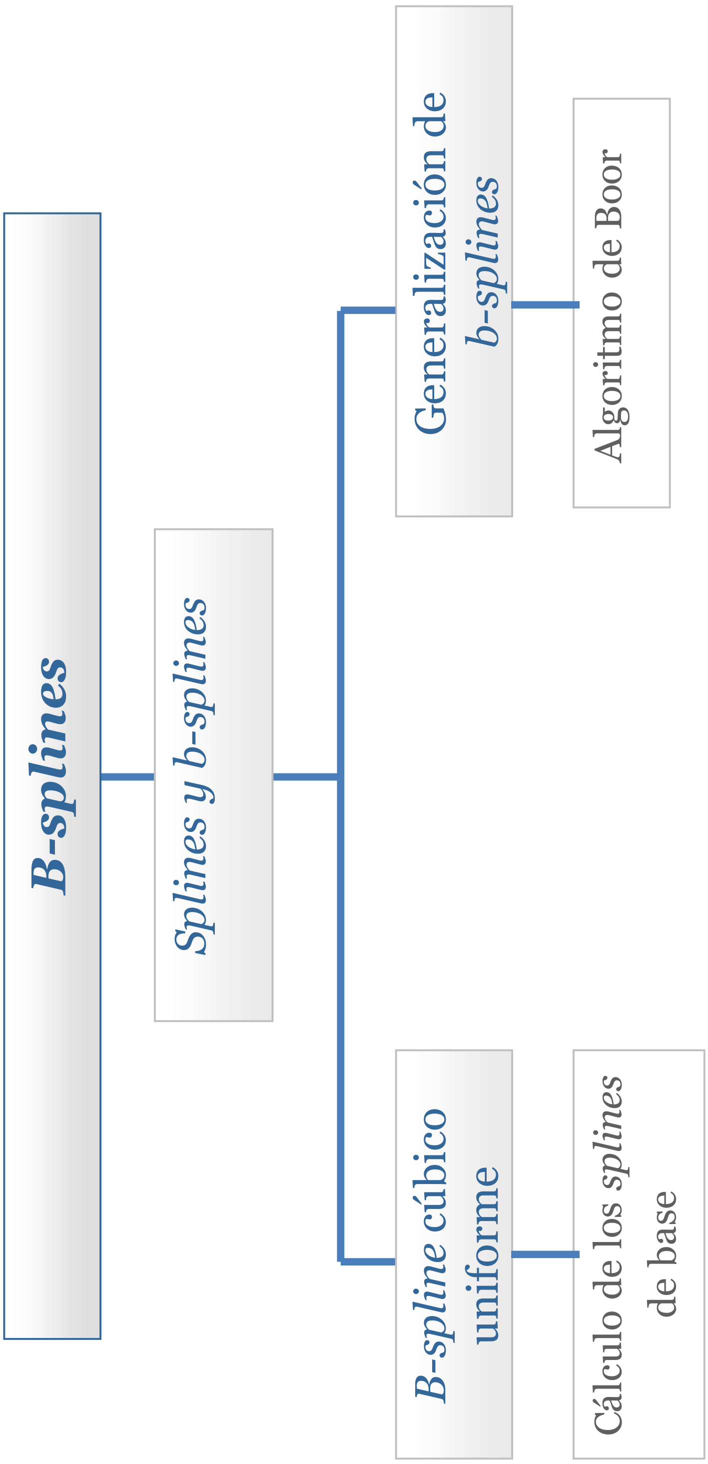
B-splines

9

TEMA

Esquema


B-splines
Splines y b-splines
B-spline cúbico uniforme
Cálculo de los *splines* de base
Generalización de *b-splines*
Algoritmo de Boor

Ideas clave

9.1. ¿Cómo estudiar este tema?

Para estudiar este tema lee las **Ideas clave** que encontrarás a continuación.

En este tema se estudia una técnica de aproximación muy utilizada: los *b-splines*.

» Qué es un *b-spline*.
» Cómo construir un *b-spline* cúbico uniforme.
» *B-splines* en dimensiones superiores.

9.2. Introducción

Aunque los *splines* permiten encontrar curvas regulares definidas a trozos que pasen por una serie de puntos dados, siguen teniendo un problema importante: el cambio de un punto implica recalcular todos los polinomios. Los *b-splines* son una suma de polinomios definidos a trozos (*splines*) que tienen la ventaja de estar definidos localmente de forma que si cambia uno de los puntos solo hay que recalcular los polinomios correspondientes a los intervalos más próximos (figura 9.1). Esos puntos pueden cerrar mucho la curva sin necesidad de aumentar la densidad de puntos en la zona.

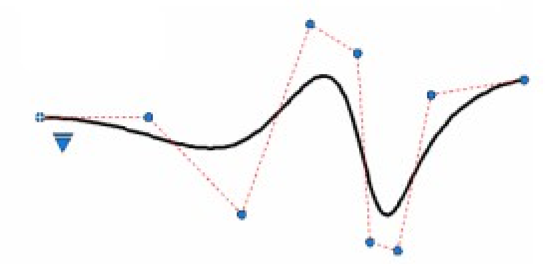


Figura 9.1. Representación de un *b-spline* bidimensional
Fuente: https://knowledge.autodesk.com/

Los *b-splines* están compuestos por una suma de polinomios de grado 3 llamados **polinomios base**. Los más comunes tienen grado 3 (Figura 9.2).

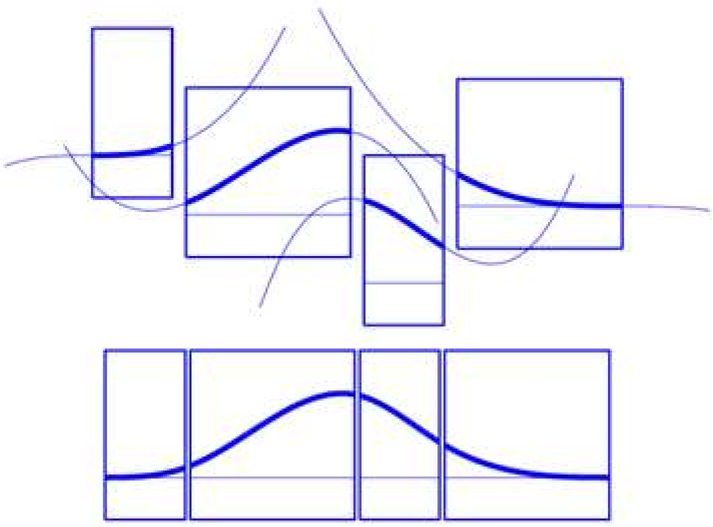


Figura 9.2. Representación de las funciones base de un b-spline.
Fuente: http://es.mathworks.com/

9.3. *B-spline* cúbico uniforme

En este apartado vamos a explicar la construcción y características del *b-spline* más utilizado: el *b-spline* cúbico uniforme. Tal y como pasaba con los *splines*, los *b-splines* cúbicos ofrecen suficiente regularidad, por lo que no se suele considerar necesario complicar los cálculos considerando polinomios de mayor grado.

La idea es la siguiente: se consideran una serie de nodos que forman una red de paso h y se definen *splines* cúbicos en cada subintervalo. Estas funciones se llaman *b-splines* de base y puede probarse que son esencialmente iguales, solo que trasladadas a lo largo de cada nodo (figura 9.3).

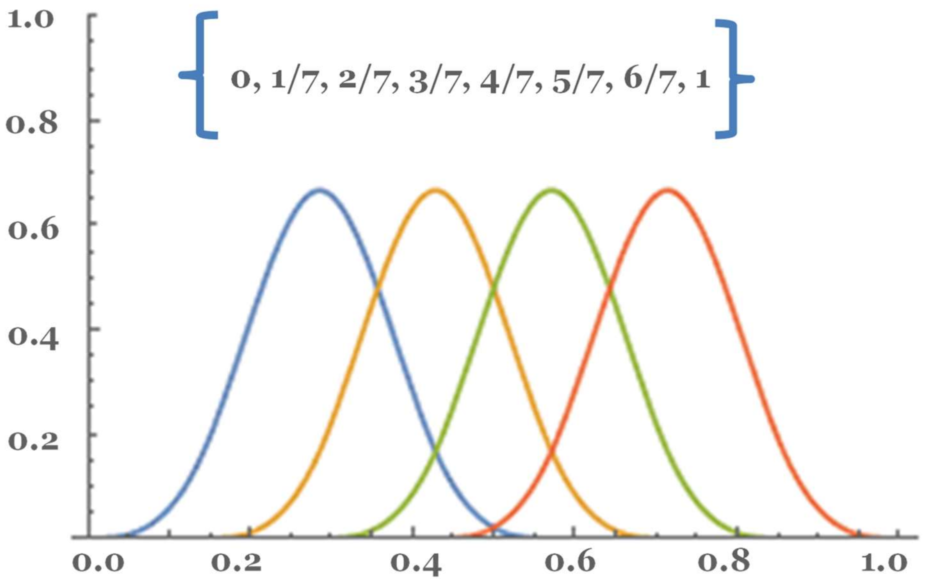


Figura 9.3. Funciones de base

Fuente: http://www.cs.mtu.edu/

Además se definen de tal forma que la suma de todas funciones de base sea 1 (figura 9.4).

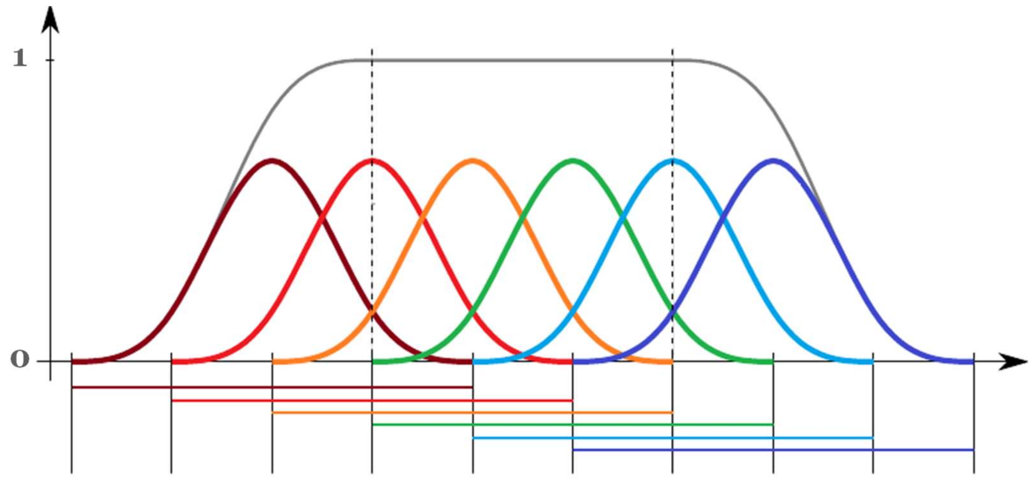


Figura 9.4. Suma de funciones de base

Fuente: http://www.brnt.eu/

A partir de lo que se vio en el tema anterior puede probarse que las ecuaciones de base de un *b-spline* cúbico normalizado con nodos en los puntos $x_i, i \in \{0, 1, n-1\}$ son de la forma:

$$B_i(x) = \begin{cases} s_{i-1}(x), & con\ x \in [i-2, i-1] \\ s_i(x), & con\ x \in [i-1, i] \\ s_{i+1}(x), & con\ x \in [i, i+1] \\ s_{i+2}(x), & con\ x \in [i+1, i+2] \\ 0, & resto \end{cases}$$

Donde:

$$s_{i-1}(x) = -\frac{1}{6h^3}(x_{i-2} - x)^3$$

$$s_i(x) = -\frac{1}{6h^3}(x_{i-2} - x)^3 + \frac{2}{3h^3}(x_{i-1} - x)^3$$

$$s_{i+1}(x) = \frac{2}{3h^3}(x_{i+1} - u)^3 + \frac{1}{6h^3}(x_{i+2} - u)^3$$

$$s_{i+2}(x) = -\frac{1}{6h^3}(x_{i+2} - u)^3$$

Aunque no lo parezca a simple vista las funciones de base se pueden obtener de la anterior aplicando la fórmula:

$$s_i(x) = s_0(x - i)$$

Por tanto, un cambio en un nodo solo afecta a los 4 *splines* más cercanos a él.

Un *b-spline* es una función de la forma:

$$S(x) = \sum_{i=0}^{n} P_i s_i(x)$$

Donde los P_i son pesos. La interpretación geométrica de los pesos es que son las coordenadas de los llamados **puntos de control**. Un *b-spline* pasa por todos los nodos y aproxima los puntos de control.

Definición recursiva

Dada una serie de nodos $a_1, \ldots, a_n$, se definen los b-splines de forma recursiva como:

$$N_i^0(u) = \begin{cases} 1, si\ u \in [a_i, a_{i+1}) \\ 0, eoc \end{cases}$$

$$N_i^n(u) = \alpha_i^{n-1} N_i^{n-1}(u) + \left(1 - \alpha_{i+1}^{n-1}\right) N_{i+1}^{n-1}(u)$$

Donde:

$$\alpha_i^{n-1} = \frac{u - a_i}{a_{i+n} - a_i}$$

Por ejemplo, si tenemos los nodos $a_0 = 0, a_1 = 1$, ..., $a_n = n$, el *b-spline* de grado uno se define:

$$N_0^0(u) = \begin{cases} 1, si\ u \in [0,1) \\ 0, eoc \end{cases}$$

Su representación (Figura 9.4) es la de una constante en el intervalo [0,1) o cero en el resto.

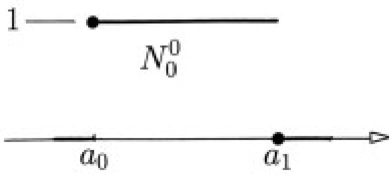


Figura 9.4. *b-spline* de grado cero.
Fuente: http://www.brnt.eu/

Si continuamos iterando, llegamos a la expresión:

$$N_0^1(u) = \alpha_0^0 N_0^0(u) + (1 - \alpha_1^0) N_1^0(u) = \begin{cases} u & , si\ u \in [0,1) \\ \dfrac{-u+3}{2} + 1, si\ u \in [1,2) \\ 0, & eoc \end{cases}$$

ya que:

$\alpha_0^0 = \frac{u-a_0}{a_1-a_0} = \frac{u-0}{1-0}$ y $\alpha_1^0 = \frac{u-a_1}{a_3-a_1} = \frac{u-1}{3-1}$

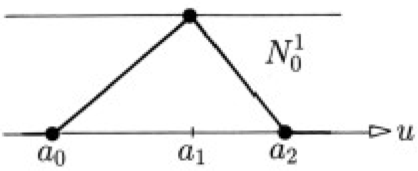


Figura 9.5. b-spline de grado uno.
Fuente: http://www.brnt.eu/

Para el polinomio de grado 2 seguimos iterando:

$N_0^2(u) = \alpha_0^1 N_0^1(u) + (1 - \alpha_1^1) N_1^1(u)$, donde:

$\alpha_1^0 = \frac{u - a_0}{a_2 - a_0} = \frac{u-0}{2-0}$ y $\alpha_1^1 = \frac{u - a_0}{a_2 - a_0} = \frac{u-0}{2-0}$

Necesitamos calcular el polinomio de grado 1 en el punto 1, es decir,

$N_1^1(u) = \alpha_1^0 N_1^0(u) + (1 - \alpha_2^0) N_2^0(u)$,

que se calcula de forma análoga.

9.4. Generalización de *b-splines*

Un *b-spline* puede tener los nodos distribuidos de manera no uniforme. Entonces la forma de los *splines* base varía ligeramente (figura 9.6).

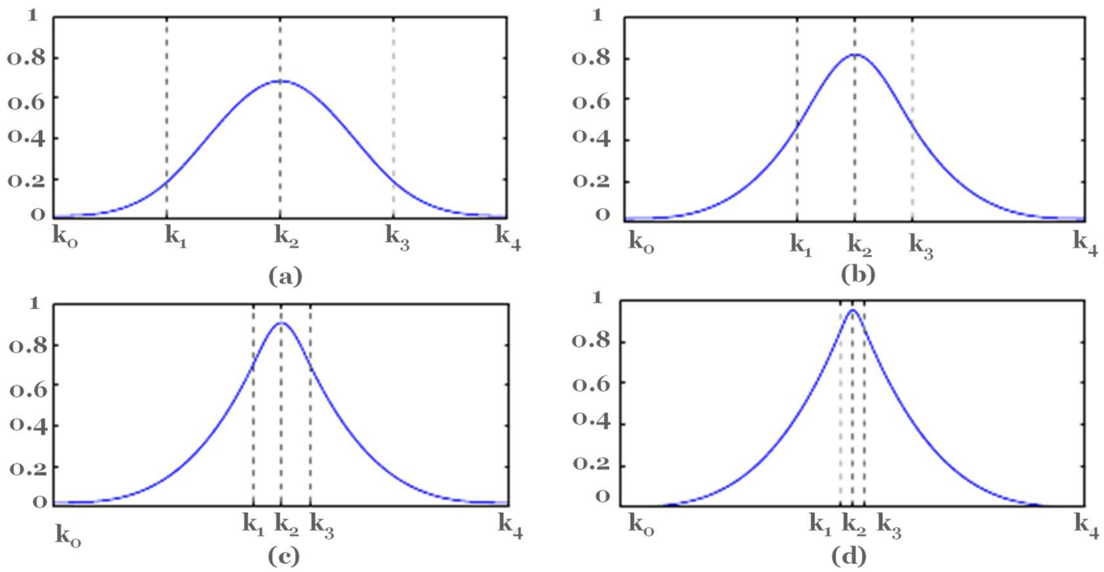


Figura 9.6. Funciones base con distintas disposiciones de nodos
Fuente: http://www.brnt.eu/

Por ejemplo, con las funciones de base de la figura 9.6 puede definirse el *b-spline* de la figura 9.7.

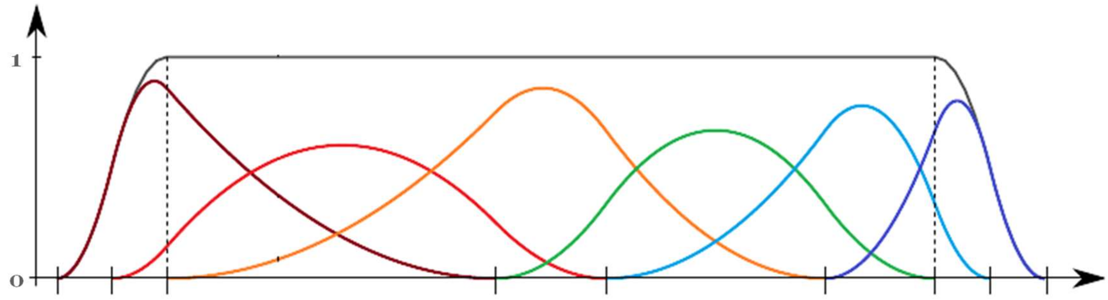


Figura 9.7. Funciones base de un *b-spline* no uniforme
Fuente: http://www.brnt.eu/

En la figura 9.8 los segmentos representan los pesos que se aplican a cada línea base.

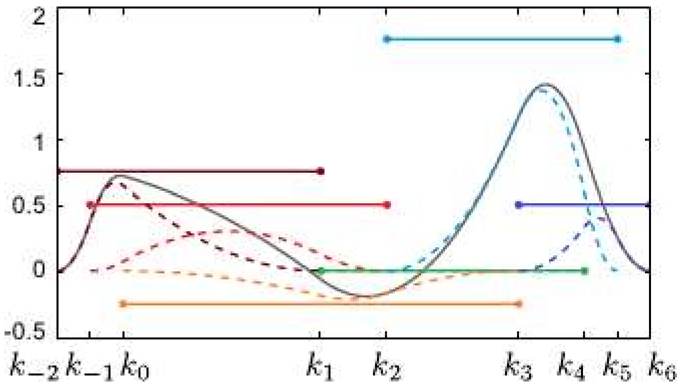


Figura 9.8. Funciones base de un *b-spline* no uniforme

Fuente: http://www.brnt.eu/

Estos *b-splines* pueden calcularse tal y como se ha descrito en el apartado anterior, simplemente considerando intervalos no uniformes. También pueden calcularse con el algoritmo de Boor, descrito en el siguiente apartado.

9.5. Algoritmo de Boor

Este algoritmo permite calcular los puntos sobre el *b-spline* sin necesidad de calcular la curva de forma explícita. Es muy útil para dibujar el detalle de la curva en un subintervalo dado.

Coordenadas polares

Se utilizan para etiquetar los puntos de control. Cada punto de un *b-spline* de grado n necesita n coordenadas. Si el *b-spline* está definido sobre un intervalo $[a, b]$, las coordenadas polares $P(u_1, u_2, \dots, u_n)$ se definen de forma que:

$u_j = a$, si $j \leq n - i$

$u_j = b$, en otro caso

Si P es un *b-spline* de grado 3, en el intervalo $[0,2]$ podemos considerar las coordenadas polares (Figura 9.9):

$$P(0,0,0), P(0,0,2), P(0,2,2), P(2,2,2)$$

y en el intervalo$[2,3]$:

$$P(2,2,2), P(2,2,3), P(2,3,3), P(3,3,3)$$

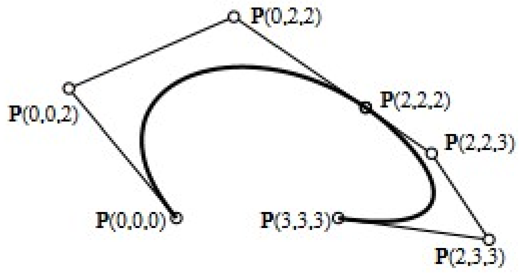


Figura 9.9. Coordenadas polares en un *b-spline*.

Fuente: http://cagd.cs.byu.edu/

Por tanto, si queremos dividir el intervalo $[0,1]$ en los intervalos $[0, a]$ y $[a, 1]$:

Las coordenadas polares que tenemos son:

$$P(0,0,0), P(0,0,1), P(0,1,1), P(1,1,1)$$

Al añadir el punto a, resulta:

$$P(0,0,0), P(0,0,a), P(0,a,a), P(a,a,a)$$

y

$$P(a,a,a), P(a,a,1), P(a,1,1), P(1,1,1)$$

Algoritmo de Boor

En cada paso, trazamos dos segmentos que dividan los segmentos que forman los puntos de control (Figura 9.10). Los coeficientes a_{pi} marcan la división del segmento en cada paso.

Paso 1:

$$P(0,0,a) = (1 - a_{10}) \cdot P(0,0,0) + a_{10} \cdot P(0,0,1)$$
$$P(0,a,1) = (1 - a_{11}) \cdot P(0,0,1) + a_{11} \cdot P(0,1,1)$$
$$P(a,1,1) = (1 - a_{12}) \cdot P(0,1,1) + a_{12} \cdot P(1,1,1)$$

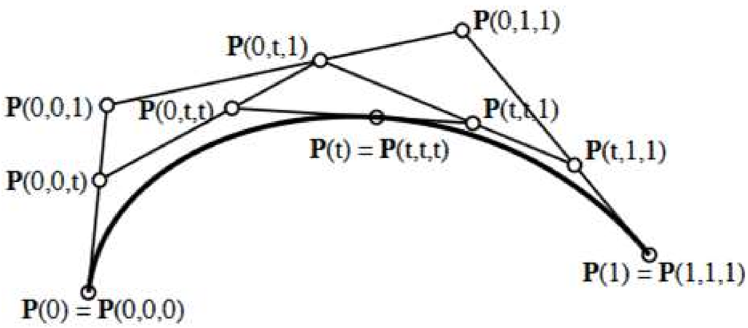


Figura 9.10. Coordenadas polares en un *b-spline*.
Fuente: http://cagd.cs.byu.edu

Paso 2:

$$P(0,a,a) = (1 - a_{21}) \cdot P(0,0,a) + a_{21} \cdot P(0,a,1)$$
$$P(a,a,1) = (1 - a_{22}) \cdot P(0,a,1) + a_{22} \cdot P(a,1,1)$$

Paso 3:

$$P(a,a,a) = (1 - a_{32}) \cdot P(0,a,a) + a_{32} \cdot P(a,a,1)$$

Lo + recomendado

Lecciones magistrales

Producto sensorial

En esta clase magistral vamos a explicar por encima el producto tensorial y vamos a detallarlo un poco más para ver cómo se utiliza para todo el tema de curvas de Bézier en el espacio o de interpolación en un sentido más general.

La lección magistral está disponible en el aula virtual

No dejes de leer…

Triedro de Frenet y *b-splines*

En este enlace se muestra un algoritmo en Mathematica 5.0 que permite generar movimiento de una superficie *b-spline* utilizando el triedro de Frenet.

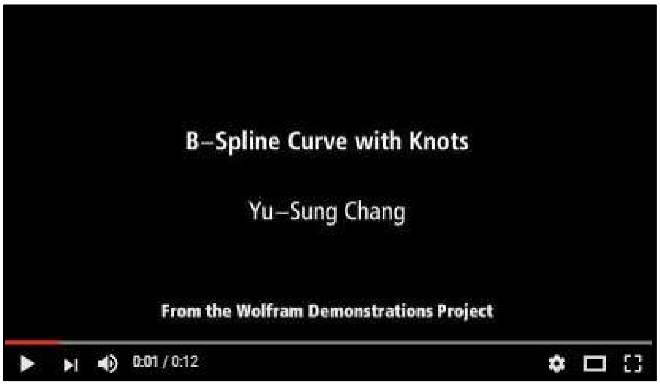
"Aplicación del Triedro Móvil de Frenet-Serret en la animación por desplazamiento de superficies Bspline"

Accede al documento desde el aula virtual o a través de la siguiente dirección web:
http://www.ime.usp.br/~jvalentm/aplicacion.pdf

No dejes de ver…

Dibujo *b-splines*

En este vídeo se muestra la utilización de una herramienta de dibujo para dibujar *b-splines* introduciendo distintos nodos.



Accede al vídeo desde el aula virtual o a través de la siguiente dirección web:
https://www.youtube.com/watch?v=EA6fptmIKZE

+ Información

A fondo

B-splines con *inkscape*

En este documento se muestra cómo dibujar *b-splines* utilizando *inkscape*.

Accede al documento desde el aula virtual o a través de la siguiente dirección web:
http://blogs.20minutos.es/mati-una-profesora-muy-particular/tag/geodesicas/

Métodos de Bézier y *b-splines*

Paluszny, M., Prautzsch, H. y Boehm, W. (2005). *Métodos de Bézier y b-splines*. Alemania: KIT Scientific Publishing.

En los capítulos 5, 6 y 7 de este libro se tratan en profundidad los contenidos de este tema.

Accede al libro desde el aula virtual o a través de la siguiente dirección web:
http://books.google.es/books?id=EP-mijyNnvgC&printsec=frontcover

Dibujar *splines* con OpenGL

En este enlace se muestra cómo dibujar *splines* con OpenGL.

Almighty Bus Error

Accede al artículo desde el aula virtual o a través de la siguiente dirección web:
http://www.almightybuserror.com/2009/12/04/drawing-splines-in-opengl.html

Test

1. Un *spline:*

A. Puede considerarse un *spline*.

B. Mejora algunos aspectos de los *splines* cúbicos estándar.

C. A y B son ciertas.

2. Un *b-spline:*

A. Puede utilizarse para aproximar valores.

B. Puede utilizarse para interpolar valores.

C. A y B son ciertas.

3. La suma de las funciones de base de un *b-spline* normalizado:

A. Es igual a 1.

B. Depende de los puntos de control.

C. Depende de las funciones de base.

4. En un *b-spline* cúbico normalizado:

A. Cada función de base puede tener una altura distinta.

B. El número de funciones de base depende del número de nodos.

C. A y B son falsas.

5. En un *b-spline* cúbico:

A. Cada función base está formado por 4 polinomios.

B. Cada función base es un polinomio cúbico.

C. A y B son falsas.

6. En un *b-spline* cúbico:

A. El apuntamiento de la función base depende de la disposición de los puntos de control.

B. El apuntamiento de la función base depende de la distancia entre los nodos.

C. A y B son ciertas.

7. Para calcular la ecuación de un *b-spline* genérico se utiliza:

A. Una fórmula recursiva.

B. La fórmula de Cox de Boor.

C. A y B son ciertas.

8. Los pesos de la fórmula de un *b-spline* pueden interpretarse como:

A. La importancia de una curva de base en la fórmula total.

B. Las coordenadas de los puntos de control.

C. A y B son ciertas.

9. La sección normal de un *b-spline* tridimensional es:

A. Un *b-spline* bidimensional.

B. Otro *b-spline* tridimensional.

C. A y B son falsas.

10. Si un *spline* cúbico tiene 10 puntos de control, ¿cuántos segmentos de curva están contenidos en la componente convexa?

A. 7.

B. 8.

C. 9.