

REMALLADO INTERACTIVO COMO MÉTODO PARA LA NAVEGACIÓN EFICIENTE DE MODELOS TOPOGRÁFICOS

MARÍA V. CIFUENTES¹, JUAN P. D'AMATO¹, CRISTIAN GARCÍA BAUZA²,
MARCELO J. VÉNERE³, ALEJANDRO CLAUSSE^{3,4}

PLADEMA-ISISTAN, Universidad Nacional del Centro, 7000 Tandil, Argentina.
{cifunte, jpdamato, crgarcia, venerem, clausse}@exa.unicen.edu.ar

¹ Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

² Universidad Nacional del Centro de la Provincia de Buenos Aires

³ Comisión Nacional de Energía Atómica

⁴ CONICET

PALABRAS CLAVE: DISCRETIZACIÓN DE SUPERFICIES, REALIDAD VIRTUAL, GEOMETRÍA COMPUTACIONAL.

RESUMEN. *Se propone un algoritmo que permite la navegación ágil de grandes escenarios, como aquellos que se basan en modelos topográficos digitales. Básicamente, la idea gravita en construir una malla poligonal aproximada de la escena cada vez que el observador (o cámara) se desplaza; donde, sin alterar la estructura del paisaje original, las zonas alejadas se representan con un conjunto más reducido de polígonos que los alrededores de la actual posición del observador. La simplificación poligonal resultante se construye con sucesivas divisiones del campo escalar de alturas que generan una estructura jerárquica quadtree con restricciones pudiéndose incluso obtener diversas aproximaciones de un modelo variando la cota de error condicionante. En particular, el indicador de error propuesto integra información de curvatura local y ubicación del observador. Se estudia un caso sintético y se presenta una aplicación a modelos de elevación reales.*

1. INTRODUCCIÓN

Tanto los nuevos juegos por computadoras como las aplicaciones de realidad virtual que se basan en escenarios exteriores demandan la representación de topografía en forma realista y eficiente. La tendencia actual es abandonar las ideas basadas en efectos artificiales como imágenes proyectadas o representaciones sintéticas y utilizar modelos digitales de elevación (*MDE*) reales de la mayor precisión posible. Básicamente un *MDE* está constituido por una grilla georeferenciada donde cada elemento (o celda) tiene un valor de cota o altura. A su vez, cada elemento de la grilla lleva asociado un par de coordenadas que se corresponde con un sistema de representación terrestre (p.e. geodésicas o Gauss Kruger). De esta manera, el *MDE* constituye una representación más o menos fiel de una determinada porción de la superficie terrestre.

Obviamente, mientras más densa sea esta grilla, mayor nivel de detalle estará representado mediante el *MDE*, siendo normal hoy utilizar modelos con decenas o incluso centenas de millones de celdas. Este volumen de información puede ser necesario para muchas aplicaciones, pero representa un serio inconveniente para su visualización interactiva y más aún si se desea “navegar” en tiempo real esta topografía^{1,2}.

En este trabajo se propone un algoritmo que permite reducir la cantidad de polígonos utilizados para representar estos modelos, sin perder calidad en la imagen generada. Para ello se utiliza una representación jerárquica basada en un *quadtree* con restricciones y *templates* en cada terminal del árbol para generar la triangulación final de la superficie. En primera instancia como indicador de cuales sectores pueden ser simplificados y cuales no, se utiliza la curvatura local. De esta forma los lugares con baja curvatura pueden ser simplificados sin perder calidad, lo cual en general permite reducir la cantidad de polígonos en un factor diez y este factor es aún mayor si se tolera una cierta pérdida³.

De todas formas estas reducciones, si bien importante, pueden no ser suficientes, por lo cual se incorporó al indicador de error el concepto de dónde está el observador. La malla poligonal resultante no permanece estática, se extrae desde la jerarquía *quadtree* que cambia dinámicamente cada vez que el observador se mueve.

El trabajo se organiza en siete secciones que abordan el detalle de la técnica de simplificación poligonal propuesta. Primeramente se describen el criterio y el método propuesto para la simplificación poligonal de un *MDE* empleando la curvatura local –secciones 2 y 3-. Luego se incorpora el concepto de distancia del observador al proceso de simplificación, detallando el nuevo indicador de error y la generación dinámica de la estructura necesaria para la navegación –sección 4-. Los resultados experimentales se exponen en la sección 5 y surgen de evaluar nuestro esquema sobre un terreno generado sintéticamente sobre el que se ofrecen vistas de las mallas aproximadas generadas cuando el observador permanece estático y cuando realiza una travesía. Además se incluyen algunos ejemplos sobre topografías reales y una implementación concreta sobre el motor gráfico *IMPROMPTU* para la visualización de escenarios virtuales. Por último, la sección 6 y 7 detallan las conclusiones y las referencias del trabajo.

2. CRITERIO PARA LA SIMPLIFICACIÓN

Decimos que una región adquiere el grado de detalle necesario cuando la información contenida en ella cumple con un dado criterio de tolerancia permitido, situación que se alcanzará después de un proceso continuo de subdivisiones sucesivas. Tanto las mallas como los criterios de simplificación serán isotrópicos, considerando una forma de curvatura media presentada en la ec. (4). En un primer momento, como indicador de cuales sectores pueden ser simplificados y cuales no, se utiliza la curvatura local de la región que compara la desviación de la geometría intrínseca de la superficie respecto a la geometría del plano. La idea es que una región con alta curvatura cambia abruptamente su pendiente y tiene que ser modelada con más triángulos por unidad de área que una región de baja curvatura que permanece relativamente plana. De esta forma los lugares con baja curvatura pueden ser simplificados sin perder calidad, lo cual en general permite reducir la cantidad de polígonos en un factor diez y este factor es aún mayor si se tolera una cierta pérdida.

Teniendo en cuenta que la superficie inicial está descrita mediante una grilla regular con tamaño de celda constante, para calcular el valor de curvatura local κ en un punto P del *MDE* se utiliza un esquema centrado de diferencias finitas. Dado el campo escalar de alturas $h(x, y)$ se calcula primero el gradiente en forma centrada para una dada celda $P_{i,j}$ como:

$$\nabla h = \left[\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y} \right]^T = [G_x, G_y]^T \quad (1)$$

$$G_x_{i,j} = \frac{\partial h_{i,j}}{\partial x} = \frac{h_{i+1,j} - h_{i-1,j}}{2 \Delta x} \quad (2)$$

$$G_y_{i,j} = \frac{\partial h_{i,j}}{\partial y} = \frac{h_{i,j+1} - h_{i,j-1}}{2 \Delta y}$$

Las componentes del tensor curvatura se calculan también en forma centrada como:

$$(3)$$

donde se asume que $\Delta x = \Delta y = \Delta$.

Dado que en nuestra aplicación interesa conocer solo la magnitud de la curvatura y no sus componentes individuales, se utilizó el siguiente indicador:

$$K_{i,j}^2 = (G_{x_{i+1,j}} - G_{x_{i-1,j}})^2 + (G_{y_{i+1,j}} - G_{y_{i-1,j}})^2 + (G_{x_{i,j+1}} - G_{x_{i,j-1}})^2 + (G_{y_{i,j+1}} - G_{y_{i,j-1}})^2 \quad (4)$$

3. MÉT
ODO

PARA LA SIMPLIFICACIÓN

El método que reduce la complejidad geométrica de superficies extensas transforma el campo de alturas original en un campo cuadrado de longitud de lado $L/2^k$, donde k identifica el nivel de refinamiento y L el tamaño del dominio⁴. El algoritmo subdivide al nodo raíz en amplitud (*breadth first*) hasta que se cumpla con el criterio de tolerancia presentado; es decir, subdivide cuando el “acumulado de curvaturas” (o suma de los valores de curvatura local de cada celda de una región del MDE) supera al valor de tolerancia permitido. Finalmente, construye una jerarquía de mallas con complejidades diferentes que representan el mismo terreno organizadas en un *quadtree* (árbol cuaternario). Cada cuadrado en la jerarquía poseerá un error determinado, que puede ser mayor o igual al error de todos sus descendientes. Los nodos más pequeños se ubican en los niveles más inferiores y con los terminales se construye la malla poligonal de máxima resolución.

La figura 1 muestra como queda esta estructura para un caso simple de cambio de pendiente y a la derecha la malla finalmente generada.



Figura 1– Vista en planta y en perspectiva rotada de la descomposición recursiva *quadtree* en un caso simple como el cambio de pendiente.

Otro problema es la no conformidad de la malla generada, presencia de huecos en la trama de la malla que son perceptibles en la visualización. Hay autores que proponen la solución de las discontinuidades insertando polígonos adicionales que cubran el área de los agujeros (con la desventaja de que estos polígonos seguramente serán muy deformados)⁵, otros que proponen seleccionar puntos restringidos de acuerdo al grafo de dependencia definido entre los vértices⁶ y otros, proponen generar un *quadtree* restringido con el requerimiento que cuadrantes adyacentes -bloques *quadtree* vecinos localizados con los algoritmos propuestos por Ibaroudene⁷ deben diferir a lo sumo en un nivel en la jerarquía *quadtree*⁸.

Nuestro algoritmo adopta la última propuesta, se acuerda que las regiones ortogonalmente adyacentes no difieran en más de un nivel de subdivisión, reduciendo a 16 las posibles situaciones o *templates* que producen huecos. La figura 2 muestra como queda representado el *quadtree* de la izquierda con esta restricción, donde las líneas punteadas identifican las

subdivisiones adicionales impuestas en el criterio de refinamiento empleado a las regiones vecinas de los vértices problemáticos.

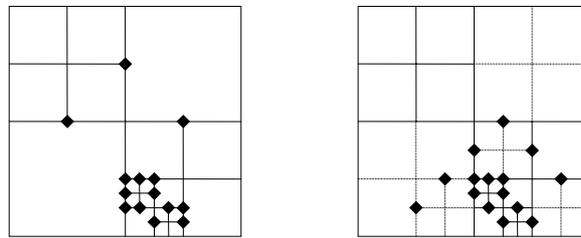


Figura 2- Izquierda: *Quadtree* sin restricciones. Derecha: *Quadtree* con restricciones.

A partir de este tipo de árbol, generar la triangulación es un proceso mucho más simple ya que una hoja solo puede encontrarse en una de 16 posibilidades: las 15 que se muestran en la figura 3 y la trivial en que la hoja se divide en dos triángulos. Es decir, para generar la triangulación basta recorrer los terminales del árbol y dividir las hojas según el *template* correspondiente. Es interesante observar que la triangulación que se obtiene en esta manera presenta solo cuatro tipos de triángulos de forma diferente, todos ellos de excelente calidad. La crítica más atendible a esta estrategia es que se agregan triángulos que a priori no eran necesarios, sin embargo en Rivara⁹ se demuestra que la cantidad de triángulos adicionales al forzar este tipo de subdivisiones es $O(\text{NumTerminales}^{1/2})$, lo cual resulta despreciable frente a la cantidad de terminales a graficar $O(\text{NumTerminales})$.

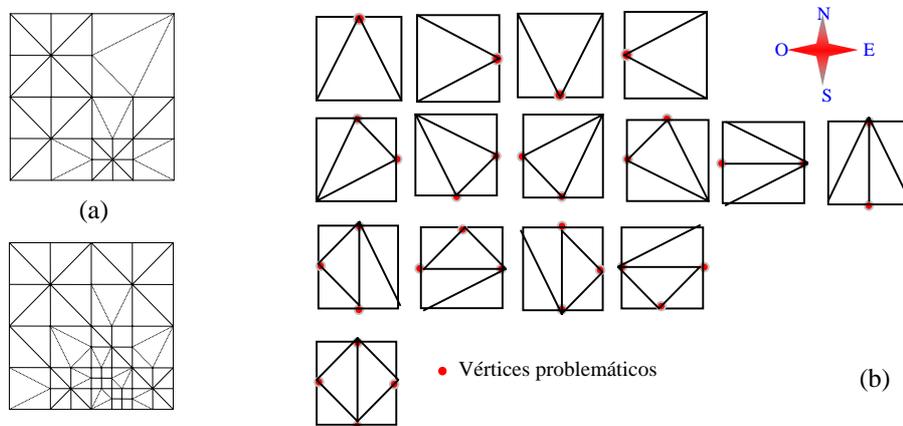


Figura 3- (a) Mallas de triángulos cerradas aplicando el esquema de *templates* de la derecha (regiones subdivididas con líneas punteadas). (b) Presentación de los 15 *templates* para discretizar los terminales.

4. GENERACIÓN DINÁMICA

Modificaciones al criterio de simplificación

De todas formas estas reducciones, si bien importantes, pueden no ser suficientes, por lo cual se incorporó al indicador de error el concepto de dónde está el observador (o cámara)^{10,11}.

La idea consiste en que las regiones cercanas al observador o con cambios abruptos de pendiente se modelen con más triángulos por unidad de área que las regiones alejadas o relativamente planas. La malla poligonal resultante no permanece estática, se extrae desde la jerarquía *quadtree* que cambia dinámicamente cada vez que el observador se mueve.

Básicamente, el algoritmo opera con la superficie descripta mediante una grilla regular con tamaño de celda constante que agrupa valores de curvatura local resultantes de aplicar un esquema centrado de diferencias finitas a los puntos del campo escalar de alturas. La subdivisión de la superficie surge al evaluar el criterio o indicador de error presentado como

$$\frac{K}{\left(a \frac{d}{D} + 1 \right)} \leq T \quad (6)$$

donde K es el acumulado de curvatura local en la región, T es la tolerancia permitida por el usuario, a es una constante y D es información estable (por ejemplo, la longitud de la diagonal principal del campo de alturas). d mide la distancia euclídea entre el centro de la región *quadtree* analizada y la ubicación actual del observador. Para cada región se evalúa $K / \left(a \frac{d}{D} + 1 \right)$ y cuando supera un valor T definido por el usuario se dispara una subdivisión.

Figura 4- Simplificación Poligonal: criterio de curvatura local (izquierda) y agregando el concepto de ubicación del observador (derecha).

A modo comparativo, en la figura 4 se aprecia la importante reducción poligonal de una dada grilla regular cuadrada de tamaño de celda constante que ha sido simplificada aplicando el criterio de curvatura local (izquierda) y el concepto de ubicación del observador (derecha). Si bien ambas emplean la misma tolerancia T en su generación, es notable la reducción del conjunto de vértices cuando se introduce la posición del punto de vista.

Modificaciones a la implementación

Inicialmente se construye la *jerarquía pasiva*: estructura jerárquica cuaternaria que almacena la superficie original con resoluciones diversas. El indicador de error usado se basa en la curvatura local, reduciendo significativamente el conjunto de vértices del modelo sujeto a condiciones de co-planaridad toleradas. Tal situación exige que la constante a se anule en

(6). Nótese que, cuando $a > 0$ se incorpora al criterio la posición del observador y, conforme se desplaza, cambia la triangulación final de la superficie resultando indispensable el remallado interactivo del modelo.

En particular, cuando $a > 0$, se aplica una estrategia *top-down* sobre la *jerarquía pasiva* construyendo reiteradamente la *jerarquía activa*. Comienza en el nodo raíz y la recorre en amplitud (*breadth first*) activando o desactivando vértices según cumplan con el criterio o indicador de error sugerido por el usuario al inicio de un recorrido (por ejemplo: valor de tolerancia deseado, ubicación inicial del observador, valor de $a > 0$, etc.). La malla a renderizar reúne a todos los nodos activos terminales de la *jerarquía activa* mediante un recorrido en profundidad. Por último, manteniendo la restricción propuesta en el método de simplificación –sección 3- se garantiza la conformidad de la malla final.

5. RESULTADOS

El terreno generado sintéticamente es una grilla regular cuadrada de 256 píxeles de lado que simbolizan el relieve asociando el color del píxel con la información de elevación para ese punto del plano. Conforman la malla original 131.072 polígonos (triángulos) que reproducen a una superficie con regiones uniformes -o planas- en los extremos mientras que en la parte central, a modo de escala monocromática, se dispone una sucesión gradual de grises que progresan desde 0 hasta 255.

Simplificar sin navegar

Los datos tabulados, expuestos a continuación, expresan los resultados derivados de aplicar sucesivas simplificaciones poligonales a la geometría del terreno sintético en estudio mediante la técnica de curvatura local propuesta al principio ($a = 0$). La solución excluye a todos los vértices redundantes, es decir, aquellos depurados por el proceso de análisis de la curvatura local y las comparaciones con valores de umbral disímiles introducidos por el usuario. Recuérdese que el valor de umbral acota superiormente a la curvatura local de una región; por ejemplo, un valor de umbral nulo restringe a que únicamente se subdividan aquellas regiones cuya curvatura local (suma de todas las curvaturas de los puntos interiores a ella) supere a cero. Nótese que, en todos los casos la cantidad de polígonos de la malla original se redujo, aún para un valor de umbral nulo, situación en la que se agrupan puntos coplanares de la superficie.

Tolerancia	0.0	0.01	0.05	0.10	0.5	1.0
Triángulos	8880	2352	432	128	8	2
% de triángulos respecto a la malla original	6,77	1,79	0,33	0,10	0,006	0,002
Divisiones forzadas	488	232	40	8	0	0

Tabla I- Criterio de Curvatura. Resultados arrojados por el algoritmo propuesto asignando distintos valores de tolerancia en el procesamiento de una malla regular cuadrada con 131.072 triángulos que representa al terreno sintético analizado.

Los resultados observados en la Tabla I, derivados de la triangulación simplificada de un dominio particular, evidencian la substancial reducción en la cantidad de polígonos de las mallas aproximadas comparadas con el tamaño de la malla original. Tal es así que, la malla aproximada más densa, generada con un umbral nulo, involucra menos del 10% de los polígonos presentes en la malla original (ver Tabla I). También es cierto que, empleando valores de umbral cada vez más grandes reducimos tanto la cantidad de polígonos como la cantidad de subdivisiones forzadas en la triangulación irregular compuesta (ver figura 5 a la izquierda y derecha respectivamente).

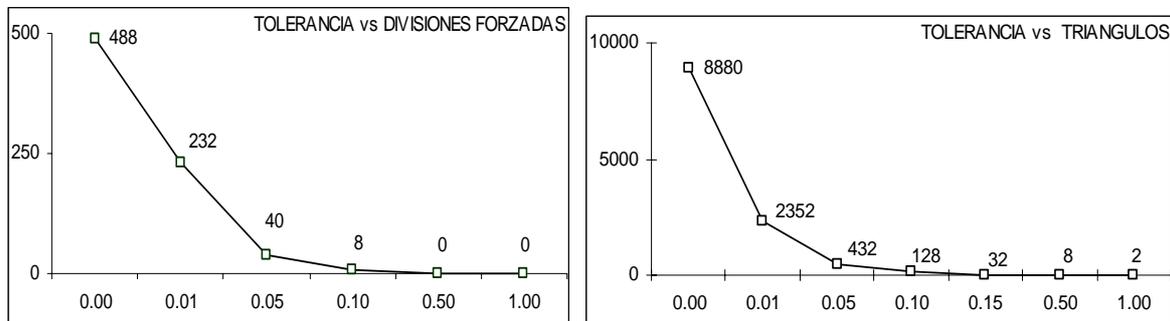


Figura 5- Representación gráfica de los resultados de la Tabla I que relacionan el número de subdivisiones forzadas (izquierda) y la cantidad de polígonos de la malla (derecha) para los valores de tolerancia permitidos.

La figura 6 reproduce varias vistas en perspectiva del terreno sintético formulado, consecuentes con los resultados presentados en la Tabla I. Las mallas irregulares de las figuras 6.b, 6.c y 6.d son vistas de simplificaciones poligonales cuyos valores de tolerancia equivalen a 0,001, 0,01 y 0,125 respectivamente. Si bien la cantidad de vértices implicados en cada malla varía considerablemente, todas las aproximaciones generadas se adaptan a la geometría del terreno y lo representan apropiadamente (comparar con la figura 6.a).

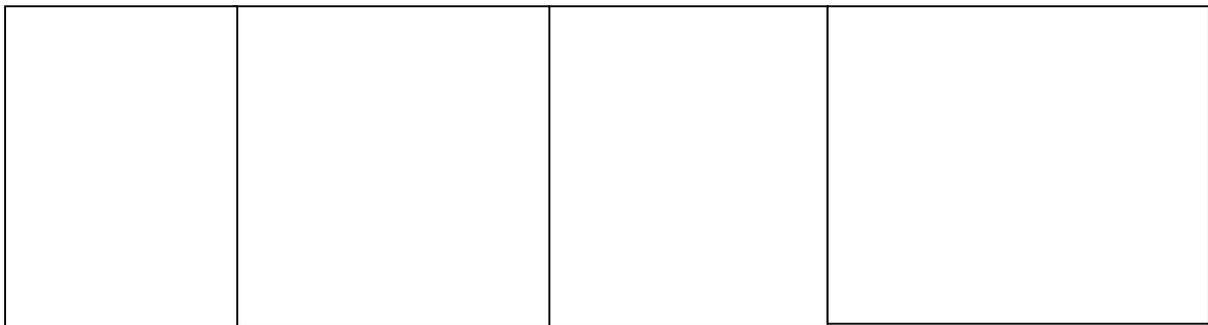


Figura 6- Representación en perspectiva del terreno generado sintéticamente. (a) Malla regular de 131.072 triángulos. Mallas Simplificadas: (b) 10.224 triángulos ($T=0,001$), (c) 1072 triángulos ($T=0,01$) y (d) 32 triángulos ($T=0,125$).

Claramente, observamos que en la representación bidimensional de la malla (ver figura 6.b) hay franjas del terreno más densas que otras: los extremos superior e inferior y la parte media reúnen espacialmente a grandes polígonos mientras que, las franjas de superficie restantes concentran a elevadas cantidades de polígonos pequeños. La causante de esta peculiaridad es la geometría del terreno y el criterio de subdivisiones del algoritmo propuesto; es decir, del gráfico se verifica que la distribución de los polígonos de la malla se mantendrá más densa en aquellas regiones que presenten cambios bruscos en la curvatura local y menos densa en otro caso.

Simplificar navegando

A medida que el observador navega la superficie, la malla experimenta repetidas simplificaciones poligonales. Tales aproximaciones satisfacen el indicador de error escogido para un dado α estable durante toda la trayectoria. La figura 7 representa a tres simplificaciones de la grilla regular sintética que surgen como consecuencia del desplazamiento del observador (punto azul). Nótese que, en la triangulación final las regiones cercanas al observador (o cámara) se modelan con una mayor cantidad de triángulos por unidad de área que las alejadas y cuanto más cercana está la cámara, más triángulos definen la superficie que rodea al observador. Además, la localización del punto de vista favorece la reducción de triángulos sin alterar la estructura original de la aproximación.

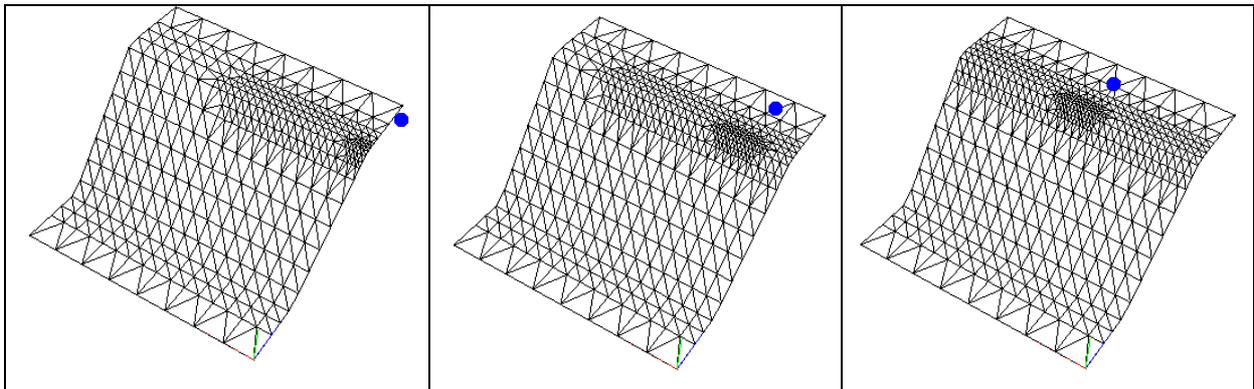


Figura 7 - Ejemplo de remallado interactivo del modelo sintético. Simplificaciones poligonales correspondientes a tres posiciones del observador que realiza una navegación eventual sobre la superficie con $\alpha=70$ y $T = 0,001$.

Aplicación a MDE reales

El motor gráfico *IMPROMPTU*, desarrollado en la UNCPBA, adopta la biblioteca gráfica OpenGL¹² para la representación de ambientes virtuales. En su navegación, la cámara está incorporada en el interior de embarcaciones o vehículos de circulación acompañada que demandan renderizados de 20 frames por segundo. Al presente, la aplicación implementa el algoritmo propuesto remallando la escena cada vez que la cámara se desplaza una cierta distancia, reduciendo el costo computacional del renderizado debido a que se ha disminuido el conjunto de vértices de la escena original, manteniéndose estable un tiempo considerable.

La figura 8 muestra una posible simplificación del *MDE* correspondiente a una región del lago Nahuel Huapi de Bariloche (Argentina) y tres vistas procedentes del motor gráfico *IMPROMPTU*, capturadas mientras se navega el modelo. Nótese que, a medida que el observador o cámara se acerca se agrega detalle a la escena y las montañas adoptan su geometría real.

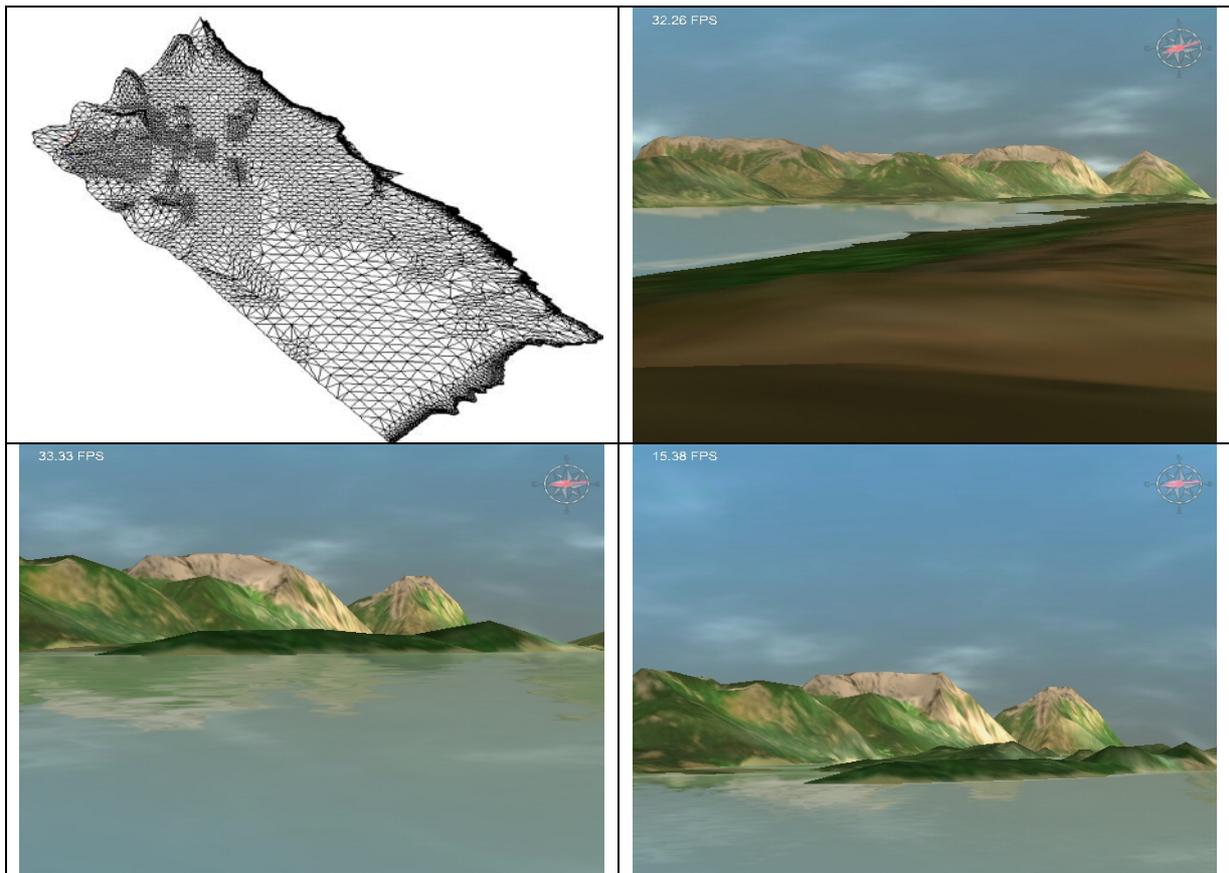


Figura 8 - *MDE* correspondiente al Lago Nahuel Huapi, Bariloche, Argentina. (a) Malla irregular de triángulos. Las zonas densas corresponden a regiones cercanas al observador. En (b), (c) y (d) vistas de la navegación con remallado interactivo del modelo.

6. CONCLUSIONES

Se ha propuesto un algoritmo que permite reducir dinámicamente la cantidad de polígonos utilizados para la representación de modelos de elevación. Las reducciones obtenidas dejan mallas con el 5% al 10% de los polígonos originales lo cual tiene un impacto directo en el costo computacional del proceso de *rendering* de la imagen.

Es importante mencionar que el costo de realizar esta simplificación es comparable a la del *rendering* sobre la superficie original, por lo que este proceso no debería utilizarse para cada

ciclo movimiento-*rendering*, sino cuando el desplazamiento de la cámara sea lo suficientemente importante como para justificar un remallado.

En las aplicaciones implementadas dentro del motor gráfico *IMPROMPTU*, los desplazamientos de la cámara están asociados a vehículos o embarcaciones, los cuales se mueven a velocidades de pocos kilómetros por hora. Si se tiene en cuenta que una aplicación de este tipo debe trabajar al menos a 20 frames por segundo, se obtiene que la necesidad de remallado ocurre cada cientos o incluso miles de *renderings*, con lo cual la aplicación del algoritmo propuesto se amortiza holgadamente.

7. REFERENCIAS

- [1] Casillas, J., Sevilla, J., Blasco, J., Pardo, C., Romero, C., Fernández, J. (1999) “*Implementación de un sistema visual para un simulador de helicóptero*”. Congreso Español de Informática Gráfica. CEIG’99, Jaen, España, Junio.
- [2] Hernández, L., Taibo, J., Seoane, A. (1999) “*Una aplicación para la navegación en tiempo real sobre grandes modelos topográficos*”. Congreso Español de Informática Gráfica. CEIG’99, Jaen, España, Junio.
- [3] Cifuentes, M.V., Venere, M.J., Clause, A. (2004) “Un algoritmo para la simplificación poligonal de modelos topográficos digitales”. 33ª JAIIO, Jornadas Argentinas de Informática e Investigación Operativa. Córdoba, Argentina, Septiembre.
- [4] Samet, H. (1984) “*The quadtree and related hierarchical data structures*”. Association for Computing Machinery. Computing Surveys, 16 (2):187-260, June 1984.
- [5] DeHaemer, M., Zyda, M. (1991) “*Simplification of objects rendered by Polygonal Approximations*”. Computers & Graphics, 15(2):175-184.
- [6] Pajarola, R. (1998) “*Large scale Terrain Visualization using the restricted quadtree triangulation*”. Reporte interno 292 del Institute of Theoretical Computer Science. ETH Swiss Federal Institute of Technology. Zurich, Suiza.
- [7] Ibaroudene, D., Demjanenko, V., Acharya, R. (1990) “*Adjacency algorithms for linear octree nodes*”. Image and Vision Computing, Butterworth & Co Publishers Ltd, 8(2):115-123, May.
- [8] Von Herzen, B., Barr, A (1987) “*Accurate triangulation of deformed, intersecting surfaces*” Computer Graphics ,SIGGRAPH’87 Proceedings. Pág. 103-110.
- [9] Rivara, M., Vénere, M. (1996) “*Cost Analysis of the longest-side refinement algorithm for triangulations*”. Engineering with Computers, Vol. 12, pág. 224-234.
- [10] Hoppe, H. (1997) “*View-Dependent Refinement of Progressive Meshes*”. Computers & Graphics. Proceedings of SIGGRAPH’97, p.189-198.
- [11] Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L., Faust, N., Turner, G. (1996) “*Real-Time, Continuous Level of Detail Rendering of Height Fields*”. Computers & Graphics. Proceedings of SIGGRAPH’96.
- [12] Woo, M.; Neider, J.; Davis, T. (1997).”*OpenGL Programming Guide: The Official Guide to Learning OpenGL*”. Version 1.1. Addison-Wesley Developers Press.