

Aumento y Discriminación de Datos en Modelos Digitales de Elevación basado en Parches Paramétricos Bicúbicos

R. Quintero, S. Levachkine, M. Moreno, M. Torres
Centro de Investigación en Computación
Instituto Politécnico Nacional
07738 México, Distrito Federal
Tel. 5729-6000 ext. 56577 e-mail quintero@cic.ipn.mx

RESUMEN

Los Sistemas de Información Geográfica modernos requieren herramientas para simulaciones tridimensionales de Modelos Digitales de Elevación. Se han desarrollado algunos algoritmos para satisfacer esta tarea, pero el problema principal es el que se refiere al uso de los recursos computacionales, ya que los algoritmos desarrollados no tienen un buen rendimiento. En este trabajo se presenta un algoritmo para la visualización de los DEM en tiempo real al reducir la cantidad de datos a procesar. Al mismo tiempo se emplea un algoritmo para cambiar la escala de los DEM para mejorar la visualización. La aplicación de los algoritmos en forma conjunta permite navegar a través de una gran cantidad de información dentro de los datos de elevación y al mismo tiempo incrementar el nivel de detalle en zonas de interés.

Palabras clave:

Modelos Digitales de Elevación, Nivel de Detalle, Parches Paramétricos, Visualización.

I. INTRODUCCIÓN

En la actualidad los Modelos Digitales de Elevación (DEM – Digital Elevation Model) han ganado terreno en aplicaciones para la simulación de desastres naturales. Las herramientas de toma de decisión para la prevención de desastres han cobrado gran importancia a nivel mundial ya que están en juego vidas humanas. Sin embargo, para desarrollar este tipo de aplicaciones se tienen principalmente dos problemas, uno de estos problemas es que se requiere manejar un enorme volumen de datos. El segundo problema es que los datos con los que se cuentan no tienen la calidad requerida para los procesos de simulación. Por ejemplo, el Instituto Nacional de Estadística Geografía e Informática (INEGI) produce DEM con una resolución de 50 metros y solo en algunos casos hasta 20 metros [1][2].

En las secciones II y III se presentan los dos algoritmos que se proponen para solucionar los

problemas mencionados en el párrafo anterior. En la sección II se presenta un algoritmo para la discriminación de datos y en la sección III se presenta un algoritmo para aumentar el nivel de detalle de los datos. En la sección IV se presentan algunos aspectos de la implementación de dichos algoritmos, así como algunos resultados obtenidos. Finalmente en la sección V se presentan las conclusiones sobre este trabajo.

II. ALGORITMO DE DISCRIMINACIÓN DE DATOS

En [3] y [4] se presenta un procedimiento para representar los DEM como mallas de polígonos. En dicho artículo se describe la malla resultante como una matriz de G de $M \times N$ puntos o vértices tridimensionales (3D).

En [5] y [6] se presentan algunos algoritmos para disminuir el Nivel de Detalle (LOD – Level of Detail) en escenas complejas. Esos algoritmos presentan principalmente tres desventajas:

- Son muy complejos e incrementan la carga de trabajo al procesador. Afectan los datos de elevación debido a que están enfocados a la apariencia final de la escena.
- Modifican los datos de elevación dependiendo del punto de vista del observador. Debido a esto no es posible el análisis espacial.
- El número de elementos a visualizar es variable, por lo que la velocidad en la visualización no es constante durante la simulación.

El algoritmo de discriminación ataca estos problemas, por lo que sus metas son tres: 1) ser simple, 2) no afectar los datos de elevación y 3) funcionar en tiempo real, es decir garantizar una velocidad de visualización mínima. Para describir el algoritmo definiremos algunos parámetros necesarios.

- Una matriz L de $H \times H$ que define el LOD discreto a utilizar. Donde H es un número impar mayor que 1, y todas las entradas de la matriz son diferentes de 0.
- Un número S que define el tamaño de la unidad de optimización. Este parámetro es el número de polígonos por región.
- Un vector o que representa la posición del observador.

En la Fig. 1 se muestra una representación visual del significado de estos parámetros.

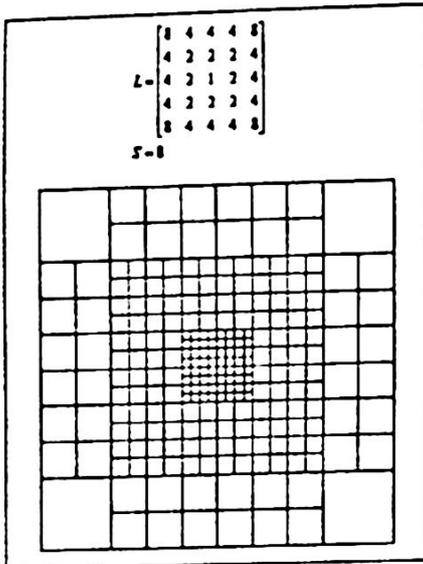


Fig. 1. Representación visual de los parámetros del algoritmo de discriminación.

Una vez establecidos los parámetros necesarios, es posible describir el algoritmo de discriminación. Recordando que los datos de elevación se encuentran en la matriz G .

RENDER (o)

- 1 $(o_x, o_y) \leftarrow \text{RELATIVE-POSITION}(o, G)$
- 2 for $i = -\frac{1}{2}H$ to $\frac{1}{2}H$
- 3 $x \leftarrow o_x + S(i - \frac{1}{2})$
- 4 for $j = -\frac{1}{2}H$ to $\frac{1}{2}H$
- 5 $y \leftarrow o_y + S(j - \frac{1}{2})$
- 6 render-block($x, y, L[i+\frac{1}{2}H, j+\frac{1}{2}H]$)

RENDER-BLOCK(x, y, lod)

- 1 if $lod > 0$
- 2 for $i = x$ to $x+S$ step lod
- 3 for $j = y$ to $y+S$ step lod
- 4 RENDER-QUAD(i, j, lod)

RENDER-QUAD(i, j, lod)

- 1 RENDER-VERTEX($G[i, j]$)
- 2 RENDER-VERTEX($G[i+lod, j]$)
- 3 RENDER-VERTEX($G[i+lod, j+lod]$)
- 4 RENDER-VERTEX($G[i, j+lod]$)

Mediante el uso de este algoritmo, determinar el número de polígonos (N_p) a para obtener un cuadro de animación, es velocidad de animación. El valor de N_p se mediante la Ec. (1).

$$N_p = \sum_{i=1}^H \sum_{j=1}^H \frac{S}{L[i, j]}$$

III. INCREMENTO DEL NIVEL DETALLE

Para el caso de las zonas de G donde se obtener mayor detalle, proponemos un algor que nos permite calcular valores intermedios zonas donde no se cuenta con información. calcular dichos valores se utilizan paramétricos bicúbicos.

Un parche paramétrico es un polígono curvo coincide con una superficie. La definición [6] de una superficie paramétrica (ya sea superf de Bezier o B-spline) $Q(u, v)$, se da en términos los parámetros u y v , donde $0 \leq u, v \leq 1$, y la funci es un polinomio cúbico. Los valores precisos coeficientes en dicho polinomio determinan Un modo conveniente para definir estos coeficie es por medio de un arreglo de 16 tridimensionales, conocidos como puntos de La forma del parche está completam determinada por la posición de esos puntos.

Una superficie bicúbica está definida por

$$Q(u, v) = \sum_{i=1}^n \sum_{j=1}^m P_{ij} B_{ij}(u, v),$$

donde P_{ij} es un arreglo de puntos de $B_{ij}(u, v)$ es la función base bivariable. Pode generar $B_{ij}(u, v)$ de la siguiente manera

$$B_{ij}(u, v) = B_i(u) B_j(v),$$

donde $B_i(u)$ y $B_j(v)$ son las funciones base de una sola variable. La definición de funciones base describen el tipo de superficie generada. A continuación damos la definición funciones base para las superficies aplicables trabajo: superficies de Bezier y B-spline.

A. Superficies de Bezier

Las propiedades de las superficies de Bezier extienden de la formulación de las curvas de em [7][8] se pueden encontrar dichas propieda Como se mencionó, una superficie bicúbica

define por su función base. En el caso de las superficies de Bezier, dicha función base está definida por los polinomios de Bernstein (para un conjunto de $n+1$ puntos de control).

$$B_k^n(u) = C(n,k)u^k(1-u)^{n-k}, \quad (4)$$

donde $C(n,k)$ son los coeficientes binomiales

$$C(n,k) = \frac{n!}{k!(n-k)!}, \quad (5)$$

por lo que la descripción de la superficie de Bezier está dada por

$$Q(u,v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} B_i^n(u) B_j^m(v), \quad (6)$$

Superficies B-spline

De manera análoga a las superficies de Bezier, las superficies B-spline están caracterizadas por su función base. En este tipo de superficies, el valor del parche depende exclusivamente de los valores de un arreglo de 4×4 tomado del conjunto de puntos de control. Así para un conjunto de $n \times m$ puntos de control tendremos $(n-3) \times (m-3)$ parches. Considerando esto, la función base para superficies B-spline está definida como

$$Q_{ij}(u,v) = \sum_{k=0}^3 \sum_{l=0}^3 P_{i+k,j+l} B_k(u) B_l(v), \quad (7)$$

para el parche Q_{ij} , con $i=0,1, \dots, n-3$ y $j=0,1, \dots, m-3$, donde

$$\begin{aligned} B_0(u) &= \frac{(1-u)^3}{6}, \\ B_1(u) &= \frac{3u^3 - 6u^2 + 4}{6}, \\ B_2(u) &= \frac{-3u^3 + 3u^2 + 3u + 1}{6}, \\ B_3(u) &= \frac{u^3}{6}, \end{aligned} \quad (8)$$

C. Aplicación de las superficies paramétricas

Para aplicar las superficies paramétricas en el incremento del LOD de los datos de elevación, realizaremos algunas modificaciones al algoritmo descrito en la sección II. En primer lugar, debemos permitir valores menores a 1 pero mayores a 0 en la matriz L . Dichos valores nos ayudarán a indicar que

en lugar de discriminar datos, deseamos obtener una mayor cantidad de información para el bloque que se está dibujando.

La segunda modificación se refiere a la necesidad de calcular la curva paramétrica. Esta curva será almacenada en una matriz alterna a la que denominaremos Q . Los valores de Q serán definidos por los puntos de control y por la matriz de transformación B que se presenta en la Ec. (9) (ver [CIARP1] para la representación matricial de las funciones base de Bezier y B-spline). En este documento sólo se presenta el procedimiento para aplicar B-splines. Para aplicar parches Bezier se deberán tomar similares consideraciones.

$$B = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \quad (9)$$

Las modificaciones al algoritmo son las siguientes:

```

RENDER-BLOCK(x, y, lod)
1  if lod ≥ 1
2    for i = x to x+S step lod
3      for j = y to y+S step lod
4        RENDER-QUAD(i,j,lod)
5  else
6    Q ← TO-SPLINE(G,x,y,lod)
7    for i = x to x+S step lod
8      for j = y to y+S step lod
9        RENDER-QUAD-SPLINE(i,j,Q)
    
```

```

TO-SPLINE(G,x,y,lod)
1  P ← CONTROL-POINTS(x,y)
2  for u=0 to 1 step lod
3    U ← [u³ u² u 1]
4    for v=0 to 1 step lod
5      V ← [v³ v² v 1]
6      Q[u,v] ← U × B × P × Bᵀ × V
    
```

```

RENDER-QUAD-SPLINE(i,j,Q)
1  RENDER-VERTEX(Q[i,j])
2  RENDER-VERTEX(Q[i+1,j])
3  RENDER-VERTEX(Q[i+1,j+1])
4  RENDER-VERTEX(Q[i,j+1])
    
```

IV. ANÁLISIS ESPACIAL 3D

Se ha desarrollado una técnica de visualización que nos proporciona suficiente rendimiento para realizar este proceso en tiempo real. En esta metodología se propone un conjunto de pasos para realizar análisis espacial 3D. Dichos pasos son los siguientes:

1. Definir las metas del análisis

2. Establecer la escala de las coordenadas de altitud (elevación)
3. Definir las características de las capas de vectores espaciales
4. Sobreponer las capas vectoriales
5. Realizar una inspección visual
6. Identificar distancias Euclidianas y mediciones de inclinaciones de las cuestas
7. Definir trayectorias automáticas (vuelos virtuales)

Los pasos 2, 6 y 7 son opcionales, debido a que dependen de análisis particulares según el caso de estudio.

V. PRUEBAS Y RESULTADOS

Se han realizado algunas pruebas de rendimiento con diferentes conjuntos de datos. Como se ha mencionado, el rendimiento del algoritmo es constante sin importar el volumen de datos que se maneje. Los resultados de la aplicación se muestran en la Fig. 2. En la Fig. 2a se muestra el resultado de procesar todos los datos de elevación. En la Fig. 2b se presenta el resultado utilizando el algoritmo utilizando los siguientes parámetros:

$$L = \begin{bmatrix} 64 & 32 & 16 & 8 & 16 & 32 & 64 \\ 32 & 16 & 8 & 4 & 8 & 16 & 32 \\ 16 & 8 & 4 & 2 & 4 & 8 & 16 \\ 8 & 4 & 2 & \frac{1}{2} & 2 & 4 & 8 \\ 16 & 8 & 4 & 2 & 4 & 8 & 16 \\ 32 & 16 & 8 & 4 & 8 & 16 & 32 \\ 64 & 32 & 16 & 8 & 16 & 32 & 64 \end{bmatrix} \quad (10)$$

$$S = 128 \\ \Rightarrow N_p = 1096$$

Como resultado emergente de la capacidad del algoritmo para obtener datos desconocidos entre los valores conocidos por interpolación bicúbica, es posible aplicar esta misma técnica para realizar acercamientos (*zoom-in*) en imágenes. Debido a esto, se han tomado otros conjuntos de datos para probar esta capacidad para realizar acercamientos en imágenes.

Los resultados de esto se muestran en la Fig. 3. La Fig. 3a muestra la imagen fuente que se utilizó para este ejemplo. En las Fig. 3b y 3c se presentan los resultados de la utilización del algoritmo con parches B-spline y Bezier respectivamente.

Como se puede observar, los parches B-spline nos permiten interpolar los valores internos de la imagen dependiendo de sus vecinos (ya que la función base de B-spline depende solamente de 16 puntos de control), así que el resultado contiene información acerca de los cambios de contraste (altitud) locales.

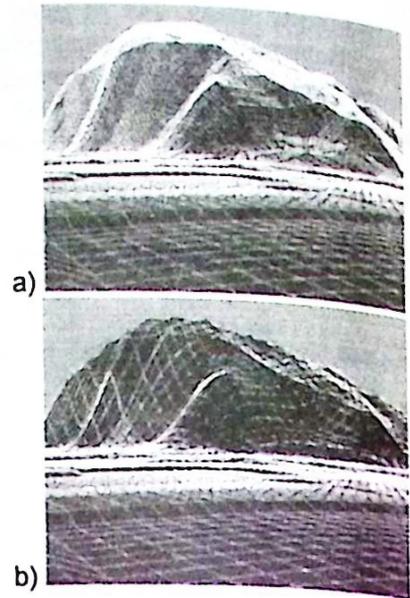


Fig. 2. a) Resultado de procesar todos los datos. b) Resultado con el algoritmo propuesto

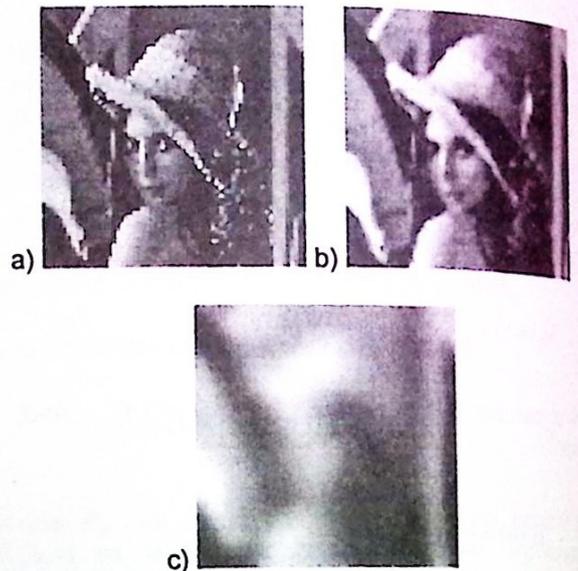


Fig. 3. Ejemplo de la aplicación del algoritmo para el acercamiento de imágenes. a) Imagen con un zoom 2x convencional. b) Imagen con zoom 2x con parches B-spline. c) Imagen con zoom 2x con parches Bezier.

Con la aplicación de parches Bezier se obtiene información sobre todo el conjunto de datos en cada uno de los valores calculados; esto nos permite aplicar a la imagen un tipo de suavizado que no depende solo de valores locales sino de toda la imagen. En la Fig. 4 se muestran los resultados de mezclar las propiedades mencionadas de los tipos de curvas utilizados.

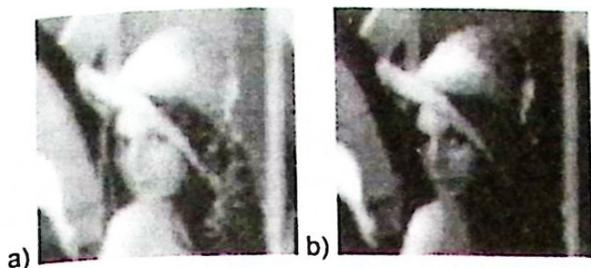


Fig 5. Mezcla de las características de los dos tipos de curvas. a) Promedio de las imágenes B-spline y Bezier. b) Multiplicación de las imágenes B-spline y Bezier.

VI. CONCLUSIONES

En este trabajo se ha presentado una metodología para realizar análisis espacial 3D con ayuda de técnicas de visualización de DEM en tiempo real. Se ha presentado un par de algoritmos para aumentar y para disminuir la cantidad de datos involucrados en el proceso de visualización. Los algoritmos presentados requieren un tiempo bajo para procesar los datos de elevación, que están almacenados en DEM. El algoritmo en su conjunto cumple con las metas establecidas de sencillez, de no afectar los datos (condición necesaria para realizar el análisis espacial) y de funcionar en tiempo real (garantizando una velocidad constante de visualización).

Adicionalmente, se puede utilizar el algoritmo, como una aplicación emergente, en procesamiento de imágenes: este puede ser utilizado para realizar acercamientos (*zoom-in*) en imágenes con un remuestreo no lineal. Utilizando parches paramétricos es posible obtener los nuevos valores para los píxeles que se encuentran entre los píxeles conocidos, conteniendo información local (por medio de parches B-spline) y comportamiento global (Por medio de Bezier), lo cual mejora la apariencia de la imagen resultante.

AGRADECIMIENTOS

Los autores de este artículo agradecen al Centro de Investigación en Computación (CIC), a la Coordinación General de Posgrado e Investigación (CGPI) y al Instituto Politécnico por las facilidades prestadas. Adicionalmente agradecen a los revisores por sus pertinentes comentarios.

REFERENCIAS

[1] "Modelos Digitales de Elevación, Generalidades y Especificaciones", Instituto Nacional de Estadística, Geografía e Informática de México. Aguascalientes, Mexico, 1999.

[2] "Normas Técnicas para la Elaboración de Ortofotos Digitales", Instituto Nacional de Estadística, Geografía e Informática de México. Aguascalientes, Mexico, 1999.

[3] Quintero, R., *et al*, "Spatial Analysis Based on 3D Rendering Techniques", Proceeding of 6th Association of Geographical Information Laboratories Europe conference on Geographical Information Science - AGILE 2003. Lyon, France, 2003.

[4] Quintero, R., *et al*, "Real-Time Elevation Map Viewer for 3D Spatial Analysis", Proceeding of International Workshop on Semantic Processing of Spatial Data, Mexico City, Mexico, 2002.

[5] M. Duchaineu, "Roaming Terrain: Real-time Optimal Adapting Meshes", Proceedings of the SIGGRAPH' 99, Denver USA, 1999, 56-67.

[6] P. Lindstrom, "Real-time, Continuous Level of Detail Rendering of Height Fields", Proceedings of the SIGGRAPH 96, Lyon, France, 1996, 89-101.

[7] Watt, A., "3D Computer Graphics", 2nd edn. Addison-Wesley New York, Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York, 1995.

[8] Heam, D., Baker, P. "Computer Graphics", 2nd edn. Prentice Hall, New York, 1994.