Integración de Modelos para Aplicaciones Geográficas

Ricardo Coppo, Leonardo Arlenghi y Claudio Delrieux

Departamento de Ing. Eléctrica y de Computadoras - Universidad Nacional del Sur - Alem 1253 (8000) Bahía Blanca - Argentina - claudio@acm.org

Resumen

Existe actualmente número muy grande de aplicaciones informáticas destinadas al procesamiento de información geográfica. Éstas se caracterizan por emplear bases de datos propietarias y no estandarizadas. Si bien son funcionalmente cada vez más completas (y complejas), es difícil concebir una aplicación que satisfaga las infinitas necesidades y preferencias de los analistas, modeladores y simuladores de información y procesos geográficos y ambientales, y menos aún que facilite el reuso o la documentación de las diferentes secuencias de procesamiento aplicadas en una tarea en particular. Por su parte, una tendencia generalmente aceptada es la de producir estándares y protocolos internacionales que apuntan a la integración de los sistemas, como lo es actualmente el OpenGIS en este campo de aplicación. En este trabajo se propone la creación de un banco de trabajo visual que brinde al usuario final la posibilidad de implementar modelos en los que se integran componentes interconectables en una red para la conformación de un sistema flexible y predecible. Estos componentes podrían ser implementados por diversos fabricantes y/o proveedores de software o encontrarse como servicios distribuidos físicamente sobre una red informática.El programa de modelación visual actúa como controlador de todo el sistema permitiendo al analista desarrollar, con una única herramienta, el modelo de su interés. Se presenta como ejemplo el modelo conceptual y la implementación de una aplicación que permite el renderizado de un DEM como superficie 3D, obtenido a partir de un archivo de curvas de nivel.

Palabras Clave: Visualización Científica - GIS - Geometría Computacional

I. INTRODUCCIÓN

Los modelos matemáticos se emplean actualmente para lograr un mayor conocimiento del entorno geográfico y ambiental en la que se mueve el hombre al desarrollar su actividad económica. Existe un sinnúmero de aplicaciones informáticas que permiten procesar la información de campo recolectada y que afectan la toma de decisiones en cuánto al uso de los recursos naturales, la planificación de su uso y su aprovechamiento sustentable.

Estas aplicaciones se caracterizan por emplear bases de datos propietarias y no estandarizadas, aunque debe reconocerse que por uso y popularidad, algunos de ellos se han convertido en estándares "de facto" o industriales. Si bien los programas son funcionalmente cada vez más completos (y complejos), es difícil concebir una aplicación que satisfaga las infinitas necesidades y preferencias de los analistas, modeladores y simuladores de procesos ambientales.

En el trabajo diario del analista lo usual es emplear "cadenas" de procesamiento en la que los resultados obtenidos en un programa son empleados en otro, continuando así hasta llegar a una

aplicación en la que se produce un resultado aceptable. Si el usuario requiere de más de un resultado, o de diferentes perspectivas de un mismo conjunto de datos, las cadenas anteriores se convierten en verdaderos "redes" de interacción entre aplicaciones y archivos intermedios de información. Al realizar un estudio particular, pocas veces se documenta con el nivel de detalle suficiente la cadena de procesamiento empleada para lograr un resultado, haciéndose prácticamente imposible repetir una secuencia de operaciones si los datos de campo se incrementan con nuevas muestras, o si los mismos se ven modificadas por un mayor conocimiento del fenómeno en estudio.

Por otro lado, la visualización científica, la teoría de grafos, y la computación en entornos distribuidos han ido avanzando hacia sistemas de abstracción de segundo orden (abstracciones que permiten la construcción de aplicaciones), conceptos que aún no son de uso masivo en el procesamiento de la información de modelos de información geográfica. El ambiente informático también se caracteriza por diferentes esfuerzos realizados con la intención de producir estándares y protocolos internacionales que apuntan a la integración de los sistemas. Se ha avanzado con especificaciones que permiten compartir recursos computacionales, realizar procesamiento remoto, e inclusive se dispone de un modelo conceptual aplicable al desarrollo de sistemas de procesamiento geográfico. Por ejemplo podemos reseñar las normativas internacionales publicadas por la ISO y el consorcio OpenGIS, que ha trabajado por más de 10 años en la especificación de estándares orientados a la integración de sistemas a través de un modelo conceptual.

En este trabajo se propone la creación de un banco de trabajo visual que brinde al usuario final la posibilidad de implementar modelos en los que se integran componentes interconectables en una red para la conformación de un sistema flexible y predecible. Estos componentes podrían ser implementados por diversos fabricantes y/o proveedores de software o encontrarse como servicios distribuidos físicamente sobre una red informática.El programa de modelación visual actúa como controlador de todo el sistema permitiendo al analista desarrollar, con una única herramienta, el modelo de su interés ya sea con componentes propios del modelador o con aplicaciones desarrollados por terceros. Se presenta un ejemplo de aplicación de un modelo que permite el renderizado de una superficie a partir de un archivo de curvas de nivel obtenidos con una mesa digitalizadora.

II. PARADIGMAS DE MODELACIÓN

Tradicionalmente se han empleado cuatro paradigmas para la modelación y procesamiento de información: el lingüístico, la planilla electrónica, el modelo jerárquico, y la de red de interconexión.

- El paradigma lingüístico incluye todos los desarrollos de genericidad de primer orden, entre las cuales se pueden mencionar los lenguajes de modelación, los grafos de dependencia, y los conjuntos de aplicaciones en tandem. En este paradigma la interacción entre los componentes se desarrolla bajo completo control del usuario, frecuentemente mediante un uso de un lenguaje de scripting como el Tcl/Tk, CGI o JavaScript. Son características las estructuras sintácticas que "conectan" la salida de un programa con la entrada de otro a través de pipes. En algunos entornos puede existir un lenguaje específico que ha dominado el paradigma como es el caso del SQL en el tratamiento de información almacenado en una base de datos relacional. La biblioteca VTK (Visualization Toolkit) también emplea este paradigma para la

definición de un pipeline gráfico de visualización por medio de la vinculación dinámica de los métodos correspondientes de cada una de las clases.

- Una mejora sustancial introduce el paradigma de las planillas de cálculo en las que, a costa de cierta rigidez en la presentación de la información (que debe poder representarse como una tabla plana) el modelo puede construirse en base a ecuaciones que relacionan "celdas" entre sí. La pérdida de flexibilidad se ve compensada por las facilidades obtenidas en la ejecución reiterada del modelo, en su clara expresión formal, su sencilla descripción funcional y a su ductilidad al cambio.

- Gracias al desarrollo de los entornos visuales, el paradigma de modelación basado en objetos ha cobrado importancia. Estas aplicaciones permiten construir una jerarquía de objetos compuestos como el resultado de la asociación de componentes que se "colocan" sobre formularios o paneles. Los objetos que forman parte del panel, en principio, no interactúan entre sí. Notorios ejemplos de este paradigma son los lenguajes "visuales" de programación, los lenguajes basados en componentes, y algunos programas de edición de imágenes y de diseño gráfico.

- El paradigma de la topología en red ha sido utilizado por diversos ramas de la ingeniería por muchos años. Los tradicionales diagramas en bloque, los diagramas de flujo de datos y diversos diagramas de proceso no solo incorporan información estructural de los componentes del modelo sino también indican su relación funcional. La teoría de grafos, ampliamente estudiada en matemática sirve de modelo conceptual sobre la cual se pueden construir diagramas de tendidos eléctricos, redes de distribución de agua, redes camineras etc. Ejemplos de la aplicación de este tipo de modelo son el SPICE y el Simulink que permiten analizar el comportamiento de una red a partir de un modelo de componentes definido por sus nodos y arcos.

Este último paradigma, junto con la de la orientación a objetos, son las elegidas para el desarrollo de un conjunto de componentes visuales geográficos que pueden ser combinados en una red para el desarrollo de modelos.

III. COMPONENTES DE UN SISTEMA INTEGRADO DE GIS

Una primera clasificación de los componentes que debería contener un sistema integrado de GIS funcionalmente completo y consistente, se basa en el tipo de función que las mismas cumplen.

- **Productores**: Producen información primaria. También son conocidos como *fuentes* o puntos de origen en los modelos. Existen diversos tipos de productor, pueden ser subdivididos a su vez por la naturaleza puntual, geométrico, o de cobertura de la información que entregan. Son ejemplos de productor: archivos de puntos en la forma de texto o de planilla electrónica, DEMS (Digital Elevation Maps), curvas de nivel digitalizadas, simbología cartográfica, imágenes satelitales, y fotografías aéreas en sus varios formatos.

- **Operadores**: También son conocidos como *transformadores*. Reciben información de una o más entradas y producen información para una o más salidas. Podemos citar los procesos de georeferenciaciación, triangulación, generalización, simplificación, procesamiento digital de

imágenes, por mencionar unos pocos, como ejemplos de transformadores de información de un formato a otro. En particular, la interpolación espacial a partir de curvas de nivel, por su importancia, y por su relación con la Visualización Científica, será desarrollado en el ejemplo.

- **Finalizadores**: Tambien conocidos como terminadores de modelo. Reciben la información y la presentan en un dispositivo gráfico o la almacenan en un archivo en un formato predeterminado. Constituyen el objetivo del estudio, ya que pueden considerarse los terminadores del proceso de modelado. Son ejemplos: los renderers, los archivos de imágen, los mapas, etc.

Una segunda clasificación posible de las componentes de un sistema integrado puede hacerse teniendo en cuenta el tipo de problema que resuelve cada uno, agrupándolos en categorías de acuerdo a la naturaleza o los aspectos de modelado comunes entre los mismos. Este enfoque es la realizado por los estándares del Consorcio OpenGIS que actualmente son usados para la definición de estándares producidos por la ISO y del ANSI. (En Argentina también están en consideración de estudio por parte del IRAM). Esta clasificación identifica las siguientes categorías de componente o tareas:

- Geometría de los objetos (*features*) (puntos, líneas, polígonos, etc.).
- Adecuación de los sistemas de referencia espaciales.
- Estandarización de las estructuras de información geométrica.
- Construcción de funciones interoperables de interpolación.
- Funciones de coberturas.
- Relaciones e interfases entre los componentes de la geometría (*feature relationship*).
- Colecciones de features.
- Administración de los metadatos.
- Arquitectura de servicios GIS (catálogos, imágenes, transformación de coordenadas).

En este trabajo se ha adoptado la primera clasificación aunque el segundo es de gran importancia por el esfuerzo internacional que se ha realizado en los últimos 10 años en esta dirección.

IV. ESTRUCTURA DE LA INFORMACIÓN

La compatibilidad entre los formatos de la información ha sido uno de los mas graves problemas en el desarrollo de las aplicaciones geográficas. Las productores entregan archivos en formatos muy diversos tales como texto plano, planillas de cálculo y en un sin número de formatos de representación de imágenes. Si bien es posible concebir componentes operadores que reciben como entrada los diversos formatos, es preferible considerar un único formato interno para la interconexión de componentes. Esto permite desarrollar componentes que se limitan a un único formato y no generar una gran cantidad de componentes diferentes pero funcionalmente iguales.

El problema se repite al considerar los resultados de cada componente. Es importante que dicho formato sea flexible y a su vez genérico, permitiéndose el desarrollo o uso de aplicaciones de diferentes proveedores. No solo las características de las entradas y las salidas sugieren la necesidad de un formato interno de la información. También debe considerarse la disparidad de

plataformas y equipos informáticos disponibles actualmente que requieren de formatos de información que permitan compartirlo entre las mismas para lograr los objetivos de interoperabilidad propuestas.

La selección de este formato interno es de fundamental importancia, ya que es muy tentadora la idea de definir un esquema propio con lo cuál rápidamente se repetirían los problemas de las estructuras propietarias mencionadas al principio. En este trabajo se propone el uso del lenguaje de marcado extensible XML, ampliamente documentado y para la cuál existen varios analizadores y traductores disponibles. La selección de un lenguaje como XML como lenguaje intermedio permite además el uso de herramientas de desarrollo independientes como por ejemplo editores, depuradores, etc. Esta selección permitirá incorporar como componente al visualizador cualquier aplicación (comercial o no) que pueda leer y escribir directamente documentos XML.

El Consorcio OpenGis ha realizado la definición de un lenguaje de marcado para fines geográficos basado en XML denominado GML (Geography Markup Language). El mismo se encuentra respaldado por las principales empresas productores de software GIS y por destacadas universidades de todo el mundo quienes se han comprometido en el desarrollo de interfases para sus aplicaciones en este lenguaje. A partir de esa definición los componentes del modelador visual pueden ser vistos como proveedores que convierten su información al formato GML, operadores que realizan transformaciones de la misma pero manteniendo el mismo formato, y como consumidores o terminadores que entregan la información procesada a los formatos requeridos por el usuario.

V. UN EJEMPLO PARTICULAR DE IMPLEMENTACIÓN

Presentamos como ejemplo de las ideas mencionadas más arriba la implementación de una componente de interpolación espacial, es decir, un componente transformadora que recibe datos en forma de curvas de nivel, y genera un conjunto de polígonos en el espacio que interpola dichos datos.

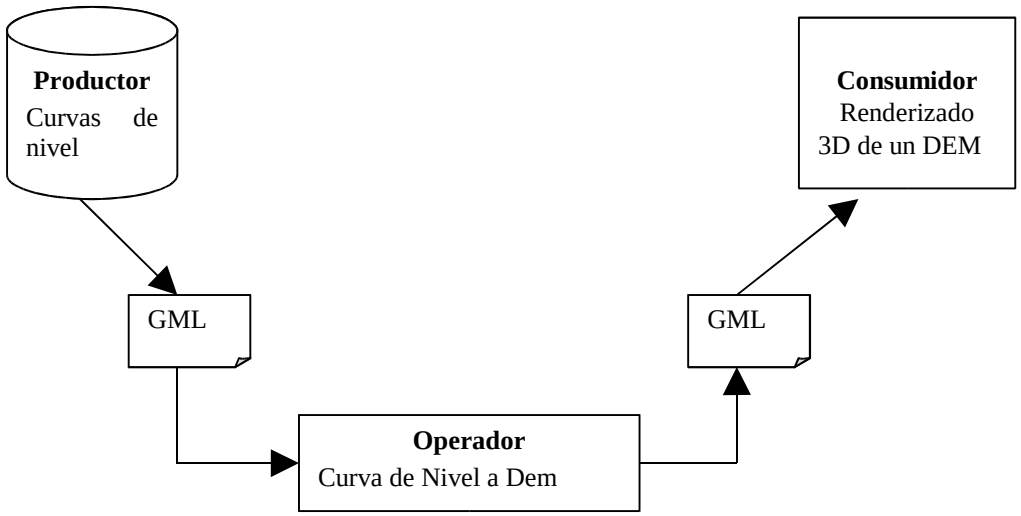


Figura 1: Ejemplo de Aplicación del modelo conceptual

El componente productor:

Este componente representa el inicio del modelo visual. En este caso es un archivo con las especificaciones de las curvas de nivel. Por ejemplo, en la Figura 2 vemos un conjunto de isolíneas -generadas con un editor para testear el comportamiento del componente operador, con los cuales se obtuvo la Figura 3 realizando además del DEM, un mapeo de texturas. Estos datos del componente productor pueden asimismo ser importados de archivos de formato estandarizado. Por ejemplo, podemos ver en la Figura 4 las isolíneas procedentes de una mesa digitalizadora a partir de datos del mar Caspio. El formato original de los datos se encuentra como un imagen formado por vectores con metadatos que identifican las alturas correspondiente. En la Figura 5 se renderizan los datos del DEM resultante utilizando además el mapeo de alturas a colores por medio de la paleta rainbow.

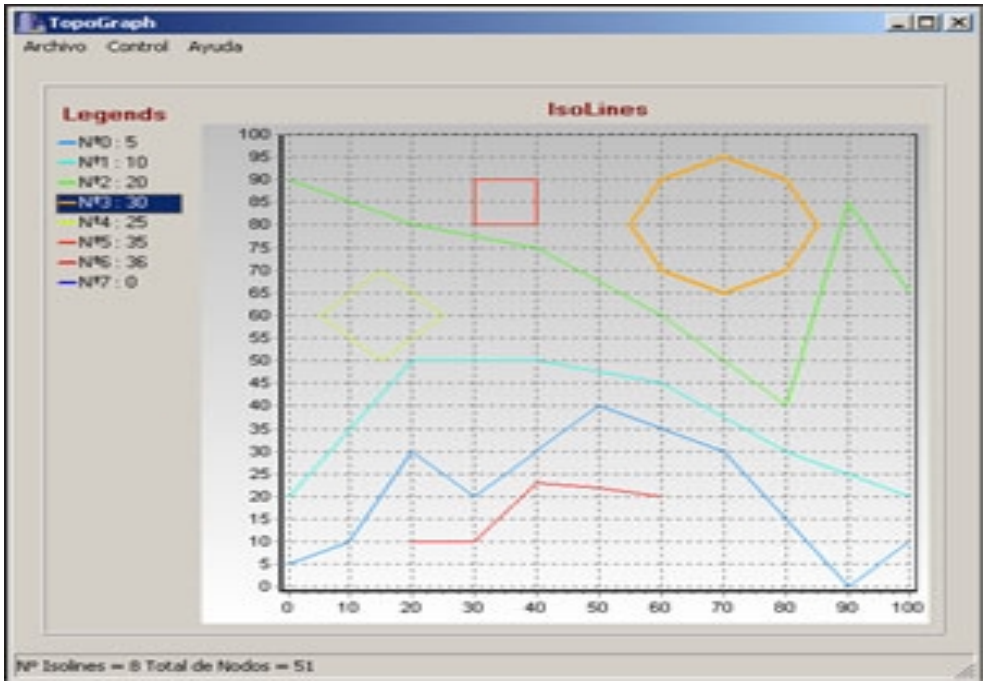


Figura 2: Un ejemplo arbitrario de Isolíneas

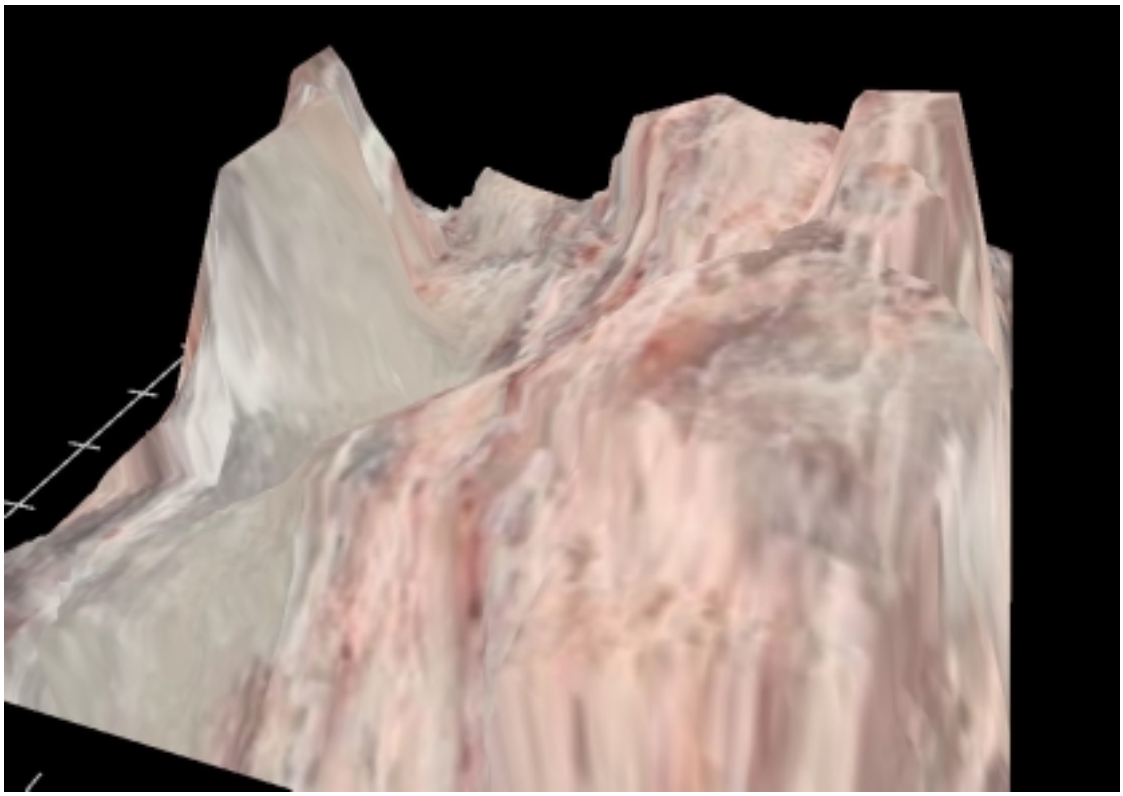


Figura 3: DEM resultante, obtenido con la componente operador, renderizado como mapa de alturas y mapeo de texturas.

El componente operador:

Implementa la conversión de la información geográfica de la superficie especificado como un conjunto de curvas de nivel (isolíneas) a un mapa digital de elevación (DEM). Este almacena la información en la forma de una matriz bidimensional en la que cada celda contiene la altura promedio de la cuadrícula. El algoritmo se presenta detalladamente en la siguiente Sección.

El componente finalizador:

Componente que recibe el archivo intermedio y realiza un renderizado en 3D de los datos del mapa empleando la biblioteca gráfica OpenGL. Las alturas se representan como coordenada espacial, y además es posible codificar algún mapeo que facilite la lectura visual de datos, como por ejemplo la paleta de colores rainbow (ver Figura 5), o el uso de mapeo de texturas para generar imágenes con fotorealismo (ver Figura 3).

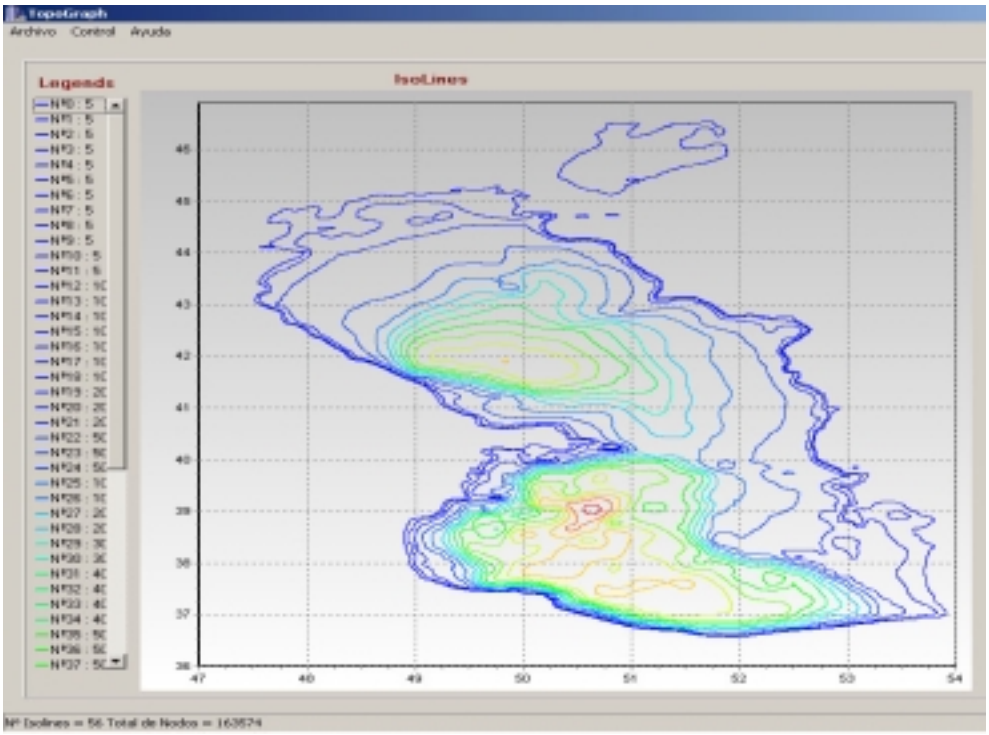


Figura 4: Isolíneas pro cedentes de un archivo de datos del Mar Caspio.

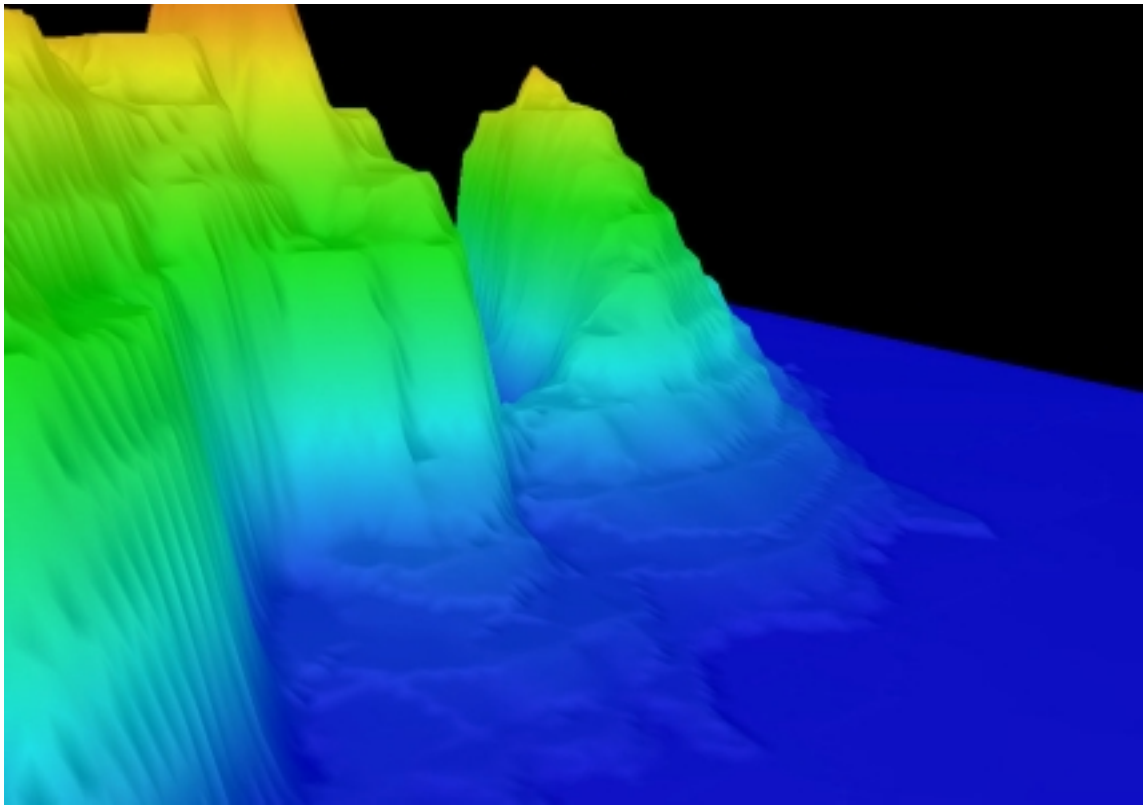


Figura 5: Renderizado 3D del DEM del Mar Caspio con paleta Rainbow.

VI. DESCRIPCIÓN DEL COMPONENTE OPERADOR

Denominamos isolínea a una poligonal plana (de nivel o altura constante). Dado el segmento $\vec{s_1 s_2}$ y el punto p, la distancia mínima entre ellos será uno de los siguientes casos:

$d_{\min} = |s_1 - p|_2$ si $\vec{s_2 p} \cdot \vec{s_1 s_2} < 0$

$d_{\min} = |s_2 - p|_2$ si $\vec{p s_2} \cdot \vec{s_1 s_2} < 0$

o en caso de no verificarse las condiciones:

$$d_{\min} = \frac{|(p_y - s_{1y})(s_{2x} - s_{1x}) - (p_x - s_{1x})(s_{2y} - s_{1y})|}{|s_1 - s_2|_2}$$

donde $\cdot$ denota producto escalar.

Dado el segmento $\vec{s_1 s_2}$, una representación paramétrica posible es:

$$\begin{cases} x = s_{1x} + t(s_{2x} - s_{1x}) \\ y = s_{1y} + t(s_{2y} - s_{1y}) \end{cases} \quad t \in [0,1]$$

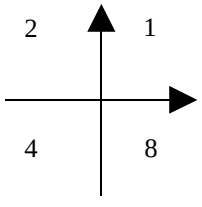
dado el punto $p = (p_x, p_y)$, la intersección del segmento con $x = p_x$, resulta:

$$\begin{cases} t_i = \dfrac{p_x - s_{1x}}{s_{2x} - s_{1x}} \\ y_i = s_{1y} + t_i(s_{2y} - s_{1y}) \end{cases} \quad (2)$$

y para $y = p_y$,

$$\begin{cases} t_i = \dfrac{p_y - s_{1y}}{s_{2y} - s_{1y}} \\ x_i = s_{1x} + t_i(s_{2x} - s_{1x}) \end{cases} \quad (3)$$

Antes de computar las intersecciones deberíamos verificar que el segmento realmente corta a los ejes. Una manera simple de hacerlo es verificar el cuadrante con centro en p que contiene a cada uno de los extremos del segmento. Dada la definición trigonométrica de cuadrante, cuadrantes cruzados son aquellos cuyos semi-ejes poseen sentidos opuestos. En nuestro caso se enumerarán los cuadrantes de la siguiente forma:



El método implementado por esta componente se basa en la siguiente idea: para cada punto p de la estructura regular, desplazar el eje de coordenadas en dicho punto y obtener la isolínea más cercana a p para cada uno de los cuadrantes con centro en él.

Para cada segmento $\vec{s_1 s_2}$, de cada isolinea, verificar que cuadrantes contienen los extremos s_1 y s_2. En caso de que pertenezcan al mismo cuadrante $c \in (1,2,4,8)$, usar (1) para computar la distancia al centro p. Si el segmento es contenido en más de un cuadrante, particionarlo usando (2) y (3) y computar la distancia para cada partición. Habiendo obtenido las distancias mínimas para cada cuadrante con centro en p y sus niveles de alturas asociados, se calcula el nivel de p interpolando con dichos valores. Para el caso de dos valores a interpolar utilizamos la siguiente ecuación (ver Fig. 6):

$$N_p = N_A k + N_B (1-k) \ , \quad k = \frac{\frac{l_A - l_B}{l_A + l_B} + 1}{2} \ ,$$

donde N representa el nivel y l la distancia.

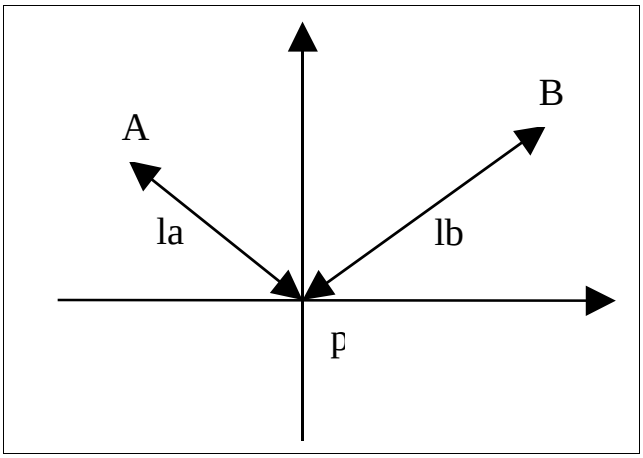


Figura 6: Geometría de la interpolación.

En el caso de segmentos cuyos extremos pertenezcan a cuadrantes cruzados generalmente se de berá particionarlo en tres segmentos (ver Fig. 7)

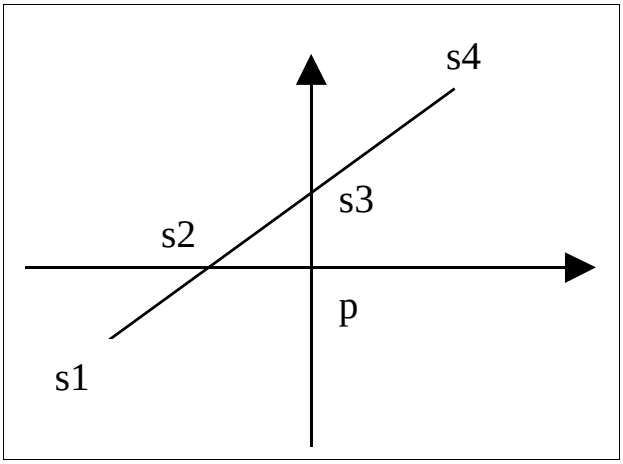


Figura 7: Interpolación de un segmento en cuadrantes cruzados.

VII. ALGUNAS TÉCNICAS DE OPTIMIZACIÓN

Para cada segmento del réctangulo que contiene a una isolínea, computar la distancia a p . Si éstas son mayores a las distancias mínimas almacendadas en un cuadrante o grupo de cuadrantes no cruzados pasar a la siguiente isolinea. En la Fig. 8 vemos un ejemplo de esta idea.

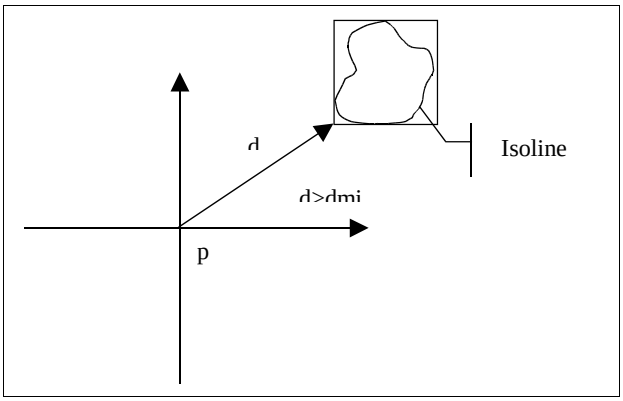


Figura 8: Simplificación de la interpolación de una isolínea.

Otra técnica de optimización consiste en crear una estructura para isolineas cerradas que contienen a otras del mismo tipo aprovechando la propiedad de que, al ser curvas de nivel, no deberían intersectarse. Por último cabe aclarar que el método puede programarse en varios hilos(threads), cada uno analizando una isolinea distinta.

VIII. CONCLUSIONES

Se han presentado las bases para un sistema de modelación conceptual y visual para la gestión de sistemas de información geográfica. La idea del mismo se bases en la interconexión de

componentes que comparten una misma estructura de información basado en GML. Se presentó como ejemplo el modelo conceptual y la implementación de una aplicación que permite el renderizado de un DEM como superficie en 3D a partir de un archivo de curvas de nivel. Este ejemplo ilustra los roles de los componentes productor, operador y consumidor para la generación de un único modelo conceptual. Se describió además en detalle el algoritmo de funcionamiento del componente operador.

BIBLIOGRAFÍA

ESRI "*ArcView and ArcInfo*", Software products. www.esri.com

Golden Software, "*Surfer Version 6*", Software product. www.goldensoftware.com

Claudio Delrieux, Gustavo Ramoscelli, Leonardo Arlenghi y Alejandro Vitale. *Image Processing Spreadsheet.* En: "Hybrid Image and Signal Processing VIII", (SPIE Proceedings Vol. 4735), págs. 67-78. The International Society for Optical Engineering Press. ISBN: 0-8194-4485-5, 2002. `www.spie.org/web/abstracts/4700/4735.html`.

Farin G., *"Curves and Surfaces for Computer Aided Geometric Design"*, Academic Press, NY, 1988.

Marchal, B., "*XML con ejemplos*", Pearson Educación de Méjico, 2001

Maling, D.H. *"Coordinate Systems and Map Projections"*, Woborn MA, Butterworth-Heinemann, 1993.

OpenGis Consortium, *"The OpenGIS Abstract Specification, Version 4"*, OpenGis Consortium, www.opengis.org, 1999.

Shroeder W., Martin K., Lorensen W., *"The Visualization Toolkit; An Object Oriented Approach to 3D Graphics"*, Prentice Hall, Old Tappau, NJ 1998.