CURVAS PARAMÉTRICAS

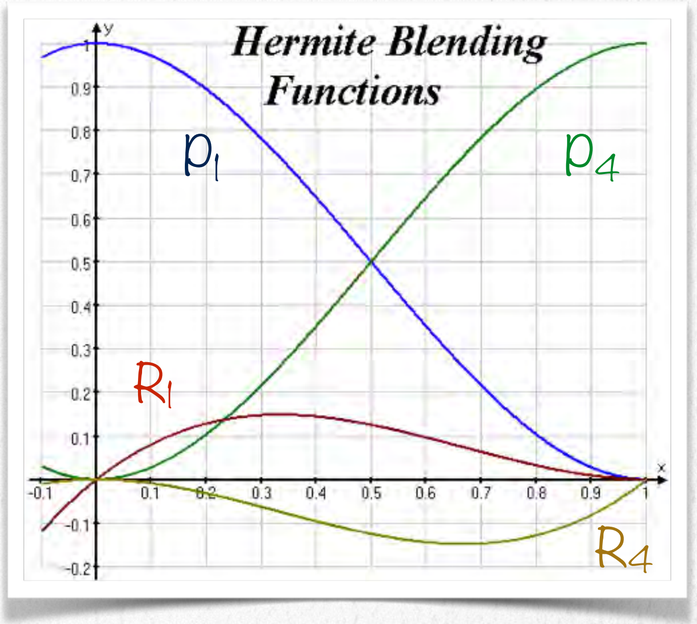
Funciones de Blending : Hermite

$$Q(t) = T.M_H.G_H$$

$$B_H(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$Q(t) = B_H(t) \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

- $B_H(t)$ indica el peso de cada elemento en G_H.



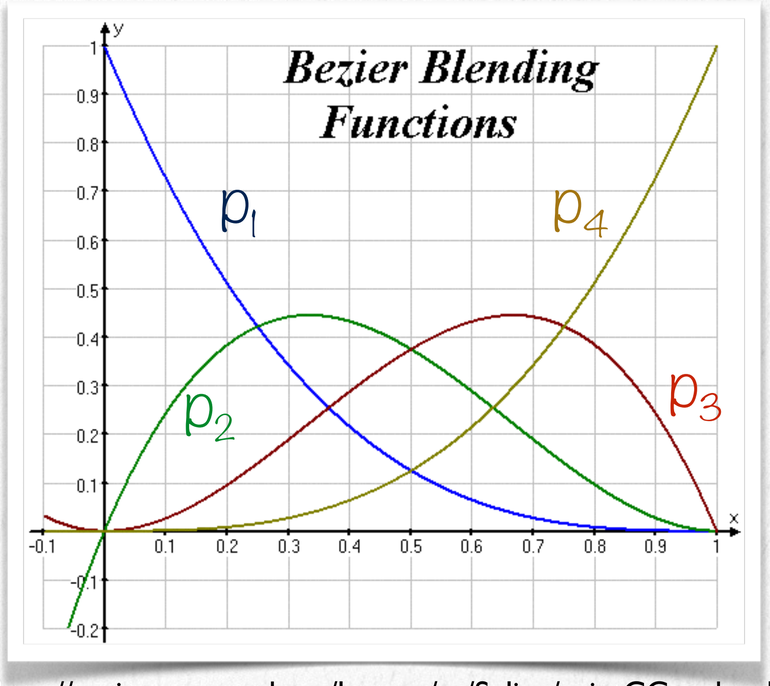
http://escience.anu.edu.au/lecture/cg/Spline/printCG.en.html

Funciones de Blending : Bézier

$$B_B(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$Q(t) = B_B(t). \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

- Funciones de blending de Bézier o Polinomios de Bernstein.



http://escience.anu.edu.au/lecture/cg/Spline/printCG.en.html

Algoritmo de Casteljau: caso cúbico

- Otra forma de construir las curvas de Bézier.
- Consideramos una polilínea con vértices P_1, P_2, P_3, P_4 y un valor $t \in [0,1]$.
- Construir para i=1,2,3 un punto:

$$P_i^{(1)}(t) = (1-t)P_i + tP_{i+1}$$

- Considerar luego la polilínea con vértices $P_1^{(1)}(t)$, $P_2^{(1)}(t)$, $P_3^{(1)}(t)$, construir para i=1,2 el punto:

$$P_i^{(2)}(t) = (1-t)P_i^{(1)} + tP_{i+1}^{(1)}$$

R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

Algoritmo de Casteljau: caso cúbico

- Finalmente tenemos:

$$Q(t) = P_i^{(3)}(t) = (1-t)P_i^{(2)} + tP_{i+1}^{(2)}$$

- La curva $Q:[0,1] \rightarrow \mathbf{R}^3$ es precisamente la curva de Bézier cúbica con P_1, P_2, P_3 y P_4 como puntos de control.

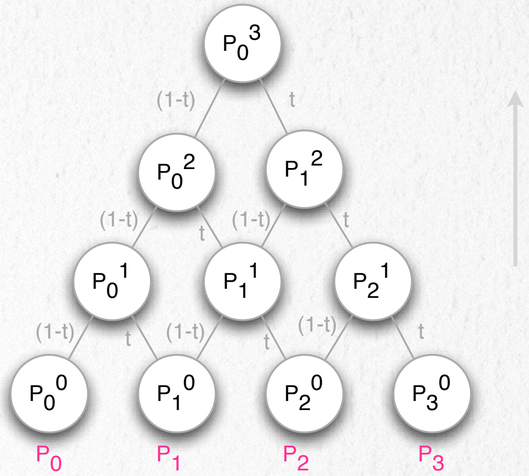
R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

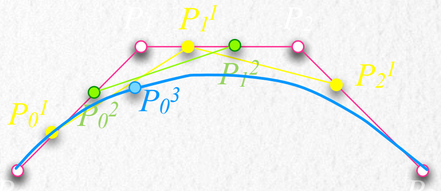
Cubic Blend (por Casteljau)

$$P_0^1(u) = (1-u)P_0 + uP_1$$
$$P_1^1(u) = (1-u)P_1 + uP_2$$
$$P_2^1(u) = (1-u)P_2 + uP_3$$

$$P_0^2(u) = (1-u)P_0^1(u) + uP_1^1(u)$$
$$P_1^2(u) = (1-u)P_1^1(u) + uP_2^1(u)$$

$$P_1^3(u) = (1-u)P_0^2(u) + uP_1^2(u)$$





R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

Prueba

- Consideramos la curva Q(t) definida por el algoritmo de Casteljau.
- Notamos que es una curva cúbica que depende linealmente de los puntos de control. Tiene entonces la forma $Q(t) = T.M_C.G_B$.
- T y B igual a Bézier.
- Mostrar entonces que $M_C = M_B$ calculando Q(t) para puntos particulares.

R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

Curvas de Bézier de orden n

- Generalizar el algoritmo de Casteljau para n≥2 puntos de control : curva de Bézier de orden n-1.
- Sean P0, ..., Pn-1 puntos de control.
- Sea t∈[0,1]
- Para definir $Q(t)$:
  - Hacemos $b_i^{(0)}(t) = P_i$ para i=0,...,n-1.
  - Para r=1, ..., n-1 e i=0,...n-1-r:

$$b_i^r(t) = (1-t)b_i^{r-1}(t) + tb_i^{r-1}(t)$$

  - definimos así por recurrenciaun punto $Q(t) = b_0^{(n-1)}(t)$

R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

Ejercicio

Comprobar que para n=4 obtenemos curvas de Bézier cúbicas.

Polinomios de Bernstein

- Los polinomios de Bernstein $B_{i,n}$ de grado n se definen para i=0,...,n con la fórmula:

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

- Donde por convención 0! = 1.

- Esto se puede escribir en términos de sus coeficientes binomiales:

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}$$

- Existen (n+1) polinomios de Bernstein de grado n.

R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

Polinomios de Bernstein

Teorema:

Los polinomios de Bernstein forman una base del espacio vectorial de polinomios de grado inferior o igual a n. Todo polinomio de grado inferior o igual puede entonces escribirse como una combinación lineal de los $B_{i,n}$. Se pueden descomponer las coordenadas de una curva de Bézier como combinaciones lineales de los $B_{i,n}$.

R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

Polinomios de Bernstein

- Propiedades que nos sirven para el estudio de las curvas de Bézier:
  - Los polinomios de Bernstein tienen valores positivos o nulos en el intervalo [0,1].
  - La suma de todos los polinomios de Bernstein es igual a 1.
- Para calcularles podemos utilizar también:

$$B_{i,n}(t) = (1-t)Bi, n-1 + tB_{i-1,n-1}$$

R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

Expresión de curvas de Bézier con los Bi,n

- Sean $P_0, \ldots, P_{n-1}$ puntos de control y $t \in [0,1]$.
- La curva de Bézier Q de orden (n-1) que tiene como puntos de control los puntos P_i se escribe:

$$Q(t) = \sum_{i=0}^{n-1} P_i B_{i,n-1}(t).$$

- De este modo, la curva Q se expresa como una suma de polinomios $B_{i,n-1}$ ponderada por los puntos de control.

R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

Derivada de una curva de Bézier

- Derivada de los polinomios de Bernstein:

$$\frac{d}{dt}B_{i,n}(t) = n\left(B_{i-1,n-1}(t) - B_{i,n-1}(t)\right).$$

- Derivada de una curva de Bézier.

$$\frac{d}{dt}Q(t) = (n-1)\sum_{i=0}^{n-1} P_i\left(B_{i-1,n-2}(t) - B_{i,n-2}(t)\right)$$

- Con la convención $B_{-1,n-2} = B_{n-1,n-2} = 0$

R. Malgouyres. Algorithmes pour la Synthèse d'images et l'animation 3D. Dunod, 2005.

Derivada de una curva de Bézier

- Utilizando el algoritmo de Casteljau se puede deducir la expresión:

$$\frac{d}{dt}Q(t) = \sum_{i=0}^{n-2} P_i' B_{i,n-2}(t).$$

- Donde $P_i' = (n-1)(P_{i+1} - P_i)$ para i=0, …, n-2.
- Vemos de esto que la derivada de la curva de Bézier de orden (n-1) y con puntos de control Pi, es otra curva de Bézier de orden (n-2) y con puntos de control Pi'.
- Podemos entonces utilizar todos los algoritmos para calcular curvas de Bézier para calcular las derivadas de una curva de Bézier.

Propiedades, ventajas e inconvenientes

- Invarianza Afín.
  - invariantes por translación, rotación y cambio de escala.
- Envolvente convexo.
  - todos los puntos de la curva se encuentran en el envolvente convexo de los puntos de control.

Propiedades, ventajas e inconvenientes

- Control global.
  - modificar un punto de control, afecta a toda la curva.
- Grado elevado.
  - el grado de las curvas de Bézier depende del número de puntos de control.