Reconstrucción de superficies apartir de contornos usando superficies B-spline.

M. en. C. Daniel A. Cervantes Cabrera
Dr. Pablo Barrera Sánchez.

1. Introducción.

En la actualidad una gran cantidad de algoritmos para la reconstrucción de superficies están basadas en técnicas spline, en este reporte técnico presentamos un algoritmo e implementación para la construcción de superficies a partir de un conjunto de contornos. Iniciaremos dando una breve descripción de los conceptos matemáticos necesarios.

Dados tres enteros positivos n, m, p y puntos de control en $\mathbb{R}^3$, $P_0, \ldots, P_p$, una curva B-spline de grado p está dada por

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \text{ para } u \in [0, 1]$$

en donde cada $N_{i,p}(u)$ es una función base B-spline definida por sobre un vector de $m + 1$ nodos estrictamente crecientes y no uniformes de la forma

$$U = \{\underbrace{0, \ldots, 0}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{1, \ldots, 1}_{p+1}\}$$

Ahora dadas dos curvas B-spline

$$C(u) = \sum_{i=0}^{n_k} N_{i,p_k}(u) P_i^k$$

definidas sobre los vectores de nodos $U^k = \{u_0^k, \ldots, u_{m_k}^k\}$ para $k = 1, 2$, se desea obtener una interpolación entre estas dos.

Es decir requerimos una superficie $S(u, v)$ cuyas isocurvas de los extremos en dirección del parámetro u, sean las curvas $C_1(u)$, $C_2(u)$, y para $u = \bar{u}$ fijo $S(\bar{u}, v)$ es una interpolación lineal entre los puntos $C_1(\bar{u})$ y $C_2(\bar{u})$.

Ahora, si hacemos $P_0 = C_1(\bar{u})$ y $P_1 = C_2(\bar{u})$, esta interpolación se puede representar en términos de curvas B-spline como

$$C(v) = P_0(1 - v) + P_1 v = \sum_{i=0}^{1} N_{i,1}(v) P_i$$

con $V = \{0, 0, 1, 1\}$.

Entonces se deduce que la superficie que necesitamos está dada por

$$S(u,v) = \sum_{i=0}^{2} \sum_{j=0}^{n_i} N_{i,p}(u) N_{j,1}(v) P_{i,j} \tag{1}$$

en donde $P_{i,0} = P_i^1$ y $P_{i,1} = P_i^2$ y el grado p está en función de p_0 y p_1.

Este proceso se puede generalizar para un conjunto de n curvas B-spline, con las que se reconstruirá la superficie. En la siguiente sección describimos esto en términos algorítmicos.

2. Descripción del algoritmo.

El algoritmo de reconstrucción de la superficie que se implementa en **Recosnt-0.0**, requieren por definición que los puntos de la curvas de nivel tengan un cierto orden. Por esta razón los primeros pasos del procedimiento se encargan de ordenar los puntos de cada curva de nivel de acuerdo a la siguiente curva. Una vez realizado esto, se reconstruye la superficies definidas en (1).

Si denotamos como $P^i = \{P_0^i, \ldots, P_{n_i}^i\}$ al conjunto de puntos de la $i-$ésima curva de nivel y $C_i(u)$ la respectiva curva de interpolación B-spline, en donde $u \in \mathbb{R}$ y $i = 0, \ldots, k$ para $k > 0$, el proceso de reconstrucción sigue los siguientes pasos.

Algoritmo.

1. Ordenar cada punto del conjunto P_{i+1} de acuerdo a la distancia más pequeña que hay a los puntos de P_i para $i = 0, \ldots, k-1$.

```
for(l=0;l<=n_i;l++)
  for(j=0;j<=n_{i+1};j++)
    P_l^{i+1} = distancia_minima{P_l^i, P_j^{i+1}}
```

2. Si $\bar{U} = \{\bar{u}_0, \ldots, \bar{u}_n\}$ es un conjunto de valores parametrales equidistantes tales que $\bar{u}_i \in [0,1]$ para $i = 0, \ldots, n$. Entonces para cada punto $P_l^i = C_i(\bar{u}_l)$ encontrar el punto $P_m^{i+1} = C_{i+1}(\bar{u}_m)$ cuya distancia es mínima. Para $i = 0, \ldots, k-1$ y $l, m = 0, \ldots, n$, es decir

```
for(l=0; l<=n; l++)
  for(m=0;m<=n; m++)
   P_l^{i+1}=distancia_minima{P_m^{i+1}, P_l^i}
```

3. Realizar la interpolación de los nuevos puntos P^i y obtener las nuevas curvas de interpolación B-spline $C_i(u)$, para $i = 0, \ldots, k$.

4. Construir la superficie con las curvas $C_i(u)$, para $i = 0, \ldots, k$ utilizando (1).

3. Implementación.

Reconst-0.0 contiene rutinas en C++ con las cuales es posible reconstruir la geometría de yacimientos simples, a partir de un conjunto de curvas de nivel usando una superficie B-spline, y la biblioteca B-spline++ desarrollada por Philippe Lavoie[1].

La rutina principal `Reconst.cpp` llama a un conjunto de funciones (ver archivo `funciones.h`), implementando el algoritmo de reconstrucción que se describe en la siguiente sección. Así, para su funcionamiento es necesario pasarle un archivo en el cual estén los puntos que componen a las curvas de nivel[2] del yacimiento que se desea reconstruir, una vez hecho esto el programa preguntará por el número de renglones y columnas de puntos que se desea tener en la superficie reconstruida, cuando estos son proporcionados el programa generará un archivo llamado `puntos.txt` con las coordenadas de estos, adicionalmente se creará un archivo en formato `VRML` con el que se puede visualizar la superficie reconstruida.

4. Modo de uso.

Para poder usar el programa `Reconst` es necesario tener instaladas las bibliotecas Nurbs++, OpenGL y Glut (o su versión libre Mesa) las cuales se pueden obtener libremente bajo la licencia GNU en la páginas de internet:

- `http://libnurbs.sourceforge.net/download.shtml`
- `http://www.mesa3d.org/`.

Así una vez instaladas todas éstas, el funcionamiento de **Reconst-0.0** requiere un archivo de datos en donde el primer valor debe indicar el número de curvas de nivel y a continuación debe estar los puntos de las curvas de nivel en coordenadas $\mathbb{R}^3$, las cuales deberán separarse con

```
1.0000000E+30 1.0000000E+30 1.0000000E+30
```

por ejemplo

```
5
3916.439 4688.086 3025.000
4012.789 4559.529 3025.000
4446.365 4286.343 3025.000
4623.007 4382.761 3025.000
4863.882 4430.970 3025.000
5313.515 4623.808 3025.000
5458.041 4896.993 3025.000
5329.574 5395.155 3025.000
4976.290 5636.201 3025.000
4767.532 5620.131 3025.000
4430.306 5491.574 3025.000
4237.606 5427.294 3025.000
```

[1] Ver en http://libnurbs.sourceforge.net/

[2] El formato de este archivo se describe en la sección 3

```
3964.614 5411.225 3025.000
3707.681 5218.388 3025.000
3755.856 4768.435 3025.000
3964.614 4672.017 3025.000
3916.439 4688.086 3025.000
1.0000000E+30 1.0000000E+30 1.0000000E+30
3787.973 6118.294 3235.000
3595.272 5957.596 3235.000
...
...
...
```

se recomienda que los archivos de datos se guarden en la carpeta `datos`, así para correr en un sistema Unix, se debe teclear

```
[Reconst-0.0]$ ./Reconst datos/nombrearchivo.txt
```

una vez hecho esto, el programa preguntará el número de puntos que se requieren para los renglones y columnas de la malla sobre la superficie reconstruida (equivalente al número de isocurvas para para los dos valores parametrales u, v [3]), una vez que el usuario los proporcione se creará un archivo `puntos.txt` en donde estarán los puntos de la superficie evaluada.

Una corrida típica de **Reconst-0.0** sería

```
[Reconst-0.0]$ ./Reconst datos/cima25.txt
Reconst: Proporciona el número de renglones y
columnas de la malla sobre superficie
40 60
Reconst: El archivo de salida es: "puntos.txt"
```

Adicionalmente **Reconst-0.0** contiene su Makefile (para ambientes UNIX), el cual se ejecuta con la instrucción:

```
[Reconst-0.0]$ make
```

Como ejemplo del funcionamiento de **Reconst-0.0**, se presentan en la figura 1 las curvas de nivel con las que se desea reconstruir las superficies contenidas en el archivo `cimas25.txt`. En las figuras 2 la malla 2D generada por **Reconst-0.0** (archivo **puntos.txt**) tomando 20×20 renglones y columnas, en la figura 3 tomando 40×40 y finalmente en la figura 4 la superficie reconstruida (archivo **superficie.wrl**).

[3]Esto es por la definición de la superficie B-spline.

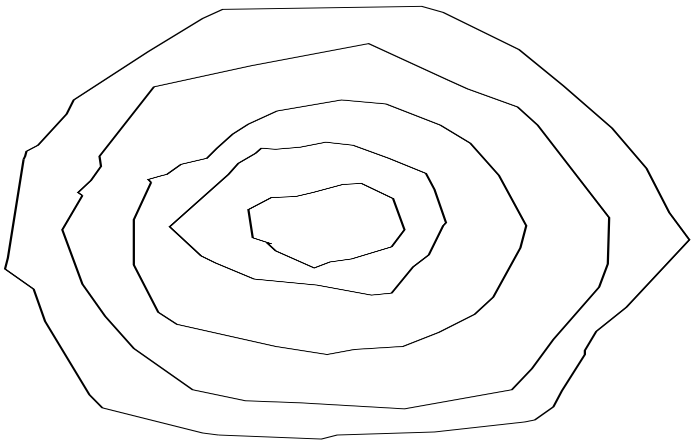

Figura 1: Curvas de nivel.

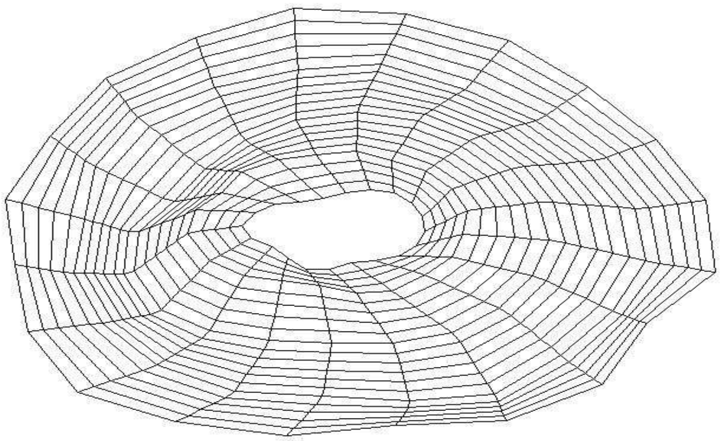

Figura 2: Malla 2D tomando 20×20

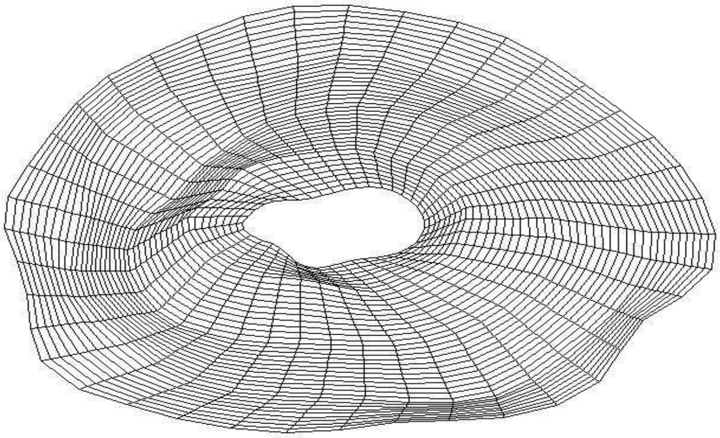

Figura 3: Malla 3D tomando 30×30.

Figura 4: Superficie resultante.

Referencias

[Blom91] *Bloomenthal M. y Riesenfeld R.F.*, **Approximation of Sweep Surfaces by Tensor Product NURBS**, Curves and Surfaces in Computer Vision and Graphics II, Society of Photo-Optical Instrumentation Enginneers, Vol. 1610 páginas 132-144, 1991.

[Coqu87] *Coquillart S.*,**A Control Point Based Sweeping Technique**, IEEE Comput. Graph and Appl., Vol 7, No.11, páginas 36-45, 1987.

[Cox82] **Practical Spline Aproximation**, *M.G. Cox.*, NPL Report Division of Information Technology and Computing, 1982.

[DeBoor78] **A Practical Guide to Splines**, *De Boor, C.*, New York: Springer-Verlag, 1978.

[Far90] **Curves and Surfaces for CAGD, a practical guide**, *Gerald Farin*, Academic Press, Inc. Third Edition, 1990.

[PT97] **The NURBS Book** *Les Piegl, Wayne Tiller*, Springer-Verlang 1997.

[Rog01] *David F. Rogers*, **An Introducction to NURBS With Historical Perspective**,,Morgan Kaufmann Pubishers 2001.

[tesis] *Cervantes Cabrera Daniel*, **Estudio de la Construcción de Superficie con NURBS, Tesis de Maestría**, Posgrado en Computación UNAM, 2005.