J. Computer-Aided Design. 38 (10) pp. 1113-1122, 2006.

Progressive Surface Modeling Scheme from Unorganised Curves

Shengfeng Qin, and David K Wright

School of Engineering and Design, Brunel University

Uxbridge, Middlesex UB8 3PH, UK

Abstract: - This paper presents a novel surface modelling scheme to construct a freeform surface progressively from unorganised curves representing the boundary and interior characteristic curves. The approach can construct a base surface model from four ordinary or composite boundary curves and support incremental surface updating from interior characteristic curves, some of which may not be on the final surface. The base surface is first constructed as a regular Coons surface and upon receiving an interior curve sketch, it is then updated. With this progressive modelling scheme, a final surface with multiple sub-surfaces can be obtained from a set of unorganised curves and transferred to commercial surface modelling software for detailed modification. The approach has been tested with examples based on 3D motion sketches; it is capable of dealing with unorganised design curves for surface modelling in conceptual design. Its limitations have been discussed.

Key-Words: - **progressive surface modeling, unorganised curves, boundary curves, interior curves, conceptual design, motion sketch.**

1 Introduction

In 3D form design, styling or industrial design, designers [1] often make 3D mock-ups such as clay models for sculpting or 3D construction references (e.g., frames) for surfacing. They can even sketch them out in 3D with the 3D references [2] just as in 2D sketching [3].

Some virtual 3D sketching systems such as 3-Draw [4], HoloSketch [5], Surface Drawing [6], have been developed to support conceptual design, they use different 3D input devices such as Responsive Workbench [5] to create 3D curves or surface strips [6]. In general, they have difficulty in making sketch strokes touch (or intersect) due to the complexity of 3D stroke boundaries and the noticeable error of the tracker system. On the other hand, in a typical iterative design process, designers may input 3D strokes over earlier ones to locally improve the surface. This means that not all sketched strokes are confined to the same surface. Hence, there is a challenging demand on surface modelling progressively from unorganised design curves.

In order to support large-sized freeform surface design, we have developed a motion-based 3D sketch method [2] as shown in Fig 1. With an optical motion capture system [7], artists or designers wearing reflective motion marks on their hands can sketch out 3D design splines when moving their body and hand in space. The sketching process can be switched on or off by covering and uncovering the motion marks. Designers can literarily combine their body and hands movements within a predefined motion capturing volume to freely perform their design tasks to any positions. During sketching, designers can use the working ground as the X and Y directional references and their body parts for the Z direction. They can also use rough physical wireframes for 3D guidance. All their design movements are recorded in the motion capturing system and transferred into corresponding 3D curves. Thus, a large-sized freeform surface based on sketched 3D curves can be obtained.

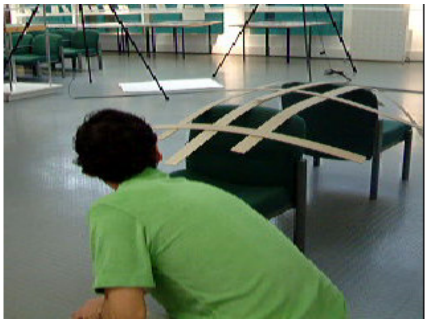


Fig. 1. Motion-based sketch system

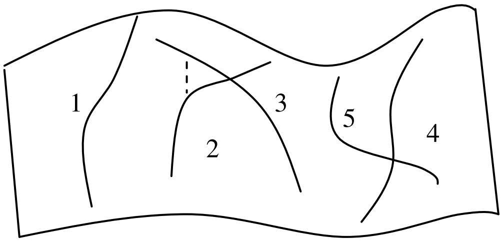


Fig. 2. Problem definition

It is in the nature of freehand sketching, that 3D sketched strokes may not be well-connected and organised. One stroke may be superimposed on earlier strokes during an iterative design process. Therefore, we define our research problem as follows (Fig. 2). Designers first “roughly” define a basic form by sketching out four boundary curves and then adding some suggestive strokes to progressively improve the model. To illustrate how the model can be created in Fig 2, the four boundary curves represent a basic form, the interior curves 1-5 (suggestive strokes) suggest some changes and they may not be extended fully across the boundary. Some suggestive curves like curves 4 and 5 may have real intersections and some may be over the others like curves 2 and 3. This means that not all suggestive curves are on the final surface.

Traditional surface construction methods are based on regular and well-organised curves [8][9] and these techniques are unsuitable for sketch-based surface modelling applications with unorganised curves [10]. Related works can be found in two categories: surface modelling from an irregular mesh (requiring boundary and interior curves connected) and surface deformation (interior curves may not be connected to the boundary). For an irregular mesh, a Gregory surface patch [11, 12, 13] is often used to generate surfaces, the irregular curves being assumed to intersect with each other. The basic Gregory technique [11] allows a design surface to be filled by free-form surface patches, which join together to make an over-all surface that is tangent plane continuous. However, the Gregory surface patch created is uniquely defined by the styling curves, and so there is no opportunity for a designer to locally refine the surface using a curve over the previous surface. Kuriyama [14] developed a curve mesh-based method for surface modelling with an irregular network of curves via sweeping and blending. The surfaces generated from the network of intersected curves are represented by multi-sided patches defined on a multivariate coordinate system. The over-all surface is a resulting interpolation from all intersected curves, but this irregular mesh-based method requires all curves to be on the final surface and therefore, may be unsuitable for our problem. For a sketch-based design interface, it is difficult to make sketch strokes always well-connected. Meanwhile, conceptual design is a typical iterative process, new sketch input can change surface design, and after this change, the earlier sketches may be on or partially on or even total off the current surface. Therefore, the capability to create a surface from curves not entirely contained on it is required.

For surface deformation, the goal is to deform a surface *s(u,v)* according to a given curve *c(t)* with known pre-image *u(t)* and *v(t)* in the domain of the surface, such attached constraints like "incidence of a curve on the surface" are satisfied and the new surface exhibits a change in consistence with the given curve [15,10]. This approach employs the "curve on the surface" concept and uses least-squares fit techniques to isolate the control vertices relevant to a curve placed on or near a surface so that motion of the curve can displace the control points, thereby changing the surface. For surface patches with a low density of control points, changing a surface by deforming control points can suffer from aliasing artefacts [15]. In addition, the determination of the pre-image of an interior curve is not straightforward. Maekawa and Ko [10] subdivided the input curve into sub-segments so that the pre-image of each segment can be a straight line in parameter space. In general, the surface deformation method has difficulty in efficiency and numerical stability. The algorithm is quite complex and difficult to apply in interactive design involving local modifications. Researches in FFD [16] focusing on small changes in the zone of influence, are not directly applicable.

In this paper, we present a novel surface modelling scheme for supporting motion-based sketch applications in conceptual design. The approach can construct a base surface model from 'rough' boundary curves and support incremental surface updating upon receiving an interior sketched curve. All interior characteristic curves may be unorganised with arbitrary orientations and positions and may cross the boundary curves or not. Following this incremental process, a final surface with multiple sub-surfaces can be obtained and transferred to commercial surface modelling software for evaluation and further modifications in detail.

The paper is organised as follows. Section 2 introduces our surface modelling strategies from unorganised interior curves and four regular boundary curves, and Section 3 presents the progressive surface modelling algorithms. Section 4 describes how to exchange a conceptual design surface into a commercial CAD system for refinement, followed by some examples in Section 5. Conclusions are finally drawn in Section 6.

2 Surface modelling strategies from unorganized interior curves

Freeform surfaces depicted by motion sketches are virtually similar to 3D sketching and have the following characteristics: (1) the boundary curves are quite ‘rough’, and may not form a closed curve network; (2) the interior characteristic curves do not always intersect the boundary or themselves to form either a well-organised curve network or an irregular curve network. This means that some sketched curves are not on the final surface. Hence, existing surface modelling methods have some difficulties in this application. The final surfaces have to be constructed progressively.

When designers conduct sketching using 3D motion, the motion velocity is varied while the motion capture rate is fixed. Therefore the density of sketch points is changed from time to time. For modelling, we smoothed each motion curve by filtering out unwanted points. For example, we used a 3D distance threshold to filter out 3D sketch points between two successive wanted points. The value of the threshold can be determined by experiments and related to surface size.

In order to create a design surface from unorganised 3D sketch curves, corresponding surface modelling strategies have been developed. They are described as follows.

2.1 Creation of a four-side boundary

Firstly, a closed boundary with four filtered curves is created. At the beginning of a design process, designers can be requested to sketch out four boundary curves, depicting a based surface. Although the curves may not be well intersected to form a network, a simple pre-processing can be conducted to make them intersect each other. Since each curve is depicted by a different set of 3D points, its type and corresponding control parameters are unknown. In order to make a pair of compatible curves, we first make them have the same number of points. In order to equalise the curves, we first count points on each pair of boundary curves and then insert more points on the curve with fewer points. Each inserted point will be located at the middle of two adjacent points with the longest chord length in each turn. We can then either fit a cubic B-spline curve to these data points or simply treat the curves as poly-lines. As a result, two pairs of the boundary curves will be obtained as *P(0,v)* and *P(1,v)*, and *P(u,0)* and *P(u,1)* as shown in Fig.3. The four boundary curves will be mapped into a unit square in the u, v two-dimensional parameter space (the pre-image plane).

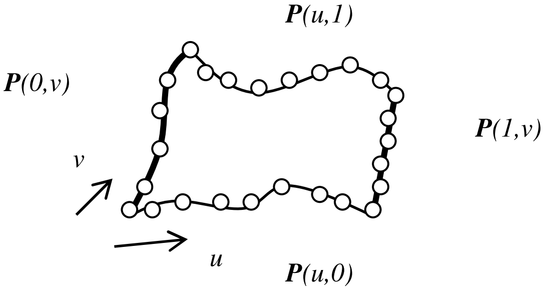


Fig. 3. Four boundary curves

Actually, if a boundary curve is taken as a poly-line, it is possible to combine some boundary curves together if necessary to form one composite curve. When the number of initial boundary curves is more than 4, we can merge some curves together into one. Finally, we will still have four composite boundary curves. The process is as follows:

(1) check if the number of curves for the boundary is more than 4, if it is so goes through the steps 2-3.

(2) compute a total chord length for each curve and its orientation.

(3) find two adjacent curves with closest orientations to merge. If there is more than one pair of curves having the same difference in their orientation, the pair consisting of a curve with shortest chord length will be merged.

(4) update the number of curves and goes back to Step 1.

In this way, we can convert an N-side boundary into a standard ‘rough’ four-side boundary. Indeed, for a broad base surface, designers often need to draw multiple strokes (curves) to represent an edge (composite boundary curve).

2.2 Construction of the base surface.

Based on the four boundary curves (poly-lines) and their pre-images, a Coons surface is built to represent the base surface as a mesh model. Each vertex on the mesh model is regarded as an intersection of the corresponding two isoparametric curves.

For construction of a base surface from boundary curves, we have chosen a bicubically blended Coons patch. In general, a Coons patch is defined by four arbitrary boundary curves: P(u,0), P(u,1) and P(0,v), P(1,v), over $u \in [0,1]$ and $v \in [0,1]$, respectively. Choosing a pair of blending functions $f_i(u)$ (i=1,2) with the restrictions $f_1(0)=1$, $f_1(1)=0$, and $f_1(u) + f_2(u) \equiv 1$, the general form of a Coons patch reads

$$P(u,v) = -\begin{bmatrix} -1 & f_1(u) & f_2(u) \end{bmatrix} \bullet \begin{bmatrix} 0 & p(u,0) & p(u,1) \\ p(0,v) & p(0,0) & p(0,1) \\ p(1,v) & p(1,0) & p(1,1) \end{bmatrix} \bullet \begin{bmatrix} -1 \\ f_1(v) \\ f_2(v) \end{bmatrix} \qquad (1)$$

Here, we choose the cubic Hermite polynomials for interpolation [8]. The blending functions are

$$f_1(u) = (2u^3 - 3u^2 + 1)$$
$$f_2(u) = -2u^3 + 3u^2$$

The reason for this choice is that a bicubically blended Coons patch with the above functions $f_1(u)$ and $f_2(u)$ guarantees C^1 continuity between patches [8] if the boundary curves are C^1 continuous. Therefore, the continuity of cross-boundary derivatives is automatically satisfied. Furthermore, the bicubic Coons patch as defined by Eq. (1) is easy to use in a design environment because only the four boundary curves are needed, although the four corners are required.

If there are 4 boundary curves, a surface from them can be generated directly with Eq. (1). For a three-side boundary, a surface can be generated as a special case of four side curves. To clarify this problem one of them is considered to have zero length, for example, $P(u,1) \equiv P1$,the corresponding surface can be gained from Eq. (1) as shown in (Fig. 4).

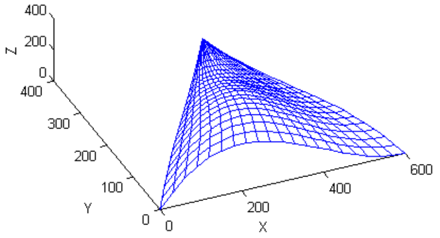


Fig. 4. Three-side boundary

This base surface modelling technique provides not only a capability of creating a surface from merged multiple boundary curves, but also keeps the pre-image of boundary curves rectangular in the parameter space (*u, v*). More importantly, this will make interpolation of the interior curves easier.

2.3. Update of the base surface

Upon receiving an interior curve sketch, it is considered on the new surface after the update. If its pre-image is a straight line, the original base surface will be sub-divided and updated correspondingly. Otherwise, the interior curve will be divided at the middle point recursively until each sub-segment's pre-image can be treated as a straight line. In this way, the design surface will be progressively updated. Details of how to obtain the pre-image of an interior curve and how to update surfaces are discussed in the next section.

2.4. Surface refinement

After obtaining a conceptual design surface (a mesh model), we can quickly transfer this final mesh model into a commercial CAD package. In a CAD system, it can be refitted with higher order continuity requirements and further be modified for manufacturing. Here, our strategy is to create a 'rough' conceptual model with reasonably good continuities and use CAD systems to refine it as in reserve engineering.

3 Progressive Surface Modeling

This modelling process includes two phases: creating a base surface and subsequently updating it progressively from interior characteristic curves. This section focuses on the progressive modelling process.

Based on the base surface obtained from boundary curves, an interior curve drawing can be then used to modify the base surface. This progressive surface modelling process is shown in Figure 5.

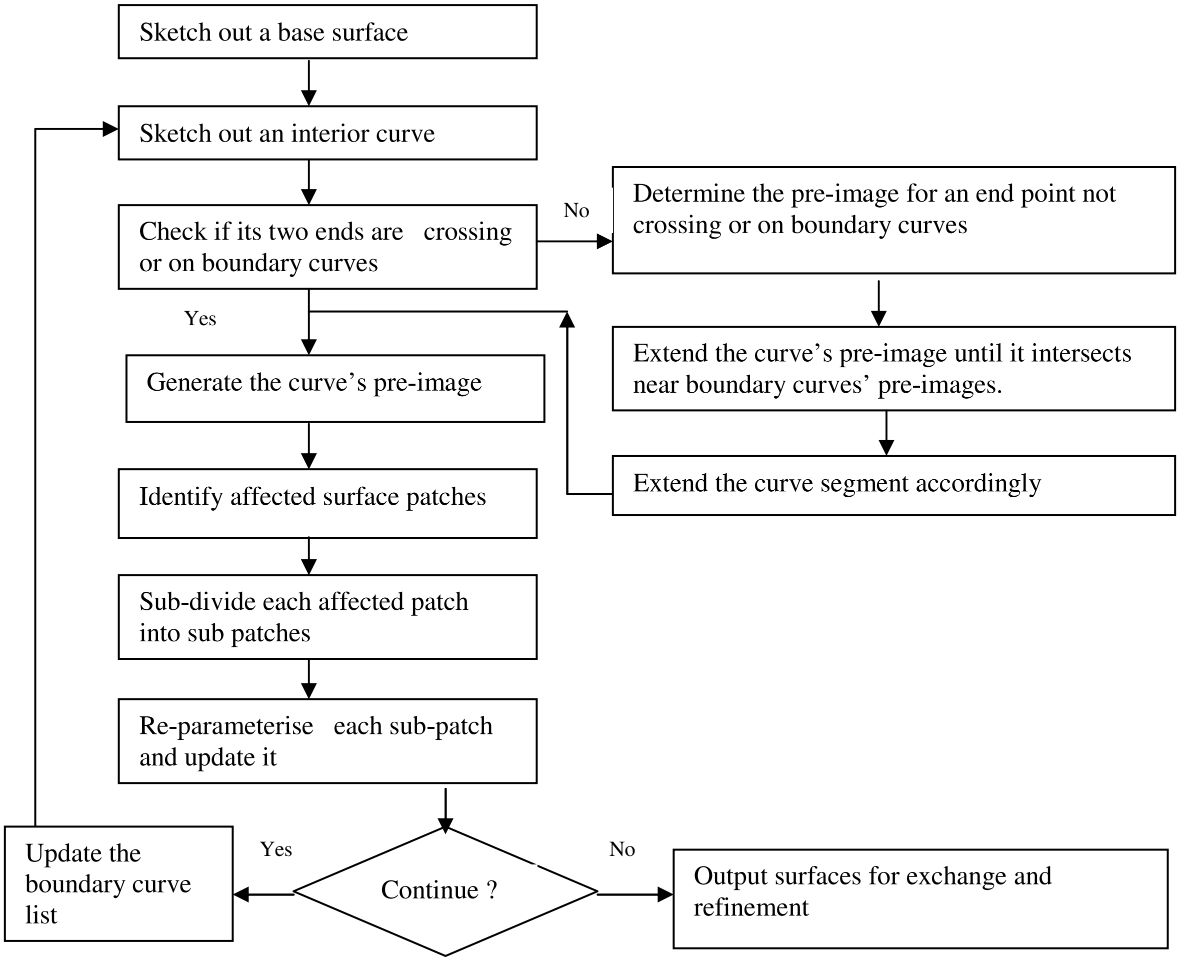

Sketch out a base surface
Sketch out an interior curve
Check if its two ends are crossing or on boundary curves
No
Determine the pre-image for an end point not crossing or on boundary curves
Yes
Generate the curve's pre-image
Extend the curve's pre-image until it intersects near boundary curves' pre-images.
Identify affected surface patches
Extend the curve segment accordingly
Sub-divide each affected patch into sub patches
Re-parameterise each sub-patch and update it
Yes
Update the boundary curve list
Continue ?
No
Output surfaces for exchange and refinement

Fig. 5. Surface updating process

The surface updating process is explained in the following steps:

(1) Obtain a new 3D stroke.

Create a new stroke by 3D motion sketching and filter the stroke by a 3D distance threshold.

(2) Check if its two ends are on, or crossing, the boundary curves, otherwise go to step (6).

The purpose of this detection is to determine the pre-image of the input stroke. First the system will check if the two ends meet together. If it is, the input stroke is not an open curve and needs to be divided into open curve segments. Here we focus on open curve detection. Boundary curves initially include the four boundary curves: p(u,0), p(u,1), p(0,v) and p(1,v), and progressively include newly input interior curves. In general, each curve can be represented by a set of points P_i, i=1,2,…,n. The new input stroke can also be described as a set of points C_j, j=1,2,…,m. In order to detect if the first end of the stroke is on a boundary curve, the detection algorithm checks the distance $D_{i,j}$ between a point P_i, i=1,2,…, n, and a point on the first half of the stroke C_j, j=1,2,…, h, (<= m/2). If the minimum distance *dm*=min($D_{i,j}$) | i=1,2,…,n; j=1,2,…, h, is smaller than a threshold, the first end of the stroke is regarded on the current searching boundary curve, detecting for the first end will stop, the corresponding point on the stroke will snap to the nearest point on the boundary curve and the over-sketched points will be trimmed off if the closest point is not the first point, and finally its pre-image will be determined. Otherwise, if the *dm* is larger than the threshold, another boundary curve will be tested. If after testing all boundary curves, the first end is not on any boundary curves, the process jumps to step (6) to find its corresponding pre-image and to extend its pre-image to intersect with the near boundary's pre-image in step (7) and the curve adjusted accordingly in step (8). For the second end of the stroke, the detection process is similar. After this detection process, a straight line on the pre-image plane for the input stroke is determined according to its two end positions on the pre-image.

(3) Identify the affected sub-surface patches based on boundary curves involved.

Once the pre-image for the input stroke is obtained from the above processes, its affected surface patches can be determined by finding the passing regions in the pre-image plane. We know that the pre-images for the boundary curves p(u,0), p(u,1), p(0,v) and p(1,v), form an unity square. Each interior curve's pre-image is a straight line segment within the square. In the pre-image plane, each closed region formed by three or four pre-image line segments corresponds to a surface patch. If a surface patch is affected, the current pre-image line should go through the corresponding region in the pre-image plane. Therefore, the identification process first checks if the pre-image line of the current curve has intersections with the boundary pre-image lines of each region in the pre-image plane. If it has, the corresponding surface patch is affected. The current curve may affect several surface patches when it has intersections with their corresponding pre-image boundaries.

(4) Divide each affected patch into two or three sub-patches and re-parameterise the related boundary curves based on their pre-images (Fig. 6). In the pre-image plane, if the new stroke intersects a sub-patch at two opposite corners (Fig 6(a)), the sub-patch will be divided into two 3-side sub-patches. When the stroke links the related patch from a corner to an edge (Fig. 6(b)), the sub-patch can be broken into a 3-side and a 4-side sub-patch. If it intersects two adjacent boundary curves of the affected patch (Fig. 6(c)), three sub-patches (two 4-side and one 3-side) can be obtained. When the stroke joins the related patch at two opposite boundary curves (Fig. 6(d)), two 4-side sub-patches will be resulted.

For each case, the current curve may have no intersections in 3D with the boundary 3D curves though their 2D pre-images have intersections. This means that although the current curve and the related boundary curve have the same position at their intersection point in the 2D pre-image plane, they may have different positions in 3D. In this case, the related boundary curves need to be updated. In general, a related boundary curve P_i, i=1,2,…,n, has a straight line pre-image between (u_0,v_0) and (u_1,v_1), and the current curve C_j, j=1,2,…,m, has a straight line pre-image between (cu_0,cv_0) and (cu_1,cv_1). Their preimages have an intersection at (u_k, v_k). The corresponding point on the boundary and the interior curves can be snapped to P_k and C_t respectively. The positional difference vector $\delta=C_t-P_k$ will be distributed gradually. If K is between 1 and n, the first segment

$P_i=P_i+(i-1)*\delta/(k-1)$, $i=1,2,...,k$; and the second segment $P_i=P_i+(n-i)*\delta/(n-k)$, $i=k, k+1,..., n$.

For each sub-divided surface patch, after updating its boundary curves if necessary, re-parameterisation process is to make corresponding boundary curves to have the same number of points. This process has been described in the section 2.1 and it is simple to implement.

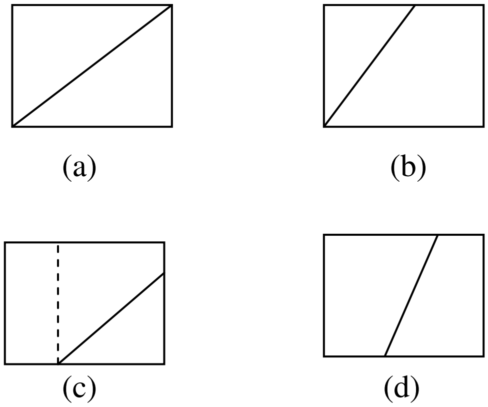


Figure 6. Surface subdivision scheme: (a) two 3 side patches; (b) one 3 and one 4 side patches; (c) two 4 side and one 3 side patches; (d) two 4 side patches

(5) Create 3- or 4-side Coons surface sub-patches and go back to step (1) until the user inputs stop.

(6) If any end of the new stroke does not touch boundary curves, the base surface is used to determine its pre-image as follows: project the end point of the stroke onto the base surface, resulting in a footpoint on the base surface, and compute parameter values for the footpoint on the base surface. The end point Q will have a corresponding footpoint F on the base surface as shown in Fig. 6. Their relation can be described by the following vector equations.

$$F = P_0(u_{i,}v_j) + k_r \bullet R + k_s \bullet S$$
$$F = Q - d \bullet N$$

The equations can be rewritten in a matrix form as

$$\begin{bmatrix} r_x & s_x & n_x \\ r_y & s_y & n_y \\ r_z & s_z & n_z \end{bmatrix} \bullet \begin{bmatrix} k_r \\ k_s \\ d \end{bmatrix} = \begin{bmatrix} x_Q - x_0 \\ y_Q - y_0 \\ z_Q - z_0 \end{bmatrix} \qquad (2)$$

where P_0 is a surface point (x_0,y_0,z_0) corresponding to parameters (u_i, v_j), r_x, r_y, and r_z are the components of the unit vector R from P_0 to the point $P(u_i, v_{j+1})$. Similarly, s_x, s_y, and s_z correspond to the unit vector S from P_0 to the point $P(u_{i+1}, v_j)$. N is a unit normal vector of the facet plane, which is defined as a cross product of the vectors **R** and **S**. The distance between the footpoint F and the end point Q is *d*. From Eq. (2), the parameters *kr*, *ks* and *d* can be solved.

The resulting parameters for the end point can be described as

$$\begin{aligned} u_f &= u_i + \Delta u = u_i + (u_{i+1} - u_i) * ks / \left\| p_0(u_i, v_j) - p(u_{i+1}, v_j) \right\| \\ v_f &= v_j + \Delta v = v_j + (v_{j+1} - v_j) * kr / \left\| p_0(u_i, v_j) - p(u_i, v_{j+1}) \right\| \end{aligned} \qquad (3)$$

If the resultant parameters u_f and v_f are within the parameter triangle (u_i, v_j), (u_{i+1}, v_j) and (u_i, v_{j+1}), the F is a real footpoint on the surface, and the current detecting surface patch is affected. Otherwise, we use the point $P(u_{i+1}, v_{j+1})$ as P_0 to detect the footpoint. If there are no real footpoints on these two triangle planes, the search continues. Finally, if the stroke is a real interior curve, a pair of parameters will be obtained for the end point. Otherwise, if the surface projection of the end point is outside the surface region, the stroke will be rejected because we assume that the stroke is not an interior curve. The time complexity of this process is linear to the number of surface meshes.

Similarly, we can find a pre-image for the middle point. On the pre-image, if the middle point is quite close to the pre-image line between the two ends, the whole curve's pre-image will be finally treated as a straight line, otherwise, the curve will be divided at the middle point recursively until each sub-segment's pre-image is a straight line.

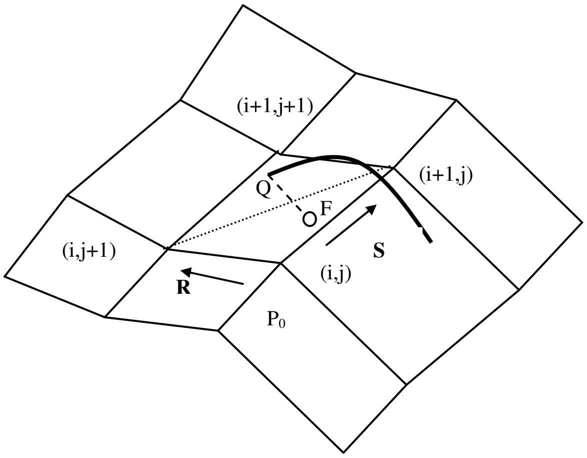


Figure 7. Curve projection in Euclidean space

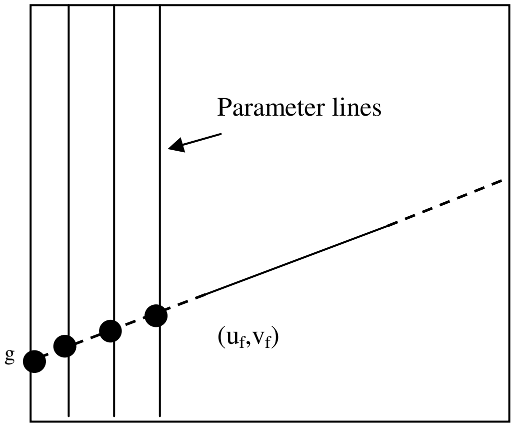


Figure 8. Curve extension in the pre-image plane.

(7) After obtaining parameters u_f and v_f for an end of the curve, the curve is extended in parameter space to intersect a boundary curve's pre-image at *g* of the affected patch (Fig. 8) and a corresponding 3D point G on the related boundary curve will be gained. The line segment between the point (u_f, v_f) and g is an extended part for the interior curve in parameter space. The line segment may intersect other parametric lines of the affected patch as shown in Fig. 8. All intersection points in parameter domain have corresponding 3D points in Euclidean space, which can be computed from their parameters related to the affected patch.

(8) The curve is updated with extended segments consisting of all 3D intersection points from Step (7). The coordinate difference between the points Q and F is used in the update of 3D intersection points between the Q and the G on a boundary curve. Actually, how to control this distribution will affect the zone of influence of the corresponding end point. Currently we suggest an even distribution because we assume that the interior curve will affect the whole affected patch.

(9) Go to Step (3).

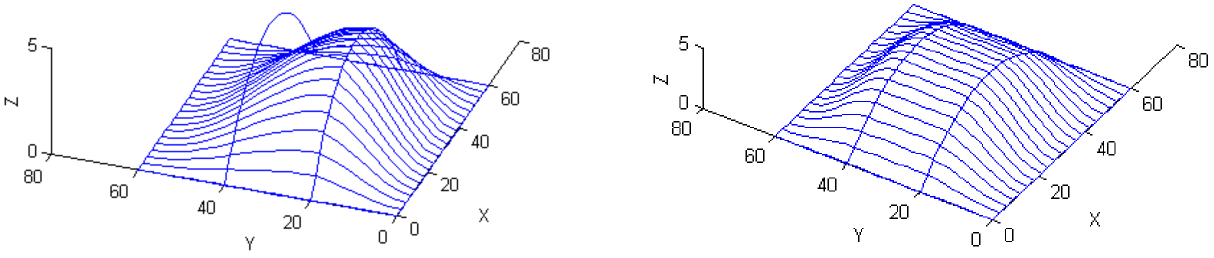
4. Creation of exchange surfaces

The surfaces obtained from rough design sketches need to be evaluated and modified in detail. Therefore, there is a need to transfer the conceptual design surfaces into a commercial CAD package. The surface can be then further modified interactively with powerful tools for various purposes and rendered in different forms for evaluation. Currently, we create exchange surfaces for Alias Studio software through its 'obj' files.

In order to exchange the conceptual design surfaces into the Alias Studio software, we transfer a set of section curves and then create a skin surface in the Alias Studio (see Fig 10g). Each section is approximated by a cubic NURBS curve with a rational knot vector [9]. Once the corresponding skin surfaces are received, they can be easily modified. Of course, if all vertices of the mesh model are exchanged into reverse engineering software, the whole surface can be refitted as well.

5. Implementation and Test

The proposed surface modelling approach and algorithm have been tested in MATLAB 5.3 with examples. Figure 9 shows an example of a surface update from section curves. The initial base surface is a flat surface and after inputting a section curve (y=20) touching the two boundary curves, the resulting surface is shown in Fig. 9a. Another section curve (y=40) was then inputted and consequently, the surface was updated as in Fig 9b.

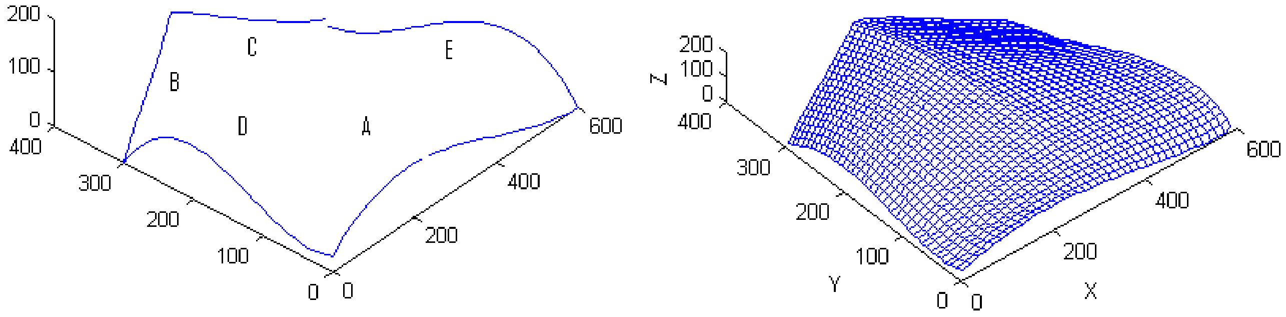


(a) First update

(b) Second update

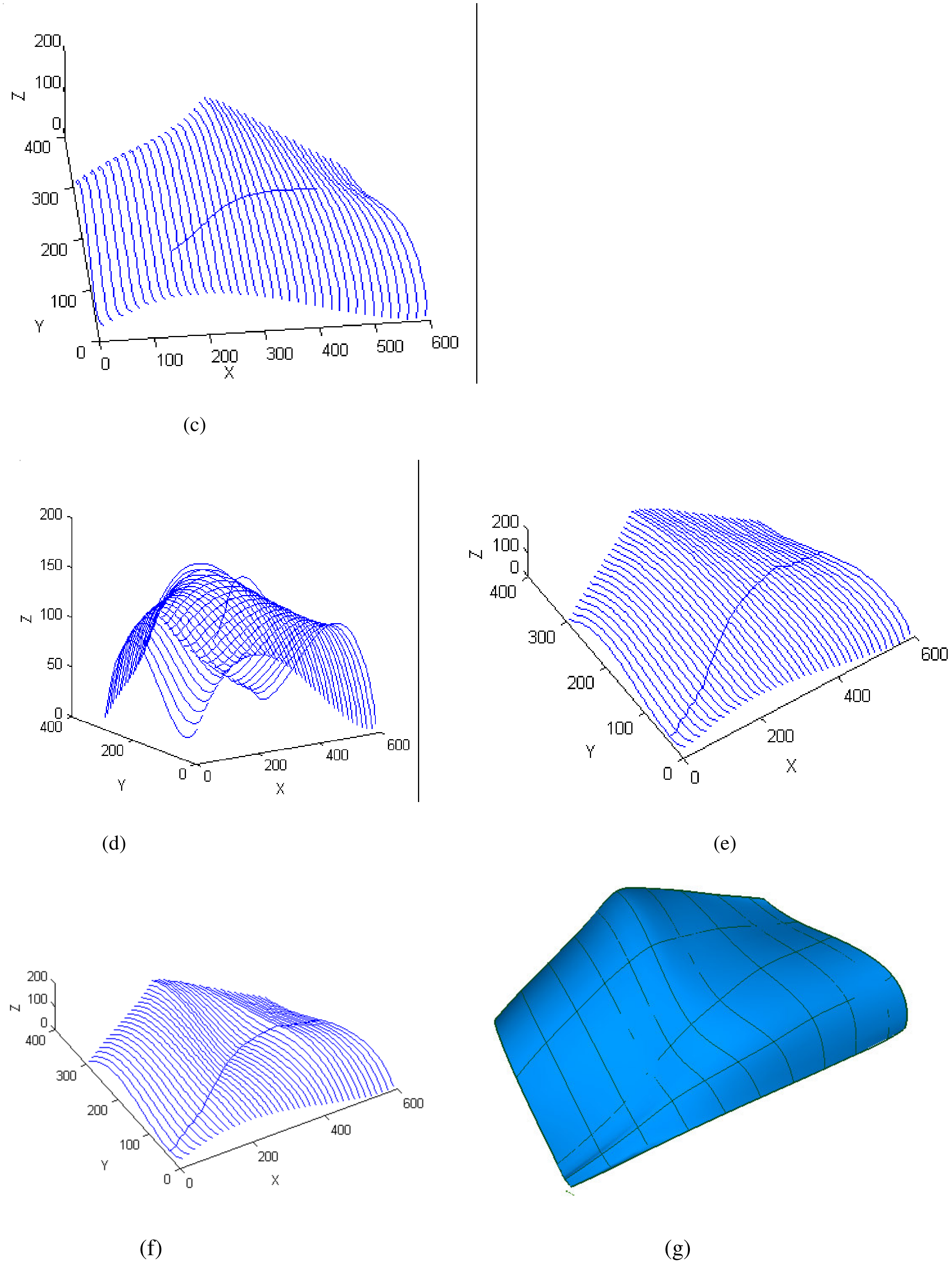
Fig. 9. Surface update from section curves

Figure 10 gives another example of surface constructions from arbitrary boundary and interior curves. The initial boundary consists of 5 curves (Fig. 10a): A, B, C, D, and E. The curves C and E form an open corner. These boundary curves were pre-processed with the intersection of curves C and E and the merge of curves B and C. After that, four new closed boundary curves from the initial interpolated section curves were obtained, and consequently a base surface for a Coons patch resulted as shown in Fig. 10b. An interior stroke was then drawn over the base surface (Fig. 10c). In order to have a clear view, Fig. 10c displayed only *u* lines and the interior stroke. Fig. 10d showed the gap between the stroke and the base surface. After identifying its projection points on the base surface, the stroke's pre-image was treated as a straight line and the curve was then extended to cross the boundary curves (Fig. 10e). Correspondingly, reparameterisation was conducted over two sub patches and the resultant surface patches are displayed in Fig. 10f. The two surface patches were exchanged into the Alias Studio. Finally, a composite surface was generated as a skin surface and was rendered with some section curves (Fig. 10g).



(a)

(b)



(c)

(d)

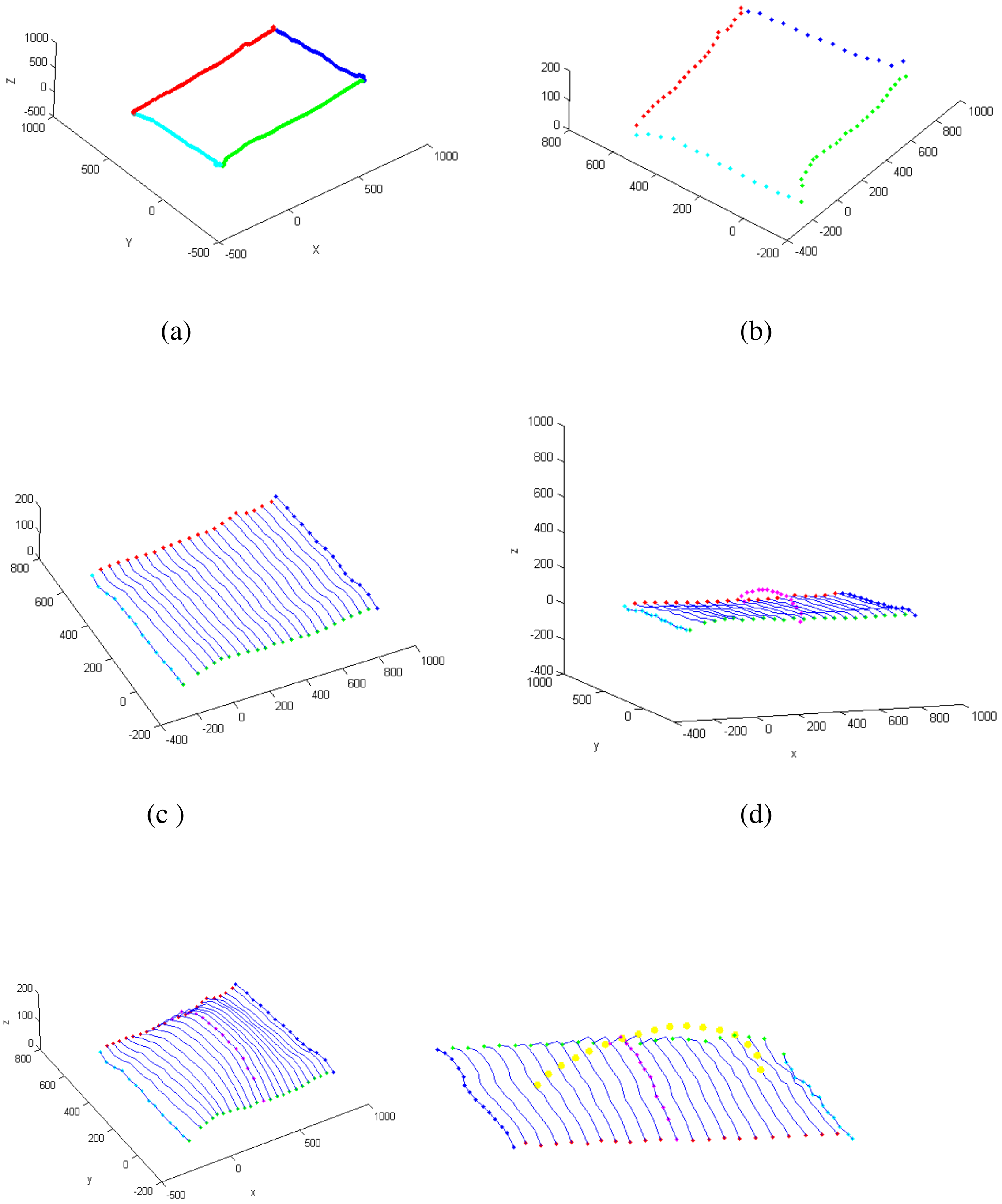
(e)

(f)

(g)

Figure 10. Surface constructions from unorganised curves: (a) 5-side boundary with an open corner; (b) creation of a base surface; (c) input of an interior curve; (d) position of the interior curve; (e) extension of the interior curve; (f) result from reparameterisation; (g) exchanged surfaces in Alias Studio

Example 3 shows a more general case. A designer first sketched out four boundary curves (Fig. 11a), which are filtered with a threshold of 40mm (we used this big threshold in order to give a clear view of sketches after the filtering). The resultant curves from the filtering are shown in Fig. 11b. After the pre-processing, four curves were properly joined together and used to create a base surface (see Fig. 11c). Based on this surface, another interior stoke was drawn out (Fig. 11d). After detection, this stroke's two ends are on the boundary curves. After trimming off one point from the second end, the pre-image for the stroke was determined as a straight line between (11/24,0) and (11/24,1). The pre-image line is within the initial pre-image square corresponding to the base surface. Thus, the affected patch is the base surface and it is divided into two sub-surface patches. Because the stroke is passing through the boundary curves, the related two boundary curves don't need to be updated. Two sub-surfaces were created based on re-parameterised sub-patches and the resulting surfaces are shown in Fig. 11e. The curve stroke was added in the boundary list with previous four boundary curves. After that, the designer sketched out another interior curve as shown in Fig. 11f in yellow. For this curve, its two ends are not on any boundary curves; therefore, its corresponding pre-image was computed from Eq. (2) as a straight line. This pre-image line has an intersection with the pre-image line of the first interior stroke, thus, the curve affects two sub-surfaces linked by the first interior stroke. Its two ends were extended accordingly to meet other boundaries (see Fig. 11g). On the other hand, the boundary curve from the first interior stroke has no intersection in 3D with the current curve, thus it was updated accordingly. Finally, each affected surface was divided into two four-side surfaces. After re-parameterisation, the final surface was obtained as in Fig. 11h. The designer completed this conceptual design in 5 minutes.


Z
1000
500
0
-500
1000
500
0
Y
-500
-500
0
500
1000
X
200
100
0
800
600
400
200
0
-200
-400
-200
0
200
400
600
800
1000
z
1000
800
600
400
200
0
-200
-400
y
x
0
500
1000

(a)

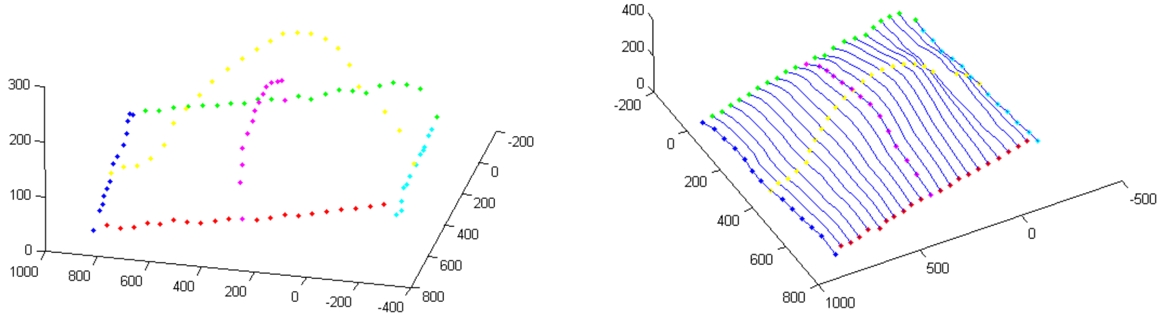
(b)

(c)

(d)

(e)

(f)



(g)

(h)

Fig 11. General case study

5 Conclusion

In this paper, we have presented a novel surface modelling scheme for supporting sketch applications in conceptual design. Based on rough boundary curves, well-organised four boundary curves are obtained through pre-processing and a corresponding base surface is then constructed based on a regular Coons patch. After that, each interior design curve is progressively incorporated to update the base surface. These interior characteristic curves may cross the boundary curves or not. Their pre-images may have arbitrary directions. Therefore, both the complex curve fitting method and the multi-sided patches approach have difficulties in this application. The proposed modelling scheme adopts the 'coarse-to-fine' principle to progressively apply a 'curve on the surface' concept to update the surface model. In order to change the base surface from an interior curve, it is checked against the previous surface patches to figure out which patches are affected and the corresponding parameters in the parameter space. Consequently, reparameterisation is conducted for refining surfaces locally. Following this incremental process, a final surface with multiple patches can be obtained and transferred to commercial surface modelling software for modification in detail.

The proposed surface modelling approach and associated algorithms have been tested with three examples. This modelling scheme is capable of dealing with unorganised design curves for surface modelling and easy to create intermediate models for design feedback. It supports an incremental design process starting by sketching out rough boundary curves for a base surface and then updating the base surface with interior curves. Accuracy depends on sketching speed and the parameter for sketch curve filtering. The creation of high quality surface with high order continuity requirements has been assigned as a typical job for a commercial CAD system.

The proposed surface modelling scheme currently limits its base surface as a four (or three)-side surface. For a closed surface like a sphere, it has difficulties in input. Similarly, for a closed interior curve or a spiral curve, the determination of its pre-image is not straightforward. Furthermore, for a base surface, if its pre-image cannot be mapped to a unit square, a pre-processing is required.

Sketching is a flexible design tool and the current CAD modelling systems have difficulties in matching its flexibility. This paper aims to reduce some assumptions used by traditional CAD systems through unorganised curves. To develop a perfect surface modelling method matching the sketching flexibility still needs further research efforts.

Acknowledgement

The authors would like to thank the anonymous referees for the valuable and very helpful comments improving the quality of this paper. This work was funded by the UK EPSRC grant (GR/S01701/01).

References:

[1] P.A. Prieto, D.K. Wright, and S.F. Qin, A novel method for early formal developments using CAD and RP technology. *Proceedings of the Institution of Mechanical Engineering Part B, Journal of Engineering Manufacture*, Vol. 217, 2003, pp. 695-698.

[2] S.F. Qin, D.K. Wright, J.S. Kang, and P.A. Prieto, Incorporating 3D body motions into large-sized freeform surface conceptual design, *Biomedical Sciences Instrumentation*, Vol. 439, No.1, 2005. pp.271-276.

[3] L Eggli, C. Hsu, B.D. Bruderlin and G. Elber, Inferring 3D models from freehand sketches and constraints, Computer-Aided Design, Vol. 29 (2), 2003, pp. 101-112.

[4] E Sachs, A Roberts, D. Stoops, 3-Draw: A tool for designing 3D shapes. *IEEE Computer Graphics & Applications*, Vol.11, No.6, 1991, pp.18-25.

[5] MF. Deering, HoloSketch: A virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction* Vol. 2&3, 1995, pp.220-238.

[6] S. Schkolne, M. Pruett, P. Schröder, Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. *Proceedings of ACM SIGCHI'01*, 2001, pp. 261-268.

[7] MotionAnalysis, http://www.motionanalysis.com, 2006.

[8] I. Zeid I, CAD/CAM Theory and Practice. Mc-Graw-Hill, Inc, 1991.

[9] L Piegl, W Tiller. The NURBS book. Springer, 1995.

[10] T Maekawa, KH Ko, Surface construction by fitting unorganized curves. *Graphical Models*, Vol. 64, 2003, pp.316-332.

[11] Gregory JA, Yuen PK. An arbitrary mesh network scheme using rational splines. In: Lyche T, Schumaker LL, editors. Mathematical methods in CAGD II, New York: Academic Press, 1992. p. 321-9.

[12] CCL. Wang, Parametrization and parametric design of mannequins. *Computer-Aided Design,* Vol. 37, 2005, Pp.83-89.

[13] G. Mullineux, Improvement of free-form surfaces for product styling applications. Computer-Aided Design, Vol. 34(12), 2002, pp.871-80.

[14] S. Kuriyama, Surface modelling with an irregular network of curves via sweeping and blending, Computer Aided Design, Vol 26(8), 1994. pp. 597-606.

[15] P. Michalik, B. Bruderlin, A constraint-based method for sculpting free-form surface, In: G. Brunett, H.P. Bieri, G. Farin(Eds), Geometric Modelling, Supplement 14 to Computing, Springer, Berlin, 2001.

[16] F. Lazarus, S, Coquillart and P. Jancene, Axial deformations: an intuititive deformation technique, Computer Aided Design, Vol 26 (8), 1994, pp. 607-613.