

DEVELOPING 3D HUMAN MODEL BY USING MATHEMATICAL TOOLS

TEAM MEMBERS

KA-LEUNG MATCH CHU, KA-HEI BOSSA WONG¹

SCHOOL

YUEN LONG MERCHANTS ASSOCIATION SECONDARY SCHOOL

ABSTRACT. In a common personal computer environment, 3D objects are projected on the screen 2-dimensionally, and the project result is based on: (1) The relative position and angle of the object and the observer; (2) Observing method, for example, alpha channel or image distortion. Among most of the 3D objects, human models usually are very high regarded; indeed, creating motive human model is the technique which the game designers want to master, as human are so curious about themselves, and the related mathematics and the technique of computer efficiency are treated as a challenge. Although this is not a fresh topic, obviously, it is still very specialized, we want to express it effectively and simply by mathematics.

Though the research of the construction, appearance design, and the limbs motions of human model, we hope we can extend to other 3D objects, and let the program designer to design a popular 3D model designing tools by using the result of the research.

1. Projecting 3D object to a surface

The technique of 3D projection is very important in 3D programming. For example, if a computer programmer needs to write a game with virtual 3D space, he must master this technique before he can create anything.

This section will discuss the usage of 3D projection in 3D space creation of a computer game.

First, there are three basic elements in 3D projection:

¹This work is done under the supervision of the authors' teacher, Mr. Chi-Keung Chan.

- 1) object;
 - 2) virtual space;
 - 3) observer.
- (1.1)

Among these three elements, Virtual space is absolute, object and observer can be treated as a point of the space. What the programmer needed to do is to grasp the relationship of them, and let the players can act as the observers, and observe the virtual space on a 2D screen.

Therefore the programmer needs to know how to project 3D object on a 2D screen.

First, the coordinate of an object and observer are defined as below:

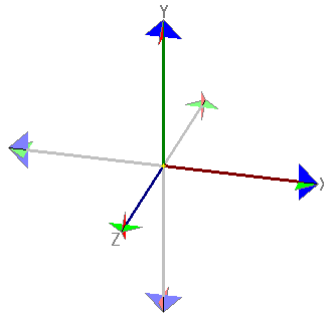


FIGURE 1

Now the object is located at the center of the virtual space, origin(0,0,0), and:

- 1) the positive value of Z -axis as 'Forward';
 - 2) the positive value of Y -axis as 'Upward';
 - 3) the positive value of X -axis as 'Left'.
- (1.2)

Let the coordinate of observer be (0,0,1) and:

- 1) the positive value of Y -axis as 'Upward';
 - 2) the observer is located toward origin.
- (1.3)

Figure 2 show the screen at this setting (without transparency), where the point in virtual space with coordinate $P(x_0, y_0, z_0)$ is displaying on the screen, and the respect displaying coordinate is (x_0, y_0) .

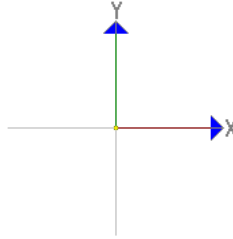


FIGURE 2

1.1. 3D transformation

Before investigating the image obtained when the observer changes the position, we will introduce three matrix formula about 3D transformation, let

$$\begin{aligned}
 R_{X \rightarrow Z}(\theta) &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \\
 R_{Y \rightarrow Z}(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \\
 R_{Y \rightarrow X}(\theta) &= \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{1.4}$$

That means:

(1) When a point $P(x_0, y_0, z_0)$ in a virtual space rotates θ degrees from X -axis to Z -axis by Y -axis, the new coordinate (x_0, y_0, z_0) can be obtained by the following formula:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_{X \rightarrow Z}(\theta) = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{1.5}$$

(2) And from Y -axis to Z -axis by X -axis:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_{Y \rightarrow Z}(\theta) = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (1.6)$$

(3) From Y -axis to X -axis by Z -axis:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_{Y \rightarrow X}(\theta) = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (1.7)$$

Assume the position of the observer change from $(0, 0, 1)$ to $(3, 4, 2)$, then the respect displaying coordinate $P'(x'_i, y'_i, z'_i)$ of any point in virtual space $P(x_i, y_i, z_i)$ can be obtained by any one of the following formulas:

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = R_{Y \rightarrow Z}(\tan^{-1} \frac{4}{\sqrt{2^2 + 3^2}}) R_{X \rightarrow Z}(\tan^{-1} \frac{3}{2}) \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (1.8)$$

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = R_{X \rightarrow Z}(\tan^{-1} \frac{3}{\sqrt{2^2 + 4^2}}) R_{Y \rightarrow Z}(\tan^{-1} \frac{4}{2}) \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (1.9)$$

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = R_{Y \rightarrow Z}(\tan^{-1} \frac{\sqrt{3^2 + 4^2}}{2}) R_{Y \rightarrow X}(\tan^{-1} \frac{3}{4}) \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (1.10)$$

Among these three formulas, (1.8) is the most suitable for calculating according to the way of human thinking based on the conditions (1.2).

Figure 3 shows the geometric meaning of (1.8), and $R_{X \rightarrow Z}(\tan^{-1} \frac{3}{2})$ represents the observer moves from $A(0, 0, \sqrt{2^2 + 3^2 + 4^2})$ to $B(\frac{3\sqrt{2^2 + 3^2 + 4^2}}{2^2 + 3^2}, 0, \frac{2\sqrt{2^2 + 3^2 + 4^2}}{2^2 + 3^2})$, and $R_{Y \rightarrow Z}(\tan^{-1} \frac{4}{\sqrt{2^2 + 3^2}})$ represents the observer moves from B to $C(3, 4, 2)$.

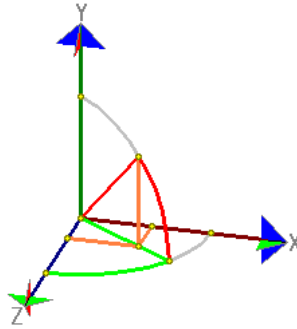


FIGURE 3

Figure 4 shows the screen after transforming:

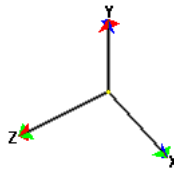


FIGURE 4

If we multiply (1.8) by $R_{Y \rightarrow X}(\theta_z)$, that means it will further rotate θ_z degrees clockwise as shown in Figure 5 :

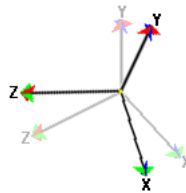


FIGURE 5

Actually the algorithm of this 3D transformation is simulating the way of using photo camera. For example, you are handling a camera and want to move from point A to point B, you first will move horizontally and then

vertically to the target.

If we want to magnify the screen by m times, and generalize (1.8), that means:

- 1) The coordinate of observer is (O_x, O_y, O_z) ;
 - 2) The observer faces (C_x, C_y, C_z)
- (1.11)

and

$$\begin{aligned} \begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} &= \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} R_{Y \rightarrow X}(\theta_z) \\ &R_{Y \rightarrow Z}(\tan^{-1} \frac{O_y - C_y}{\sqrt{(O_x - C_x)^2 + (O_z - C_z)^2}}) \\ &R_{X \rightarrow Z}(\tan^{-1} \frac{O_x - C_x}{O_z - C_z}) \begin{bmatrix} x_i - C_x \\ y_i - C_y \\ z_i - C_z \end{bmatrix} \end{aligned} \quad (1.12)$$

In the view of observer, if we only consider the degree of moving of the default position (observing degree), we can simplify (1.12) as below:

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} R_{Y \rightarrow X}(\theta_z) R_{Y \rightarrow Z}(\theta_x) R_{X \rightarrow Z}(\theta_y) \begin{bmatrix} x_i - C_x \\ y_i - C_y \\ z_i - C_z \end{bmatrix} \quad (1.13)$$

If we require the result become more transparent, we can process x'_i and y'_i in (1.13) as below:

$$\begin{aligned} x''_i &= \frac{x'_i W}{2 \tan \frac{\phi}{2} (D - z'_i)} \\ y''_i &= \frac{y'_i W}{2 \tan \frac{\phi}{2} (D - z'_i)} \end{aligned} \quad (1.14)$$

And W is the width of the screen, ϕ is the viewing angle of the observer. $D = \sqrt{(O_x - C_x)^2 + (O_y - C_y)^2 + (O_z - C_z)^2}$ is the distance between the observer and the observing point. Therefore the screen can only display the point which $D > z'_i$.

1.2. Controlling the view of a 3D virtual space

When we are observing a certain point in the virtual space in a program and want to change the view, drag-and-drop by mouse is a common method by which we can change the viewing angle arbitrarily, in this section, we will construct a controlling algorithm and the respect mathematical formulas.

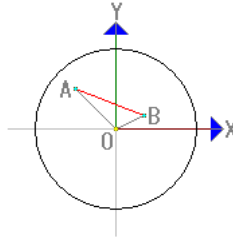


FIGURE 6

In Figure 6, assume the user drag from $A(x_0, y_0)$ to $B(x_1, y_1)$, and $|OA| \leq D$, $|OB| \leq D$. Let A and B be the image of A' and B' respectively, and A' and B' are located on the surface of a sphere with centre $(0, 0, 0)$ and radius D , so the coordinate of A' and B' are:

$$\begin{aligned} A'(x_0, y_0, \sqrt{D^2 - |OA|^2}) \\ B'(x_1, y_1, \sqrt{D^2 - |OB|^2}) \end{aligned} \quad (1.15)$$

When we drag the point, the whole space will rotate certain degrees by an axis passing through the origin, and then A' and B' will move to their original location. (The location of A' and B' is related to the computer screen, that is the centre of this coordinate system is defined by the point that the observer observes, but not the center of the virtual space. In this section, we called this type of coordinate system 'observer space'.)

The following are the calculating algorithm:

(1) Rotate the point in the observer space by Z -axis so that the Y -coordinate of A' becomes 0.

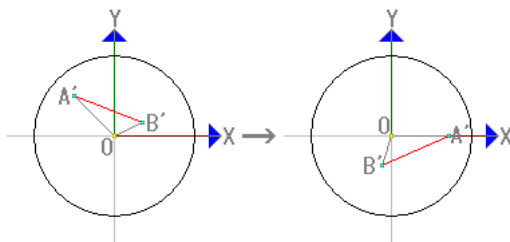


FIGURE 7

(2) Rotate the point in the observer space by Y -axis so that the coordinate of A' becomes $(0, 0, D)$.

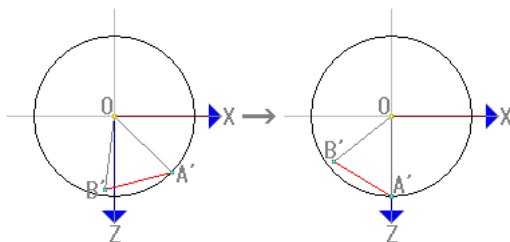


FIGURE 8

(3) Rotate the point in the observer space by Z -axis so that the Y -coordinate of B' becomes 0.

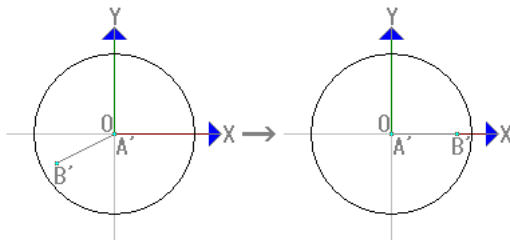


FIGURE 9

(4) Rotate the point in the observer space by Y -axis so that A' can arrive the position of B' .

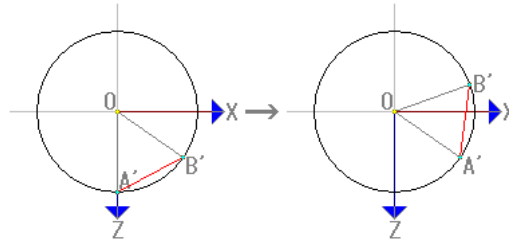


FIGURE 10

(5) Reverse the procedure (3), (2) and (1), so that A' moves to the original position of B' , $(x_1, y_1, \sqrt{D^2 - |OB|^2})$.

From these five procedures, we obtained the formula of a certain point in the observing space $P(x_i, y_i, z_i)$ moves to new coordinate $P'(x'_i, y'_i, z'_i)$ as $A'(x_0, y_0, z_0)$ moved to the original position of $B'(x_1, y_1, z_1)$.

Let:

radius of the sphere = D

$$\theta_1 = \tan^{-1}\left(\frac{y_0}{x_0}\right)$$

$$\theta_2 = \cos^{-1}\left(\frac{z_0}{D}\right)$$

$$\theta_3 = \tan^{-1}\left(\frac{y_T}{x_T}\right)$$

$$\theta_4 = \cos^{-1}\left(\frac{z_T}{D}\right)$$

$$\begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix} = R_{X \rightarrow Z}(\theta_2) R_{Y \rightarrow Z}(\theta_1) \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

then:

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = R_{Y \rightarrow Z}(-\theta_1) R_{X \rightarrow Z}(-\theta_2) R_{Y \rightarrow X}(-\theta_3) R_{X \rightarrow Z}(-\theta_4) \\ R_{Y \rightarrow X}(\theta_3) R_{X \rightarrow Z}(\theta_2) R_{Y \rightarrow X}(\theta_1) \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (1.16)$$

Obviously, if every point in the observer space needs to calculate, the amount of calculation will be very huge, and observer space is the projection of virtual space, if we change the position of the point in observer space directly, the projection relationship will be terminated, and we cannot change the view anymore, so this formula is used to find the position of observer in virtual space only.

First, the observer faces the origin, and the distance to the origin is 1, assume we do not know the observing angle, that is $\theta_x, \theta_y, \theta_z$ in (1.13), but we know the position of the points $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$ in the observer space are (x_0, y_0, z_0) , (x_1, y_1, z_1) , (x_2, y_2, z_2) respectively, so we can use the following simultaneous equation to find out $\theta_x, \theta_y, \theta_z$.

$$\left\{ \begin{aligned} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} &= \begin{bmatrix} \cos \theta_y \cos \theta_z - \sin \theta_y \sin \theta_x \sin \theta_z \\ -\cos \theta_y \sin \theta_z - \sin \theta_y \sin \theta_x \cos \theta_z \\ \sin \theta_y \cos \theta_x \end{bmatrix} \\ &= R_{Y \rightarrow X}(\theta_z) R_{Y \rightarrow Z}(\theta_x) R_{X \rightarrow Z}(\theta_y) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \right. \quad (1.17a)$$

$$\left\{ \begin{aligned} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} &= \begin{bmatrix} \cos \theta_x \sin \theta_z \\ \cos \theta_x \cos \theta_z \\ \sin \theta_x \end{bmatrix} \\ &= R_{Y \rightarrow X}(\theta_z) R_{Y \rightarrow Z}(\theta_x) R_{X \rightarrow Z}(\theta_y) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \end{aligned} \right. \quad (1.17b)$$

$$\left\{ \begin{aligned} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} &= \begin{bmatrix} -\sin \theta_y \cos \theta_z - \cos \theta_y \sin \theta_x \sin \theta_z \\ \sin \theta_y \sin \theta_z - \cos \theta_y \sin \theta_x \cos \theta_z \\ \cos \theta_y \cos \theta_x \end{bmatrix} \\ &= R_{Y \rightarrow X}(\theta_z) R_{Y \rightarrow Z}(\theta_x) R_{X \rightarrow Z}(\theta_y) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \right. \quad (1.17c)$$

When the viewing angle changes, we can substitute (x_0, y_0, z_0) , (x_1, y_1, z_1) , (x_2, y_2, z_2) into (1.16) to obtain (x'_0, y'_0, z'_0) , (x'_1, y'_1, z'_1) , (x'_2, y'_2, z'_2) , and substitute them into (1.17) to obtain new viewing angle $\theta_x, \theta_y, \theta_z$, so that we can use the new viewing angle to calculate the projection image of the virtual space.

That's all of the method of controlling the view of observer.

2. Constructing curves on a plane surface

In section 1, we have obtained the formula about the relationship between virtual space and observers. In this section, we will investigate a basic relationship between virtual space and object: Creating objects in the virtual space. In practice, some basic mathematical functions or its varied form such as spherical surface formulas $x^2 + y^2 + z^2 = R^2$, hyperbolic paraboloid formulas $y^2 = x^2 - z^2$ are not enough. To create objects in virtual space, a curved surface formula is necessary. But before calculating complex curved surface formula, it is essential for us to master the 2D curve, and this will help 3D model designer to understand the characteristic of curved surface

and basic requirement of curved surface, also help us to master some basic mathematical methods. So the main theme of this section is curves.

2.1. Requirements of constructing a curve

First, we need to decide an operating method to construct curves, after comparing with some commercial software such as Photoshop, painter, the following method is the most efficient and convenient[1]:

First, drag point A to B' , and B will appear which $\overrightarrow{AB'} = -\overrightarrow{AB}$, then drag D to C' , and C will appear which $\overrightarrow{DC'} = -\overrightarrow{DC}$. Finally a curve is constructed according to point A, B, C and D shown in Figure 11.

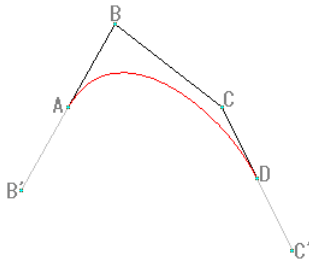


FIGURE 11

This curve has the following characteristics:

- (1) Starts from point A , and ends at point D .
- (2) Initial direction is same as \overrightarrow{AB} , and ending direction is same as \overrightarrow{CD} .

Now, if we drag G to F' , F will appear which $\overrightarrow{GF'} = -\overrightarrow{GF}$, so another curve defined by D, C', G, F will appear in Figure 12.

We can see that if we continuously drag the points, we can easily and accurately construct a series of smooth curves.

After confirming the operating method, we need to find the mathematical formula which is suitable to confirm the shape of the curve. And a suitable mathematical formula can be found in the following two ways:

- (i) Find a formula which can represent a curve defined by A, B, C and D four points.

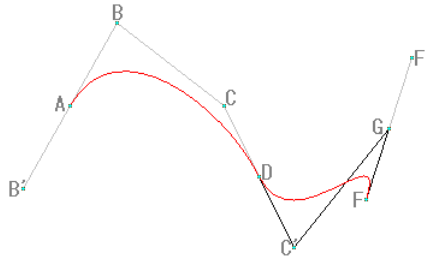


FIGURE 12

(ii) Add a point E other than A, B, C, D , and find a formula which can represent two part of curves defined by A, B, E and E, C, D respectively, and two curves are joint at E and E is on BC where the slopes of these two curves at E are the same.

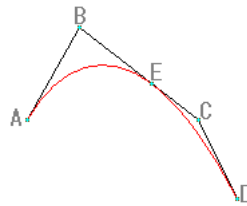


FIGURE 13. Method 2

We choose method (ii) to investigate, as there are three reasons:

- (1) The research can be done from simple to complex sequentially; first we find some formulas which represent the curves that can be defined by 3 points only, and try to convert the formulas to 4-point version.
- (2) The curve mentioned in method (ii) can be treated as two curves defined by three points, and it can also be treated as a curve defined by five points, so the shape of the curve will be more various than the curves that directly defined by four points.
- (3) The curve mentioned in method (i) has been used by common commercial drawing software nowadays: Bezier Curve [1]. (The curve in Figure 11 is Bezier Curve). This curve has been introduced long time ago, but after comparing the curves in figure 11 and figure 13, we hope these 4-point-defined curves can also be tangent to BC at certain point that can let the

users control the position of the curve easier, moreover, after considering the situation that BCD lies on the same straight line, we believe that only the curve constructed by method (ii) can satisfy our requirement.

So in this section, we need to consider the following problems:

Let ABC be an triangle, if we need to construct a smooth curve which satisfies the following conditions:

- 1) Start from A and end at B ;
- 2) Initial direction is same as \overrightarrow{AB} , and ending direction is same as \overrightarrow{CD} .

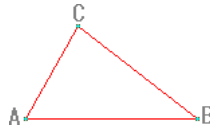


FIGURE 14

2.2. The methods of constructing curves

2.2.1. Method 1 - Circle array

In the view of geometry, we first consider the method of constructing curve by the arc of circle as below:

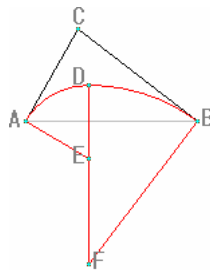


FIGURE 15. Circle array

The curve was drawn as the figure, the curve was formed by two parts of arc

of the circle, E and F are the centers of two arcs, and D is the connecting point of the arcs. As it is required that the slopes of two arc at D are equal, so D, E, F three points will be on the same straight line. Obviously, a curve can be constructed if we locate the position of D .

Now we consider the position of D :

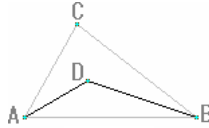


FIGURE 16

Let

$$\begin{aligned}
 \angle CAB &= a \\
 \angle ABC &= b \\
 \angle BCA &= c \\
 \angle DAB &= p \\
 \angle DBA &= q
 \end{aligned} \tag{2.1}$$

From Figure 16, we know that:

$$\angle ADB = \angle DAE + \angle DBF \tag{2.2}$$

$$\angle ABD = 180^\circ - (p + q) \tag{2.3}$$

Since $\angle CAE = \angle CBE = 90^\circ$, we have

$$\begin{aligned}
 \angle BAE &= 90^\circ - a \\
 \angle ABF &= 90^\circ - b
 \end{aligned} \tag{2.4}$$

$$\angle DAE = 90^\circ - a + p$$

$$\angle DBF = 90^\circ - b + q$$

Therefore,

$$\begin{aligned}
 \angle ADB &= 180^\circ - (a + b) + (p + q) \\
 &= c + (p + q)
 \end{aligned} \tag{2.5}$$

$$\begin{aligned}
 180^\circ - (p + q) &= c + (p + q) \\
 p + q &= \frac{180^\circ - c}{2}
 \end{aligned} \tag{2.6}$$

$$\angle ADB = \frac{180^\circ + c}{2} \tag{2.7}$$

From (2.7), we know that D is a moving point with a circle arc locus, and the shape as shown in Figure 17.

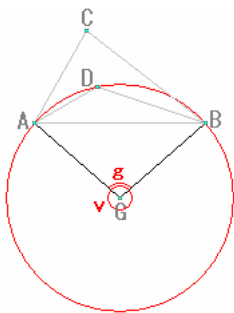


FIGURE 17

Because angle at centre is twice of angle at circumference,

$$v = 180^\circ + c \tag{2.8}$$

$$g = 180^\circ - c \tag{2.9}$$

Therefore, A, C, B, G are concyclic.

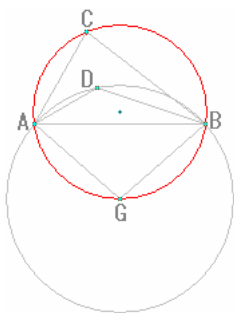


FIGURE 18. A, C, B, G are concyclic.

From the above result, we can find out the position of G by the following method:

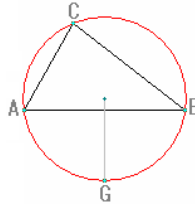


FIGURE 19. Circumscribed circle

1) Draw the circumscribed circle of triangle ABC , and let the centre of the arc AB be G .

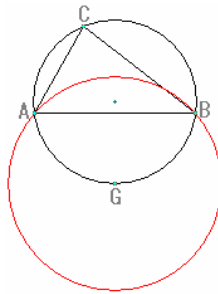


FIGURE 20. Circle with radius AG and center G

2) Draw a circle with radius AG and center G .

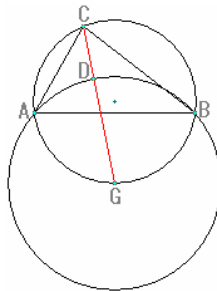


FIGURE 21. The intersection point of CG and the big circle should be considered.

3) As D is necessarily inside triangle ABC , in order to choose the best point D , the intersection point of CG and the big circle should be considered.

As the angle at circumference and the arc length are directly proportional, CG is the angle bisector of angle ACB .

But the location of D is still not trivial enough, so we rearrange the data:

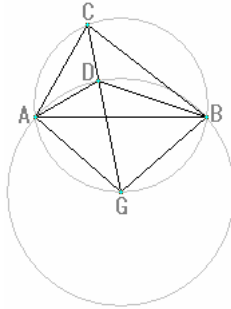


FIGURE 22

Because the angles are equal in same segment,

$$\angle CGB = \angle CAB = a \quad (2.10)$$

$$\angle CGA = \angle CBA = b \quad (2.11)$$

$$\angle BAG = \angle BCG = \frac{c}{2} \quad (2.12)$$

$$\angle GDA = \angle GAD = \frac{180^\circ - b}{2} \quad (2.13)$$

$$\angle DAB = \frac{180^\circ - b}{2} - \frac{c}{2} = \frac{a}{2} \quad (2.14)$$

Similarly,

$$\angle DBA = \frac{b}{2} \quad (2.15)$$

Therefore D is the center of the inscribed circle of triangle ABC .

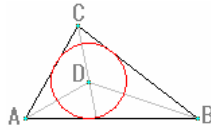


FIGURE 23. D is the center of the inscribed circle of triangle ABC .

2.2.2. Method 2 - Elliptic arc

From method 1, in order to construct the curve which satisfies the conditions of Figure 15, at most two circle arcs are needed, and only one is needed if we switch to elliptic arc. And Figure 24 shows the concept:

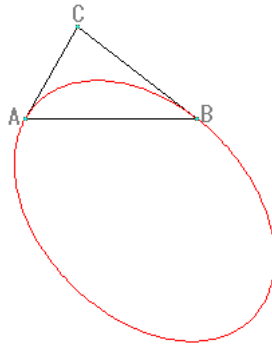


FIGURE 24. Elliptic arc

Actually, we can treat ellipse as “Distorted circle” as shown in Figure 25.

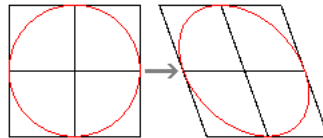


FIGURE 25. Distorted circle

We have the process as below:

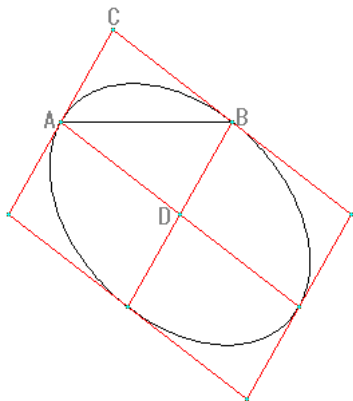


FIGURE 26. Parallelogram

Add a point D inside the ellipse which $ABCD$ form a parallelogram, and extend the parallelogram for four times to gain a larger parallelogram, so that D is the center of the large parallelogram and the inscribed ellipse. Finally we need to find the parameter of the ellipse.

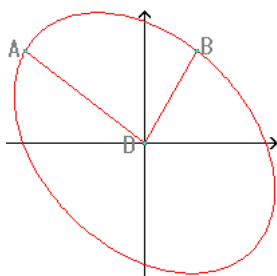


FIGURE 27. Ellipse A

Let the ellipse be a functional curve Ellipse A (Figure 27) on the coordinate system, and let the coordinate as below:

$$\begin{aligned}
 &A(a, b) \\
 &B(c, d) \\
 &C(a + c, b + d) \\
 &D(0, 0)
 \end{aligned} \tag{2.16}$$

And we can know that the slope at point A and B are

$$\begin{aligned}
 A &: \frac{d}{c} \\
 B &: \frac{b}{a}
 \end{aligned} \tag{2.17}$$

Let ellipse A be

$$Px^2 + Qy^2 + Rxy = 1 \tag{2.18}$$

By differentiating both sides, we have

$$2Pxdx + 2Qydy + R(xdy + ydx) = 0 \tag{2.19}$$

And we have the following simultaneous equations to solve for P , Q and R :

$$\begin{cases}
 Pa^2 + Qb^2 + Rab = 1 & (2.20a) \\
 Pc^2 + Qd^2 + Rcd = 1 & (2.20b) \\
 2Pac + 2Qbd + R(ad + bc) = 0 & (2.20c)
 \end{cases}$$

By solving, we have

$$\begin{aligned}
 P &= \frac{b^2 + d^2}{(ad - bc)^2} \\
 Q &= \frac{a^2 + c^2}{(ad - bc)^2}
 \end{aligned} \tag{2.21}$$

Sub P and Q into (2.20c):

$$\begin{aligned}
 R &= -\frac{2(Pac + Qbd)}{ad + bc} \\
 &= -\frac{2[(b^2 + d^2)ac + (a^2 + c^2)bd]}{(ad + bc)(ad - bc)^2} \\
 &= -2(ab + cd)
 \end{aligned} \tag{2.22}$$

And Ellipse A can be rewritten as:

$$(b^2 + d^2)x^2 + (a^2 + c^2)y^2 - 2(ab + cd)xy = (ad - bc)^2 \quad (2.23)$$

$$(ay - bx)^2 + (cy - dx)^2 = (ad - bc)^2 \quad (2.24)$$

The formula of Ellipse A has been found, but it cannot be used directly in curve constructing as:

- (1) The coordinate of drawing point cannot be found easily;
- (2) The start point and end point of the curve cannot not be located easily.

The best way is to transform (2.24) into parametric function:

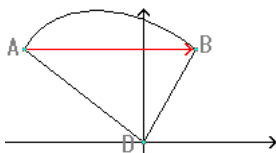


FIGURE 28

The parametric formula of a straight line starting from A to B : ($0 \leq t \leq 1$)

$$\begin{aligned} x &= (1 - t)a + tc \\ y &= (1 - t)b + td \end{aligned} \quad (2.25)$$

Sub (2.25) into LHS of (2.24)

$$\begin{aligned} & \{a[(1 - t)b + td] - b[(1 - t)a + tc]\}^2 + \{c[(1 - t)b + td] \\ & \quad - d[(1 - t)a + tc]\}^2 \\ &= [ab + a(d - b)t - ab - b(c - a)t]^2 + [cb + c(d - b)t - ad - d(c - a)t]^2 \\ &= (ad - bc)^2 t^2 + [(cb - ad) - (cb - ad)t]^2 \\ &= (ad - bc)^2 (2t^2 - 2t + 1) \end{aligned} \quad (2.26)$$

Comparing (2.26) with RHS of (2.24), the parametric formula of drawing arc AB can be gained as below:

$$\begin{aligned}x &= \frac{(1-t)a + tc}{\sqrt{2t^2 - 2t + 1}} \\y &= \frac{(1-t)b + td}{\sqrt{2t^2 - 2t + 1}}\end{aligned}\tag{2.27}$$

The obstacle met in (2.24) has been overcome.

Finally, substitute the arbitrary values into A , B , C .

$$\begin{aligned}A(x_0, y_0) \\B(x_1, y_1) \\C(x_2, y_2)\end{aligned}\tag{2.28}$$

$$\text{The coordinate of } D = (x_0 + x_1 - x_2, y_0 + y_1 - y_2)\tag{2.29}$$

Therefore, the parameter in (2.27) should be changed into:

$$\begin{aligned}a &= x_2 - x_1 \\b &= y_2 - y_1 \\c &= x_2 - x_0 \\d &= y_2 - y_0\end{aligned}\tag{2.30}$$

And the result:

$$\begin{aligned}x &= x_0 + x_1 - x_2 + \frac{(1-t)(x_2 - x_1) + t(x_2 - x_0)}{\sqrt{2t^2 - 2t + 1}} \\y &= y_0 + y_1 - y_2 + \frac{(1-t)(y_2 - y_1) + t(y_2 - y_0)}{\sqrt{2t^2 - 2t + 1}}\end{aligned}\tag{2.31}$$

From these two methods, we know that if we switch to common conic curves to be a curve constructing tool, we can guarantee the curves drawn satisfy the conditions of Figure 15, also by controlling the eccentricity e we can change the bending rate of the curve as shown in Figure 29:

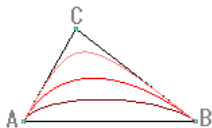


FIGURE 29. controlling the eccentricity

However, it is tedious and ineffective if we use this method to find the suitable parametric function of conic formula, so we will not consider here.

2.2.3. Method 3 - Distortion of circle

In method 2, we have mentioned that “Ellipse may be treated as a distorted circle”, beside ellipse, actually we have also tried to distort known circle formula as below:

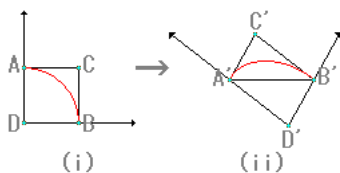


FIGURE 30

As shown in Figure 30, “Distortion” is only the transformation of coordinate system, if the transformed curve (coordinate system (ii)) can be displayed directly on the coordinate system (i), that means the aim is achieved.

First, let the formula of curve in coordinate system (i) be

$$y = f(x) = \sqrt{1 - x^2} \quad (0 \leq x \leq 1) \tag{2.32}$$

Let $P(x_i, y_i)$ be a point on the curve, then:

$$\overrightarrow{DP} = t\overrightarrow{DB} + f(t)\overrightarrow{DA} \quad (0 \leq t \leq 1) \tag{2.33}$$

Similarly in Figure 2.2.3.2, let $P'(x_j, y_j)$ be a point on the curve of coordinate system (ii), then:

$$\overrightarrow{D'P'} = t\overrightarrow{D'B'} + f(t)\overrightarrow{D'A'} \quad (0 \leq t \leq 1) \tag{2.34}$$

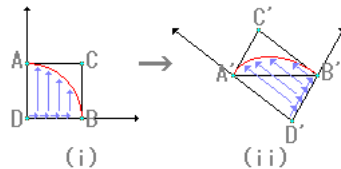


FIGURE 31

So we can easily find out the parametric formula of distorted curve now. Let the coordinate of A', B', C', D' in coordinate system (i) be

$$\begin{aligned} &A(x_0, y_0) \\ &B(x_1, y_1) \\ &C(x_2, y_2) \\ &D(x_0 + x_1 - x_2, y_0 + y_1 - y_2) \end{aligned} \tag{2.35}$$

By (2.34), we can get a parametric formula:

$$\begin{aligned} X &= x_0 + x_1 - x_2 + t(x_2 - x_0) + f(t)(x_2 - x_1) \\ Y &= y_0 + y_1 - y_2 + t(y_2 - y_0) + f(t)(y_2 - y_1) \end{aligned} \tag{2.36}$$

This method can also be used on the others common functional curve, such as $y = x^n$ or $y = \sqrt[3]{1 - x^n}$. Before them, we can find more other methods of distorting the circle.

This circle distorting method has a weak point, as the formulas need to run by computer, and then the variable t in the formulas will be substituted by a sequence of increasing numbers from 0 to 1, for example, if it is required that the curve is formed by five straight line segment, the program will substitute the following numbers into t :

$$0, 0.2, 0.4, 0.6, 0.8, 1 \tag{2.37}$$

And the result as Figure 32.

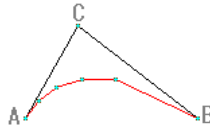


FIGURE 32. Curve formed by five straight line segments

We can see that the lengths of the line segments vary, and we have found two methods to improve the situation:

(1) Replace t with

$$\begin{aligned} & \sin\left(\frac{\pi t}{2}\right) \quad (\text{Upper curve in Figure 33}) \\ \text{or } & \left(\frac{\pi}{2} - 2\right)t^3 - (\pi - 3)t^2 + \frac{\pi}{2}t \quad (\text{Lower curve in Figure 33}). \end{aligned} \quad (2.38)$$

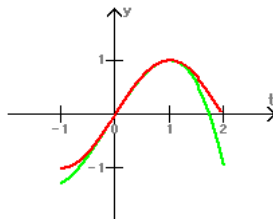


FIGURE 33

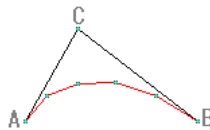


FIGURE 34. Result

(2) Change the distortion method by changing the grouping method of vectors of Figure 31 as Figure 35:

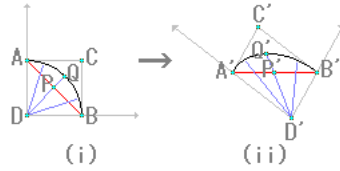


FIGURE 35

Let $\overrightarrow{AP} = t\overrightarrow{AB}$, then

$$|\overrightarrow{DP}| = \sqrt{2t^2 - 2t + 1}. \tag{2.39}$$

Therefore,

$$\begin{aligned} \overrightarrow{DQ} &= \frac{\overrightarrow{DP}}{\sqrt{2t^2 - 2t + 1}} \\ &= \frac{\overrightarrow{DA} + t\overrightarrow{AB}}{\sqrt{2t^2 - 2t + 1}}. \end{aligned} \tag{2.40}$$

Similarly, the parametric function can be made by making the assumption same as (2.35):

Since

$$\begin{aligned} \overrightarrow{OQ'} &= \overrightarrow{OD'} + \overrightarrow{D'Q'} \\ &= \overrightarrow{OD'} + \frac{\overrightarrow{D'A} + t\overrightarrow{A'B'}}{\sqrt{2t^2 - 2t + 1}}, \end{aligned} \tag{2.41}$$

we have

$$\begin{aligned} X &= x_0 + x_1 - x_2 + \frac{x_2 - x_1 + t(x_1 - x_0)}{\sqrt{2t^2 - 2t + 1}} \\ Y &= y_0 + y_1 - y_2 + \frac{y_2 - y_1 + t(y_1 - y_0)}{\sqrt{2t^2 - 2t + 1}} \end{aligned} \tag{2.42}$$

and the result is the same as (2.31).

2.2.4. Method 4 - Distortion of general function

As mentioned in method 3, curves such as $y = x^n$ or $y = \sqrt[n]{1 - x^n}$ can also be distorted. Actually every formula of $y = f(x)$ can be distorted in same

way if they satisfy the following conditions:

$$\begin{aligned} 1) f(0) &= 0 \\ 2) f'(0) &= 0 \end{aligned} \tag{2.43}$$

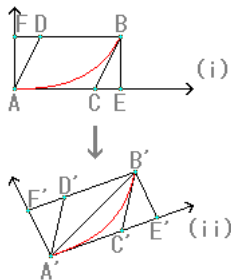


FIGURE 36

In Figure 36, let the coordinates of the points in coordinate system (i) be

$$\begin{aligned} A(0, 0) \\ B(a, f(a)) \\ C\left(a - \frac{f(a)}{f'(a)}, 0\right) \\ D\left(\frac{f(a)}{f'(a)}, f(a)\right) \\ E(a, 0) \\ F(0, f(a)) \end{aligned} \tag{2.44}$$

Let $P(x_i, y_i)$ be a point on the curves, then

$$\begin{aligned}
 \overrightarrow{AP} &= \frac{x_i}{a} \overrightarrow{AE} + \frac{f(x_i)}{f(a)} \overrightarrow{EB} \\
 &= \frac{x_i}{a} \overrightarrow{AC} + \frac{x_i}{a} \overrightarrow{CE} + \frac{f(x_i)}{f(a)} \overrightarrow{EB} \\
 &= \frac{x_i}{a} \overrightarrow{AC} + \left(\frac{x_i}{a} - \frac{f(x_i)}{f(a)} \right) \overrightarrow{CE} + \frac{f(x_i)}{f(a)} (\overrightarrow{CE} + \overrightarrow{EB}) \\
 &= \frac{x_i}{a} \overrightarrow{AC} + \left(\frac{x_i}{a} - \frac{f(x_i)}{f(a)} \right) \left(\frac{\frac{f(a)}{f'(a)}}{a - \frac{f(a)}{f'(a)}} \right) \overrightarrow{AC} + \frac{f(x_i)}{f(a)} (\overrightarrow{CB}) \\
 &= \frac{x_i f'(a) - f(x_i)}{a f'(a) - f(a)} \overrightarrow{AC} + \frac{f(x_i)}{f(a)} \overrightarrow{CB}
 \end{aligned} \tag{2.45}$$

And we make the assumption same as (2.35), and let $P'(at, f(at))$ be a point on the curve of the coordinate system (ii), then by

$$\begin{aligned}
 \overrightarrow{AP'} &= \overrightarrow{AA'} + \overrightarrow{A'P'} \\
 &= \overrightarrow{AA'} + \frac{at f'(a) - f(at)}{a f'(a) - f(a)} \overrightarrow{A'C'} + \frac{f(at)}{f(a)} \overrightarrow{C'B'} \quad (0 \leq t \leq 1)
 \end{aligned} \tag{2.46}$$

We can write down the parametric formula:

$$\begin{aligned}
 X &= x_0 + \frac{at f'(a) - f(at)}{a f'(a) - f(a)} (x_2 - x_0) + \frac{f(at)}{f(a)} (x_1 - x_2) \\
 Y &= y_0 + \frac{at f'(a) - f(at)}{a f'(a) - f(a)} (y_2 - y_0) + \frac{f(at)}{f(a)} (y_1 - y_2)
 \end{aligned} \tag{2.47}$$

Because a is treated as a constant, (2.47) can be rewritten as

$$\begin{aligned}
 X &= P_x + Q_x t + R_x g(t) \\
 Y &= P_y + Q_y t + R_y g(t)
 \end{aligned} \tag{2.48}$$

where

$$\begin{aligned}
 g(t) &= f(at) \\
 P_x &= x_0 \\
 Q_x &= \frac{g'(1)(x_2 - x_0)}{g'(1) - g(1)} \\
 R_x &= \frac{x_1 - x_2}{g(1)} - \frac{x_2 - x_0}{g'(1) - g(1)} \\
 P_y &= y_0 \\
 Q_y &= \frac{g'(1)(y_2 - y_0)}{g'(1) - g(1)} \\
 R_y &= \frac{y_1 - y_2}{g(1)} - \frac{y_2 - y_0}{g'(1) - g(1)}
 \end{aligned} \tag{2.49}$$

However, because the method of finding this formula is similar to (2.36), and it has the same problem (Vary straight line segment length), and it motives us to find another curve.

2.2.5. Method 5 - Distortion of conic curve

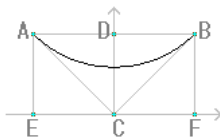


FIGURE 37

In Figure 31, if we change the circle arc into a common conic curve, a curve which can be distorted more free will be gained. Let ConeA be a functional curve (and straight line $x = 0$ be the axis of symmetry) (Figure 37) which

$$Px^2 + Qy^2 + Rx + Sy + Txy = 1 \tag{2.50}$$

By differentiating both sides, we have

$$2Pxdx + 2Qydy + Rdx + Sdy + T(dx + dy) = 0 \tag{2.51}$$

And the curve satisfies the following conditions:

- 1) Tangent with $y = x$ at $(1, 1)$
 - 2) Tangent with $y = -x$ at $(-1, 1)$
- (2.52)

Then the simultaneous equation can be gained:

$$\begin{cases} P + Q + R + S + T = 1 & (2.53a) \\ P + Q + R - S - T = 1 & (2.53b) \\ 2P + 2Q + R + S + 2T = 0 & (2.53c) \\ 2P + 2Q - R + S = 0 & (2.53d) \end{cases}$$

Therefore,

$$\begin{aligned} R &= T = 0 \\ S &= 2 \\ P &= -(Q + 1) \end{aligned}$$

$$\begin{aligned} Qy^2 + 2y - [1 + (Q + 1)x^2] &= 0 \\ y &= \frac{\sqrt{(1 + Q)(1 + Qx^2)} - 1}{Q} \end{aligned} \quad (2.54)$$

Let a certain point $(0, k)$ on the curve ConeA, then

$$k = \frac{\sqrt{1 + Q} - 1}{Q}. \quad (2.55)$$

So (2.54) can be rewritten as

$$\begin{aligned} y = f(x) &= \frac{\sqrt{(1 - k)^2 [k^2 + (1 - 2k)x^2]} - k^2}{1 - 2k} \quad (0 \leq k \leq 1, k \neq 0.5) \\ \text{or } y = f(x) &= \frac{x^2 + 1}{2} \quad (k = 0.5). \end{aligned} \quad (2.56)$$

Let $V(2t - 1, f(2t - 1))$ be a point on ConeA, then:

$$\begin{aligned}
 \overrightarrow{AV} &= \overrightarrow{AE} + \overrightarrow{EV} \\
 &= -\overrightarrow{EA} + t\overrightarrow{EF} + f(2t - 1)\overrightarrow{EA} \\
 &= t\overrightarrow{AB} + [f(2t - 1) - 1]\overrightarrow{CD} \\
 &= t\overrightarrow{AB} + [f(2t - 1) - 1](\overrightarrow{CA} + \frac{1}{2}\overrightarrow{AB}) \quad (0 \leq t \leq 1) \quad (2.57)
 \end{aligned}$$

Now, change the coordinates of A, B, C to

$$\begin{aligned}
 &A(x_0, y_0) \\
 &B(x_1, y_1) \\
 &C(x_2, y_2) \quad (2.58)
 \end{aligned}$$

So the respect parametric formula of distorted conic curve will be

$$\begin{aligned}
 X &= x_0 + t(x_1 - x_0) + [f(2t - 1) - 1]\left(\frac{x_1 + x_0}{2} - x_2\right) \\
 Y &= y_0 + t(y_1 - y_0) + [f(2t - 1) - 1]\left(\frac{y_1 + y_0}{2} - y_2\right) \quad (2.59)
 \end{aligned}$$

We can see that if we change the value of k in (2.56) when we constructing the curve, the bending rate of the curve will be changed greatly. (Reference to Figure 29)

2.2.6. Method 6 - Parametric formula of polynomial

The results gained by method 2–5 have a common characteristic, all of their curves are represented by parametric formula. This phenomenon is due to the calculating method. And this type of formula can construct the curve which the formula of common type such as $y = f(x)$ cannot construct. Because it has such an advantage, in this section, we will try to find out a new parametric formula polynomial directly.

Let A, B, C be three points on the ordinary coordinate system

$$\begin{aligned} &A(x_0, y_0) \\ &B(x_1, y_1) \\ &C(x_2, y_2) \end{aligned} \tag{2.60}$$

And there exists a parametric formula

$$\begin{aligned} X &= f_1(t) = a_1t^2 + b_1t + c_1 \\ Y &= f_2(t) = a_2t^2 + b_2t + c_2 \quad (0 \leq t \leq 1) \end{aligned} \tag{2.61}$$

which

$$\begin{aligned} f_1(0) &= x_0 & f_1(1) &= x_1 \\ f_2(0) &= y_0 & f_2(1) &= y_1 \\ f'_1(0) &= p(x_2 - x_0) & f'_1(1) &= q(x_2 - x_1) \\ f'_2(0) &= p(y_2 - y_0) & f'_2(1) &= q(y_2 - y_1) \end{aligned} \tag{2.62}$$

Therefore,

$$\begin{aligned} f_1(t) &= x_0(1-t)G_1(t) + x_1tG_2(t) + c_1t(t-1)x_2 \\ f_2(t) &= y_0(1-t)H_1(t) + y_1tH_2(t) + c_2t(t-1)y_2 \end{aligned} \tag{2.63}$$

Also,

$$\begin{aligned} G_1(0) &= 1 & G_2(1) &= 1 \\ H_1(0) &= 1 & H_2(1) &= 1 \end{aligned} \tag{2.64}$$

Therefore,

$$\begin{aligned} f'_1(t) &= x_0(1-t)G'_1(t) - x_0G_1(t) + x_1tG'_2(t) + x_1G_2(t) + c_1(2t-1)x_2 \\ f'_2(t) &= y_0(1-t)H'_1(t) - y_0H_1(t) + y_1tH'_2(t) + y_1H_2(t) + c_1(2t-1)y_2 \end{aligned} \tag{2.65}$$

$$\begin{aligned}
p(x_2 - x_0) &= x_0 G_1'(0) - x_0 + x_1 G_2(0) - c_1 x_2 \\
q(x_2 - x_1) &= -x_0 G_1(1) + x_1 G_2'(1) + x_1 + c_1 x_2 \\
p(y_2 - y_0) &= y_0 H_1'(0) - y_0 + y_1 H_2(0) - c_2 y_2 \\
q(y_2 - y_1) &= -y_0 H_1(1) + y_1 H_2'(1) + y_1 + c_2 y_2
\end{aligned} \tag{2.66}$$

$$\begin{aligned}
G_1 &= (0) & G_2(0) &= 0 \\
H_1 &= (0) & H_2(0) &= 0 \\
1 - G_1'(0) &= 1 - H_1'(0) = p \\
1 + G_2'(1) &= 1 + H_2'(1) = -q \\
c_1 = c_2 &= -p = q
\end{aligned} \tag{2.67}$$

Since $\deg(f_1(t)) = \deg(f_2(t)) = 2$, we have

$$\begin{aligned}
G_1'(t) &= H_1'(t) = 1 - p \\
G_2'(t) &= H_2'(t) = -(1 + q)
\end{aligned} \tag{2.68}$$

Then

$$\begin{aligned}
G_1(t) &= H_1(t) = (1 - p)t + k \\
G_2(t) &= H_2(t) = -(1 + q)t
\end{aligned} \tag{2.69}$$

Therefore,

$$k = 1, \quad p = 2, \quad q = -2 \tag{2.70}$$

$$\begin{aligned}
f_1(t) &= x_0(1 - t)^2 + x_1 t^2 + 2t(1 - t)x_2 \\
f_2(t) &= y_0(1 - t)^2 + y_1 t^2 + 2t(1 - t)y_2
\end{aligned} \tag{2.71}$$

2.2.7. Method 7 - Bezier Curve

Let four points:

$$\begin{aligned}
 &A(x_0, y_0) \\
 &B(x_1, y_1) \\
 &C(x_2, y_2) \\
 &D(x_3, y_3)
 \end{aligned}
 \tag{2.72}$$

If we need to construct a curve which

- 1) Start from A and ends at D ;
 - 2) Initial direction is same as \overrightarrow{AB} , and ending direction is same as \overrightarrow{CD} ,
- (2.73)

then we can use the parametric formula:

$$\begin{aligned}
 X &= x_0(1-t)^3 + 3x_1(1-t)^2t + 2x_2(1-t)t^2 + x_3t^3 \\
 Y &= y_0(1-t)^3 + 3y_1(1-t)^2t + 2y_2(1-t)t^2 + y_3t^3
 \end{aligned}
 \tag{2.74}$$

to define a curve shown in Figure 38:

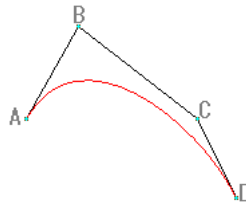


FIGURE 38

The curve defined by above method is a type of the Cubic Bezier Curves. Actually (2.71) is Quadratic Bezier Curves [1].

Although we will not use Bezier curve as the tool of constructing 4-point defined

curve, we can remake the Bezier curve, and become a tool similar to method 5, that means we can change the value of k in the formula to tune the rate of bending.

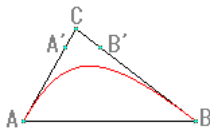


FIGURE 39

We have designed the method of constructing curves as shown in Figure 39:

- 1) Add a point A' on AC that $\overrightarrow{AA'} = k\overrightarrow{AC}$
- 2) Add a point B' on BC that $\overrightarrow{BB'} = k\overrightarrow{BC}$ (2.75)

And draw a Bezier curve defined by A, A', B', B four points. Therefore if the coordinates of A, B, C are

$$\begin{aligned} A(x_0, y_0) \\ B(x_1, y_1) \\ C(x_2, y_2) \end{aligned} \tag{2.76}$$

The following parametric formula can be gained:

$$\begin{aligned} X &= x_0(1-t)^3 + 3[x_0 + k(x_2 - x_0)](1-t)^2t \\ &\quad + 3[x_1 + k(x_2 - x_1)](1-t)t^2 + x_1t^3 \\ Y &= y_0(1-t)^3 + 3[y_0 + k(y_2 - y_0)](1-t)^2t \\ &\quad + 3[y_1 + k(y_2 - y_1)](1-t)t^2 + y_1t^3 \end{aligned} \tag{2.77}$$

2.3. Comparison of curves

After completing the constructing method of above seven types of curves, we will first conclude the formulas found: (Actually we can easily prove that all these formulas can construct the curves that satisfy the conditions of Figure 15, but the procedures of proving are not important so we will skip this part.)

Assume there are three points on a plane surface:

$$\begin{aligned} &A(x_0, y_0) \\ &B(x_1, y_1) \\ &C(x_2, y_2) \end{aligned} \tag{2.78}$$

And construct a curve that satisfies the following conditions:

- 1) Start from A and end at B ;
- 2) Tangent to AC at A , and tangent to BC at B .

The formulas we can use are:

Curve A (by (2.31), (2.42))

$$\begin{aligned} x &= x_0 + x_1 - x_2 + \frac{x_2 - x_1 + t(x_1 - x_0)}{\sqrt{2t^2 - 2t + 1}} \\ y &= y_0 + y_1 - y_2 + \frac{y_2 - y_1 + t(y_1 - y_0)}{\sqrt{2t^2 - 2t + 1}} \\ &= (0 \leq t \leq 1), (0 \leq K \leq 1) \end{aligned} \tag{2.79}$$

Curve B (by (2.48))

$$\begin{aligned} X &= P_x + Q_x t + R_x g(t) \\ Y &= P_y + Q_y t + R_y g(t) \end{aligned} \tag{2.80}$$

and:

$$\begin{aligned} P_x &= x_0 & Q_x &= \frac{g'(1)(x_2 - x_0)}{g'(1) - g(1)} & R_x &= \frac{x_1 - x_2}{g(1)} - \frac{x_2 - x_0}{g'(1) - g(1)} \\ P_y &= y_0 & Q_y &= \frac{g'(1)(y_2 - y_0)}{g'(1) - g(1)} & R_y &= \frac{y_1 - y_2}{g(1)} - \frac{y_2 - y_0}{g'(1) - g(1)} \\ g(t) &= t^{\frac{3+5K}{2}} \end{aligned} \tag{2.81}$$

Curve C (by (2.48), (2.38))

$$\begin{aligned} X &= P_x + Q_x t + R_x g(t) \\ Y &= P_y + Q_y t + R_y g(t) \end{aligned} \quad (2.82)$$

and:

$$\begin{aligned} P_x &= x_0 & Q_x &= x_2 - x_0 & R_x &= x_1 - x_2 \\ P_y &= y_0 & Q_y &= y_2 - y_0 & R_y &= y_1 - y_2 \\ g(t) &= 1 - \sqrt[3+5K]{1 - t^{\frac{3+5K}{2}}} \end{aligned} \quad (2.83)$$

Curve C2

Variable t in (2.82) and (2.83) can be rewritten by the method similar to (2.38):

$$\begin{aligned} T(1 - K) + [(\frac{\pi}{2} - 2)T^3 - (\pi - 3)T^2 + \frac{\pi}{2}T]K \\ (T = (\frac{\pi}{2} - 2)t^3 - (\pi - 3)t^2 + \frac{\pi}{2}t) \end{aligned} \quad (2.84)$$

to equalize the length of the line segments.

Curve D (by (2.71))

$$\begin{aligned} X &= x_0(1 - t)^2 + 2t(1 - t)x_2 + x_1 t^2 \\ Y &= y_0(1 - t)^2 + 2t(1 - t)y_2 + y_1 t^2 \end{aligned} \quad (2.85)$$

Curve E (by 2.59))

$$\begin{aligned} X &= x_0 + t(x_1 - x_0) + [f(2t - 1) - 1](\frac{x_1 + x_0}{2} - x_2) \\ Y &= y_0 + t(y_1 - y_0) + [f(2t - 1) - 1](\frac{y_1 + y_0}{2} - y_2) \end{aligned} \quad (2.86)$$

and:

$$\begin{aligned} f(x) &= \frac{1 - k}{1 - 2k} (\sqrt{k^2 + (1 - 2k)(2x - 1)^2} + k - 1) \quad (0 \leq k \leq 1, k \neq 0.5) \\ \text{or } f(x) &= 2x(x - 1) \quad (k = 0.5) \end{aligned}$$

that is

$$\begin{aligned} f(x) &= -\frac{4(1-k)x(1-x)}{1-k+\sqrt{k^2+(1-2k)(2x-1)^2}} \quad (0 \leq k \leq 1) \\ f(x) &= 0 \quad (k=1) \end{aligned} \quad (2.87)$$

Curve F (by (2.77))

$$\begin{aligned} X &= x_0(1-t)^3 + 3[x_0 + k(x_2 - x_0)](1-t)^2t + 3[x_1 + k(x_2 - x_1)] \\ &\quad (1-t)t^2 + x_1t^3 \\ Y &= y_0(1-t)^3 + 3[y_0 + k(y_2 - y_0)](1-t)^2t + 3[y_1 + k(y_2 - y_1)] \\ &\quad (1-t)t^2 + y_1t^3 \end{aligned} \quad (2.88)$$

We can see that, in above 7 types of curve (C, C2 are counted as 2 types), no one is relative to method 1, it is because if we want to draw a straight line by method 1, it is difficult to rearrange $X^2 + Y^2 = R^2$ to a suitable one (R is sufficiently large), therefore we only compare Curve A to F which are no special limits when constructing curves.

In this part, we will compare these 7 types of curve in the following three ways:

(1) Speed of constructing curves

The formula will be entry to a computer program, the source code is optimized for computer to calculate. And the computer will continuously draw the curve within 30 seconds, finally compare the numbers of curve drawn by different formula. Every curve are formed by 100 line segments, the coordinate of controlling points of curves (A, B, C in (2.78)) and the value of parameter K (if any) will be randomly generated with a value between 0 and 1, moreover, the computer will not show the curve graphically to improve the speed of calculating. We will count the number of curves drawn in 1 second with the best result, more curves can be drawn within 1 second is the better curve.

(2) Degree of average:

The computer will simulate to draw 10100 curves, every curve has completely been drawn, the program will calculate the percentage of difference between the length of each line segments and the mean length of line segment of curve as the degree of average of the curve, and compare the mean of degree of average, top, least value of these 10100 results. About the value setting of the curves, parameter K will be sequentially substituted from 0.00 to 1.00 total 101 values 100 times each, while other values are same as speed test. The curve with lower mean of degree of average and the top of mean of degree of average and not on the high side is better.

(3) Rate of change:

We will compare rate of change of height of these 7 types of curve when the parameter K changes between 0 and 1, also we will discuss the problem of changing height of curve when K changes. The curve is better with greater rate of change. (Figure 40 shows the definition of 'height', and $CD \parallel AB$, $DF \perp AB$, height of a curve means the greatest value of $\frac{EF}{DF}$.)

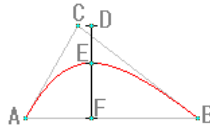


FIGURE 40

The followings are the experiment data:

* The numbers inside () are the rank.

(1) Speed of constructing curves

Curve	Number of Curve drwan per second	The fastest reaches 1000 curves
A	91152 (3)	511
B	32067 (5)	180
C	18710 (6)	105
C2	17474 (7)	98
D	178447 (1)	1000
E	64314 (4)	360
F	158996 (2)	891

(2) Degree of average

Curve	Mean of mean deviation (%)	Greatest mean deviation (%)	Lowest mean deviation (%)
A	30.18 (2)	32.0 (1)	27.5 (4)
B	37.83 (6)	44.5 (5)	31.5 (6)
C	59.62 (7)	76.0 (7)	34.5 (7)
C2	35.91 (5)	42.0 (4)	27.0 (3)
D	34.77 (4)	37.0 (3)	31.0 (5)
E	28.27 (1)	36.5 (2)	0.0 (1)
F	33.17 (3)	47.5 (6)	23.0 (2)

(3) Rate of change

Curve	Rate of change (%)	Greatest height (%)	Lowest height (%)
A	0 (6)	-	-
B	19 (5)	63	44
C	40 (3)	67	27
C2	40 (3)	67	27
D	0 (6)	-	-
E	100 (1)	100	0
F	75 (2)	75	0

In this experiment, the performance of curve E and F are satisfactory, and the rate of change of curve E reaches 100%, and their heights are directly proportional to parameter K because these two types of curves are designed for this condition. For curve F, we cannot increase the height to 100% because if we force to increase the height of curve F, the curve will bend to outside as shown in Figure 41.

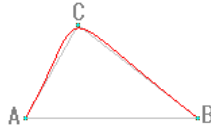


FIGURE 41

For curve B and C, they are originally the distortion of known functions, because the functions have not been carefully designed, when we tune the value higher ($K > 1$), the rate of increase of height decreases sharply. Certainly we can replace K to exponential form such as a^k , but the speed of drawing will decrease to lower than 50% of original. (The speed of drawing curve B and C are lowest originally) And the rates of average of the curve are also worse. The only solution of this problem is to change the function used to ones that are similar to curve E and F. Trivially, using curve E and F directly is better for sure.

After the three comparisons, trivially, only curve E and F are worth using, and we need to prove that these two types of curve satisfy the following conditions before we continue to use them:

If the control point of the curve $A(x_0, y_0)$, $B(x_1, y_1)$, $C(x_2, y_2)$ are changed to $A'(x'_0, y'_0)$, $B'(x'_1, y'_1)$, $C'(x'_2, y'_2)$ respectively by the same method of moving, rotating, or zooming, and every point (X'_n, Y'_n) on the curve defined by A', B', C' (and $t = n$ in the formula) is the result of moving, rotating, or zooming of every point (X_n, Y_n) on the curve defined by A, B, C , we can define these types of curves as 'movable'.

We first prove curve E satisfies above condition:

(1) When A , B , C move to right by M_X units, and move upward by M_Y units:

$$\begin{aligned}
 X'_n &= x'_0 + t(x'_1 - x'_0) + [f(2t - 1) - 1]\left(\frac{x'_1 + x'_0}{2} - x'_2\right) \\
 &= x_0 + M_X + t(x_1 + M_X - x_0 - M_X) \\
 &\quad + [f(2t - 1) - 1]\left(\frac{x_1 + M_X + x_0 + M_X}{2} - x_2 - M_X\right) \\
 &= x_0 + t(x_1 - x_0) + [f(2t - 1) - 1]\left(\frac{x_1 + x_0}{2} - x_2\right)M_X \\
 &= X_n + M_X \\
 Y'_n &= y'_0 + t(y'_1 - y'_0) + [f(2t - 1) - 1]\left(\frac{y'_1 + y'_0}{2} - y'_2\right) \\
 &= y_0 + M_Y + t(y_1 + M_Y - y_0 - M_Y) \\
 &\quad + [f(2t - 1) - 1]\left(\frac{y_1 + M_Y + y_0 + M_Y}{2} - y_2 - M_Y\right) \\
 &= y_0 + t(y_1 - y_0) + [f(2t - 1) - 1]\left(\frac{y_1 + y_0}{2} - y_2\right)M_Y \\
 &= Y_n + M_Y
 \end{aligned} \tag{2.89}$$

Therefore curve E can move.

(2) When A, B, C are magnified by m times relative to origin:

$$\begin{aligned}
 X'_n &= x'_0 + t(x'_1 - x'_0) + [f(2t - 1) - 1]\left(\frac{x'_1 + x'_0}{2} - x'_2\right) \\
 &= mx_0 + t(mx_1 - mx_0) + [f(2t - 1) - 1]\left(\frac{mx_1 + mx_0}{2} - mx_2\right) \\
 &= m \left\{ x_0 + t(x_1 - x_0) + [f(2t - 1) - 1]\left(\frac{x_1 + x_0}{2} - mx_2\right) \right\} \\
 &= mX_n \\
 Y'_n &= y'_0 + t(y'_1 - y'_0) + [f(2t - 1) - 1]\left(\frac{y'_1 + y'_0}{2} - y'_2\right) \\
 &= my_0 + t(my_1 - my_0) + [f(2t - 1) - 1]\left(\frac{my_1 + my_0}{2} - my_2\right) \\
 &= m \left\{ y_0 + t(y_1 - y_0) + [f(2t - 1) - 1]\left(\frac{y_1 + y_0}{2} - my_2\right) \right\} \\
 &= mY_n
 \end{aligned} \tag{2.90}$$

Therefore curve E can zoom.

(3) when A, B, C are rotated anti-clockwise by θ degrees, and let the origin be the centre:

$$\begin{aligned}
 X'_n &= x'_0 + t(x'_1 - x'_0) + [f(2t - 1) - 1] \left(\frac{x'_1 + x'_0}{2} - x'_2 \right) \\
 &= (x_0 \cos \theta - y_0 \sin \theta) + t[(x_1 \cos \theta - y_1 \sin \theta) - x_0 \cos \theta - y_0 \sin \theta] \\
 &\quad + [f(2t - 1) - 1] \\
 &\quad \left(\frac{(x_1 \cos \theta - y_1 \sin \theta) - (x_0 \cos \theta - y_0 \sin \theta)}{2} - (x_0 \cos \theta - y_0 \sin \theta) \right) \\
 &= \cos \theta \left\{ x_0 + t(x_1 - x_0) + [f(2t - 1) - 1] \left(\frac{x_1 + x_0}{2} - x_2 \right) \right\} \\
 &\quad - \sin \theta \left\{ y_0 + t(y_1 - y_0) + [f(2t - 1) - 1] \left(\frac{y_1 + y_0}{2} - y_2 \right) \right\} \\
 &= X_n \cos \theta - Y_n \sin \theta \\
 Y'_n &= y'_0 + t(y'_1 - y'_0) + [f(2t - 1) - 1] \left(\frac{y'_1 + y'_0}{2} - y'_2 \right) \\
 &= (x_0 \sin \theta - y_0 \cos \theta) + t[(x_1 \sin \theta - y_1 \cos \theta) - x_0 \sin \theta - y_0 \cos \theta] \\
 &\quad + [f(2t - 1) - 1] \\
 &\quad \left(\frac{(x_1 \sin \theta - y_1 \cos \theta) - (x_0 \sin \theta - y_0 \cos \theta)}{2} - (x_0 \sin \theta - y_0 \cos \theta) \right) \\
 &= \sin \theta \left\{ x_0 + t(x_1 - x_0) + [f(2t - 1) - 1] \left(\frac{x_1 + x_0}{2} - x_2 \right) \right\} \\
 &\quad - \cos \theta \left\{ y_0 + t(y_1 - y_0) + [f(2t - 1) - 1] \left(\frac{y_1 + y_0}{2} - y_2 \right) \right\} \\
 &= X_n \sin \theta - Y_n \cos \theta \tag{2.91}
 \end{aligned}$$

Therefore curve E can be rotated.

Combining three results, curve E is ‘movable’.

Obviously, curve F can be easily proved as ‘movable’ if we apply the proving method that curve E used, so the process will not be shown here.

We need to prove the curve is ‘movable’, because when we create a human model (by about 10000 small triangle surfaces) in a virtual space, the position of a construction points of the 3D object are mainly (over 90%) defined by the control points of other construction points, curves or curved

surface formulas. When a human model constructed in a virtual space is movable, we need to use 3D transformation formulas (similar to (1.13), but the calculating sequence of matrix is different, for details please refer to section 4), to move the construction points to new position, so we only apply the curve formula to control points will better than apply the complex 3D transformation formula to all points in the 3D object, and use the curve or curved surfaces which is simpler to define the position of rest of construction points, this method can reduce the calculation procedure of computer at least 80%, and the curve or curved surface formulas are movable can prevent unreasonable distortion when the human object is moving.

Generally, curve E and F can both obtain a satisfactory performance, but we need not to decide which curve is better in this section, may be we have the conclusion after investigating of curved surface. By the way, curves still have some characteristics that we can investigate.

2.4. Extra problems about curves

In the investigation of 2.2 and 2.3, we used method (ii) mentioned in 2.1 as the base of experiments, in this section, we will go extra investigation the point E mentioned in method (ii).

Assume the coordinates of points A , B , C , D and E in method (ii) are:

$$\begin{aligned} A(x_0, y_0) \\ B(x_1, y_1) \\ C(x_2, y_2) \\ D(x_3, y_3) \end{aligned} \tag{2.92}$$

And $E(x_4, y_4)$ can also be rewritten as:

$$E(x_1 + n(x_2 - x_1), y_1 + n(y_2 - y_1)) \quad (0 \leq n \leq 1) \tag{2.93}$$

Being a joining point of two curves, although the position of E can be any one point on BC, we found that the whole complete curve will become the most 'beautiful' when E is located at a certain point.

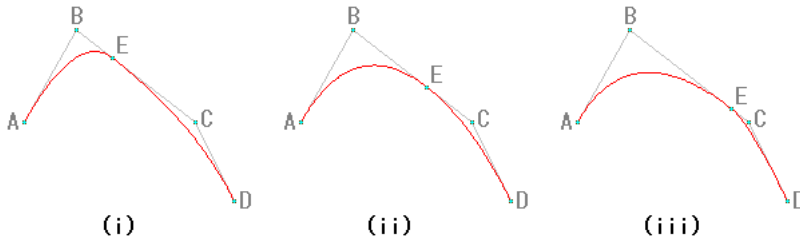


FIGURE 42

In Figure 42, when E is located in the position (i) or (iii), the rate of bending of the curve seems to be changed suddenly; In the situation (ii), the curve seems to be natural that it looks like it has no joining point.

We have discovered similar problems when we investigate the circle array curves, that is: when the incenters of the triangles of two parts of arcs are joining together, unless the radii of two arcs are equal, otherwise the whole curve will become unnatural.

We think we can explain and solve the above two problems by the curvature:

For the explicit function $y = f(x)$, the curvature when $x = n[2]$:

$$K = \frac{y_n''}{\sqrt[3]{1 + (y_n')^2}} \quad (2.94)$$

And for some continuous function such as $y = ax^b + c$, $y = a \sin x + b \cos x$, $y = b \log_a x$, obviously, their curvatures are also continuous, and their shapes look natural too, therefore we conclude that if we want two parts of curves can be connected well, the slopes and the curvatures, in addition, of two curves at joining point are required to become equal.

Assume the parametric of two curves are:

$$\begin{aligned} X &= f_X(t) \\ Y &= f_Y(t) \end{aligned} \quad (2.95)$$

Therefore,

$$\begin{aligned} \frac{d^2y}{dx^2} &= \frac{f'_Y(t)}{f'_X(t)} & (2.96) \\ d\left(\frac{d^2y}{dx^2}\right) &= d\left(\frac{f'_Y(t)}{f'_X(t)}\right) \\ &= \frac{f'_X(t)f''_Y(t) - f''_X(t)f'_Y(t)}{[f'_X(t)]^2} dt \\ &= \frac{f'_X(t)f''_Y(t) - f''_X(t)f'_Y(t)}{[f'_X(t)]^3} dx \end{aligned}$$

Therefore,

$$K = \frac{Y''}{\sqrt[3]{1 + (y')^2}} = \frac{f'_X(t)f''_Y(t) - f''_X(t)f'_Y(t)}{[f'_X(t)]^3} \div \sqrt[3]{1 + \left(\frac{f'_Y(t)}{f'_X(t)}\right)^2} \quad (2.97)$$

Because the slopes of two curves are equal when they join each other, when we want to solve n in (2.93) by $K_1 = K_2$, we only need to solve the formula $Y''_1 = Y''_2$. (It should be noted that the value of n is between 0 and 1.)

However, sometimes formula $Y''_1 = Y''_2$ has no solutions, one of the reasons is that the curvature of two curves cannot be equal, such as the situation in Figure 43, two curves are bended to different directions. That means unless the curvatures of two curves can both become 0 at certain point, otherwise we cannot find a perfect joining point.

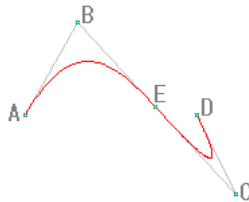


FIGURE 43

Besides, in practice, we do not require the curvature of two curves at the joining point are necessarily the same as it will limit the shapes of the curve, being a software designer, we will let the user decide the position of E on BC arbitrarily, and the software should display the curvature centers [2] of two curves at the joining point as reference. (Nearer the curvatures centers

at the joining point, better the results.)

And we have completed all the research about curves here.

3. Creating curved surfaces

After obtaining some methods of constructing curves and the relative experience, in this section, we will discuss how to create curved surfaces in virtual spaces.

The goal of this section is to obtain the method and formulas of creating curved surfaces.

3.1. Requirement of creating curved surfaces

There are two common types of curved surfaces:

- 1) Three vertexes
- 2) Four vertexes

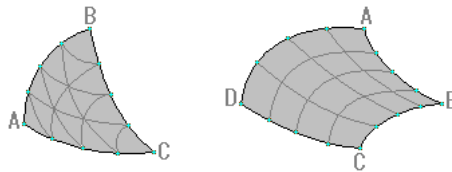


FIGURE 44. three vertexes and four vertexes

In this section we will only consider three-vertex curve surfaces because:

- (1) Two three-vertex curve surfaces can form a four-vertex curve surface but not vice versa as shown in Figure 45

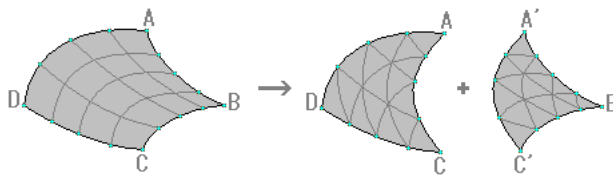


FIGURE 45

(2) Three-vertex curve surfaces seem to be easier to be created.

For the designed three-vertex curve surfaces, we hope it can satisfy at least the following two conditions:

(1) Three sides of the curved surface are defined by one curve which is defined by three points. (Such as curve E and F in section 2);

(2) If there are two curve surfaces and they have a common side, when they are combined to each other, a new four-vertex curve surfaces will be formed and certain two sides can be treated as one smooth curve, then the four-vertex curve surfaces is a smooth curved surface.

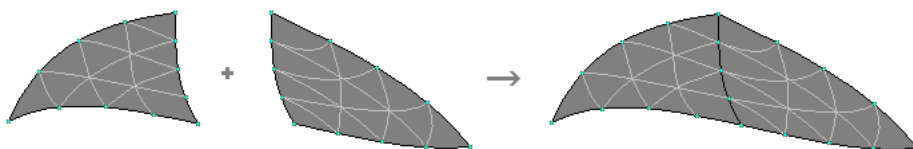


FIGURE 46

3.2. Methods of creating curved surfaces

From above conditions, a curved surface is defined by at least six points, and if the curves are the results of the following formula:

$$\begin{aligned}
 X &= f_x(t, u) \\
 Y &= f_y(t, u) \quad (0 \leq t \leq 1, 0 \leq u \leq 1, t + u = 1)
 \end{aligned}
 \tag{3.1}$$

Then we can treat curved surfaces as:

$$\begin{aligned} X &= f_x(t, u, v) \\ Y &= f_y(t, u, v) \\ Z &= f_z(t, u, v) \quad (0 \leq t \leq 1, 0 \leq u \leq 1, 0 \leq v \leq 1, t + u + v = 1) \end{aligned} \quad (3.2)$$

Assume the coordinates of six points which define the curved surface are[3]:

$$\begin{aligned} A(x_0, y_0, z_0) \quad B(x_1, y_1, z_1) \quad C(x_2, y_2, z_2) \\ D(x_3, y_3, z_3) \quad E(x_4, y_4, z_4) \quad F(x_5, y_5, z_5) \end{aligned} \quad (3.3)$$

And the definition points of three sides of the curved surface S_0, S_1, S_2 are

$$\begin{aligned} S_0 &: (A, B, C) \\ S_1 &: (C, D, E) \\ S_2 &: (E, F, A) \end{aligned}$$

As shown in Figure 47:

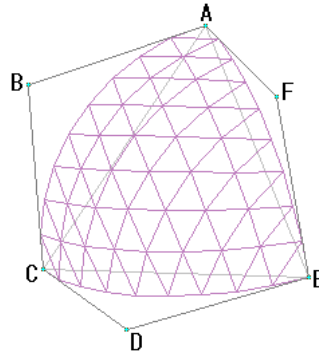


FIGURE 47

Now if the formula of the curve can be represented by curve F in section 2, then S_0, S_1, S_2 can be rewritten as following:

$$\begin{aligned} S_0 : F(t, u) &= At^3 + 3[A + k(B - A)]t^2u + 3[C + k(B - C)]tu^2 + Cu \\ S_1 : F(u, v) &= Cu^3 + 3[C + k(D - C)]u^2v + 3[E + k(D - E)]uv^2 + Ev \\ S_2 : F(v, t) &= Ev^3 + 3[E + k(F - E)]v^2t + 3[A + k(F - A)]vt^2 + At \\ (0 \leq t \leq 1, 0 \leq u \leq 1, 0 \leq v \leq 1, t + u + v = 1) \end{aligned} \quad (3.4)$$

And we try to combine the above equations to $F(t, u, v)$ to form a curved surface formula:

$$\begin{aligned}
 F(t, u, v) = & At^3 + 3[A + k(B - A)]t^2u + 3[C + k(B - C)]tu^2 \\
 & + Cu^3 + 3[C + k(D - C)]u^2v + 3[E + k(D - E)]uv^2 \\
 & + Ev^3 + 3[E + k(F - E)]v^2t + 3[A + k(F - A)]vt^2 \quad (3.5)
 \end{aligned}$$

To simplify the process of examination of the practicability of above formula, we firstly use the formula to create a one eighth sphere, and let the coordinate of six points in (3.3) be

$$\begin{aligned}
 & A(1, 0, 0) \quad B(1, 1, 0) \quad C(0, 1, 0) \\
 & D(0, 1, 1) \quad E(0, 0, 1) \quad F(1, 0, 1) \\
 & \left(k = \frac{4}{3}(\sqrt{2} - 1)\right) \quad (3.6)
 \end{aligned}$$

And the drawing result:

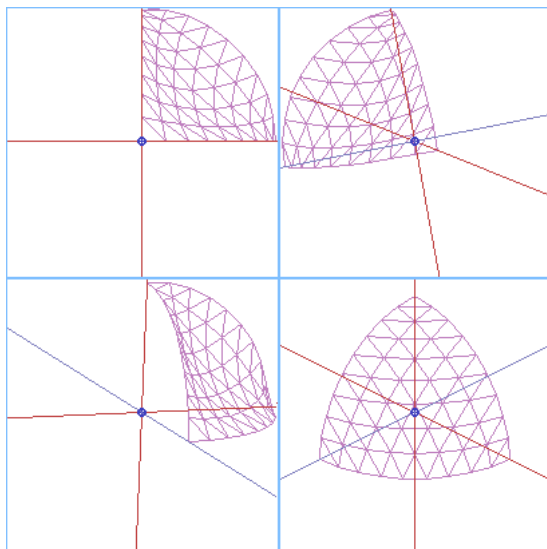


FIGURE 48

Obviously the result is not satisfactory, so after the reference of formula of Bezier Triangle[3], we decided to increase a control point $G(x_6, y_6, z_6)$ to curved surfaces, and rewrite (3.5) as below: (The value of g is to be decided)

$$\begin{aligned}
 F(t, u, v) = & At^3 + 3[A + k(B - A)]t^2u + 3[C + k(B - C)]tu^2 \\
 & + Cu^3 + 3[C + k(D - C)]u^2v + 3[E + k(D - E)]uv^2 \\
 & + Ev^3 + 3[E + k(F - E)]v^2t + 3[A + k(F - A)]vt^2 \\
 & + G \cdot 6gtuv
 \end{aligned}
 \tag{3.7}$$

We use (3.7) to create a one eighth sphere again, and the coordinate of G becomes $(1, 1, 1)$ naturally. For the best results, we firstly substitute $\frac{1}{3}$ into t, u, v in (3.7), and confirm $G = \frac{9\sqrt{3} - 6k - 7}{6}$.

And Figure 49 shows the result:

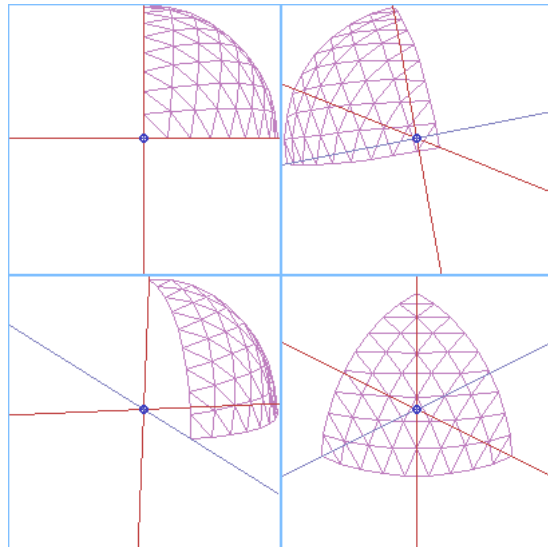


FIGURE 49

Because we are unable to design a mathematical method to verify whether (3.7) satisfies the conditions of Figure 46, and the practicability of (3.7), so it was left until we have improved the relevant ability.

By the way, we are interested in creating one more curved surface formula, by using curve E in section 2, (3.4) can be rewritten as

$$\begin{aligned}
 S_0 : F(t, u) &= At + Cu + tu\left(\frac{A + C}{2} - B\right)f(t, u) \\
 S_1 : F(u, v) &= Cu + Ev + uv\left(\frac{C + E}{2} - D\right)f(u, v) \\
 S_2 : F(v, t) &= Ev + At + vt\left(\frac{E + A}{2} - F\right)f(v, t)
 \end{aligned} \tag{3.8}$$

and

$$f(a, b) = -\frac{4(1 - k)}{1 - k + \sqrt{k^2 + (1 - 2k)(1 - 4ab)}} \quad (0 \leq k < 1) \tag{3.9}$$

By using the method similar to obtaining (3.7):

$$\begin{aligned}
 F(t, u, v) &= At + Cu + Ev + 9(\sqrt{3} - 1)Gtuv \\
 &\quad \times \left[tu\left(\frac{A + C}{2} - B\right) + uv\left(\frac{C + E}{2} - D\right) \right. \\
 &\quad \left. + vt\left(\frac{E + A}{2} - F\right) + 3Gtuv \right] f(t, u, v)
 \end{aligned} \tag{3.10}$$

and

$$f(t, u, v) = -\frac{4(1 - k)}{1 - k + \sqrt{k^2 + (1 - 2k)[1 - 4(tu + uv + vt)]}} \tag{3.11}$$

And the result is shown in Figure 50:

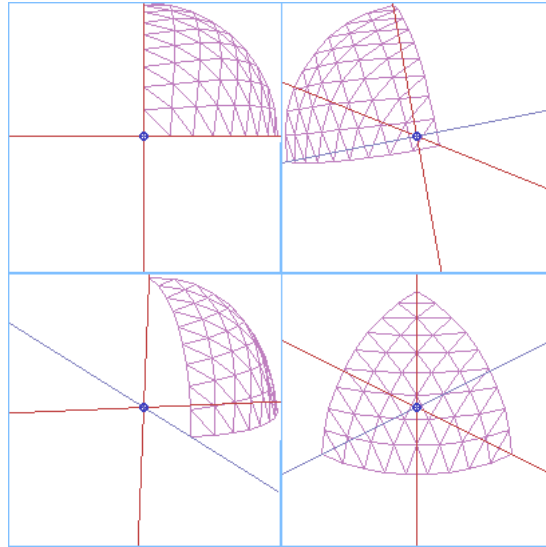


FIGURE 50

If we create the whole sphere by the same method, we will discover that the sphere created by (3.7) looks more like a sphere, and the lines of the sphere created by (3.10) look more natural.

3.3. Conclusion

When we just start to investigate curved surfaces, we want to generalize the method of creating curves in section 2.2 to creating curved surfaces, such as dividing a curved surface into four ellipsoid is the generalization of dividing a curves to two arcs in method 1, distorting sphere is the generalization of distorting circle in method 3, and method 6 is also constructing polynomial parametric formula directly. But all of them have their critical problems, for example, the generalization of method 1 to 4 can only create limited curved surfaces, that means they cannot create both sphere and hyperbolic paraboloid by single formula. Also we have to face some extremely complex calculating process when we generalize the method of creating curves to creating curved surfaces directly. When we investigate curved surface, we have only considered curve E and F, although they may have the weak points that the others curves have, but they have the maximum flexibility in shape, and

the drawing speed is faster than other curves, also they can generalize to 3-dimensional space by vector method, and this can confirm that the curved surfaces created by these curve is 'movable' by the method similar to section 2.4. Finally, although there are a lot of functional formulas such as sinusoid, pendulum curves, spiral etc, but the calculating speed is very important to computer graphics, and personal computer software are commonly using rectangular coordinate system, so we cannot consider trigonometric functions or other complex functions to create curves or curved surfaces, square root calculating is the acceptable upper limit.

We want the method of creating curved surface to be ocular and easy to understand, we want the sides of curved surfaces to be created by the curves we have used well, and we can define a curved surface by only these curves, so we want to create a curved surface by only six points. And because of the same reason, we have not considered using Bezier Surface or Bezier Triangle to be a tool of creating curved surfaces in this section. Also we need to consider the practical use, seven points are needed to define a curved surface, because we do not want a curved surface formula not to be able to simulate even a common sphere, but the effect to the shape of curved surfaces by the seventh point is not ocular, and this motives us to develop a better curved surface formula, but due to the limit of our ability, the research of curved surfaces will stop here.

4. Human model

After finishing the above three sections research, by the reference of some shapes of the objects, we can create 3D objects in virtual space, but our main goal is creating human model, not only a model having a human shape, the optimal result is that we can change the pose of the model by changing only few parameters, then if we continuously change the parameters, a series of motions can be done.

The above problem can be treated as the research of interactive relationship of the objects, and this relationship is about the virtual space and objects, because no matter how the related position of the objects has been changed, we need to know their position related to virtual space in order to let the observers observe.

4.1. Interactive relationship between objects of human models

In this section, we will first consider the interactive relationship of which objects is frequently used when we create human model. We group these relations as below:

(1) Oscillating: The relations between upper arm and lower arm can be grouped as this type. These relations can be treated as object A let a line of object B as axis, and change its position by rotating by the axis. Other objects in human model such as the relation between fingers and palms, thigh and shank can be grouped to this type.

(2) Rotating: The relations between upper arm and wrist can be grouped as this type. These relations can be treated as object A let a line of object B as axis, and change its position by rotating by the axis. Obviously, this type of relation can be treated as same type of (1), but in the view of human, the rotating direction is different, and upper arm and wrist are let the whole lower arm to be axis, so the sharp of lower arm will be distorted.

(3) Free transformation: The relations between upper arm and body can be grouped as this type. This type is the fusion of (1) and (2). The interactive relationship of most complex part of human body is the parts of backbone (including cervical vertebra), as they include the above three types of relationships, also backbone can process S shape distortion, and lead the motions of relative parts more varied. But if we abandon precise human motion, we only need to decompose backbone part to few relation of type (3) and simulate.

Therefore we will investigate type (3) only.

4.2. Formulas about free transformation

Assume a man is holding a rectangular shape touch and lighting, he discovered an interesting object and point to it with the torch, and for the best lighting result, he let the light emitted by the rectangular touch as axis, and rotated it by certain degrees (θ_z). Now if this man defines a coordinate system for the center, how to describe the direction that the touch pointed to?

We have discovered that the relationship between this object and the space is the opposite of the relationship between observer and virtual space in section 1, so the following formula that describes this situation is obtained:

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} + R_{X \rightarrow Z}(\theta_y)R_{Y \rightarrow Z}(\theta_x)R_{Y \rightarrow X}(\theta_z) \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \tag{4.1}$$

And when we define a coordinate system for an object (object coordinate), (x_i, y_i, z_i) is the coordinate that belonged to a point of the object related to the object coordinate. m is the ratio of unit of object coordinate and unit of space. (C_x, C_y, C_z) is the center of object coordinate related to space, (x'_i, y'_i, z'_i) is the location of the point of an object in the space.

From above formula, we can initially simulate human motion, we only need to treat the objects that required moving as few independent objects and then create it. And use (4.1) to arrange the relationship of those object, finally we can create a formula based on time(t) for $\theta_x, \theta_y, \theta_z$ in (4.1) according to the required motions, so that when we use the software to simulate the human model, we can preview the scheduled motions.

When we design the motion degree of the components of human model, for the users convenient, we will use the following graph as the designing tools:

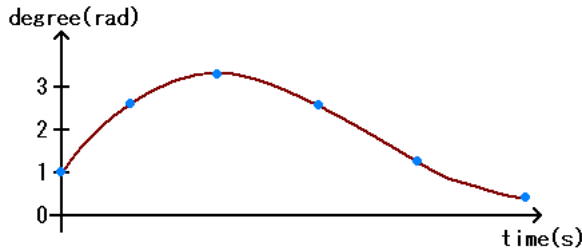


FIGURE 51

In the above graph, we will use the result obtained in section 2 as the tools of creating curves. Due to time limit, the investigation of developing human model will stop here.

Appendix A. Computer Program

Here is the implementations used to plot the curves in Visual Basic. It is assumed that the constant PI has been declared to be 3.1415926535.

A.1. The program for plotting Curve B

```

1 Private Type Vector2
2     sngX As Single
3     sngY As Single
4 End Type
5
6 Private Sub vr2OutB( _
7     vr2InputPoints() As Vector2, _
8     sngK As Single, _
9     vr2Output() As Vector2)
10
11     Dim intI As Integer
12     Dim sngT As Single
13     Dim sngTempT As Single
14     Dim sngTempK As Single
15     Dim vr2Const(1 To 2) As Vector2
16
17     sngTempK = 1.5 + sngK * 2.5
18     vr2Const(1).sngX = sngDifNormal(1, sngTempK) * _
19         (vr2InputPoints.sngX(2) - vr2InputPoints.sngX(0)) _
20         / (sngDifNormal(1, sngTempK) - sngNormal(1, sngTempK))
21     vr2Const(2).sngX = _
22         ((vr2InputPoints.sngX(1) - vr2InputPoints.sngX(2)) _
23         / sngNormal(1, sngTempK) - _
24         vr2InputPoints.sngX(2) - vr2InputPoints.sngX(0)) _
25         / (sngDifNormal(1, sngTempK) - sngNormal(1, sngTempK)))
26     vr2Const(1).sngY = sngDifNormal(1, sngTempK) * _
27         (vr2InputPoints.sngY(2) - vr2InputPoints.sngY(0)) _
28         / (sngDifNormal(1, sngTempK) - sngNormal(1, sngTempK))
29     vr2Const(2).sngY = _
30         ((vr2InputPoints.sngY(1) - vr2InputPoints.sngY(2)) _
31         / sngNormal(1, sngTempK) - _
32         (vr2InputPoints.sngY(2) - vr2InputPoints.sngY(0)) _
33         / (sngDifNormal(1, sngTempK) - sngNormal(1, sngTempK)))
34     For intI = 0 To 100
35         sngT = intI / 100
36         sngTempT = sngT ^ sngTempK
37         vr2Output(intI).sngX = vr2InputPoints.sngX(0) + _
38             vr2Const(1).sngX * sngT + vr2Const(2).sngX * sngTempT
39         vr2Output(intI).sngY = vr2InputPoints.sngY(0) + _

```

```

40         vr2Const(1).sngY * sngT + vr2Const(2).sngY * sngTempT
41     Next intI
42 End Sub
43
44 Private Function sngNormal(sngT As Single, _
45 ByVal sngK As Single) As Single
46     sngNormal = sngT ^ sngK
47 End Function
48
49 Private Function sngDifNormal(sngT As Single, _
50 ByVal sngK As Single) As Single
51     sngDifNormal = sngK * sngT ^ (sngK - 1)
52 End Function

```

A.2. The program for plotting Curve C2

```

1 Private Sub vr2OutC2( _
2     vr2InputPoints() As Vector2, _
3     sngK As Single, _
4     vr2Output() As Vector2)
5
6     Dim intI As Integer
7     Dim sngT As Single
8     Dim sngTempT As Single
9     Dim sngTempK As Single
10    Dim vr2Const(1 To 2) As Vector2
11
12    sngTempK = 1.5 + sngK * 2.5
13    vr2Const(1).sngX = vr2InputPoints.sngX(2) - vr2InputPoints.sngX(0)
14    vr2Const(2).sngX = vr2InputPoints.sngX(1) - vr2InputPoints.sngX(2)
15    vr2Const(1).sngY = vr2InputPoints.sngY(2) - vr2InputPoints.sngY(0)
16    vr2Const(2).sngY = vr2InputPoints.sngY(1) - vr2InputPoints.sngY(2)
17    For intI = 0 To 100
18        sngT = intI / 100
19        sngT = sngT * ((Pi / 2 - 2) * sngT * sngT + _
20            (3 - Pi) * sngT + Pi / 2)
21        sngT = sngT * (1 - sngK) + _
22            (sngT * ((Pi / 2 - 2) * sngT * sngT + _
23                (3 - Pi) * sngT + Pi / 2)) * sngK
24        sngTempT = (1 - (1 - sngT ^ sngTempK) ^ (1 / sngTempK))
25        vr2Output(intI).sngX = vr2InputPoints.sngX(0) + _
26            vr2Const(1).sngX * sngT + vr2Const(2).sngX * sngTempT
27        vr2Output(intI).sngY = vr2InputPoints.sngY(0) + _
28            vr2Const(1).sngY * sngT + vr2Const(2).sngY * sngTempT
29    Next intI
30 End Sub
31
32 Private Function sngInfiniteSlope(sngT As Single, _

```

```

33 ByVal sngK As Single) As Single
34     sngInfiniteSlope = 1 - (1 - sngT ^ sngK) ^ (1 / sngK)
35 End Function

```

A.3. The program for plotting Curve E

```

1 Private Sub vr2OutE( _
2     vr2InputPoints() As Vector2, _
3     sngK As Single, _
4     vr2Output() As Vector2)
5
6     Dim intI As Integer
7     Dim sngT As Single
8     Dim sngTempT As Single
9     Dim sngTempK As Single
10    Dim vr2Const(1 To 2) As Vector2
11
12    sngTempK = (1-sngK) * 0.999
13    vr2Const(1).sngX = vr2InputPoints.sngX(1) - vr2InputPoints.sngX(0)
14    vr2Const(2).sngX = (vr2InputPoints.sngX(1) _
15        + vr2InputPoints.sngX(0)) * 0.5 - vr2InputPoints.sngX(2)
16    vr2Const(1).sngY = vr2InputPoints.sngY(1) - vr2InputPoints.sngY(0)
17    vr2Const(2).sngY = (vr2InputPoints.sngY(1) _
18        + vr2InputPoints.sngY(0)) * 0.5 - vr2InputPoints.sngY(2)
19    For intI = 0 To 100
20        sngT = intI / 100
21        sngTempT = sngConeCurve(4 * sngT * (1 - sngT), sngTempK)
22        vr2Output(intI).sngX = vr2InputPoints.sngX(0) + _
23            vr2Const(1).sngX * sngT + vr2Const(2).sngX * sngTempT
24        vr2Output(intI).sngY = vr2InputPoints.sngY(0) + _
25            vr2Const(1).sngY * sngT + vr2Const(2).sngY * sngTempT
26    Next intI
27 End Sub
28
29 Private Function sngConeCurve(sngT As Single, _
30 ByVal sngK As Single) As Single
31     sngConeCurve = - sngT * (1 - sngK) _
32         / (1 - sngK + Sqr(sngK * sngK + (1 - 2 * sngK) * (1 - sngT)))
33 End Function

```

A.4. The program for plotting Curve F

```

1 Private Sub vr2OutF( _
2     vr2InputPoints() As Vector2, _
3     sngK As Single, _
4     vr2Output() As Vector2)
5

```

```

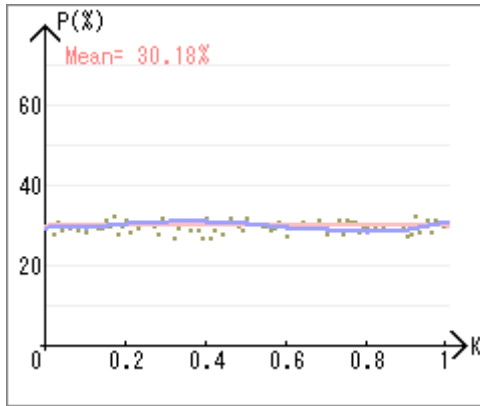
6   Dim intI As Integer
7   Dim sngT As Single
8   Dim sngTempT As Single
9   Dim vr2Const(0 To 2) As Vector2
10
11  vr2Const(0).sngX = 3 * sngK * (vr2InputPoints.sngX(2) - _
12    vr2InputPoints.sngX(0))
13  vr2Const(1).sngX = 3 * (vr2InputPoints.sngX(1) - _
14    vr2InputPoints.sngX(0) + sngK * (2 * vr2InputPoints.sngX(0) - _
15    vr2InputPoints.sngX(2) - vr2InputPoints.sngX(1)))
16  vr2Const(2).sngX = 2 * (vr2InputPoints.sngX(0) - _
17    vr2InputPoints.sngX(1)) + 3 * sngK * (vr2InputPoints.sngX(1) - _
18    vr2InputPoints.sngX(0))
19  vr2Const(0).sngY = 3 * sngK * (vr2InputPoints.sngY(2) - _
20    vr2InputPoints.sngY(0))
21  vr2Const(1).sngY = 3 * (vr2InputPoints.sngY(1) - _
22    vr2InputPoints.sngY(0) + sngK * (2 * vr2InputPoints.sngY(0) - _
23    vr2InputPoints.sngY(2) - vr2InputPoints.sngY(1)))
24  vr2Const(2).sngY = 2 * (vr2InputPoints.sngY(0) - _
25    vr2InputPoints.sngY(1)) + 3 * sngK * (vr2InputPoints.sngY(1) - _
26    vr2InputPoints.sngY(0))
27  For intI = 0 To 100
28    sngT = intI / 100
29    vr2Output(intI).sngX = sngX(0) + vr2Const(0).sngX * sngT + _
30      vr2Const(1).sngX * sngT * sngT + _
31      vr2Const(2).sngX * sngT * sngT * sngT
32    vr2Output(intI).sngY = sngY(0) + vr2Const(0).sngY * sngT + _
33      vr2Const(1).sngY * sngT * sngT + _
34      vr2Const(2).sngY * sngT * sngT * sngT
35  Next intI
36 End Sub

```

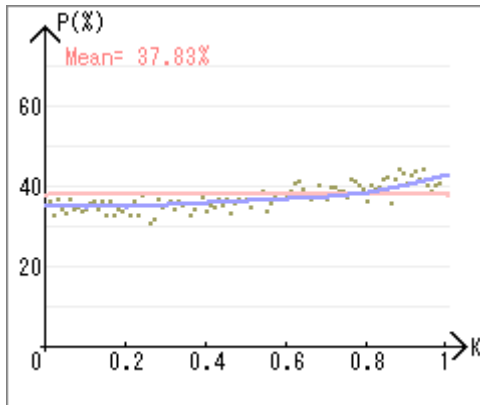
A.5. Reference graphs of section 2.3

The dots (Total 101 dots) represent the mean deviation when the program calculating. (Every dot is the percentage of mean deviation of 100 results of putting the corresponding value of K into the formula and calculate.) And the curve is generated by extracting the $(N + 1)$ th point (Total 11 points) and substituting into Bernstein polynomials[1][4].

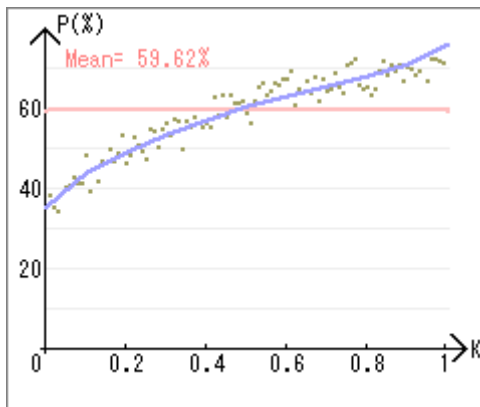
Curve A



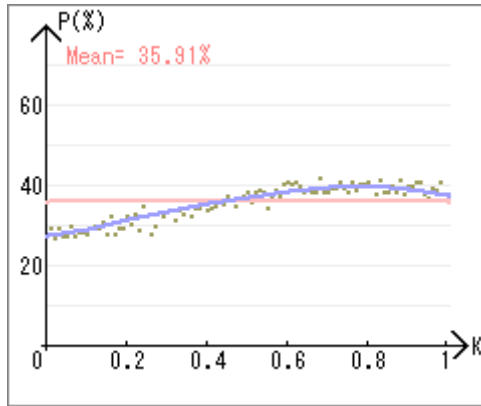
Curve B



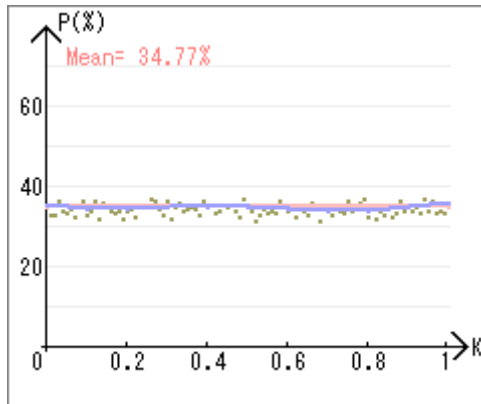
Curve C



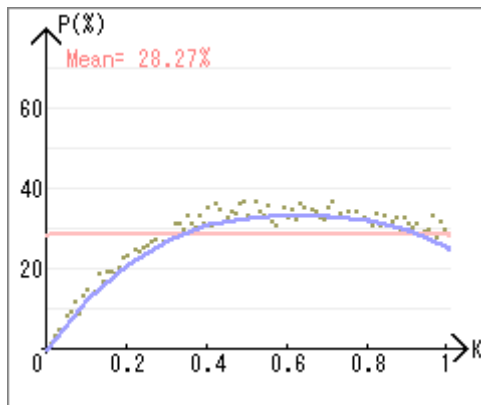
Curve C2

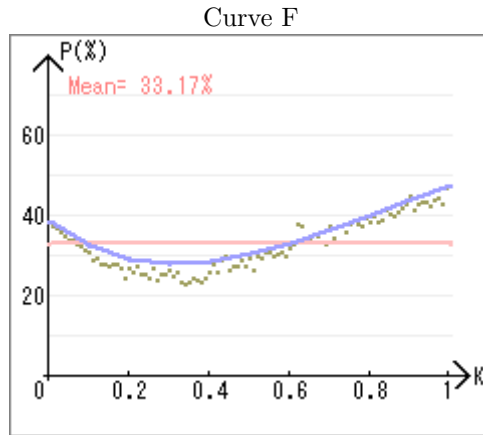


Curve D



Curve E





REFERENCES

- [1] *Bezier curve*, Wikipedia, [http://en.wikipedia.org/wiki/Bezier curve](http://en.wikipedia.org/wiki/Bezier_curve).
- [2] *Curvature*, Wikipedia, <http://en.wikipedia.org/wiki/Curvature>.
- [3] *Bezier triangle*, [http://en.wikipedia.org/wiki/Bezier triangle](http://en.wikipedia.org/wiki/Bezier_triangle).
- [4] <http://www.simplexnumerica.com/Simplex%20%20Manual%20WebG7-Dateien/image224.jpg>.
- [5] Mizuno, R., *Bezier surface*, <http://www.mizuno.org/gl/bs/>.

Reviewer's Comments

The reviewer has only comments on the wordings, which have been amended in this paper.