

动态绘制任意阶 Bezier 曲线的算法及其实现方法*

王福元^{1,2}, 徐家文¹, 王 琪²

(1. 南京航空航天大学 机电学院 江苏 南京 210016 2. 盐城工学院 机械工程学院 江苏 盐城 224003)

摘 要: 介绍了如何利用 VC++ 编程技术在图形系统中动态绘制任意阶 Bezier 工程曲线方法, 并且使绘制的曲线具有捕捉、修改、动态增加型值点等功能。利用同样的方法, 可以实现 B 样条、三次参数等其它工程曲线的绘制, 从而扩大了 Bezier 等曲线的工程应用。

关键词: Bezier 曲线; 动态绘制; 高阶; VC++

中图分类号: TP391

文献标识码: A

文章编号: 1671-532X(2004)04-0029-05

UG、CATIA 等图形系统可以实现 B 样条曲线的动态绘制方法, 但没有 Bezier、三次参数样条等曲线的绘制方法。虽然 B 样条曲线使用最广泛, 但 Bezier、三次参数样条在工程上也有较多的运用。为拓宽工程曲线的应用范围, 在结合现行 Bezier 等曲线的算法的基础上, 利用 VC++ 开发工具实现了高阶 Bezier 等曲线的绘制方法, 较方便地动态实现了曲线的修改。本设计模块可以单独使用, 也可作为一个模块嵌入其它图形系统中使用。从绘制 Bezier 曲线的方法中可了解用现行绘图软件实现曲线的动态绘制、曲线的识别、拖动型值点修改曲线等编程方法, 该方法对于开发其它专业图形系统有极大的帮助。

1 Bezier 曲线的算法

1.1 Bezier 曲线的描述

设 $P_0, P_1, P_2, \dots, P_n$ 为 n 阶 Bezier 曲线的特征多边形控制点, 则 Bezier 曲线一般表达式为

$$P(t) = \sum_{i=0}^n B_{i,n}(t) P_i \quad 0 \leq t \leq 1 \quad (1)$$

其中: $B_{i,n}(t) = C_i^n (1-t)^{n-i} t^i \quad (i=0, 1, \dots, n)$

1.2 绘制 Bezier 曲线的条件

绘制 Bezier 曲线的条件有两种, 一是给出型值点坐标, 另一是给出控制点坐标。在计算中, 一般最终是利用控制点坐标计算来绘制曲线的。当给出的数据是型值点时, 要利用型值点的坐标值反求出控制点坐标, 再利用控制点坐标进行插补运算求出曲线的坐标, 反求控制点的算法为:

设给出的型值点为 $Q_0, Q_1, Q_2, \dots, Q_n$, 要反求的控制点为 $P_0, P_1, P_2, \dots, P_n$, 设参数 $t = 0, 1/n, 2/n, \dots, 1$ 时分别经过 $Q_0, Q_1, Q_2, \dots, Q_n$, 根据 Bezier 定义有

$$\begin{aligned} Q_0 &= P(0) = P_0 \\ Q_1 &= P(1/n) = B_{0,n}(1/n)P_0 + B_{1,n}(1/n)P_1 + \dots \\ &\quad + B_{n,n}(1/n)P_n \end{aligned} \quad (2)$$

$$\begin{aligned} Q_{n-1} &= P((n-1)/n) = B_{0,n}((n-1)/n)P_0 + \\ &\quad B_{1,n}((n-1)/n)P_1 + \dots + B_{n,n}P_n \end{aligned}$$

$$Q_n = P_n$$

把上式写成矩阵形式, 利用矩阵的求逆运算, 求出控制点为式(3)。

$$\begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ B_{0,n}(1/n) & B_{1,n}(1/n) & \dots & B_{n-1,n}(1/n) & B_{n,n}(1/n) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ B_{0,n}((n-1)/n) & B_{1,n}((n-1)/n) & \dots & B_{n-1,n}((n-1)/n) & B_{n,n}((n-1)/n) \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{n-1} \\ Q_n \end{bmatrix} \quad (3)$$

* 收稿日期: 2004-06-24

作者简介: 王福元(1969-), 男, 江苏盐城市人, 盐城工学院讲师, 南京航空航天大学硕士研究生。

2 绘制曲线类的方法

2.1 构造 Bezier 曲线类

在 VC++ 中绘制 Bezier 曲线,首先建立 Bezier 曲线类,利用 Bezier 类的指针来对 Bezier 曲线进行控制。在 VC++ 中所构造 Bezier 的曲线类为

```
CBezier::CBezier( short ColorPen, short ColorBrush, float LineWide, short LineType, float x-Scale, short Layer, int id-only, bool Delete, BpointStruct * BpointList ):CDraw( ColorPen, ColorBrush, LineWide, LineType, x-Scale, Layer, id-only, Delete )
```

各参数的控制对象分别为线型、颜色、线宽、比例、图层、曲线 ID 号、删除状态、型值点数目、存放型值点的结构指针,其中 BpointStruct 结构有 x 和 y 两个成员,存放型值点坐标,CDraw 类为 CBezier 的基类,该类的各个参数在构造时完成初始化。CBezier 类成员有:

```
int m-Numble //存放结点的数目
```

```
BPointStruct m-BpointList //存放型值点的坐标
```

```
BPointStruct m-CpointList //存放反求的控制点的坐标
```

```
void Draw( CDC *pDC, int m-DrawMode, int m-DrawModel, short BackColor ) //绘制曲线函数
```

```
BOOL IsPoint( float x, float y, float jl, float blc, int * No ); //判断曲线是否选中函数
```

```
virtual void MDpoin( int num, float x, float y ) //修改控制点函数
```

```
void GetRec( float *minX, float *minY, float *maxX, float *maxY ); //获得曲线区域函数
```

```
virtual ~CBezier( ) // CBezier 的析函数
```

CBezier 类成员在类构造时完成参数数值的传递,绘制曲线时构造一个 CBezier 类对象。为了存储构造的对象指针,在文档类中建立一个存放曲线指针的指针 m-BezierArray。

```
CTypedPtrArray< CObArray, CBezier* > m-BezierArray;
```

2.2 动态绘制 Bezier 曲线

Bezier 曲线类由 Draw 函数完成曲线的绘制,在 Draw 中完成曲线的反求控制点、插补运算、区域的计算,设置线型、颜色、线宽等功能。Bezier 曲线的绘制方法是当用户选择绘制 Bezier 曲线的功能时,用户在屏幕上拾取型值点时或输入型值点坐标,在屏幕上动态绘制曲线,绘制中每增加一个型值点时,Bezier 曲线的阶次增加一阶,此方

法可以实现任意阶的 Bezier 曲线的绘制,这和 B 样条的曲线绘制方法不同,B 样条曲线可用多个三次 B 样条曲线绘制 N 个型值点样条曲线,得到的曲线是光滑连续的,而 Bezier 曲线随着型值点个数的增加,若用升阶的方法计算,则绘制程序要根据型值点的个数重新计算曲线的坐标。Draw 函数绘制的程序流程如图 1 所示。

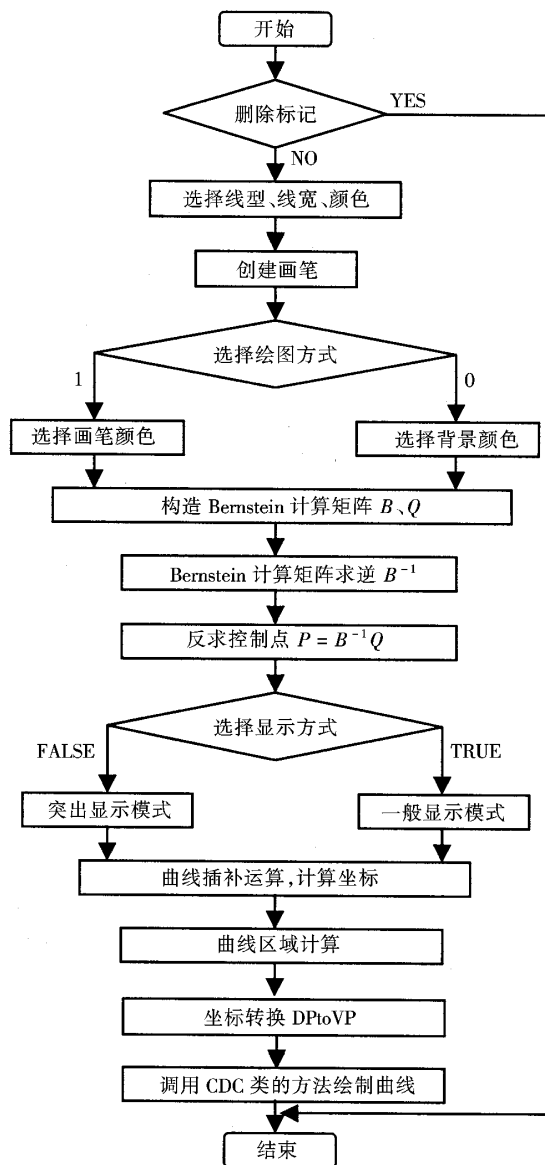


图 1 Draw 函数的流程图

Fig.1 Process flow for draw function

在程序中构造成计算矩阵时,把与坐标值对应的参数 t 的值代入伯恩斯坦基函数计算出与各型值点对应的各个基函数值,其中调用了求阶乘、求幂等函数,然后组合成一个基函数矩阵,再调用矩阵的求逆运算,本模块中采用全选主元高斯消去法进行求逆,得到计算的系数矩阵,求出控制点

坐标,根据 Bezier 曲线的计算公式计算出曲线的坐标。绘制 Bezier 曲线调用了 CDC 类的绘制直线方法,用直线段来近似代替曲线,图形系统中绘制曲线大多采用这种方法,曲线的近似程度和所采用的插值间距有关。为了得到好的显示效果,要把间距控制视觉能接受的范围内。由于无论曲线多长, t 的值都是在 $0 \sim 1$ 之间,当 t 的分隔间距恒定时,在动态绘制过程中,随着曲线长度的增加,直线的逼近效果越差。 t 的划分方法很多,可以利用等间距、等弦长方法来解决这一问题,一般来说, t 的插值间距为非均匀,而让每一段代替曲线的直线段的距离为定值,获得好的视觉效果。

2.3 增加 Bezier 曲线

在动态绘制过程中,当鼠标移动过程中,曲线也随着鼠标的位置不同实现重画。这时的图形指针并没有存储到 Bezier 曲线类的 m-BezierArray 数组中。只有当完成曲线绘制时,发出命令后才进行曲线的最后绘制,对象指针才加入到 m-BezierArray 中存储。当用鼠标的右键结束绘制时,在鼠标的消息函数中增加以下语句

```
pDoc > AddCBezier( m-pColor ,m-brColor ,m-LineWidth , m-LineType , m-xScale , m-Layer , id-only , 0 ,
PushNumb ,BPointXyz) -> Draw( &ddd ,0 ,1 ,m-bColor );
```

AddCBezier 函数在文档类中定义,它完成对象的增加,存储指针。得到指针后,调用 Draw 函数绘制直线。

3 实现曲线类动态修改

3.1 曲线的捕捉

曲线的捕捉功能是图形系统的基本功能,要实现曲线的捕捉前提是获得曲线类的对象指针,建立 Bezier 曲线类的目的就是为了实现曲线的捕捉、移动、修改、删除、恢复,移动和缩放功能。Bezier 类中 IsPoint 函数就是为了实现鼠标移动过程中鼠标位置与曲线之间距离的计算,距离判别最简单的方法是计算鼠标点位置和对象中的存储的型值点距离,若距离在设定的范围内,则把选中的图形突出显示。IsPoint 函数不但返回曲线是否选中的逻辑值,而且利用参数返回选中曲线的型值点的序号。更为复杂的方法是把鼠标点和曲线的坐标进行判别,这种判别方法可用的判别点多,曲线易选中,但计算复杂,系统采用第一种方法实现捕捉功能。

3.2 曲线的修改

在绘图过程中,有时需要修改图形,一般是通过修改型值或控制点的位置来达到所要求的曲线形状。为了修改型值或控制点,在 Bezier 类中增加了 Mdpoint 函数,在函数中根据 IsPoint 返回的型值点或控制点的序号替换它们的坐标,然后重画曲线,这样就实现了动态修改的功能。

3.3 曲线的删除、恢复

曲线删除有两种情况,一种是可以恢复的,另一种是不可恢复的。不可恢复的删除,只要在存放对象指针的 m-BezierArray 数组中删除所要删除的曲线指针。如果是可恢复的删除,只是在对象的成员中增加了删除标志,在重画曲线时,做了删除标志的曲线不绘制,而对象指针并没有从指针数组中删除。

3.4 曲线的移动和缩放

曲线移动是让曲线做平移。平移时,根据平移的 X, Y 方向的距离在曲线的型值或控制点上加上一个平移值,该值的正负号表示移动方向。

曲线的缩放是根据图形的缩放比例对图形放大或缩小。在程序中,增加了缩放的比例因子和从逻辑坐标到实际坐标转换函数 VPtoDP 以及从实际坐标到逻辑坐标的转换函数 DPtoVP,在坐标转换函数中,通过缩放的比例因子改变坐标值达到缩放目的。VPtoDP 函数定义如下

```
void CVDDrawView::DPtoVP( float x ,float y ,int
* X ,int * Y )
{
    * X =( int )( x - m-xStart )/blc );
    if ( m-MapMode = 1 )
        * Y = m-hScreen -( int )( y - m-yStart )/blc );
    else
        * Y =( int )( y - m-yStart )/blc )- m-hScreen ;
}
```

其中,blc 就是坐标变换中的缩放因子,m-xStart、m-yStart 是调节整个图形的平行移动坐标。

4 实现曲线动态绘制的过程

在实现 Bezier 曲线的绘制过程中,绘制过程会调用了许多函数,利用了鼠标的消息函数来实现绘图操作。具体实现的过程是先用鼠标单击左键在屏幕上拾取曲线的型值点,当拖动鼠标时,曲线随着鼠标的移动在不断的变化。鼠标左键每拾取一个型值点时,Bezier 曲线的阶次增加一次,这

样绘制的曲线连续光滑。另外一种方法是用几个低阶的 Bezier 曲线来连接由多个型值点确定的曲线,它的拼接条件和曲线要求的光滑的程度有关系,曲线光滑的条件是在连接点的坐标值相同,曲线斜率相同,即 C^1 连续,但在绘制曲线时,程序较难实现。在绘制曲线时,多采用升阶的方法。

当所有的型值点确定后,单击右键发出结束绘制消息,绘制出曲线的最终形状,若需要修改曲线时,用鼠标选中曲线和要修改的型值点,按住左键,拖动型值点修改曲线至所要求的形状,松开左键结束修改。绘制和修改 Bezier 曲线的过程如图 2 所示。其它功能的实现过程在这不做详细介绍。

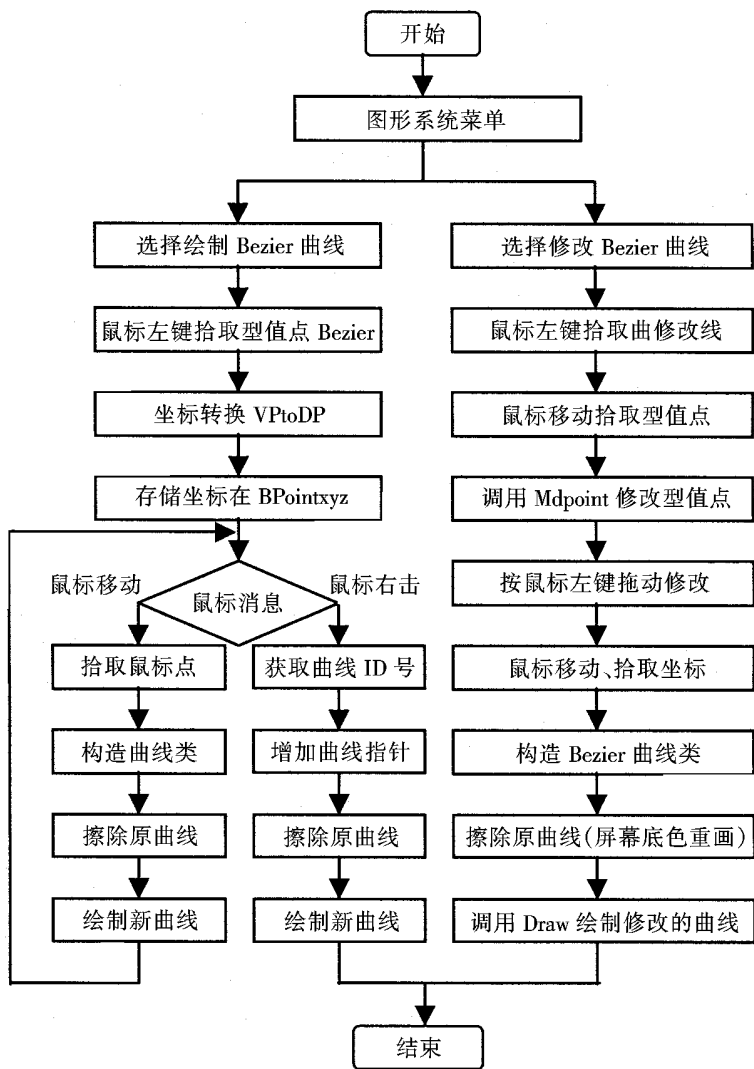


图 2 Bezier 曲线的动态绘制过程

Fig.2 Dynamic Drawing Process of Bezier Curve

5 结束语

在 VC++ 中,利用 VC 的编程技术实现了绘制任意阶 Bezier 曲线,曲线具有捕捉、删除、修改、移动、缩放等功能,可对绘制的曲线形状进行任意的修改,编写的绘图程序具备图形系统的基本操

作功能,在几何造型系统中可作为一个模块使用,从而扩大几何造系统的应用范围。利用 Bezier 曲线的绘制方法,同样可以实现 B 样条,三次参数曲线等曲线的绘制,它们与 Bezier 曲线的算法不同,在编程方法和实现手段上相同。

参考文献：

- [1] 刘根洪, 何仁杰. 微分几何和计算几何[M]. 成都: 四川林学出版社, 2000.
- [2] 关履泰, 罗笑南等. 计算机辅助几何图形设计[M]. 北京: 高等教育出版社, 1999.
- [3] 陈建春. 矢量图形系统开发和编程[M]. 北京: 电子工业出版社, 2004.
- [4] 施法中. 计算机辅助几何设计与非均匀有理 B 样条[M]. 北京: 北京航空航天大学出版社, 1989.
- [5] 李友辉. Visual C++ .NET. 图形图像编程[M]. 北京: 北京电子工业出版社, 2001.

Arithmetic and Method of Dynamically Drawing Random Order Bezier Curve

WANG Fu-yuan^{1,2}, XU Jia-wen¹, WANG Qi²

- (1. College of Mechanical & Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Jiangsu Nanjing 210016, China;
2. College of Mechanical Engineering, Yangcheng Institute of Technology, Jiangsu Yangcheng 224003, China)

Abstract In this paper, the method and arithmetic, how to dynamically draw random order Bezier curves by VC++ programming technology in graph system, are introduced. And these curves drew by this way have characters of snapping, modifying, adding control points. Similarly, B-spline and three order parameter curves can be drawn in same way. Consequently, the drawing method extends the application of these curves for engineering.

Keywords Bezier curve; dynamically drawing; high order; VC++

(上接第 21 页)

Generalized Finite Element Mrthod For Elasto - Plastic Problem

ZHANG Zhen-yu, ZHANG Qing, DONG Xing-ping

(College of Civil Engineering, Hohai University, Jiangsu Nanjing 210098, China)

Abstract In this paper, the authors propose the generalized finite element method by applying the concept of manifold method to the modification of the conventional FEM, advance the fundamental theory for elasto-plastic problem and illustrate the effectiveness of this method by numerical examples.

Keywords Finite Element Method; elasto-plastic problem; generalized finite element method