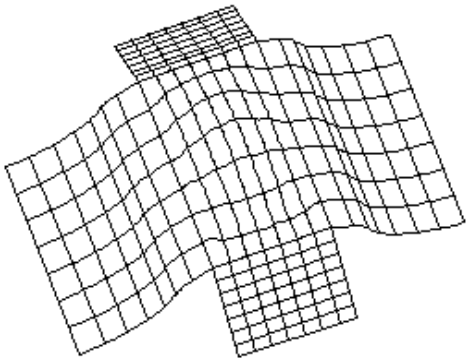

Fig. (5). The constructed B-spline surface and its surrounding surfaces.

have constructed the four surrounding B-spline surfaces, shown in Fig. (**3**). We want to construct a B-spline surface $S(u,v)$ to approximate the measured data, at the same time, we want the surface to be smoothly connected with its four surrounding B-spline surfaces with C^1 continuity.

First, the four boundary curves and cross-boundary derivatives of the constructed surface are calculated from the four surrounding B-spline surfaces according to the C^1 continuity. Then, the initial surface which interpolates the given boundary curves and cross-boundary derivatives is constructed as shown in Fig. (**4**). Finally, the inner control vertices are modified with optimization problem (16) to obtain the final B-spline surface, shown in Fig. (**5**).

5. CONCLUSIONS

In this paper, reconstruction of a B-spline surface via interpolating the boundary geometric information and approximating the inner data points is studied. First, we give an algorithm for reconstruction of the B-spline surface to interpolate the four given boundary curves and simultaneously to approximate some given inner points. Then, we extend our algorithm to reconstruct the B-spline surface which can interpolate four given boundary curves and the cross-boundary derivatives, meanwhile approximate the given inner points. The main idea of our method is: first, we construct an initial surface which interpolates the four given boundary curves (or even the cross-boundary derivatives on the boundaries), then, while keeping the boundary control vertices of the initial surface unchanged, we reposition the inner control vertices of the surface with optimization. Our algorithm is very useful in reverse engineering of CAD models. Examples show that our algorithm is practicable and effective.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

This work was financially supported by the Natural Science Foundation of Hebei Province, CHINA (No.F20122-02041) and Youth Research Foundation of Science and Technology of Hebei Education Department,CHINA(No. Q2012022). Finally, the first author would like to thank Dr. Lazhu Wang for his great help on this work.

REFERENCES

[1] L. Piegl and W. Tiller, *The NURBS book,* 2nd ed, Springer-Verlag: New York, 1996.

[2] F. Shi, *Computer Aided Geometric Design and NURBS*, Higher Education Press, Beijing, China, 2001 (in Chinese).

[3] G. Mu, "*On the Reconstruction of NURBS Surfaces in Reverse Engineering"*, Ph.D. thesis, Beihang University, Beijing, China, 2001(in Chinese).

[4] W. Gordon, "Spline-blended surface interpolation through curve networks", *Journal of Math and Mechanics*, vol.18, no.10, pp. 931-952, 1969.

[5] J. Gregory, and J. Zhou, "Filling polygonal holes with bicubic patches", *Computer Aided Geometric Design*, vol. 11, no.4, pp.391-410, 1994.

[6] L. Wang, and X. Zhu, "Local Interpolating Blended B-spline Surfaces and Its Conversion to NURBS Surfaces", *Computer Aided Drafting, Design and Manufacturing*, vol. 4, no.2, pp.5-17, 1994.

[7] L. Ma, X. Zhang and X. Zhu, "Construction of Blending Surface with PDE", *Journal of Engineering Graphics*, no.1, pp.1-8, 1995 (in Chinese).

[8] D. Song, and X. Zhu, "Construction of N-sided surface with minimum energy", *Journal of Engineering Graphics*, no.1, pp.41-47, 1998(in Chinese).

[9] G. Li, and H. Li, "Blending Parametric Patches with Subdivision Surfaces", *Journal of Computer Science and Technology*, vol. 17, no.4, pp. 498-506, 2002.

[10] J. Gregory, and J. Zhou, "Irregular C2 surface construction using bi-polynomial rectangular patches", *Computer Aided Geometric Design*, vol. 16, no.5, pp. 423-435, 1999.

[11] L. Piegl, and W. Tiller, "Filling n-sided regions with NURBS patches", *The Visual Computer*, vol. 15, no.2, pp.77-89, 1999.

[12] N. Pla-Garcia, and M. Vigo-Anglada, "N-sided patches with B-spline boundaries", *Computer & Graphics*, vol. 30, no.6, pp. 959-970, 2006.

[13] K. Shi, and J. Yong, "Filling n-sided regions with G1 triangular Coons B-spline patches", *The Visual Computer*, vol. 26, no. 6-8, pp. 791-800, 2010.

[14] T. Varady, A. Rockwood, and P. Salvi, "Transfinite surface interpolation over irregular n-sided domains", *Computer Aided Design*, vol. 43, no.11, pp.1330-1340, 2011.

[15] D. Song, and X. Zhu, "B-spline Interpolation based on optimization", *Computer Aided Drafting, Design and Manufacturing*, vol. 7, no.2, pp.1-8, 1997.

[16] W. Ma, and J. P. Kruth, "Parametrization of randomly measured points for least squares fitting of B-spline curve and surfaces", *Computer-Aided Design*, vol. 27, no.9, pp. 663-675,1995.

Received: April 10, 2013 Revised: September 17, 2013 Accepted: September 28, 2013