Proceedings of the 6th WSEAS International Conference on Robotics, Control and Manufacturing Technology, Hangzhou, China, April 16-18, 2006 (pp271-276)

Designing Bézier Surfaces Minimizing the Gaussian Curvature

MO GUOLIANG, ZHAO YANAN
Information and Computational Science
Zhejiang University City College
Huzhou Street, Hangzhou, 310015
PEOPLE'S REPUBLIC OF CHINA

Abstract: In the freeform surface design, developable surfaces have much application value. But, in 3D space, there is not always a regular developable surface which interpolates the given boundary of an arbitrary piecewise smooth closed curve. In this paper, tensor product Bézier surfaces interpolating the closed curves are determined and the result surface is a minimum of the functional defined by the L^2 -integral norm of the Gaussian curvature. The Gaussian curvature of the surfaces is minimized by the method of solving nonlinear optimization problems. An improved approach is proposed: the Pseudo-Newtonian method. A simple application example is also given.

Key-Words: approximate developable surfaces, L^2 -integral norm, nonlinear optimization

1 Introduction

A ruled surface is generated by continuous motion of a straight line in 3D-space along a spatial curve called a directrix. These straight lines are called generators, or rulings, of the surface. A developable surface is a special ruled surface which has the same tangent plane at all points along a generator. The Gaussian Curvature of a developable surface is zero, and vice versa. There is only a developable surface that can be isometrically mapped onto a plane. Therefore developable surfaces are particularly interesting and appealing, owing to the simplicity in the manufacturing process required to fabricate them. It plays an important role in the geometric design and modeling. It can be conveniently formed by bending or rolling a planar surface without any stretching or contraction. Based on these, developable surfaces are widely used in manufacturing items from materials that are not amenable to stretching, such as ship hulls, ducts, shoes, clothing, and automobiles parts. Thereby developable surfaces have been widely used in CAD/CAM systems. Many papers presented application of developable surfaces in industry [1-4].

When a piecewise-smooth closed curve is given as a boundary curve, there is not always a regular developable surface interpolating the boundary in theory [5-6]. There is still not a simple and effective method for constructing a developable surface in Computer Graphics (CG) and Computer Aided Geometric Design (CAGD). In this paper, when the boundary control points of a Bézier surface are given, we'll determine the interior control points in order where the resulting surface has a minimum of the functional defined by the L^2 -integral norm of the Gaussian curvature. There are approximations to the extremals of the functional defined by the Gauss curvature. The key is how to solve the extreme value problem. The parameter domain can be subdivided into small ones and the continuous functional becomes a concrete one. We adopt a well-known optimization algorithm (the BFGS method) to the concrete functional in order to get the approximations. The Pseudo-Newtonian method is improved and corrected for this special problem [7-11]. Examples show that computation of the approximation is simple and fast.

In Section 2, the definition of an approximate is presented. The algorithm of designing an approximate developable surface is given in Section 3. In Section 4, some examples are given in detail. An important application is described in Section 5. Finally, conclusions are drawn.

2 Approximate developable surfaces

In this section, the definition of an approximate developable surface is presented.

Definition: In 3D space, a piecewise-smooth closed space curve $\Gamma : \varphi(u,v)$ is given. Ω is the domain enclosed by $\partial\Omega$, the projection of Γ in parameter plane. For any $(u,v) \in \Omega$, there is a surface set:

Proceedings of the 6th WSEAS International Conference on Robotics, Control and Manufacturing Technology, Hangzhou, China, April 16-18, 2006 (pp271-276)

$$Y := \{ \boldsymbol{r} = \boldsymbol{r}(u,v) = (x(u,v), y(u,v), z(u,v)) : (u,v) \in \Omega \quad \boldsymbol{r}(\partial\Omega) = \Gamma \}.$$

If $\|K^*\| = \min_{\boldsymbol{r}\in Y} \|K\|$, then $\boldsymbol{r}^* \in Y$ is called an approximate developable surface, where $\|K\|$ is a norm of K, K is the Gaussian Curvature of $\boldsymbol{r}$, K^* is the Gaussian Curvature of $\boldsymbol{r}^*$.

In classical differential geometry, the Gaussian Curvature is defined as:

$$K = \frac{LN - M^2}{EG - F^2},$$

where E, F, G are coefficients of the first fundamental form of $\boldsymbol{r}$, L, M, N are coefficients of the second fundamental form of $\boldsymbol{r}$. They are defined as

$$E = \langle \boldsymbol{r}_u, \boldsymbol{r}_u \rangle, F = \langle \boldsymbol{r}_u, \boldsymbol{r}_v \rangle, G = \langle \boldsymbol{r}_v, \boldsymbol{r}_v \rangle,$$

$$L = \langle \boldsymbol{r}_{uu}, \boldsymbol{n} \rangle, M = \langle \boldsymbol{r}_{uv}, \boldsymbol{n} \rangle, N = \langle \boldsymbol{r}_{vv}, \boldsymbol{n} \rangle, \boldsymbol{n} = \frac{\boldsymbol{r}_u \times \boldsymbol{r}_v}{\|\boldsymbol{r}_u \times \boldsymbol{r}_v\|},$$

where $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ denotes the inner product of two vectors $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{n}$ is the unit normal vector of $\boldsymbol{r}$.

In this paper, the discussed problem is that given a piecewise-smooth closed curve $\Gamma : \varphi(u,v)$ in 3D, an approximate developable surface with the curve Γ as its boundary is constructed and the surface is smooth or at least 1-order smooth. In the definition of an approximate developable surface, the key is how to minimize the norm of the Gaussian Curvature and this is a nonlinear optimization problem. To overcome this, the problem can be simplified. Supposed that Γ is composed of four end-to-end curves $\{\Gamma_i\}_{i=1}^4$, and the norm of Gaussian Curvature K is $L^2(\Omega)$

$$\|K\|_{L^2} = \sqrt{\iint_\Omega \left(\frac{LN - M^2}{EG - F^2} \right)^2 dudv}.$$

The problem becomes a minimization problem in a function space:

$$\min_{\boldsymbol{r}\in Y} \|K\|_{L^2} = \min_{\boldsymbol{r}\in Y} \sqrt{\iint_\Omega \left(\frac{LN - M^2}{EG - F^2} \right)^2 dudv}$$

where

$$Y := \{ \boldsymbol{r} = \boldsymbol{r}(u,v) = (x(u,v), y(u,v), z(u,v)) : (u,v) \in \Omega \quad \boldsymbol{r}(\partial\Omega) = \Gamma \}.$$

$\boldsymbol{r}^* \in Y$ is a minimizer and called an approximate developable surface when $\|K\|_{L^2}$ is a minimum value.

By variation principles, an Euler-Lagrange equation can be gained. It is a partial differential equation (PDE). The problem further becomes how to solve a PDE. But the Euler-Lagrange equation is nonlinear and has 4-order partial derivatives. The form of the PDE is complex and difficult to solve [12]. In this paper, the optimization method is presented to solve the problem directly.

3 The algorithm of designing an approximate developable surface

Next, the space Y can be appointed as a space of $m \times n$ tensor product polynomial surfaces. The task is how to solve a nonlinear optimization problem. In this section, how to construct an optimization object function and choose an initial value is discussed in detail. Then two methods are also given to solve the optimization problem. At last, the process of solving the model is presented.

3.1 $m \times n$ tensor product polynomial surfaces as approximate developable surfaces

When an approximate developable surface is constructed, it is smooth and not too complex. So it is a better choice to appoint the space Y as a space of $m \times n$ tensor product polynomial surfaces. A tensor product polynomial is smooth and easy to design. The Bézier form of an $m \times n$ tensor product polynomial surface is usually used.

Supposed that Γ is composed of four end-to-end Bézier curves $\{\Gamma_i\}_{i=1}^4$, that is, $\Gamma_1 : \boldsymbol{c}_0(u)$, $\Gamma_2 : \boldsymbol{d}_0(v)$, $\Gamma_3 : \boldsymbol{c}_1(u)$, $\Gamma_4 : \boldsymbol{d}_1(v)$, $(0 \le u, v \le 1)$ and the parameter domain is $\Omega = [0,1] \otimes [0,1]$. The approximate solution of the minimum optimization problem can be expressed as:

$$\boldsymbol{r}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \boldsymbol{P}_{ij} B_i^m(u) B_j^n(v)$$

where $B_i^m(u) = \binom{m}{i} (1-u)^{m-i} u^i \, (i = 0,1,\ldots m)$ are Bernstein basis functions and $\boldsymbol{P}_{ij} (i = 0,1,\ldots,m;\ j = 0,1,\ldots,n)$ are control points in 3D. The control points $\boldsymbol{P}_{i0}, \boldsymbol{P}_{in} (i = 0,1,\ldots,m)$ and $\boldsymbol{P}_{0j}, \boldsymbol{P}_{mj} (j = 0,1,\ldots,n)$ are determined by the boundary curve Γ. Another control points $\boldsymbol{P}_{ij}(i=1,\cdots,m-1;\ j=1,\cdots,n-1)$ are undetermined. The process of determining control points $\boldsymbol{P}_{ij}$ is a nonlinear optimization problem.

3.2 Forming object functions and choosing initial values

A nonlinear optimization object function is formed. The surface $\boldsymbol{r}(u,v)$ is undetermined and the expression of the Gaussian Curvature is complex. It is difficult to express explicitly the integral of the object function $\|K\|_{L^2}$. The parameter domain Ω is uniformly divided $p\times q$ small sub-patches. The integral of $\iint_\Omega \left(\frac{LN-M^2}{EG-F^2}\right)^2 dudv$ can be substituted by Riemann Sum, that is,

$$\|K\|_{L^2} = \sqrt{\iint_\Omega \left(\frac{LN-M^2}{EG-F^2}\right)^2 (u,v) dudv}$$

$$\approx \sqrt{\frac{1}{pq}\sum_{i=1}^{p}\sum_{j=1}^{q}\left(\frac{LN-M^2}{EG-F^2}\right)^2 \left(\frac{i}{p},\frac{j}{q}\right)}$$

If the domain is divided more densely, the norm of $\|K\|_{L^2}$ is more accurate. But the computation work increases quickly. In this paper, the 20×20 division is adopted.

For a nonlinear optimization problem, iteration method is often adopted. The initial value $\boldsymbol{X}_0$ is given and the optimal point is gained as certain iteration form. The common iteration form is $\boldsymbol{X}_{i+1}=\boldsymbol{X}_i+\lambda_i\boldsymbol{D}_i, i=0,1\cdots$, where λ_i is the step length and $\boldsymbol{D}_i$ is the descent direction. It is fairly important to construct an iteration form and choose an initial value. These determine the speed of the algorithm and the result. In Computer Aided Geometric Design (CAGD), a Coons surface interpolating boundary curves is simply constructed and is smooth. Especially, it has been proved that if a pair of opposite boundary curves are straight lines, then the corresponding Coons surface is a ruled surface or if the degree of the curves is not more than 3 and there exists a ruled surface interpolating these boundary curves (the directions of the rulings are not always parallel to a parameter direction), then the corresponding Coons surface is also a ruled surface. So it is a better choice to choose a Coons surface as an initial surface. Hoschek expressed a Coons surface as [13]:

$$\boldsymbol{p}(u,v)=\boldsymbol{p}_1(u,v)+\boldsymbol{p}_2(u,v)-\boldsymbol{p}_3(u,v)$$

where

$$\begin{aligned}
\boldsymbol{p}_1(u,v) &= (1-v)\boldsymbol{c}_0(u)+v\boldsymbol{c}_1(u),\\
\boldsymbol{p}_2(u,v) &= (1-u)\boldsymbol{d}_0(v)+u\boldsymbol{d}_1(v),\\
\boldsymbol{p}_3(u,v) &= (1-u)\left[(1-v)\boldsymbol{c}_0(0)+v\boldsymbol{c}_1(0)\right]\\
&\quad +u\left[(1-v)\boldsymbol{d}_0(0)+v\boldsymbol{d}_1(0)\right].
\end{aligned}$$

$\boldsymbol{c}_0(u),\boldsymbol{c}_1(u),\boldsymbol{d}_0(v),\boldsymbol{d}_1(v)$ are end-to-end boundary curves, that is

$$\boldsymbol{c}_0(0)=\boldsymbol{d}_0(0),\boldsymbol{c}_0(1)=\boldsymbol{d}_0(1),\boldsymbol{c}_1(0)=\boldsymbol{d}_1(0),\boldsymbol{c}_1(1)=\boldsymbol{d}_1(1).$$

3.3 An optimization algorithm

In this section, the Pseudo-Newtonian method is presented to solve the optimization problem. The method is mature and efficient in optimization theory. But some details need to be improved appropriately in order to match the optimization problem in this paper.

Let

$$\boldsymbol{X}=\left(\boldsymbol{P}_{11},\boldsymbol{P}_{12},\cdots,\boldsymbol{P}_{1,n-1},\boldsymbol{P}_{21},\boldsymbol{P}_{22},\cdots,\boldsymbol{P}_{2,n-1},\cdots\right.$$
$$\left.\boldsymbol{P}_{m-1,1},\boldsymbol{P}_{m-1,2},\cdots,\boldsymbol{P}_{m-1,n-1}\right)$$

be a one dimension vector, and

$$F(\boldsymbol{X})=\sqrt{\iint_\Omega\left(\frac{LN-M^2}{EG-F^2}\right)^2 dudv}$$

be the object function. The fundamental idea is that a quadric function approximates origin optimization problem in the iteration point $\boldsymbol{X}_k$.

The object function $F(\boldsymbol{X})$ can be approximated in the iteration point $\boldsymbol{X}_k$ by a quadric function $Q_k(\boldsymbol{d})=F(\boldsymbol{X}_k)+\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}+\boldsymbol{d}^{\mathrm{T}}\boldsymbol{G}_k\boldsymbol{d}$, where $\boldsymbol{g}_k=\nabla F(\boldsymbol{X}_k)$, $\boldsymbol{G}_k=\nabla\left(\nabla F(\boldsymbol{X}_k)\right)$, ∇ is Hamilton operator, defined as $\nabla=\frac{\partial}{\partial x}\boldsymbol{i}+\frac{\partial}{\partial y}\boldsymbol{j}+\frac{\partial}{\partial z}\boldsymbol{k}$.

Because there are so many unknowns and the object function is complex, the gradient $\nabla F(\boldsymbol{X}_k)$ becomes complex and the Hessian matrix $\nabla\left(\nabla F(\boldsymbol{X}_k)\right)$ is hard to compute. It is not sure the matrix $\nabla\left(\nabla F(\boldsymbol{X}_k)\right)$ is semi-positive definite. Here, the Hessian matrix is approximated with the method BFGS in Pseudo-Newtonian method. The formula was proposed by Broyden, Fletcher, Goldfarb and Shanno in 1970. About this, you can refer to in the reference [10, 11].

$$\boldsymbol{G}_{k+1}^{\mathrm{BFGS}} = \boldsymbol{G}_k + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^{\mathrm{T}}}{\boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{y}_k} - \frac{\boldsymbol{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{B}_k}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{B}_k \boldsymbol{s}_k},$$

$$\boldsymbol{H}_{k+1}^{\mathrm{BFGS}} = \left(\boldsymbol{I} - \frac{\boldsymbol{s}_k \boldsymbol{y}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}_k} \right) \boldsymbol{H}_k \left(\boldsymbol{I} - \frac{\boldsymbol{y}_k \boldsymbol{s}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}_k} \right) + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}_k},$$

where $\boldsymbol{y}_k = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k, \boldsymbol{s}_k = \boldsymbol{d}_k, \boldsymbol{G}_{k+1}^{\mathrm{BFGS}}$ is the approximate matrix of the Hessian matrix $\boldsymbol{G}_k$ and $\boldsymbol{H}_{k+1}^{\mathrm{BFGS}}$ is the approximate matrix of the inverse matrix of the Hessian $\boldsymbol{G}_k$, $\boldsymbol{I}$ is an identity matrix. BFGS rank-one correction makes sure the Hessian matrix is semi-positive definite and it only depends on the rank-one information, and makes computation simple.

The descent direction is the direction $\boldsymbol{d}_k = -\boldsymbol{H}_k^{\mathrm{BFGS}} \boldsymbol{g}_k$ used in the Newtonian method. The step length is

$$\alpha_k = \frac{\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{H}_k^{\mathrm{BFGS}} \boldsymbol{g}_k}{\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{H}_k^{\mathrm{BFGS}} \boldsymbol{G}_k^{\mathrm{BFGS}} \boldsymbol{H}_k^{\mathrm{BFGS}} \boldsymbol{g}_k}.$$

Where $\boldsymbol{g}_k = \nabla F(\boldsymbol{X}_k)$, $\boldsymbol{H}_k^{\mathrm{BFGS}}, \boldsymbol{G}_k^{\mathrm{BFGS}}$ are corrected Hessian matrix respectively.

The final algorithm is presented as following.

Algorithm:

(1) Initialization, set $\boldsymbol{X}_0, \mu_0, 0 < \beta < 1, \varepsilon > 0, \boldsymbol{G}_0 = \boldsymbol{I}, \boldsymbol{H}_0 = \boldsymbol{I}, k = 0$.

(2) $\boldsymbol{X}_k$ are known, compute $\boldsymbol{g}_k$ and $\boldsymbol{G}_k^{\mathrm{BFGS}}, \boldsymbol{H}_k^{\mathrm{BFGS}}$. If $\|\boldsymbol{g}_k\| < \varepsilon$, then stop.

(3) Compute $\boldsymbol{d}_k = -\boldsymbol{H}_k^{\mathrm{BFGS}} \boldsymbol{g}_k$ and

$$\alpha_k = \frac{\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{H}_k^{\mathrm{BFGS}} \boldsymbol{g}_k}{\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{H}_k^{\mathrm{BFGS}} \boldsymbol{G}_k^{\mathrm{BFGS}} \boldsymbol{H}_k^{\mathrm{BFGS}} \boldsymbol{g}_k}.$$

(4) Let $\boldsymbol{X} = \boldsymbol{X}_k + \alpha_k \boldsymbol{d}_k$ and compute $F(\boldsymbol{X})$ and $F(\boldsymbol{X}_k)$.

(i)If $F(\boldsymbol{X}) \geq F(\boldsymbol{X}_k)$: if $\|F(\boldsymbol{X})\| < \varepsilon$, stop; otherwise, $\alpha_{k+1} = \beta \alpha_k$, $\boldsymbol{d}_{k+1} = \boldsymbol{d}_k$, $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k$, $k \leftarrow k+1$, go to step (4);

(ii) If $F(\boldsymbol{X}) < F(\boldsymbol{X}_k)$, $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + \alpha_k \boldsymbol{d}_k$

(5) Let $k \leftarrow k+1$, go to step (2).

The algorithm has the global convergence property and the detail analysis of Pseudo-Newtonian method convergence can be found in the reference. The algorithm has the global convergence property and the detail analysis of Pseudo-Newtonian method convergence can be found in Reference [10] and Reference [11].

3.4 The algorithm of solving the model

After some discussion about optimization problem, a new method for constructing an approximate developable surface is presented. The following are the steps.

(1) Determine the boundary curves, that is the control points $\boldsymbol{P}_{i0}, \boldsymbol{P}_{in} (i = 0,1,\ldots,m)$ and $\boldsymbol{P}_{0j}, \boldsymbol{P}_{mj} (j = 0,1,\ldots,n)$.

(2) Determine the object function $F(\boldsymbol{X}) = \|K\|_{L^2}$, where

$\boldsymbol{X} = \left(\boldsymbol{P}_{11}, \boldsymbol{P}_{12}, \ldots, \boldsymbol{P}_{1,n-1}, \boldsymbol{P}_{21}, \boldsymbol{P}_{22}, \ldots, \boldsymbol{P}_{2,n-1}, \ldots \boldsymbol{P}_{m-1,1}, \boldsymbol{P}_{m-1,2}, \ldots, \boldsymbol{P}_{m-1,n-1} \right)$

(3) Give the initial $\boldsymbol{X}_0$ which is gained from a Coons surface interpolating $\boldsymbol{c}_0(u), \boldsymbol{c}_1(u), \boldsymbol{d}_0(v), \boldsymbol{d}_1(v)$.

(4) Solve the optimization problem $\min F(\boldsymbol{X})$ and get the points $\boldsymbol{P}_{ij} (i = 1,\ldots,m-1; j = 1,\ldots,n-1)$.

(5) Depict the pictures.

4 Some examples

A bi-quadric tensor product surface and a bi-cubic one are constructed. Developable surfaces and non-developable ones are presented as examples. Approximate developable surface approximate these two types and the approximation errors are given.

As depicted in Fig.1 and Fig.2, the approximate developable surface whose boundary curves are given is in the left and the Gaussian Curvature is in the right. In Fig.1, the surface is a part of a cylinder, which is developable. The Gaussian Curvature of the approximate developable surface is 10^{-18} . In Fig.2, there is not certain whether there exists a developable surface interpolating the boundary curves. The Gaussian Curvature of the approximate developable surface is 10^{-2} . So there isn't an accurate developable surface interpolating the boundaries. An approximate has been constructed.

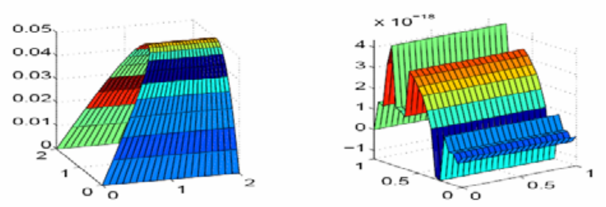

Fig.1 an approximate developable surface in the left, the Gaussian Curvature distribution in the right, $\max\|K\|_{L^2} \approx 10^{-18}$

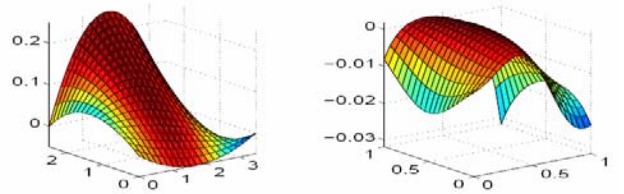

Fig.2 an approximate developable surface in the left, the Gaussian Curvature distribution in the right, $\max\|K\|_{L^2} \approx 10^{-2}$

A bi-cubic tensor product has more free control points than a bi-quadric one. There are 16 control points $\boldsymbol{P}_{ij} = (x_{ij}, y_{ij}, z_{ij})(i, j = 0,1,2,3)$ in bi-cubic control mesh where the points $\boldsymbol{P}_{i0}, \boldsymbol{P}_{i3}, \boldsymbol{P}_{0j}, \boldsymbol{P}_{3j} (i, j = 0,1,2,3)$ are determined by the boundary curves and the left points $\boldsymbol{P}_{11}, \boldsymbol{P}_{12}, \boldsymbol{P}_{21}, \boldsymbol{P}_{22}$ are undetermined. There are 12 variables to be solved by optimization algorithm.

Let four end-to-end curves in a cone as boundary curves and the Coons surface interpolating the boundaries as an initial, and then iterate it. In Fig.3, the result is obtained using the Pseudo-Newtonian method and the order of magnitude of the Gaussian Curvature is 10^{-8}. In Fig.4, the four boundary curves are common smooth curves. There is not certain whether there exists a developable surface interpolating the boundary curves. An approximate has been constructed and the maximum of the Gaussian Curvature is 0.0134. It is trivial that there isn't a developable surface interpolating the four curves.

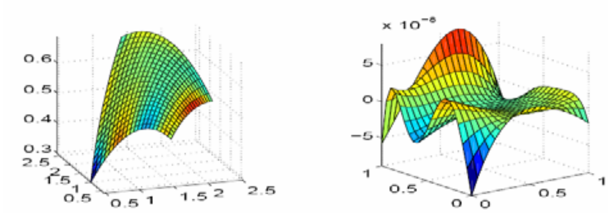

Fig.3 an approximate developable surface in the left, the Gaussian Curvature distribution in the right, $\max\|K\|_{L^2} \approx 10^{-8}$

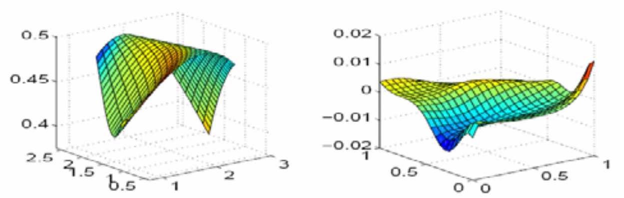

Fig.4 an approximate developable surface in the left, the Gaussian Curvature distribution in the right, $\max\|K\|_{L^2} \approx 10^{-2}$

5. Application

In this section, an approximate developable surface is applied to texture mapping. A surface can be approximated by an approximate developable surface. Then some information in a texture can be defined in the approximate developable surface. Finally, the approximate developable surface maps to an appointed surface. The result has less distortion because the texture information lies in an approximate developable. In Fig.5, texture information is defined in parameter domain and then mapped to the appointed surface. In Fig.6, texture information is defined in the approximated developable surface and then mapped to the appointed surface. The result adopted the latter method has better view effect.

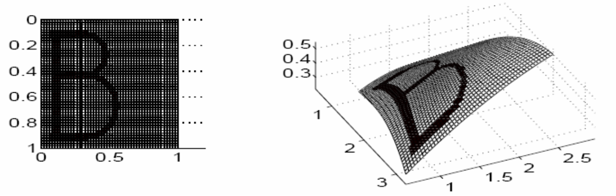

Fig.5 'B' character as texture information defined in the parameter domain in the left, texture is mapped to an appointed surface in the right

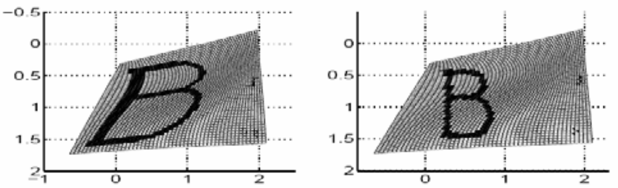

Fig.6 'B' character as texture information defined in the approximate developable surface in the left, texture is mapped to an appointed surface in the right

6. Conclusion

In this paper, the definition of an approximate developable surface is given. Then how to construct a surface using improved Pseudo-Newtonian method is also presented. An important application to texture mapping as an example is depicted. Theory and practice show that the optimization method in this paper for designing an approximate developable surface is simple to compute. This is a new algorithm for surface modelling. In the future, there is still a hard task to construct a developable B-Spline surface interpolating B-Spline curves.

References:

[1] Mancewicz M-J, Frey W-H, Developable surfaces properties, representations and methods of design, *GM Research Publications GMR-7637*, 1992.

[2] Frey W-H., Bindschadler D, Computer aided design of a class of developable Bézier surfaces *GM Research Publications R&D-8057*, 1993.

[3] Pottmann H., Wallner J., *Computational Line Geometry*, Springer, Berlin, 2001.

[4] Chu,C-H, Séquin C-H, Developable Bézier patches properties and design, *Computer-Aided Design*, Vol.34, No.7, 2002, pp:511-527.

[5] Do Carmo, M.P, *Differential Geometry of Curves and Surfaces*, Pearson Education, 1976.

[6] Struik, D.J, *Lectures on Classical Differential Geometry*, Dover, New-York, 1988.

[7] Aumann G., Interpolation with developable Bézier patches, *Computer Aided Geometric Design*, Vol.8, 1991, pp:409-420.

[8] Pottmann H., Wallner J., Approximation algorithms for developable surfaces, *Computer Aided Geometric Design*, Vol.16, 1999, pp:539-566.

[9] Monterde, J., Bézier surfaces of minimal area: The Dirichlet approach, *Computer Aided Geometric Design* Vol.21, 2004, pp:117-136.

[10] Yuan Y.-X, Sun W.-Y, *Theory and Method in Optimization* (in Chinese), Beijing Science Press, January 1997.

[11] Anthony L.P, Francis E.S, J.J.Uhl,Jr, *The Mathematics of Nonlinear Programming*, Springer-Verlag, 1988.

[12] Xu, G.L, Surface fairing and featuring by mean curvature motions, *Journal of Computational and* Applied Mathematics, Vol.163, 2004, pp:295-309.

[13] Hoschek, J., Lasser, D., *Fundamentals of Computer Aided Geometric Design*, Wellesley, MA, 1993.