# Free-form Sketching with Ball B-Splines[*]

Rongqing Song, Zhongke Wu, Mingquan Zhou,
and Xuefeng Ao

Institute of Virtual Reality and Visualization, Beijing Normal University,
BeiJing 100875
rongqingsong.11@163.com, zhongkewu@gmail.com,
mqzhou@bnu.edu.cn, aoxuefeng@mail.bnu.edu.cn

**Abstract.** Quickly and conveniently generating a 3D freeform model is a challenging problem in computer graphics field. This paper proposes a new approach for rapid building 3D freeform shapes through sketching based on ball B-splines.

**Keywords:** Sketch, Ball B-Spline, Free-form Shapes.

## 1 Introduction

It is a challenge to quickly and conveniently generate a 3D freeform model in computer graphics field. Despite the 3D scanning technique has been widely used, it is not suitable for model building in the initial or conceptual stage of design. The goal of the paper is to build 3D model rapidly and intuitively through sketching silhouette curves. Sketching is a fast and intuitive method of building 3D models from freehand 2D input. 3D shapes are generated through regarding the sketching 2D curves as silhouettes from current points of view.

There exist two well-known sketching systems to create 3D geometric models by 2D sketching, SKETCH and Teddy [1, 2]. SKETCH is based gesture recognition. Teddy uses polygonal mesh model. Recent works propose some systems based on implicit function [6, 7].

In this paper, a new sketching method using ball B-Spline curves and surfaces is proposed. Ball B-Spline curves and surfaces represent 3D objects as well as its skeleton, explicitly and intuitively. This gives more potential for prototype modeling. Therefore, we primarily use ball B-Spline curves to represent 3D objects in our method. The rest of this paper is organized as follows: In section 2, fundamentals of ball B-Spline curves are introduced; in section 3, the skeleton algorithm on sketching contours is discussed; in section 4, Sketching methods are described; in section 5, Editing methods are further discussed as more convenient way for solid generation; finally some conclusions are given.

[*] The work is partly supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry and the National High Technology Research and Development Program of China (863 Program. No: 2008AA01Z301).

C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCII 2009, LNCS 5615, pp. 583–592, 2009.

## 2 Introduction to Ball B-Spline Curve

### 2.1 Sphere Geometry

A sphere is defined as a set

$$< C;r >= \{x \in R^3 \mid \left|x - C\right| \leq r, C \in R^3, r \in R^+\} \tag{1}$$

Here $C$ is the center of the sphere and $r$ is the radius.

### 2.2 Ball B-Spline Curve

$N_{i,p}(t)$ is the i-th B-Spline basis of degree $p$ with knot vector.

$$[u_0, \ldots, u_m] = \{\underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{b \ldots, b}_{p+1}\} \tag{2}$$

Now the Ball B-Spline Curve can be defined:

$$< B > (t) = \sum_{i=0}^{n} N_{i,p}(t) < P_i; r_i > \tag{3}$$

$P_i$ is called control points, $r_i$ is called control radius. As shown in Fig.1

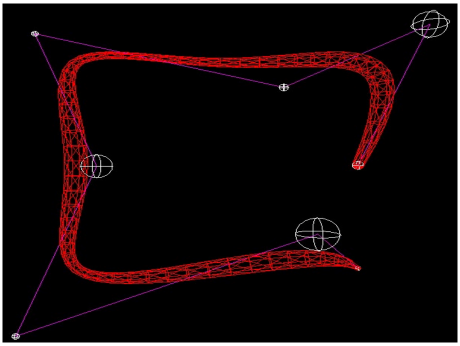


**Fig. 1.** Control polygon, radius (white) and center curve (red)

### 2.3 Ball B-Spline surface

$N_{i,p}(u)$ is the i-th B-Spline basis of degree $p$ with knot vector

$$[u_0, \ldots, u_m] = \{\underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{b \ldots, b}_{p+1}\} \tag{4}$$

And $N_{i,q}(v)$ is the j-th B-Spline basis of degree $q$ with knot vector

$$[v_0,...,v_m]=\{\underbrace{d,...,d}_{q+1},u_{q+1},...,u_{m-q-1},\underbrace{e...,e}_{q+1}\} \tag{5}$$

Then the ball B-Spline Surface is defined as follows:

$$<s>(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}N_{i,p}(u)N_{j,q}(v)<P_{i,j};r_{i,j}> \tag{6}$$

$P_{i,j}$ is called control points, $r_{i,j}$ is called control radii. As shown in Fig.2

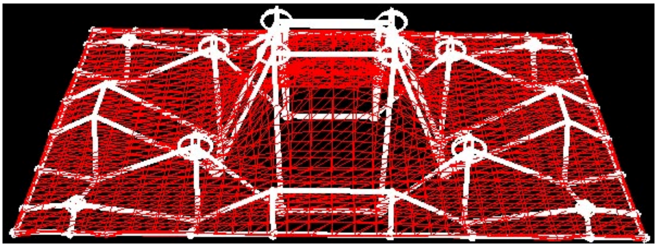


**Fig. 2.** Control polygon, radius (white) and center surface (red)

## 3 Computing Skeleton

User can create a new object by drawing a closed polygon, and then a 3D model will be automatically constructed based on the 2D input. In order to use ball B-Spline curves and surfaces, the control points and control radius must be given, these information can extract from the skeleton of the 2D polygon. In our system, we use voronoi-based method to extract skeleton.

Blum was the first person who introduced the concise definition of skeleton or medial axis as the set of centers of maximal disks (2D) or balls (3D) [8].

Medial Axis: The set of points of a region P with more than one closest point among the boundary points $\partial$ P of the region. Equivalently, it is the set of centers of maximal balls, i.e. of balls in P that are themselves not enclosed in another ball in P [9].

The skeleton has a wide field of applications. It has been proven useful in automatic navigation, compression, shape description and abstraction and tracking etc. there is a large variety of techniques for extracting skeletons and centerlines. We use voronoi-based method to extract skeleton. The method is more suitable for polygon mesh.

The skeleton we extract has some restrictions

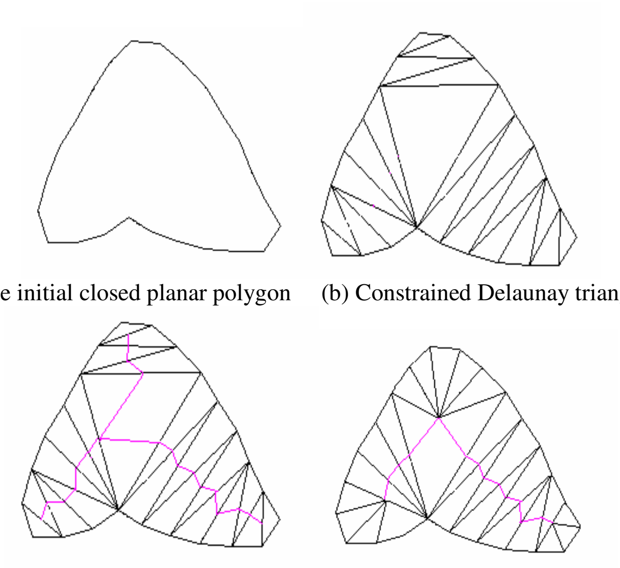
(I) The centerline should be a sequential single-voxel chain

(I I)The order of centerline points is crucial for reconstruction.

(I I I)The skeleton need to be pruned, the less the branches are it will be better for our model.

The skeleton of the 2D sketch can be extracted follow the next steps

First, while drawing the 2D sketching, there are many restrictions: it should not be self-intersecting, the start point and end point of the stroke will be automatically connected, and the operation will fail if the stroke is self-intersecting. The algorithm to create the 3D shape is described in detail in section 4. When constructing the initial closed planar polygon, the system makes all edges a predefined unit length (Fig.3 (a)).



(a) The initial closed planar polygon (b) Constrained Delaunay triangulation

(c) The skeleton without prune (d) The pruned skeleton

**Fig. 3.** The process of extract the skeleton

Next, construct a Constrained Delaunay Triangulation (Fig.3 (b)) of the polygon point's .after the triangulation, the edges can be classified to two categories: The edges of the initial polygon are called external edges, while edges added in the triangulation process are called internal edges. The skeleton can be obtained by line the midpoint of the internal edges which may have many branches (Fig.3(c)), in order to get one main centerline, it should be pruned. Our skeleton extraction algorithm and pruning are based on the ones described in [2].after the pruning ,only the main skeleton will be remain Figure (Fig.3 (d)).The link points of the skeleton are control points of the Ball B-Spine cures and surfaces.

Third, we calculate the radius of each point on the skeleton use the method introduced in [2]. Then let the radius of the point be the control radius.

After the above operation, an orderly main skeleton (include control points and control radius) will be extracted.

# 4 Sketching Methods

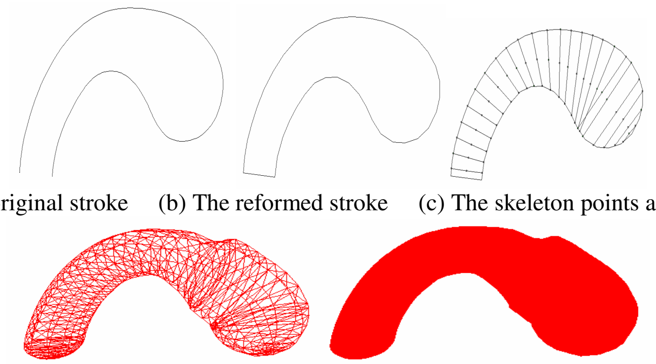
## 4.1 2D Input and Data Obtaining

Our system accepts two ways of input. The first method is drawing using a digital pen on the tablet, while the other one is importing a sketching image, which is stored after previous sketching. As generating 3D model from the sketching by digital pen on tablet in time is more natural to the idea of sketch, we discussed this method mainly. By drawing each stroke, the points on each stroke are stored in independent data structure for following processing.

In order to avoid taking bother to compute the intersection points of the stroke, the constraint for inputting is set for users that the stroke should be open. So the stroke closing is conducted by the system automatically later by simply connect the end point and the start point of the drawing.

As the drawing rates of the users can be arbitrary but the precision of the tablet is certain, the data points stored can be totally irregular in density, which means the points can be very dense if the user draw slower or can be very sparse if the user draw faster in local places. Therefore the stroke reforming is inevitable for following processing. By setting the gate value for the distance between two adjacent points in the reformed stroke, some too close points will be sorted out and only the points whose distances with the previous ones are larger than the gate value can be stored.

Then based on these points in the reformed stroke, the skeleton points and the corresponding radius can be extracted out for later 3D surface construction.



(a) The original stroke (b) The reformed stroke (c) The skeleton points and radius

(d) The 3D object in mesh mode (e) The 3D object in face mode

**Fig. 4.** Input and data obtaining

The fig.4 shows the process of inputting and obtaining data used for later 3D model generation. (Fig.4 (a)) is a original drawn stroke using the digital pen, (Fig.4 (b)) is the stroke after reforming using the above way, (Fig.4(c)) shows the skeleton points and corresponding diameters of the data spheres used for following interpolation; (Fig.4 (d)) is the 3D mesh model generated from the skeleton points and corresponding radius by interpolation; (Fig.4 (e)) is the rendering result in face form.

### 4.2 Reconstruction of the 3D Surface

As the solid in reality usually has the disk cross section, it is very natural to use BBSC to construct the 3D surfaces using the skeleton points and the corresponding radius.

**BBSC Interpolation.** In fact, these skeleton points combined with their corresponding radius can be regarded as several spheres, from which a whole BBSC whose center curve passes these central points and their corresponding radius can be obtained through interpolation. We can use B-spline interpolation method to interpolate the center curve and B-Spline scalar function method to interpolate the radius in a BBSC respectively to obtain BBSC interpolation. For more information on B-Spline interpolation, please refer to [10]. For the detail interpolation effect see the following figure.

**Boundary surface computing.** To evaluate the boundary of the region represented by a BBSC, we can regard the boundary as the envelope surface of a one-parameter family of spheres.

For example, for any t, center at

$$\begin{cases} x(t)=\sum_{i=0}^{n} N_{i,p}(t)x_i \\ y(t)=\sum_{i=0}^{n} N_{i,p}(t)y_i \\ z(t)=\sum_{i=0}^{n} N_{i,p}(t)z_i \\ r(t)=\sum_{i=0}^{n} N_{i,p}(t)r_i \end{cases} \tag{7}$$

Then $(x-x(t))^2+(y-y(t))^2+(z-z(t))^2-(r(t))^2=0$ (8)

According to envelope theorem [11],

$$\begin{cases} F(x,y,z,t)=0 \\ \dfrac{\partial F(x,y,z,t)}{\partial t}=0 \end{cases} \tag{9}$$

The second equation is obtained:

$$(x-x(t))x^{'}(t)+(y-y(t))y^{'}(t)+(z-z(t))z^{'}(t)+r(t)r^{'}(t)=0 \tag{10}$$

For any t, we can compute $x(t)$, $y(t)$, $z(t)$, $r(t)$ and their derivatives $x^{'}(t)$, $y^{'}(t)$, $z^{'}(t)$, $r^{'}(t)$ according to the boor algorithms of a B-Spline curve.

Now, let $X=x-x(t)$, $Y=y-y(t)$, $Z=z-z(t)$, $C=r(t)$,

$$D = x'(t)\,,\ E = y'(t)\,,\ F = z'(t)\,,\ G = r'(t) \tag{11}$$

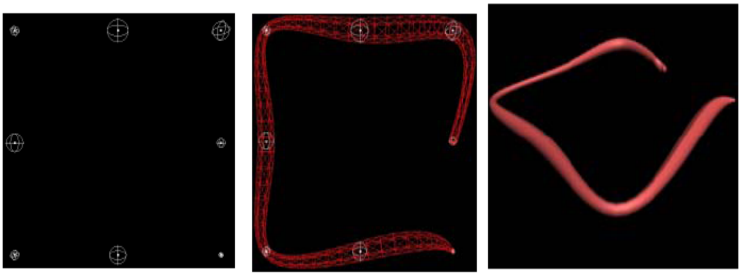
So we get the following equation group:

$$\begin{cases} X^2 + Y^2 + Z^2 = C^2 \\ DX + EY + FZ = -CG \end{cases} \tag{12}$$

This is an equation group of the intersection of a sphere and a plane. So its solution is a circle. We call it characteristic circle.

When $r_i$=constant for any i, for example, $r(t)$=constant, then the boundary is the offsets in 3D space of the center curve by the constant. When the center curve is open and the end sphere's radius is not zero, the end part is a sphere cap, whose center is the end point of the center curve and the intersection plane can be obtained from above computation. Therefore any point on the boundary of the 3D region can be obtained.

Therefore, by computing the boundary points and then triangulate these points, we get a 3D mesh surface which can represent the 3D object. As shown in the following figure, (Fig.5.(a)) is the data spheres used for later interpolation; (Fig.5 (b)) is the result of interpolation and boundary evaluation of the data spheres in the left;(Fig.5(c)) is the rendering effect of the 3D boundary surface which can represent the 3D object.



(a) Data Spheres (b) Interpolating and (c) Rendered boundary surface
Boundary evaluation (mesh)

**Fig. 5.** Interpolation of BBSC and boundary evaluation

### 4.3 System Diagram

Use our system,first,we draw a 2D input use the tablet, then the input points will be reformed, then the skeleton will be extract use CDT algorithm, with the skeleton information use ball B-Spline a 3D model will be create., and many rules will be defined to made our model more complex. The diagram in figure 6 describes the system flow.

## 5 Editing Operations

Our system also provides various editing methods which help the user to generate the free-form objects more conveniently. The editing methods include extrusion, deformation and cutting.

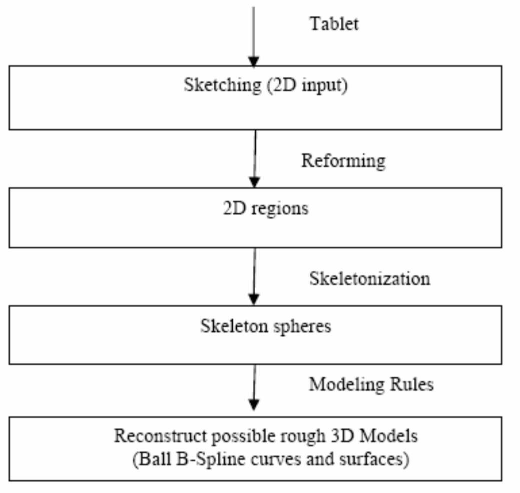


**Fig. 6.** Diagram of our system

### 5.1 Extrusion

Extrusion is an editing method provided for users to generate an extra part added to the original 3D object. Firstly, the users are allowed to draw a stroke on the surface of the already generated objects, which is regarded as the base stroke. Later, the users need to draw a second extrusion stroke just in the panel which is orthogonal to the panel the base stroke surface lies in. Assume the base stroke is similar to a circle, the center point and the corresponding radii can be figured out. Furthermore, by applying the above method of computing the skeleton, the skeleton points and the corresponding radius of the second stroke can be obtained. Then the center and radii of the base stroke is put at the head of the skeleton points list and radius list for following interpolation. Finally, by interpolation these skeleton spheres in the list, the result 3D object with added part can be generated.

The following figure shows some results generated by extrusion.

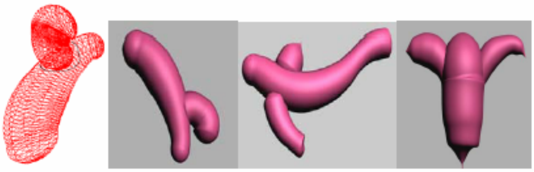
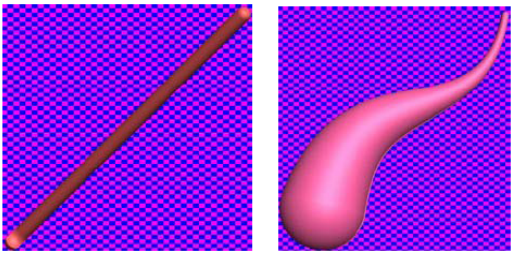


**Fig. 7.** Solid generated by extrusion

### 5.2 Deformation

As the BBSC has very good features for its solid mathematical representation, the deformation can be easily operated by moving its control spheres or data spheres or by changing its control radius or data radius. After movement or changing, the interpolation can be operated again to get the new BBSC.

The following figure shows the deformation effects when generating new objects.



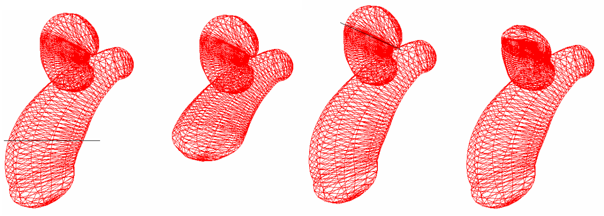
(a) Original object (b) deformed object

**Fig. 8.** Object generated by deformation

### 5.3 Cutting

The system provides also cutting method for users to cut the unwanted part of the solid. The detail operation is drawing a line across the solid and specifying which side besides line should be cut. Then the system can figure out the data spheres within the cutting part and throw them away from the original list. Later just by interpolating these left data spheres, the new solid after cutting can be obtained very easily.

The following figure shows the how the cutting method is used to get new solid. In (Fig.9 (a)), the cross line is drawn and the lower side of the cross line was specified to cut; in (Fig.9 (b)) the upper side of the cross line was specified to throw away.



**(a)** **(b)**

**Fig. 9.** Object generated by cutting method

## 6 Conclusions

As the BBSC have very good features as very sound mathematical representation and precise evaluation. It can be very proper to present 3D free-form objects.

**Acknowledgments.** I would like to thank Dr. Zhongke Wu, my supervisor, for his time, effort, and constant support during this research. His personal enthusiasm and serious attitude to research work benefit me well. I am also thankful to Xuefeng Ao, who gave me a lot of help in my research.

## References

1. Zeleznik, R.C., Herndon, K.P., Hughes, J.F.: SKETCH: an interface for sketching 3D scenes. In: Proceedings of SIGGRAPH 1996, pp. 163–170. ACM, New York (1996)
2. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: A Sketching Interface for 3D Freeform Design. In: SIGGRAPH 1999 Conference Proceedings, pp. 409–416 (1999)
3. Lee, H.J., Chen, Z.: Determination of 3D Human Body Postures from a Single View. Computer Vision, Graphics, and Image Processing 30, 148–168 (1985)
4. Taylor, C.T.: Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image. Computer Vision and Image Understanding 80, 349–363 (2000)
5. Davis, J., Agrawala, M., Chuang, E., Popović, Z., Salesin, D.: A Sketching Interface for Articulated Figure Animation. In: Proceedings of the ACM SIGGRAPH/Euro graphics Symposium on Computer Animation, pp. 320–328 (2003)
6. Tai, C.-L., Zhang, H., Fong, C.-K.: Prototype Modeling from Sketched Silhouettes based on Convolution Surfaces. Computer Graphics Forum (2004)
7. Karpenko, O., Hughes, J.F., Raskar, R.: Free-form sketching with variational implicit surfaces. Computer Graphics Forum (2002)
8. Blum, H.: A transformation for extracting new parameter of shape. Models for the Perception of Speech and Visual Form (1967)
9. Goodman, J.E., O'Rourke, J.: Handbook of Discrete and Computational Geometry. CRC Press, Boca Raton (1997)
10. Piegl, L., Tiller, W.: The NURBS Book. Springer, Heidelberg (1995)
11. Pottmann, H., Peternell, M.: Envelopes - computational theory and applications. In: Falcidieno, B. (ed.) Spring Conference on Computer Graphics 2000. Proceedings of the conference in Budmerice, Comenius University, Bratislava, May 3-6, 2000, pp. 3–23 (2000) ISBN 80-223-1486-2
12. Wu, Z., Seah, H.S., Zhou, M.: Skeleton based Parametric Solid Models: Ball B-Spline Surfaces. In: Proceedings of 10th IEEE International Conference on Computer-Aided Design and Computer Graphics, Beijing, pp. 421–424 (2007)
13. Wu, Z., Zhou, M., Wang, X., Ao, X., Song, R.: An Interactive System of Modeling 3D Trees with Ball B-Spline Curves. In: PMA 2006: The Second International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications. Beijing (CHINA PR), November 13-17 (2006)
14. Prasad, L.: Morphological analysis of shapes. CNLS Newsletter 139, 1–18 (1997)
15. Seah, H.S., Wu, Z.: Ball B-Spline Based Geometric Models in Distributed Virtual Environments. In: Workshop towards semantic virtual environments, SVE, Villans, Switzerland, pp. 1–8 (March 2005)
16. Fei, G., Li, X.: An adaptable sketch-based modeling system. In: Computer-Aided Design and Computer Graphics Conference Proceedings, pp. 371–376 (2007)
17. Masry, M., Lipson Aug, P.H.: A Sketch-Based Interface for Iterative Design and Analysis of 3D Objects. ACM SIGGRAPH 2007 courses (2007)
18. Sheng, B., Wu, E.: Laplacian-based Design: Sketching 3D Shapes. The International Journal of Virtual Reality 5(3), 59–65 (2006)
19. Sun, L., Jin, X., Feng, J., Peng, Q.: Generate 3D Models from 2D Silhouette Curves Using Metaballs. In: CCVRV 2003 (2003)