第4-3章Bezier曲线

课程目标

- Bezier曲线的定义
- Bezier曲线的数学表达式
- 二次Bezier曲线
- 三次Bezier曲线
- Bezier曲线生成算法

4.3 Bezier曲线

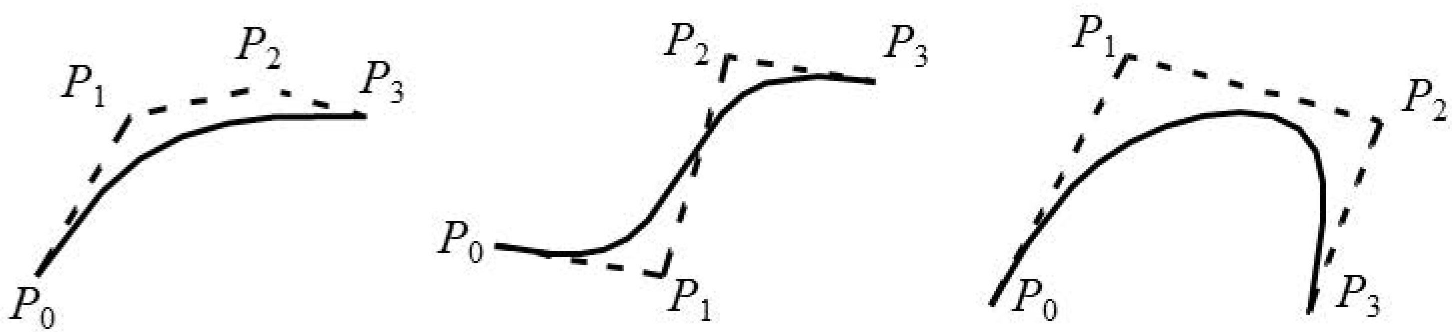
- 抛物样条曲线的特点
  - 曲线通过所有型值点，“点点通过”
- 在外形设计中，
  - 初始给的型值点往往不精确
  - 希望局部修改型值点能在外形上得到直观的反映

4.3 Bezier曲线

- 为了较好地适用于外形设计的特殊要求
  - 法国Bezier提出新的参数曲线逼近方法，称作Bezier曲线。
  - Gordon， Riesenfeld和Forrest等人对Bezier曲线进行修改和发展，提出了B样条曲线。

4.3 Bezier曲线

- **Bezier曲线的定义**
  - 只有第一点和最后一点在曲线上的，其余的顶点则用来定义曲线的导数、阶次和形状。
  - 第一条边和最后一条边分别和曲线在起点和终点处相切。
  - 改变多边折线的顶点位置和曲线形状的变化有联系。



4.3.1 Bezier曲线的数学表达式

Bezier曲线由多项式调和函数推导出,**n+1**个顶点定义一个**n**次多项式

其参数向量表达式为:

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

P_i为顶点位置向量，$B_{i,n}(t)$为伯恩斯坦基函数

即**Bezier**曲线的调和函数

$$B_{i,n}(t)=\frac{n!}{i!(n-i)!}\cdot t^i\cdot(1-t)^{n-i}=C^i{}_n\cdot t^i\cdot(1-t)^{n-i}$$

4.3.1 Bezier曲线的数学表达式

规定：0^0和0！均为1

t=0时

$$P(0) = P_0 B_{0,n}(0) + P_1 B_{1,n}(0) + P_2 B_{2,n}(0) + \ldots + P_n B_{n,n}(0)$$

上式除了第一项外其余各项均为0，所以得

$$P(0) = P_0 B_{0,n}(0) = \frac{n!}{1 \cdot n!} \cdot 0^0 \cdot (1-0)^{n-0} P_0 = P_0$$

可得，曲线通过多边折线的起点

4.3.1 Bezier曲线的数学表达式

t=1时

$$\mathbf{P}(1) = \mathbf{P}_0\mathbf{B}_{0,n}(1) + \mathbf{P}_1\mathbf{B}_{1,n}(1) + \mathbf{P}_2\mathbf{B}_{2,n}(1) + \ldots + \mathbf{P}_n\mathbf{B}_{n,n}(1)$$

上式除了最后一项外其余各项均为0，所以得

$$P(1) = P_n B_{n,n}(1) = \frac{n!}{n! \cdot 1} \cdot 1^n \cdot (1-1)^{n-n} \cdot P_n = P_n$$

可得，曲线通过多边折线的终点

4.3.1 Bezier曲线的数学表达式

- 曲线在两端点处的切矢情况

$$
\begin{aligned}
B'_{i,n}(t) &= \frac{n!}{i!(n-i)!}\cdot\left[i\cdot t^{i-1}\cdot(1-t)^{n-i}-(n-i)\cdot t^{i}\cdot(1-t)^{n-i-1}\right] \\
&= n\left[\frac{(n-1)!}{(i-1)!(n-i)!}\cdot t^{i-1}\cdot(1-t)^{n-i}-\frac{(n-1)!}{i!(n-i-1)!}\cdot t^{i}\cdot(1-t)^{n-i-1}\right] \\
&= n\left[B_{i-1,n-1}(t)-B_{i,n-1}(t)\right]
\end{aligned}
$$

- 于是

$$
P'(t)=n\sum_{i=0}^{n-1}P_i\left[B_{i-1,n-1}(t)-B_{i,n-1}(t)\right]
$$

4.3.1 Bezier曲线的数学表达式

- t=0时，起始点的切矢情况

$$P'(0)=n\{P_0\left[\frac{(n-1)!}{(0-1)!(n-0)!}\cdot 0^{0-1}\cdot(1-0)^{n-0}-\frac{(n-1)!}{0!(n-1-0)!}\cdot 0^{0}\cdot(1-0)^{n-1-0}\right]$$

$$+P_1\left[\frac{(n-1)!}{(1-1)!(n-1)!}\cdot 0^{1-1}\cdot(1-0)^{n-1}-\frac{(n-1)!}{1!(n-1-1)!}\cdot 0^{1}\cdot(1-0)^{n-1-1}\right]$$

$$+P_2\left[\frac{(n-1)!}{(2-1)!(n-2)!}\cdot 0^{2-1}\cdot(1-0)^{n-2}-\frac{(n-1)!}{2!(n-1-2)!}\cdot 0^{2}\cdot(1-0)^{n-1-2}\right]$$

$$=n(P_1-P_0)$$

- 可得，**Bezier**曲线在起点的切矢方向和**Bezier**多边形折线的第一条边相一致

4.3.1 Bezier曲线的数学表达式

- t=1，公式中只有i=n-1,n两项有效，同理

$$P'(1)=n(P_n-P_{n-1})$$

- 可得，Bezier曲线在终点的切矢方向和Bezier多边形折线的最后一条边相一致

4.3.2 二次Bezier曲线

- 顶点P_0，P_1，P_2可定义一条二次Bezier曲线
- 调和函数$B_{i,n}(t)$分别为：

$$B_{0,2}(t) = 1-2t+t^2$$
$$B_{1,2}(t) = 2t-t^2$$
$$B_{2,2}(t) = t^2$$

- 二次Bezier曲线的表达式

$$P(t) = (1-t)^2P_0+2t(1-t)P_1+t^2P_2 \quad (0\leqslant t\leqslant 1)$$

4.3.2 二次Bezier曲线

- 二次Bezier曲线的矩阵形式

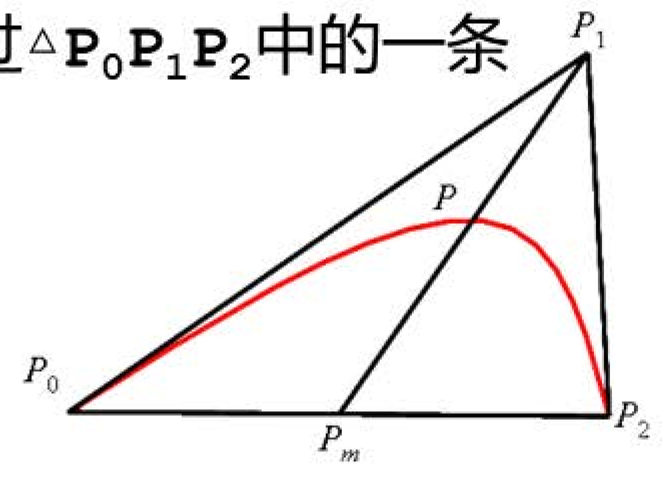
$$P(t) = [t^2\ t\ \ 1] \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix}$$

4.3.2 二次Bezier曲线

- 当**n=2**时，二次**Bezier**曲线有：
  - 起点P_0处有切向量$P'_0=P'_{(t=0)}=2(P_1-P_0)$
  - 终点P_2处有切向量$P'_2=P'_{(t=1)}=2(P_2-P_1)$
  - 当**t=1/2**时：

$$P\left(\frac{1}{2}\right)=\frac{1}{4}P_0+\frac{1}{2}P_1+\frac{1}{4}P_2=\frac{1}{2}\left[P_1+\frac{1}{2}(P_0+P_2)\right]$$

  - 该式说明，二次**Bezier**曲线经过$\triangle P_0P_1P_2$中的一条中线P_1P_m的中点P。



4.3.3 三次Bezier曲线

- P_0,P_1,P_2 ,P_4四点可定义一条三次Bezier曲线
- 调和函数$B_{i,n}(t)$分别为:

$$B_{0,3}(t) = 1-3t+3t^2-t^3$$

$$B_{1,3}(t) = 3t-6t^2+3t^3$$

$$B_{2,3}(t) = 3t^2-3t^3$$

$$B_{3,3}(t) = t^3$$

- 则三次Bezier曲线的表达式

$$P(t)= (1-t)^3P_0+3t(1-t)^2P_1+3t^2(1-t)P_2+t^3P_3$$

$$= (1-3t+3t^2-t)^3P_0+(3t-6t^2+3t^3)P_1+(3t^2-3t^3)P_2 + t^3P_3$$

$$(0\leqslant t\leqslant 1)$$

4.3.3 三次Bezier曲线

- 三次Bezier曲线的矩阵形式

$$P(t) = [t^3\ t^2\ t\ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P0 \\ p1 \\ p2 \\ P3 \end{bmatrix}$$

4.3.3 三次Bezier曲线

- 三次Bezier曲线是二阶连续的。

$$\because\ P'(t)=3(1-t)^2P_0+(3-12t+9t^2)P_1+(6t-9t^2)P_2+3t^2P_3$$

$$P''(t)=6(1-t)P_0+(-12+18t)P_1+(6-18t)P_2+6tP_3$$

$$P''_i(1)=6(1-1)P_0+(-12+18)P_1+(6-18)P_2+6P_3=6(P_1+P_3-2P_2)$$

$$P''_{i+1}(0)=6(1-0)P_1+(-12+0)P_2+(6-0)P_3+0P_4=6(P_1+P_3-2P_2)$$

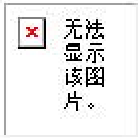
$$\therefore\ P''_i(1)=P''_{i+1}(0)$$

4.3.4 Bezier曲线生成算法

```
double powi(double v,int k)     // 计算v^k
{
    double temp=1.0;
    if(k==0 || v==0)return 1;     // 00=1
    else
    {     for(int i=1;i<=k;i++)
           temp=temp*v;
    }
    return temp;
}
long fac(int m)    //计算m!
{
    int i;
    long temp=1;
    if(m==0)return 1;        // 0!=1
    else
    {      for(i=2;i<=m;i++)
             temp=temp*i;
    }
    return temp;
}
```

4.3.4 Bezier曲线生成算法

```
void Bezier(POINT *p,int n){
      int x,y,i,j,k=100;
      double t,t1,u,v;
      double temp,temp1,temp2,bi;
      t=1.0/k;
      moveto(p[0].x,p[0].y);
      for(j=1;j<k;j++){
          t1=j*t;u=t1;v=1-u;x=0;y=0;
          for(i=0;i<=n;i++){
               temp=(double)fac(n)/fac(i)/fac(n-i);
               temp1=powi(u,i);temp2=powi(v,n-i);
               bi=temp*temp1*temp2;
               x=x+bi*p[i].x;y=y+bi*p[i].y;
          }
          lineto(x,y);
      }
      lineto(p[n].x,p[n].y);
}
```



桔子酒店

- 橘子钟表程序主要分成三个部分：画表盘、画表针、显示当前时间。
- 画表盘部分运用到了三次贝塞尔曲线、HSL 颜色模型以及字符串格式化命令。画表针主要涉及到计算各表针运动的弧度。显示当前时间所用字体为等宽字体，其作用在于居中后效果更均匀。
- 程序当中计算三次贝塞尔曲线坐标部分，定义了 13 个点，其中 0 点和 11 点 12 点重合，3 点和 4 点重合，5 点和 6 点重合，10 点和 9 点重合。这样做的目的是便于确定起始点、控制点和终点。

orangeClock.cpp