Computer-Aided Design 38 (2006) 703–712

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Cone spline approximation via fat conic spline fitting

Xunnian Yang *, Weiping Yang

Department of Mathematics, Zhejiang University, Yuquan, Hangzhou 310027, People's Republic of China

Received 20 November 2005; accepted 3 March 2006

Abstract

Fat conic section and fat conic spline are defined. With well established properties of fat conic splines, the problem of approximating a ruled surface by a tangent smooth cone spline can then be changed as the problem of fitting a plane fat curve by a fat conic spline. Moreover, the fitting error between the ruled surface and the cone spline can be estimated explicitly via fat conic spline fitting. An efficient fitting algorithm is also proposed for fat conic spline fitting with controllable tolerances. Several examples about approximation of general developable surfaces or other types of ruled surfaces by cone spline surfaces are presented.

Keywords: Cone spline; Fat conic spline; Ruled surface; Developable surface; Approximation

1. Introduction

Developable surface is a kind of ruled surface that can be mapped to a plane isometrically. From the viewpoint of differential geometry, developable surfaces are composed of general cylinders, general cones, tangent surface of a spatial curve or a composition of these types of surfaces [1]. With well established properties, developable surfaces find wide applications in shape modeling and manufacturing with non-stretchable material such as paper, leather and steel, etc. [2–7]. Within the types of developable surfaces, cones of revolution and cone spline of revolution are of special interests. These surfaces can be represented in parametric form as well as algebraic form. The offsets are the same type of the original surfaces and the development of a cone of revolution into a plane is elementary.

Besides developable surfaces, general types of ruled surfaces play important roles in geometric modeling and shape optimization too. With more degrees of freedom, they can be used to approximate a set of measured data [8,9] or can be used as intermediate approximation to double curved surfaces [10,11]. The main goal of this paper is focusing on the approximation of a ruled surface by a cone spline within prescribed tolerances.

To incorporate developable surfaces into a CAD/CAM system, many researches have been devoted to the representation of developable surfaces in Bézier or B-spline forms. Aumann [12] and Lang and Röschel [13] derived sufficient conditions when a tensor product Bézier surface is developable. Because the tangent plane at every point on a generator of a developable surface are the same, then the developable surface is also the envelope of the set of tangent planes along all generators. With this property, the definition of a developable surface is equivalent to the computation of a curve in duality space. Bodduluri and Ravani [14], Pottmann and Farin [15] have studied developable B-spline surface and developable rational B-spline surfaces using duality methods.

The approximation of a given surface by a developable Bézier or B-spline surface have important application in shape modeling and manufacturing. By representing the fitted planes or points in projective space, Hoscheck and Pottmann [16], Pottmann and Farin [13] presented several methods for developable surface fitting. These methods can be applied easily, but the fitting error cannot be controlled efficiently. Pottmann and Wallner [17] have designed an approximation algorithm under a proper metric in projective space. Recently, Park et al. [18] proposed an optimal control algorithm for developable surface design with which the base curve has to be solved when the directions of the rules are given.

For the purpose of fabrication, ruled surfaces or a general type of developable surfaces are preferred to be approximated by cone spline surfaces or planes. Aumann [2] and Elber [11] proposed to approximate a skew ruled surface by a set of triangles. Though the fitting error can be controlled, but

* Corresponding author. Tel.: +86 571 87951609.
E-mail address: yxn@zju.edu.cn (X. Yang).

0010-4485//$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2006.03.001

the approximating surfaces are not smooth any more. Leopoldseder and Pottmann [19], Leopoldseder [20,21] have developed several interesting algorithms for approximating general types of developable surfaces with cone splines. The algorithms compute the fitting cones using either a Hermite like interpolation technique or space arc spline method. However, this method cannot be extended for the approximation of general types of ruled surfaces directly.

In this paper, we propose an efficient new algorithm for ruled surface approximation with cone splines. Inspired by plane curve approximation with conic splines [22–24], we show that the problem of cone spline approximation can be changed as fat conic spline fitting. For a ruled surface bounded by two parallel planes, when it is projected onto one of these two paralleling planes, we will then obtain a planar fat curve, of which the boundaries are just the projection of the surface boundaries. Similarly, the boundaries of cone spline between two paralleling planes are two conic splines, and the projection of the cone spline is a fat conic spline. Moreover, the fat conic spline can be represented as piecewise rational surfaces and the control polygons of two boundary conic splines are piecewise parallel. As discussed in the following text, the approximation of a planar fat curve by a fat conic spline can be implemented efficiently and the fitting error between two surfaces can be estimated by these two planar fat curves. Finally, the fitting cone spline can be obtained by elevating the boundaries of the fat conic spline onto the original two parallel planes, respectively.

The organization of the paper is as follows. In Section 2, we will give the definition of fat conic arc and fat conic spline. In Section 3, we will present the approximation algorithm. The fitting error will be analyzed in Section 4. Some numerical examples are presented in Section 5 and we conclude the paper in Section 6.

2. Fat conic arc

It is well known that a conic section is the intersection curve between a plane and a cone of revolution (see Fig. 1). The conic section can be an ellipse, a parabola or a hyperbola according to the position that the plane lie with respect to the cone. In the field of computer-aided design, a conic segment can be represented as one or several pieces of rational quadratic Bézier curves and on the contrary any rational quadratic Bézier curve is a conic segment [25,26].

Within this paper, we mean the conic segment by a standard rational quadratic Bézier curve

$$R(t) = \frac{R_0 B_{0,2}(t) + R_1 w B_{1,2}(t) + R_2 B_{2,2}(t)}{B_{0,2}(t) + w B_{1,2}(t) + B_{2,2}(t)}, \quad t \in [0, 1]$$

where $R_i (i=0,1,2)$ are the control points of the Bézier curve, $B_{i,2}(t) = C_2^i t^i (1-t)^{2-i}$ are the Bernstein basis functions of the curve. Even more, the positive weight w can be used to characterize the conic type, the curve is a segment of parabola when the weight $w=1$, and the curve is a segment of ellipse or a segment of hyperbola when w is less than or greater than 1, respectively. Even more, when the control polygon $R_0R_1R_2$ is an isosceles triangle and $w = \cos\theta$ where $\theta = \angle R_0R_1R_2$, then the quadratic rational Bézier curve is a circular arc.

In a similar way as the definition of a conic section, we can now define a fat conic arc geometrically. At first, we can intersect a cone of revolution by two parallel planes, then we have two conic sections on the paralleling planes (see Fig. 2). Let the two conic segments between the two planes and two rulings on the cone be

$$P(t) = \frac{P_0 B_{0,2}(t) + P_1 w B_{1,2}(t) + P_2 B_{2,2}(t)}{B_{0,2}(t) + w B_{1,2}(t) + B_{2,2}(t)}, \quad t \in [0, 1] \qquad (1)$$

and

$$Q(t) = \frac{Q_0 B_{0,2}(t) + Q_1 w B_{1,2}(t) + Q_2 B_{2,2}(t)}{B_{0,2}(t) + w B_{1,2}(t) + B_{2,2}(t)}, \quad t \in [0, 1], \qquad (2)$$

respectively, then the cone surface between the two paralleling planes can be represented as a rational Bézier surface $c(s,t) = (1-s)P(t) + sQ(t)$. When we project the rational Bézier surface onto one of the paralleling planes, we obtain a fat conic arc on the plane (see Fig. 3). For the efficiency of fat conic arc fitting,

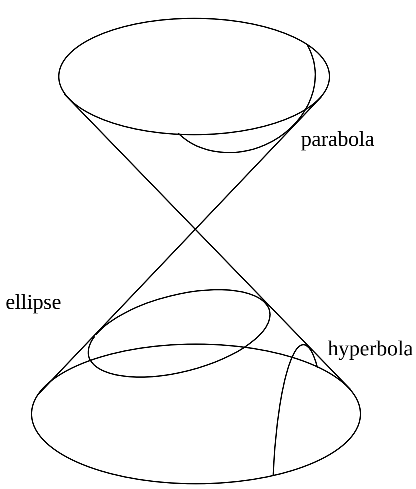

Fig. 1. Conic section.

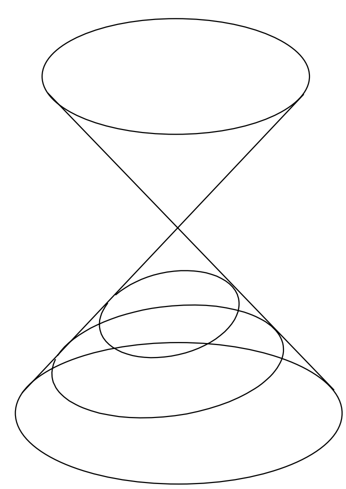
Fig. 2. Fat conic section.

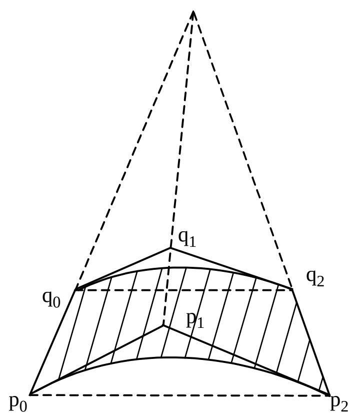

Fig. 3. Control mesh of a fat conic arc.

in this paper we consider only the case that $P(t)$ and $Q(t)$ lie at just one side of the apex of the cone.

Without loss of generality, we assume that the projection plane is just the xy-plane. When the rational Bézier surface $c(s,t)$ has been projected onto the plane, the projected area is bounded by two conic sections $p(t)$, $q(t)$ and two lines p_0q_0 and p_2q_2, where $p(t)$ and $q(t)$ are the projections of $P(t)$ and $Q(t)$, $p_i(i=0,1,2)$ and $q_i(i=0,1,2)$ are the projections of $P_i(i=0,1,2)$ and $Q_i(i=0,1,2)$, respectively.

From the above definition, we know that the rational Bézier surface $c(s,t)$ is a developable surface, then the tangent plane at every point on one generator of the surface are the same. This implies that the points P_0, Q_0, P_1 and Q_1 are on a plane. Similarly, the points P_1, Q_1, P_2 and Q_2 are coplanar too. Moreover, when the surface $c(s,t)$ is lying on a cone, it can be easily verified that the three lines P_0Q_0, P_1Q_1 and P_2Q_2 all pass through the apex of the cone. If the surface $c(s,t)$ lies on a circular cylinder, these three lines are parallel with each other and the intersection point becomes infinity. With this analysis, we have

Theorem 1. *Let $c(s,t)$ be a cone patch bounded by two parallel planes with boundary conics as $P(t)$ and $Q(t)$, then we have $P_0P_1 \| Q_0Q_1$, $P_1P_2 \| Q_1Q_2$ and $P_0P_2 \| Q_0Q_2$.*

From Theorem 1, we know that $\Delta P_0P_1P_2$ and $\Delta Q_0Q_1Q_2$ are similar triangles and the control polygons of two boundary conics are piecewise parallel. As to the plane fat conic arc we have

Theorem 2. *The conic segments $p(t)$ and $q(t)$ are two boundary curves of a fat conic arc if and only if the corresponding weights of these two conics are the same and the control points satisfy the following conditions: $p_0p_1 \| q_0q_1$, $p_1p_2 \| q_1q_2$ and $p_0p_2 \| q_0q_2$.*

Proof. From the definition of the fat conic arc, if $p(t)$ and $q(t)$ are two boundary conics of a fat conic arc, the weights of $p(t)$ and $q(t)$ are equal to the weights of $P(t)$ and $Q(t)$ which are the boundary curves of the original cone surface, then the corresponding weights of $p(t)$ and $q(t)$ are the same. Furthermore, the control polygons of $p(t)$ and $q(t)$ are the projections of $P(t)$ and $Q(t)$, respectively, and from Theorem 1, we have $p_0p_1 \| q_0q_1$, $p_1p_2 \| q_1q_2$ *and* $p_0p_2 \| q_0q_2$. On the other hand, if the corresponding edges of $\Delta p_0p_1p_2$ and $\Delta q_0q_1q_2$ are parallel, then these two triangles are similar, and it can be easily derived that the three lines p_0q_0, p_1q_1 and p_2q_2 either intersect at one common point or are parallel with each other. When we elevate the triangles on two parallel planes with different heights, the two space triangles form the control mesh of a cone patch. If the corresponding weights of $p(t)$ and $q(t)$ are equal and the weights of the cone surface are defined same with those as $p(t)$, then $p(t)$ and $q(t)$ are just the projections of the boundaries of the cone surface bounded by two parallel planes. So, $p(t)$ and $q(t)$ are two boundary curves of a fat conic segment.

The theorem is proven. □

In the same way as the definition of fat conic arcs, the projection of a tangent smooth cone spline surface bounded by two parallel planes forms a fat conic spline. Suppose that none boundary conic arc within a fat conic spline collapses, and from Theorems 1 and 2 we have

Corollary. *The control polygons of two boundary curves of a G^1 fat conic spline are piecewise parallel.*

3. Fat conic spline fitting

For a ruled surface $r(u,v)$ bounded by two parallel planes we will have a planar fat curve by projecting the ruled surface onto a plane. Without loss of generality we assume that the ruled surface is $r(u,v)=(1-u)b_0(v)+ub_1(v)$, where $b_0(v)$ and $b_1(v)$ are two boundary curves lying on the planes $z=h_0$ and $z=h_1$, respectively. We choose the xy-plane as the projection plane, and obtain a plane fat curve by projection. The boundary curves of the plane fat curve are just the projection of $b_0(v)$ and $b_1(v)$ and we still denote the boundaries as $b_0(v)$ and $b_1(v)$ in the following text without special declaration.

To compute a cone spline surface approximating the original ruled surface $r(u,v)$, we should just fit a fat conic spline curve to the projected fat curve of $r(u,v)$. With a plane fat conic spline obtained, we will obtain the approximating cone spline surface by elevating the boundary conic arcs of each fat conic arc onto planes $z=h_0$ and $z=h_1$, respectively. Within the rest of this section, we will pay attention to the approximation of a plane fat curve by a fat conic spline.

3.1. Interpolation by parallel polygons

A planar fat curve is bounded by two curves $b_0(v)$, $b_1(v)$ as well as two straight lines connecting the ends of $b_0(v)$ and $b_1(v)$ at $v=0$ and 1. Because the two straight lines $b_0(0)b_1(0)$ and $b_0(1)b_1(1)$ are determined by $b_0(v)$ and $b_1(v)$, then we mean $b_0(v)$ and $b_1(v)$ when we refer the boundary curves of a fat curve. Instead of constructing the control polygons or boundary conic splines directly, we first interpolate the boundary curves of the original fat curve by two piecewise parallel polygons in this subsection. The construction of boundary conic splines for a fitting fat conic spline will be presented in next subsection.

For two boundary curves $b_0(v)$ and $b_1(v)$ of a fat curve, we can interpolate these two curves by two polygons $L_0L_1\ldots L_n$ and $\bar{L}_0\bar{L}_1\ldots\bar{L}_n$ with pairs of paralleling lines $L_{i-1}L_i\|\bar{L}_{i-1}\bar{L}_i \quad (i=1,2,\ldots,n)$. Because L_i and $\bar{L}_i$ are lying on $b_0(v)$ and $b_1(v)$, respectively, we can then assume that $L_i=b_0(v_i)$ and $\bar{L}_i=b_1(\bar{v}_i)$, where $0=v_0\le v_1\le\ldots\le v_n=1$ and $0=\bar{v}_0\le\bar{v}_1\le\ldots\le\bar{v}_n=1$. When L_i is connected to $\bar{L}_i$ for $i=0,1,\ldots,n$, we will obtain a set of trapezoids or triangles interpolating the original fat curve (see Fig. 4).

To construct two interpolating parallel polygons, we start with the beginning points of curves $b_0(v)$ and $b_1(v)$, i.e. $v_0=0$, $\bar{v}_0=0$ and $L_0=b_0(v_0)$, $\bar{L}_0=b_1(\bar{v}_0)$. After that, we choose another point on one of the boundary curves such as $b_0(v)$ with a default forward step δ. Let the parameter corresponding to the point be v_1, then a new interpolating point is computed as $L_1=b_0(v_1)$. To compute the corresponding point on the curve $b_1(v)$, we should compute the intersection points between the curve $b_1(v)$ and a line through $\bar{L}_0$ while paralleling L_0L_1. We check the solution within the parameter interval $(\bar{v}_0,\bar{v}_0+k_f\delta)$, where k_f is a constant, which can be picked between 1 and 2 for most practical cases. If the solutions exist within the interval, we choose the smallest one for the new point. Then we have $\bar{v}_1>\bar{v}_0$ and $L_{i-1}L_i\|\bar{L}_{i-1}\bar{L}_i$ (see Fig. 4). If there is no solution within the interval, we can just choose $\bar{v}_1=\bar{v}_0$ and the line $\bar{L}_0\bar{L}_1$ collapses to a point.

When we have obtained a pair of parallel lines $L_{i-1}L_i$ and $\bar{L}_{i-1}\bar{L}_i$ interpolating two boundary curves $b_0(v)$ and $b_1(v)$, we can then choose the end points L_i and $\bar{L}_i$ as the beginning points of next pair of interpolating parallel lines. Again we compute a line interpolating the curve $b_0(v)$ or $b_1(v)$ firstly for new parallel line computation, and a default rule for curve selection is that we choose a curve with smaller parameter v_i or $\bar{v}_i$. This choice will make the generators of the fat conic arc more compatible with those of the original fat curve. Whether L_i and $\bar{L}_i$ can be accepted or should be recomputed depends on the fitting accuracy of a corresponding cone surface. This procedure can be implemented independently or along with cone spline fitting procedure, and will end until the last two end points of $b_0(v)$ and $b_1(v)$ have been interpolated.

3.2. Construction of boundary conic splines

With a set of parallel lines interpolating two boundaries of the original fat curve, every line segment can be considered as the chord of an interpolating conic arc. To determine a conic arc fitting one of the original boundary curves, we should construct its control polygon and compute weights for all control points.

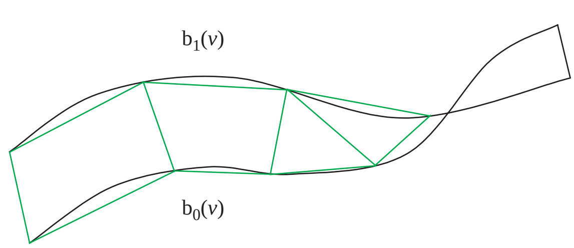

Fig. 4. Interpolating a fat curve by a pair of parallel polygons.

Let $L_{i-1}L_i$ and $\bar{L}_{i-1}\bar{L}_i$ $(i=1,2,\ldots,n)$ be the set of parallel lines interpolating the boundary curves $b_0(v)$ and $b_1(v)$, we should compute a common tangent direction T_i at L_i or $\bar{L}_i$ for $i=1,2,\ldots,n$. To guarantee the existence of conic segments between every two consecutive interpolation points, we compute the tangent directions according to the following two rules. The first rule is local convexity. Assume that the chord $L_{i-1}L_i$ does not collapse, the angle from T_{i-1} to the chord and the angle from the chord to T_i are with the same sign. Moreover the intersection point of the tangent line along T_{i-1} and the tangent line along T_i exists and lies at the same side with T_{i-1} to the chord $L_{i-1}L_i$. Secondly, the tangent T_i should also be close to the derivative direction of $b_0'(v_i)$ and $b_0'(\bar{v}_i)$ simultaneously so that the final approximating conic sections are close to the boundary curves. When local convexity property holds, it means the existence of a conic arc interpolating the ends as well as the end tangents at $L_{i-1}L_i$.

To compute the tangents satisfying the above two rules, we can just compute the tangents sequently. A default tangent direction at L_i and $\bar{L}_i$ is computed as the average of derivatives at these two points

$$V_i=\frac{b_0'(v_i)+b_1'(\bar{v}_i)}{\|b_0'(v_i)+b_1'(\bar{v}_i)\|}. \tag{3}$$

If $i=0$ we choose $T_i=V_i$. For $i>0$, if $L_{i-1}L_i$ or $\bar{L}_{i-1}\bar{L}_i$ collapses, we compute vector T_i at $\bar{L}_i$ or L_i. In the following paragraph we compute T_i at L_i under the assumption that $L_{i-1}L_i$ does not collapse. If T_{i-1}, $L_{i-1}L_i$ and V_i satisfy the local convexity condition we choose $T_i=V_i$. If these three vectors do not satisfy the local convexity property, there are two cases, which should be dealt independently. The first case is that the angle from T_{i-1} to $L_{i-1}L_i$ and the angle from $L_{i-1}L_i$ to V_i have different signs. The second case is that even though the above two angles have same sign but two tangent lines through chord ends do not intersect or the intersection point lies on an unexpected side. If the second case holds, we can disturb vector V_i so that the disturbed V_i satisfies the local convexity condition. Then the disturbed vector will be picked as T_i. If the first case holds we reflect the initial V_i with respect to the chord $L_{i-1}L_i$. Let W_i be the reflected vector we have

$$W_i=\frac{2(V_iU_i)U_i-V_i}{\|2(V_iU_i)U_i-V_i\|}, \tag{4}$$

where $U_i=L_i-L_{i-1}/\|L_i-L_{i-1}\|$. For most practical cases T_{i-1}, $L_{i-1}L_i$ and W_i will satisfy the local convexity property, and then we can choose $T_i=W_i$. In case W_i still cannot satisfy the requirement it should be disturbed further as in second case and a disturbed vector will be picked as T_i.

With all the tangent directions at the interpolation points properly defined, we can compute the intersection point R_i between the line through the point L_{i-1} with direction T_{i-1} and the line through L_i with direction T_i. In a similar way, we compute the intersection point $\bar{R}_i$ between the line through the point $\bar{L}_{i-1}$ with direction T_{i-1} and the line through $\bar{L}_i$ with

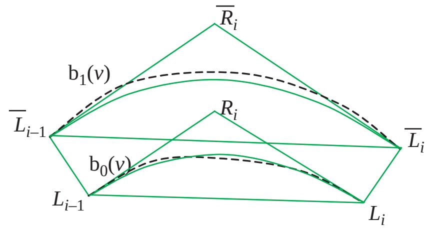

Fig. 5. Construction of a fat conic arc. The dashed curves are boundaries of a fat curve and the solid curves are of a fat conic arc.

direction T_i. If the positions of L_{i-1} and L_i are different, then the points L_{i-1}, R_i and L_i form the control polygon of a fitting conic segment. Similarly, $\bar{L}_{i-1}$, $\bar{R}_i$ and $\bar{L}_i$ form the control polygon of another conic arc when two points $\bar{L}_{i-1}$ and $\bar{L}_i$ are different from each other. If none of these two control polygons collapses to a point, the two triangles $\Delta L_{i-1}R_iL_i$ and $\Delta\bar{L}_{i-1}\bar{R}_i\bar{L}_i$ are similar triangles. From Theorem 2, the fat curve defined by these two conic segments with consistent weights is a fat conic arc (see Fig. 5).

With the control polygon obtained for each conic arc, we compute the weights for these conic arcs by optimal curve fitting. Suppose that all conic segments are in standard form, we should then just compute the weight corresponding to each middle control point. As illustrated in Fig. 5, $L_{i-1}R_iL_i$ and $\bar{L}_{i-1}\bar{R}_i\bar{L}_i$ are the control polygons of a pair of conic arcs $A_i(t)$ and $B_i(t)$. The weights corresponding to the middle control points R_i and R_i are both w_i, and we have these two conic arcs as

$$A_i(t) = \frac{L_{i-1}B_{0,2}(t) + R_iw_iB_{1,2}(t) + L_iB_{2,2}(t)}{B_{0,2}(t) + w_iB_{1,2}(t) + B_{2,2}(t)}, \quad t \in [0,1]$$

and

$$B_i(t) = \frac{L_{i-1}B_{0,2}(t) + R_iw_iB_{1,2}(t) + \bar{L}_iB_{2,2}(t)}{B_{0,2}(t) + w_iB_{1,2}(t) + B_{2,2}(t)}, \quad t \in [0,1]$$

The weight w_i should be computed so that both of conics $A_i(t)$ and $B_i(t)$ are as close as possible to $b_0(v)$ and $b_1(v)$, respectively.

Floater [23] has presented an error estimation formula for conic approximation. The main idea of Floater's method is to represent the conic segment in implicit form and compute the bound of a point to the implicit curve. Let point $b_0(v)$ be a point within the triangle $\Delta L_{i-1}R_iL_i$, and τ_0, τ_1 and τ_2 are the barycentric coordinates of the point with respect to the triangle, then one can define an algebraic function as $f_w(b_0(v)) = 4w_i^2\tau_0\tau_2 - \tau_1^2$. The point $b_0(v)$ lies on the conic segment $A_i(t)$ if and only if $f_w(b_0(v)) = 0$. In a similar way, if point $b_1(v)$ lies within the triangle $\Delta L_{i-1}R_iL_i$, one can define another function $f_w(b_1(v))$ by using the barycentric coordinates. The weight w_i can be computed under the assumption that the sum of $f_w(b_0(v))$ and $f_w(b_1(v))$ are zero for all those points lying within either of the two triangles. By denoting $\Delta L_{i-1}R_iL_i$ and $\Delta\bar{L}_{i-1}\bar{R}_i\bar{L}_i$ as Δ_i and $\bar{\Delta}_i$, respectively, we have

$$\int_{b_0(v)\in\Delta_i} f_w(b_0(v))\mathrm{d}v + \int_{b_1(v)\in\bar{\Delta}_i} f_w(b_1(v))\mathrm{d}v = 0. \quad (5)$$

Though the solution exists theoretically for Eq. (5), but it is difficult to solve the equation exactly. For practical computation, we can just sample several points within the triangles from the original two boundary curves $b_0(v)$ and $b_1(v)$ and solve the discrete equation directly.

3.3. The fitting algorithm

To construct a cone spline surface within a prescribed tolerance, we should compute the fitting cone spline globally or sequently. For global approach, we can construct an initial approximating cone spline surface using a default forward step. And then we pick segments with larger fitting error on the initial fat curve and refit the corresponding fat curve segments with more fat conic arcs. As for the sequent method we compute and test every new cone patch, and a new patch will be joined to current cone spline only when it is within the prescribed tolerance. If the fitting error is larger than a predefined criterion, we can just shorten the forward step δ by a factor k_s and compute another cone surface. This procedure can be repeated until a cone patch with the given tolerance has been found. In this way, we obtain a cone spline within a given tolerance sequently. In this paper, we adopt the second method for cone spline surface approximation.

The fat conic spline fitting algorithm for cone spline approximation within a given tolerance is summarized as follows.

Fat conic spline fitting algorithm.
Input ($r(u,v)$, tol, δ_0)
Output ($c_i(s,t)$, $i = 1,2,\ldots,n$)
$v_0 = \bar{v}_0 = 0$, $L_0 = b_0(v_0)$, $\bar{L}_0 = b_1(\bar{v}_0)$;
Compute T_0;
$i = 1$; set $\delta = \delta_0$;
while ($v_{i-1} < 1$ or $\bar{v}_{i-1} < 1$) {
  Compute v_i, $\bar{v}_i$, L_i and $\bar{L}_i$;
  Compute T_i;
  Compute $A_i(t)$ and $B_i(t)$;
  Compute fitting error err($A_i(t)$, $B_i(t)$, $r(u,v)$);
  if ($err <$ tol) {output $c_i(s,t)$; i++; reset $\delta = \delta_0$;}
  else $\delta = k_s\delta$;}

4. Error estimation for cone spline approximation

In previous sections, we have shown that the approximation of a ruled surface by a cone spline surface can be reduced to the problem of fitting a fat plane curve with a fat conic spline curve. In Section 4, we will show how to compute the fitting error for cone spline approximation via fat conic spline fitting.

The fitting error between a ruled surface and a cone spline is usually consisting of two parts: boundary error and skew error. As to the boundary error, we notice that the distance from

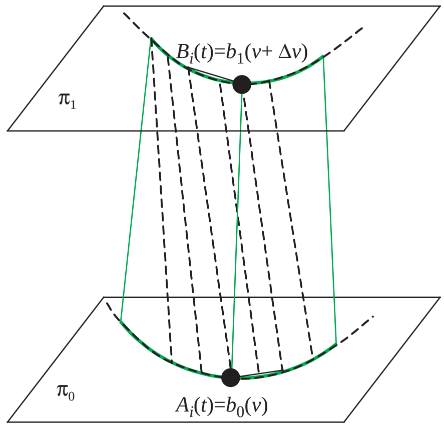

Fig. 6. The skew error between the ruled surface and the cone surface. The dashed curves are boundaries of a ruled surface while the solid curves are of a cone spline.

a boundary curve of one surface to the other one is no larger than the distance between two corresponding boundary curves on a projection plane. Then, we will compute the boundary error between a ruled surface and a cone spline as the distance between corresponding boundary curves of two projected fat curves. On another hand, even though the boundary error vanishes, there may still exists a gap between the original ruled surface and the fitting cone spline surface because of the inconsistency between the projected generators from these two surfaces. We refer this type of fitting error as skew error (see Fig. 6).

As introduced in Section 3, $A_i(t)$ and $B_i(t)$, $t\in[0,1]$ are two boundary conics of a fat conic segment interpolating the original fat curve with boundaries $b_0(v)$ and $b_1(v)$ (see Fig. 5). The distance from $A_i(t)$ to $b_0(v)$ and the distance from $b_0(v)$ to $A_i(t)$ can be defined as

$$d_H(A_i(t), b_0(v)) = \max_{t\in[0,1]} \min_{v\in[v_{i-1},v_i]} \text{dist}(A_i(t), b_0(v))$$

and

$$d_H(b_0(v), A_i(t)) = \max_{v\in[v_{i-1},v_i]} \min_{t\in[0,1]} \text{dist}(b_0(v), A_i(t)),$$

respectively. Then the Hausdorf distance between $A_i(t)$ and $b_0(v)$ is defined as

$$D_H(A_i(t), b_0(v)) = \max\{d_H(A_i(t), b_0(v)), d_H(b_0(v), A_i(t))\}. \quad (6)$$

By replacing $A_i(t)$ with $B_i(t)$, $b_0(v)$ with $b_1(v)$ and the interval $[v_i-1,v_i]$ with $[\bar{v}_{i-1}, \bar{v}_i]$ we have the definition of Hausdorf distance $D_H(B_i(t), b_1(v))$ between $B_i(t)$ and $b_1(v)$.

It should be noted that the distance from a point $b_0(v)$ to the conic section $A_i(t)$ can be estimated explicitly when $b_0(v)$ lies in the triangle formed by the control polygon of $A_i(t)$ (see Ref. [23]). However, not all points on the original boundary curves satisfy this condition in practice. Then we should compute the distance between two curves numerically, i.e. by computing the maximum distance of densely sampled points from $b_0(v)$ to the interpolating conic arc $A_i(t)$. By the same way we compute the distance from curve $b_1(v)$ to $B_i(t)$.

Besides boundary error, we should also compute the interior deviation between the ruled surface and the cone surface. In the following text, we compute the distance from the ruled surface to the cone surface and the distance from the cone surface to the ruled surface under the assumption that the boundary error vanishes. Moreover, we will show that both of these two types of distances can be reduced to the distance computation from a line to a surface.

Let $A_i(t)B_i(t)$ be a generator on the cone surface, the two end points of the line lie on two paralleling planes, respectively, we compute the bound of the distance from line $A_i(t)B_i(t)$ to the ruled surface $r(u,v)$ (see Fig. 6). Assume that points $A_i(t)$ and $B_i(t)$ also lie on the ruled surface, i.e. there exist v and Δv such that $A_i(t)=b_0(v)$ and $B_i(t)=b_1(v+\Delta v)$ hold, and the four end points of generators $b_0(v)b_1(v)$ and $b_0(v+\Delta v)b_1(v+\Delta v)$ of the ruled surface form a bilinear surface. To compute the bound of the distance from the line $A_i(t)B_i(t)$ to the original ruled surface, we compute the distance from the line $A_i(t)B_i(t)$ to the bilinear surface and the distance from the bilinear surface to the ruled surface. Firstly, we define the thickness $h(v)$ of the bilinear surface as the distance between two diagonals of the quadrangle defined by $b_0(v)b_1(v)$ and $b_0(v+\Delta v)b_1(v+\Delta v)$. Then the distance from the line $A_i(t)B_i(t)$ to the bilinear surface is bounded by $h(v)$. It is clear that $h(v)$ vanishes just when this quadrangle is on a plane. Let $L_0(v)$ be the line connecting $b_0(v)$ and $b_0(v+\Delta v)$, and $L_1(v)$ is the line connecting $b_1(v)$ and $b_1(v+\Delta v)$, then the bound of the distance from the line $L_0(v)$ to the curve $b_0(v)$ and the bound of distance from the line $L_1(v)$ to the curve $b_1(v)$ can be computed as follows [27]:

$$d_H(b_0(v), L_0(v)) \le e_0(v) = \frac{1}{8}\Delta v^2 \sup_v \|b_0''(v)\|,$$

$$d_H(b_1(v), L_1(v)) \le e_1(v) = \frac{1}{8}\Delta v^2 \sup_v \|b_1''(v)\|.$$

At this time, the error from the bilinear surface to the ruled surface is bounded by $\max(e_0(v),e_1(v))$. Finally we have the error bound from the line $A_i(t)B_i(t)$ to the ruled surface as

$$\mathrm{d}(v) = \max(e_0(v), e_1(v)) + h(v). \quad (7)$$

As to the distance from a line $b_0(v)b_1(v)$ on the ruled surface to the cone spline surface, it can be computed in a similar way. Without loss of generality, we assume that $b_0(v)=A_i(t)$ and $b_1(v)=B_i(t-\Delta t)$, the fitting error can also be divided into two components. The first part of fitting error is due to the inconsistency of parameters, and the error bounds on either boundary curves are as

$$d_H(A_i(t), \bar{L}_0(t)) \le \bar{e}_0(t) = \frac{1}{8}\Delta t^2 \sup_t \|A_i''(t)\|,$$

$$d_H(B_i(t), \bar{L}_1(t)) \le \bar{e}_1(t) = \frac{1}{8}\Delta t^2 \sup_t \|B_i''(t)\|,$$

where $\bar{L}_0(t)$ is a line connecting two points $A_i(t)$ and $A_i(t-\Delta t)$, $\bar{L}_1(t)$ is a line connecting two points $B_i(t)$ and $B_i(t-\Delta t)$. Let $\bar{h}(t)$ be the distance between two lines

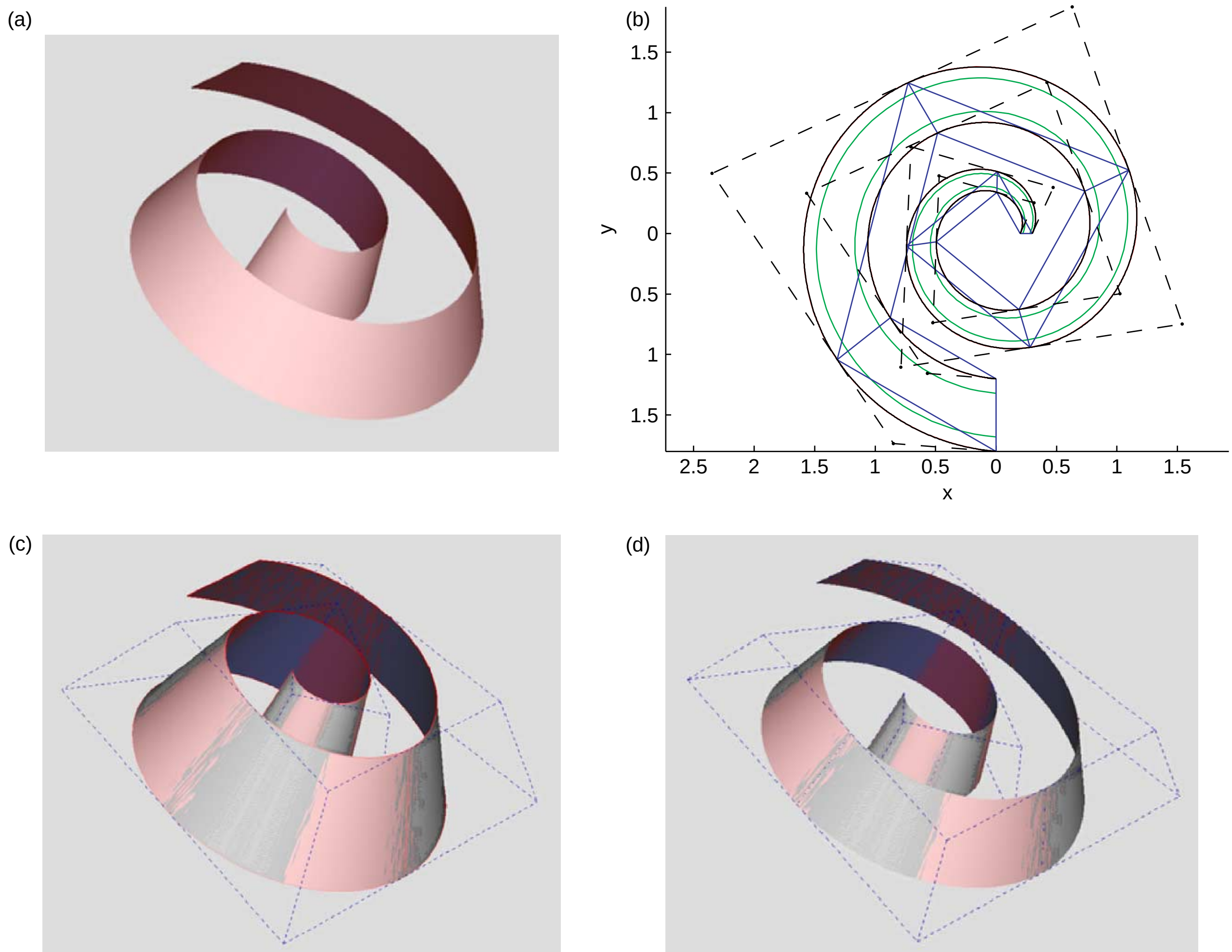

Fig. 7. Approximating a generalized cone surface by cone spline surface: (a) the original surface; (b) fitting the extended fat curve by a fat conic spline; (c) the extended fitting cone spline surface with control mesh; (d) the approximating cone spline surface.

$A_i(t-\Delta t)B_i(t)$ and $A_i(t)B_i(t-\Delta t)$, then the skew error from the line $b_0(v)b_1(v)$ to the cone surface is bounded by

$$\bar{d}(t) = \max(\bar{e}_0(t), \bar{e}_1(t)) + \bar{h}(t). \tag{8}$$

The total bound for the distance between the original ruled surface and the fitting cone spline surface can now be computed as

$$\max\{D_{\mathrm{H}}(A_i, b_0), D_{\mathrm{H}}(B_i, b_1)\} + \max(d_0, d_1), \tag{9}$$

where $d_0 = \sup_v d(v)$ and $d_1 = \sup_t \bar{d}(t)$.

5. Implementation and examples

As discussed in Section 3, when we interpolate a plane fat curve by two polygons with piecewise parallel lines, some line segments may collapse to points, and the trapezoids defined by parallel lines will degenerate to triangles (see Fig. 4). Though a fat conic section can still be defined for each trapezoid or triangle, but for collapsed lines the boundary conic arcs collapse to points, and the fitting conic arcs join with only C^0 continuity at the degenerate points. If the boundaries of a planar fat conic spline are not smooth, the corresponding cone spline surface will have no tangent plane at the degenerate boundary points. To obtain a tangent smooth cone spline surface in the end, we should fit the boundary curves of original fat curve by fat conic spline with smooth boundaries.

To remove boundary singularities of a cone spline surface bounded by two parallel planes, one can just trim the cone spline surface with two new parallel planes lying between the original ones. Then, if we want to construct a tangent continuous cone spline surface we can extend the original ruled surface in either directions and trim the fitting cone spline surface with original two parallel planes. Assume that $z=h_0$ and h_1 be two parallel bounding planes, we extend the ruled surface to two new parallel planes $z=\bar{h}_0$ and $\bar{h}_1$, where $\bar{h}_0\langle h_0$ and $\bar{h}_1\rangle h_1$. In the following examples, we choose $h_0=(4/3)h_0-(h_1/3)$ and $h_1=(4/3)h_1-(h_0/3)$. Let $\bar{p}_i$ and $\bar{q}_i$ be the control points of boundary conic arcs of an extended fat conic arc, the control points of the final fat conic spline can be obtained as

$$p_i = \frac{\bar{h}_1 - h_0}{\bar{h}_1 - \bar{h}_0}\bar{p}_i + \frac{h_0 - \bar{h}_0}{\bar{h}_1 - \bar{h}_0}\bar{q}_i,$$

$$q_i = \frac{h_1 - \bar{h}_0}{\bar{h}_1 - \bar{h}_0}\bar{p}_i + \frac{\bar{h}_1 - h_1}{\bar{h}_1 - \bar{h}_0}\bar{q}_i.$$

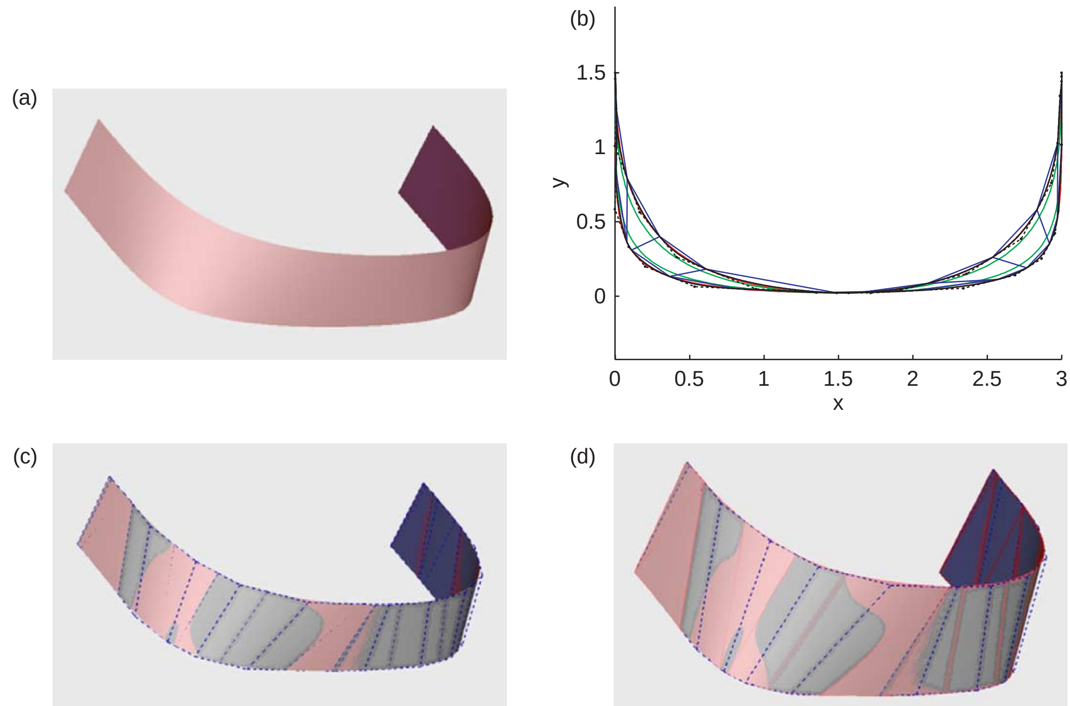

Fig. 8. Rule surface approximation by cone spline surface: (a) the original ruled surface; (b) the extended fitting cone spline surface; (c) the extended fitting cone spline surface; (d) the cone spline surface fitting the original surface.

Within the rest of this section we present several interesting examples concerning the approximation of ruled surfaces by cone spline surfaces. To be consistent with the fitting algorithm we use $b_0(v)$ and $b_1(v)$ to stand for the boundaries of extended fat curves while the boundaries of original fat curves are just $0.8b_0(v)+0.2b_1(v)$ and $0.2b_0(v)+0.8b_1(v)$. We shorten the forward step δ with 0.9δ when the fitting error of a cone patch is larger than a predefined criterion. In Figs. 7–9, surfaces with grey color are the fitting cone spline surfaces while the red ones are the original ruled surfaces. For plane fat curve fitting, the outside curves are the boundaries of extended fat curves (light color) and the fitting conic splines (deep color) while the interior curves are boundaries of original fat curves. The dashed polygons are control polygons of the fitting conic splines and the solid polygons are interpolating parallel polygons.

In the first example, we approximate a section of a generalized cone surface by a spline of cone surface of revolution (see Fig. 7). The boundary curves of the original ruled surface are two spirals, and the extended boundaries are also spirals which are $b_0(v)=((0.2+v)\cos(3.5\pi v)),(0.2+v)\sin(3.5\pi v)$ and $b_1(v)=b_0(v)\times 1.5$ (see Fig. 7(b)). When the height between two bounding planes of the extended ruled surface is 1 and the permission tolerance is 0.001, we obtain an extended cone spline with seven patches (Fig. 7(c)). The cone spline surface fitting the original ruled surface can be obtained by trimming the extended cone spline surface (Fig. 7(d)). For this example no boundary conic arc collapses even for the extended fat conic spline, and the weights w_is for these fat conic arcs are 0.6444, 0.6153, 0.6318, 0.6389, 0.67, 0.6487 and 0.8992, respectively.

In the second example, we approximate a non-developable ruled surface by cone spline surface (see Fig. 8). The boundaries of the original ruled surface are two Bézier curves of degree 9. Then the extended boundary curves $b_0(v)$ and $b_1(v)$ are also Bézier curves. The control points for $b_0(v)$ are (0,1.5), (0,1.5), (0,0), (0,0), (1,−0.075), (1,−0.075), (3,0), (3,0), (3,1.5), (3,1.5) and the control points for $b_1(v)$ are (0,1.5), (0,0.5), (0,0), (0,0), (0.5,0), (2.5,0), (3,0), (3,0), (3,0.5), (3,1.5). The height for the extended ruled surface is 1 and the given tolerance is 0.06 for this example. Though $b_0(v)$ and $b_1(v)$ coincide at two end points, a plane fat conic spline curve with 16 pieces has been constructed fitting this extended fat curve (Fig. 8(b)). When some interpolating trapezoids degenerate to triangles for plane curve fitting, there exist some singularities at the boundaries of the fitting cone spline surface (Fig. 8(c)). Finally, a tangent smooth cone spline surface is obtained by trimming the extended cone spline surface (Fig. 8(d)). The maximum fitting error is 0.0545 and the weights w_is for the fat conic spline are 6.1257, 2, 0.6376, 1.3328, 1.0636, 1.0342, 7.7169, 1.1272, 1.0006, 1.0322, 1.0251, 1.0541, 60.9157, 2, 2.2879, 2. When $b_0(v)$ and $b_1(v)$ lie outside of the triangles formed by the control polygons of a fat conic arc, the corresponding weight for this fat arc is set a default value 2.

In the third example, we approximate another bounded ruled surface with height 1.8 by a cone spline surface. The height of extended ruled surface is 3 and the extended boundary curve $b_0(v)$ is a cubic Bézier curve with control points (−2,−2), (4,8), (5,−8) and (12,1). The extended boundary $b_1(v)=b_0(v)+p(v)$, where $p(v)=3(\tan[\pi/8\cos(\pi v)]\cos[\pi/2+0.1$

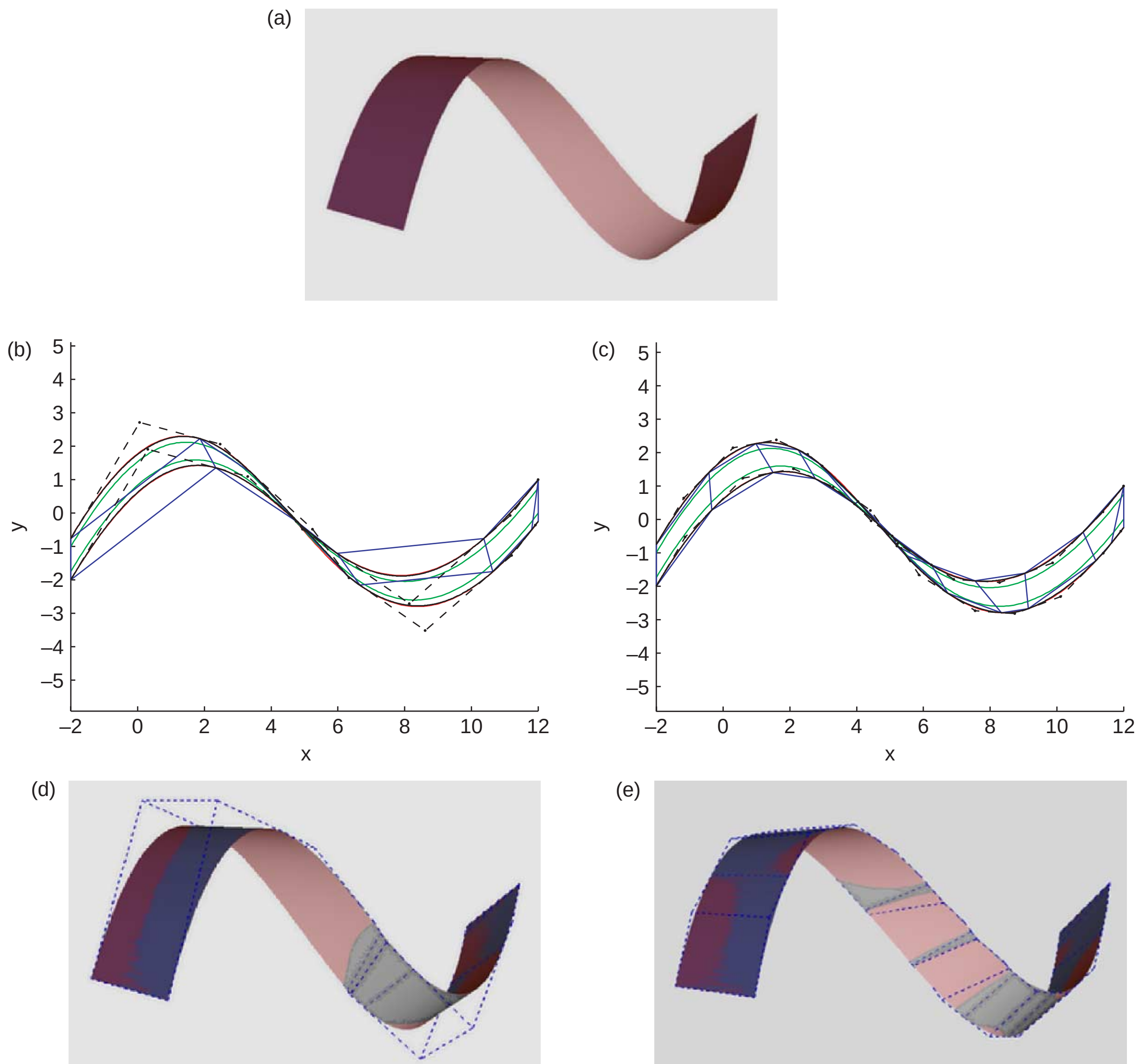

Fig. 9. Approximating a ruled surface by cone spline surfaces: (a) the original ruled surface between two parallel planes; (b) plane fat curve fitting with a larger tolerance; (c) plane fat curve fitting with a smaller tolerance; (d) the approximating cone spline surface under (b); (e) the approximating cone spline surface under (c).

$\sin(\pi v)]$, $\tan[\pi/8\cos(\pi v)]\sin[\pi/2+0.1\sin(\pi v)])$. We apply two different forward steps and fitting tolerances for this example. With forward step 0.3 and permission tolerance 0.2, we obtain a fat conic spline with seven fat arcs (Fig. 9(b)). A fat conic spline with 15 fat arcs is obtained by choosing forward step 0.1 and permission tolerance 0.1 (Fig. 9(c)). The fitting cone spline surfaces under different forward steps and tolerances are illustrated in Fig. 9(d) and (e), respectively. The weight w_is for the first fat conic spline are 1.0574, 1.1139, 0.0737, 0.3805, 1.0682, 1.0301, 2.0 and the weights for the second fat conic spline are 1.0031, 1.0299, 1.0043, 1.1244, 22.7854, 0.0048, 0.0166, 0.5616, 2, 0.3319, 1.0078, 1.1335, 1.0143, 1.1865, 2.

Table 1

Example	Figure	Forward step	Tolerance	Max error	Patch number
1	7	0.5	0.001	9.717×10^{-4}	7
2	8	0.1	0.06	0.0545	16
3a	9	0.3	0.2	0.1992	7
3b	9	0.1	0.1	0.0998	15

The forward steps, tolerances, max fitting error and patch numbers for the above mentioned examples are listed in Table 1.

6. Conclusion and discussions

In this paper, we have presented a new method for approximating a ruled surface by cone spline surfaces within prescribed tolerances. Assume that the original ruled surface lies between two parallel planes, we have a plane fat curve by projecting the ruled surface onto a plane paralleling the original bounding planes. On the selected plane, we fit the projected fat curve by a fat conic spline. The fitting error for cone surface approximation can be estimated efficiently via fat conic spline approximation.

The assumption that a ruled surface to be approximated is bounded by two paralleling planes may not be valid for general types of ruled surfaces. If this case do occur, we can just divide the original ruled surface into two or several sub-ruled surfaces of which every sub-ruled surface can be bounded by a pair of paralleling planes. If one want to approximate a double curved surface by a cone spline surface, a ruled surface can first be constructed fitting the original surface, and then a developable surface is obtained by approximating the ruled surface.

Acknowledgements

We thank anonymous referees for their helpful comments and suggestions on an early version of the paper. This work is supported by 973 program (2002CB312101) and Natural Science Foundation of China (60303015,60333010).

References

[1] do Carmo MP. Differential geometry of curves and surfaces. New Jersy: Prentice Hall; 1976.
[2] Aumann G. Approximate development of skew ruled surfaces. Comput Graph 1989;13(3):361–6.
[3] Redont P. Representation and deformation of developable surfaces. Comput Aided Des 1989;21(1):13–20.
[4] Randrup T. Approximation of surfaces by cylinders. Comput Aided Des 1998;30(10):807–12.
[5] Chu CH, Sequin CH. Developable Bézier patches: properties and design. Comput Aided Des 2002;34(7):511–27.
[6] Maekawa T, Chalfant J. Design and tessellation of B-spline developable surfaces. J Mech Des Trans ASME 1998;120(3):453–61.
[7] Frey WH. Modeling buckled developable surfaces by triangulation. Comput Aided Des 2004;36(4):299–313.
[8] Chen HY, Pottmann H. Approximation by ruled surfaces. J Comput Appl Math 1999;102:143–56.
[9] Chen HY, Lee IK, Leopoldseder S, Pottmann H, Randrup T, Wallner J. On surface approximation using developable surfaces. Graph Models Image Process 1999;61:110–24.
[10] Hoscheck J, Schwanecke U. Interpolation and approximation with ruled surfaces. In: Cripps R, editor. The mathematics of surfaces VIII. Birmingham: Information Geometers; 1998. p. 213–31.
[11] Elber G. Model fabrication using layout projection. Comput Aided Des 1995;27(4):283–91.
[12] Aumann G. Interpolation with developable Bézier patches. Comput Aided Geom Des 1991;8(5):409–20.
[13] Lang J, Röschel O. Developable (1,*n*)-Bézier surfaces. Comput Aided Geom Des 1992;9:291–8.
[14] Bodduluri RMC, Ravani B. Design of developable surfaces using duality between plane and point geometries. Comput Aided Des 1993;25(10): 621–32.
[15] Pottmann H, Farin G. Developable rational Bézier and B-spline surfaces. Comput Aided Geom Des 1995;12:513–31.
[16] Hoscheck J, Pottmann H. Interpolation and approximation with developable B-spline surfaces. In: Daehlen M, Lyche T, Schumaker L, editors. Mathematical methods in CAGD III. Vanderbilt University Press; 1995.
[17] Pottmann H, Wallner J. Approximation algorithms for developable surfaces. Comput Aided Geom Des 1999;16(6):539–56.
[18] Park FC, Yu J, Chun C, Ravani B. Design of developable surfaces using optimal control. J Mech Des Trans ASME 2002;124(4): 602–8.
[19] Leopoldseder S, Pottmann H. Approximation of developable surfaces with cone spline surfaces. Comput Aided Des 1998;30:571–82.
[20] Leopoldseder S. Algorithms on cone spline surfaces and spatial osculating arc splines. Comput Aided Geom Des 2001;18:505–30.
[21] Leopoldseder S. Cone spline surfaces and spatial osculating arc splines—a sphere geometric approach. Adv Comput Math 2002;17:49–66.
[22] Ahn YJ. Conic approximation of planar curves. Comput Aided Des 2001; 33(12):867–72.
[23] Floater M. An $O(h^{2n})$ Hermite approximation for conic sections. Comput Aided Geom Des 1997;14:135–51.
[24] Yang X. Curve fitting and fairing using conic splines. Comput Aided Des 2004;36(5):461–72.
[25] Lee ET. The rational Bézier representation for conics, Geometric modeling: algorithms and new trends. SIAM. Academic Press, PA; 1987. p. 3–19.
[26] Farin G. From conics to NURBS: a tutorial and survey. IEEE Comput Graph Appl 1992;12(5):78–86.
[27] Filip D, Magedson R, Markot R. Surface algorithms using bounds on derivatives. Comput Aided Geom Des 1986;3:295–311.

Xunnian Yang is now an associate professor in the department of mathematics and institute of Computer Graphics and Image Processing at Zhejiang University, China. He obtained a BS in applied mathematics from Anhui University and a PhD in CAGD and Computer graphics from Zhejiang University in 1993 and 1998, respectively. His research interests include geometric design and processing, computer graphics and image processing.

Weiping Yang got a bachelor's degree of science from Zhejiang University, China in 2003. He is now a graduate student in the Department of Mathematics at Zhejiang University, and his research interests are computer graphics and geometric modeling.