

Tipo de artículo: Artículo original  
Temática: Matemática computacional  
Recibido: 11/07/2016 | Aceptado: 23/09/2016

## **Visualización en tres dimensiones de capas de información geográfica a partir de modelos digitales de elevación**

### *Three dimensional visualization of geographic information layers from digital elevation models*

José Luis Castrillón Garrido <sup>1\*</sup>, Nelson González Peñate<sup>1</sup>, Gilberto Arias Naranjo<sup>1</sup>

<sup>1</sup>Centro de Estudios de Matemática Computacional. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370. Correo-e: [{jlcastrillon, npenate, gilberto}@uci.cu](mailto:{jlcastrillon, npenate, gilberto}@uci.cu)

\* Autor para correspondencia: [jlcastrillon@uci.cu](mailto:jlcastrillon@uci.cu)

---

#### **Resumen**

La representación en tres dimensiones de información geográfica georreferenciada es de gran utilidad en diversas áreas de la ciencia. Los modelos digitales de elevación utilizados con este propósito pueden alcanzar elevadas dimensiones, lo que dificulta la visualización directa de los mismos. Otro elemento que favorece la representación realista de la escena es la introducción de fenómenos ópticos en las superficies de agua, como es el caso de la reflexión y refracción de la luz. El principal objetivo del presente trabajo se enmarcó en la visualización realista de modelos digitales de elevación de altas dimensiones incluyendo aquellos con superficies de agua. Con este fin se utilizó el modelo de Mallas Dinámicas Adaptativas por Lotes para la obtención en tiempo real de la geometría en distintos niveles de detalle y la técnica de Mapeo de Textura de Proyección con el objetivo de dotar a las capas de agua de los fenómenos ópticos antes mencionados. Los resultados obtenidos demostraron la efectividad de las técnicas utilizadas, permitiendo la visualización realista de superficies de terreno y de agua a partir de modelos digitales de elevación en el orden de los cientos de millones de puntos.

**Palabras clave:** Modelo Digital de Elevación, modelo multi-resolución, realismo, tres dimensiones.

#### *Abstract*

*Three dimensional representation of georeferenced geographic information is quite useful in several science fields. The digital elevation models used on that purpose may be large in size, making impossible their direct visualization. Another element favoring the realistic representation of the scene is the insertion of optical phenomena in the water surfaces, e.g. light reflection and refraction. The main goal of the present work is focusing on the realistic visualization of large digital elevation models including the ones with water surfaces. The batched dynamic adaptive meshes model was used to obtain the geometry in different levels of detail and the Projection Texture Mapping technique to supply the water layers with the already mentioned optical properties. Results shown the effectiveness of the employed techniques, allowing the realistic visualization of terrain and water surfaces from large digital elevation models in the order of hundreds of millions of samples.*

**Keywords:** Digital elevation model, multiresolution model, realism, three dimensions

---

## Introducción

La visualización de información geográfica en tres dimensiones a partir de modelos digitales de elevación (MDE) tiene gran utilidad en diversas áreas tales como la visualización científica, los sistemas de información geográfica, los juegos y simulaciones en tres dimensiones. El aumento constante de las tecnologías satelitales y de las técnicas para la extracción de las elevaciones ha hecho posible la disponibilidad de MDE con enormes cantidades de datos (en el orden de  $10^6$  hasta  $10^9$  datos de elevación aproximadamente) y con texturas de alta definición, haciendo imposible la visualización directa de los mismos de manera interactiva. Los MDE también han sido utilizados para representar superficies de agua, un ejemplo es la simulación de fenómenos hidrológicos (tsunamis, crecidas de río), haciendo posible la coexistencia de enormes superficies de terreno y de agua en una misma escena. Otro elemento a tener cuenta durante la representación de estos fenómenos es el realismo, donde las propiedades físicas de reflexión y refracción de la luz juegan un papel fundamental (DARLES et al., 2011).

La triangulación de los MDE mediante modelos multi-resolución ha sido objeto de un gran número de estudios tal como se refleja en los artículos de revisión (Cohen y Manocha, 2005) (De Floriani et al., 2005) y (Pajarola y Gobbeti, 2007). Estos modelos, en su mayoría, constan de tres elementos fundamentales: una malla base, que define la aproximación de menor resolución del terreno, un conjunto de actualizaciones, que al ser aplicadas a la malla base se obtienen variables resoluciones de la superficie del terreno, y por último un conjunto de dependencias entre las actualizaciones. Un modelo multi-resolución permite extraer en tiempo de ejecución una representación consistente y con la menor complejidad posible minimizando una métrica de error específica a través de una consulta dependiente de los parámetros de visión.

Entre las formas existentes para la representación multi-resolución de MDE, en el caso más general, se encuentran las técnicas irregulares. *Mallas Progresivas* (HOPPE, 1997) es uno de los ejemplos más representativos de esta clase, donde se comprimen y se almacenan valores históricos de un conjunto de aproximaciones de la red irregular de triángulos (TIN<sup>1</sup>) inicial que son obtenidas mediante el colapso progresivo de las aristas, a partir de estos valores y una TIN base se generan representaciones continuas mutli-resolución de la superficie original. Este concepto fue extendido en (Sander, 2001), haciendo posible la aplicación de una misma imagen de textura para toda la secuencia de aproximaciones. La naturaleza irregular de estas técnicas les permite obtener una representación con la menor complejidad posible para un umbral de error específico, sin embargo, tienen como costo adicional que necesitan una codificación explícita de la conectividad de las TINs, de la jerarquía y de las dependencias (Pajarola, 2007).

Otro grupo considerable de modelos multi-resolución explotan ciertas características de semi-regularidad de los MDE con el fin de crear conexiones más estrictas entre los nodos mediante criterios fijos de subdivisión. Entre los ejemplos más exitosos de esta clase se encuentra el *quadtree* restringido (Pajarola, 1998), donde se generan triangulaciones a partir de la subdivisión de cuadrantes, imponiendo como restricción que dos cuadrantes pueden ser adyacentes si la diferencia entre sus niveles es a lo sumo 1. Los modelos ROAM (Duchaineau et al., 1997) y RTIN (Evans et al., 2001) utilizan como estructura base los *bintrees*<sup>2</sup> que a partir de nodos en forma de triángulos rectángulos generan una jerarquía mediante la bisección de los mismos por la hipotenusa. Las restricciones establecidas para la subdivisión en los modelos semi-regulares permiten obtener una triangulación con menor costo computacional que los modelos irregulares.

Con el incremento del poder de cómputo de las unidades gráficas de procesamiento (GPU) en los últimos años, se han propuesto otras alternativas que aprovechan el *rendering* de grupos de triángulos (denominados parches). CABTT (Levenberg, 2002) aprovecha este concepto mediante la creación de parches a partir de subárboles del *bintree* que son mantenidos en la caché del GPU y reusados entre tramas (*frames*) consecutivas. El modelo BDAM (Cignoni et al., 2003) construye un bosque de *bintrees* de triángulos rectángulos, donde cada nodo representa una triangulación perteneciente al dominio de las redes irregulares de triángulos. Uno de los ejemplos más reciente de los modelos basados en parches es RASTER (Bosch et al., 2009), donde se introduce el concepto de K-parches y M-bloques representando las unidades de un *bintree* y de un *quadtree* respectivamente.

---

<sup>1</sup> TIN: del inglés *triangulated irregular network*

<sup>2</sup> *bintree*: Se utiliza en la literatura para denominar a los árboles binarios de triángulos

Una de las técnicas comunes para dotar las escenas con propiedades de reflexión y refracción de la luz es la técnica Trazado de Rayos (Whitted, 1980). En esta técnica se emplea una simulación óptica avanzada trazando un rayo desde las fuentes de luz que colisiona con los objetos en la escena y causa la creación de nuevos rayos de luz. Este rayo de reflexión se calcula recursivamente en dependencia de la calidad que se quiera lograr (Rodgers, 2014). Mapeo de Textura de Proyección (Jensen y Goliás, 2001) es otra de las técnicas comúnmente utilizada para lograr reflexión en las superficies. Esta técnica permite reflexiones locales y globales en la escena, donde la geometría en 3D puede ser proyectada a un mapa de textura de igual forma que se proyecta la geometría en 3D al monitor en 2D (de la Fuente, 2011).

El presente trabajo se enmarca en el problema de la visualización realista de superficies de agua y de terreno representadas por modelos digitales de elevación que por su tamaño no pueden ser representados en la memoria interna. Con este propósito, se utiliza el modelo BDAM para la obtención eficiente de las geometrías mediante aproximaciones multi-resolución y la técnica de Mapeo de Textura de Proyección para dotar a las superficies aproximadas de las propiedades físicas de reflexión y refracción. Además, se adapta el modelo BDAM para permitir la representación de modelos digitales de elevación con valores nulos (*no data values*).

## **Materiales y métodos**

### **Representación en 3D de modelos digitales de elevación usando BDAM**

El modelo BDAM realiza una partición del terreno en un bosque de *bintrées* de triángulos rectángulos donde la unión de todas las raíces cubre la superficie completa del MDE. BDAM utiliza bisección de triángulos como criterio de subdivisión para la obtención de los nodos intermedios (ver figura 1a), esta operación consiste en reemplazar una región triangular  $\beta$  por las dos regiones triangulares que se obtienen al separar  $\beta$  por el punto medio de la hipotenusa.

Se garantiza una partición conforme del terreno (las aristas son compartidas por a lo sumo dos triángulos) forzando la bisección de las dos (o una, cuando la hipotenusa está en el borde de la superficie) regiones triangulares que comparten la hipotenusa al mismo tiempo, a la unión de estas regiones que comparten la hipotenusa se le denomina diamante y representa un cuadrado. Aplicando la operación de bisección por diamantes se obtienen varias representaciones multi-resolución del dominio (ver figura 1b).

BDAM tiene la particularidad de que se le asigna a cada región triangular una TIN (ver figura 1b) con una cantidad constante de triángulos (unos pocos cientos) que aproxima la región contenida por esta. Cada nodo hoja representa

una triangulación de los puntos originales del modelo digital de elevación, mientras que los nodos intermedios se obtienen mediante algoritmos de simplificación de mallas triangulares a partir del nivel hoja o mediante refinamiento a partir del nivel raíz. A cada nodo se le asigna el valor del error espacio-objeto de la aproximación con respecto al MDE original y una esfera delimitadora que abarca en su interior todos los puntos de la TIN del nodo. Una de las formas más comunes de hallar este error es calculando la máxima diferencia en altura entre la TIN aproximada y la superficie original correspondiente a la misma región del nodo.

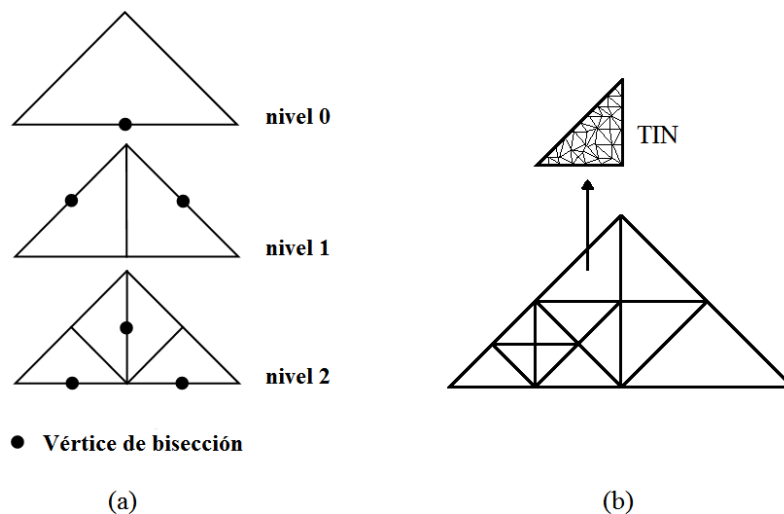


Figura 1. (a) Jerarquía del *bintree* de triángulos rectángulos definida mediante la bisección de la hipotenusa (b) Una partición consistente a partir de varias bisecciones en BDAM, cada nodo del *bintree* representa una red irregular de triángulos.

La selección del nivel de detalle se realiza a partir de la métrica de error espacio-pantalla, una cuota superior de este error se obtiene al medir el tamaño aparente en la pantalla de una esfera con centro en el punto más cercano del volumen delimitador del nodo al punto de visión de la cámara y con radio igual al error espacio-objeto del mismo. Un nodo necesita ser refinado (sustituido por sus dos hijos) si el error espacio-pantalla sobrepasa un umbral predefinido. Para lograr una partición multi-resolución consistente, se impone la restricción de que los nodos pertenecientes a un mismo diamante tienen que compartir el mismo error espacio-objeto y volumen delimitador.

### Tratamiento de los valores de altura nulos de los MDE en la construcción de BDAM

Los valores nulos en los modelos digitales de elevación indican la ausencia de altura para un punto específico (Easson y Robinson, 2001). Otra causa por la que aparecen valores nulos en el MDE es cuando las dimensiones del mismo no son cuadradas ni de la forma  $2^k + 1 \times 2^k + 1$ , incumpliendo las restricciones impuestas por varios modelos multi-resolución (por ejemplo, el *quadtree* restringido, ROAM, BDAM) y se rellena la matriz de altura con valores nulos

para cumplir dicho requerimiento. Los valores nulos suelen ser representados con alturas muy distantes del resto (en general -9999 o -32766), por lo que pueden generar inconsistencias durante la simplificación de los parches del modelo BDAM y durante el cálculo de la información asociada a los mismos (error espacio-objeto, volumen delimitador). Una alternativa es la modificación de este valor por uno o la interpolación de varios de los valores no nulos más cercanos, sin embargo, cuando interactúan múltiples capas de información geográfica en una misma escena estas modificaciones pueden afectar la visualización íntegra del fenómeno que se intente representar, un ejemplo de esto es la utilización de modelos digitales de elevación para la simulación hidrológica sobre terrenos (Domingos da Silva et al. 2014).

En este trabajo se realiza el mismo proceso *bottom-up* de “marca, mezcla y simplifica” para la construcción de los parches del modelo BDAM que se describe en (Cignoni et.al, 2003). Durante el pre-procesamiento de un MDE con valores de altura nulos se descartan los vértices, triángulos y aristas asociados a estos valores en todos los parches, la eliminación de los mismos conlleva a que las TINs obtenidas para la aplicación del algoritmo de simplificación en los diamantes tengan vértices frontera no necesariamente en los bordes de los parches (ver figura 2).

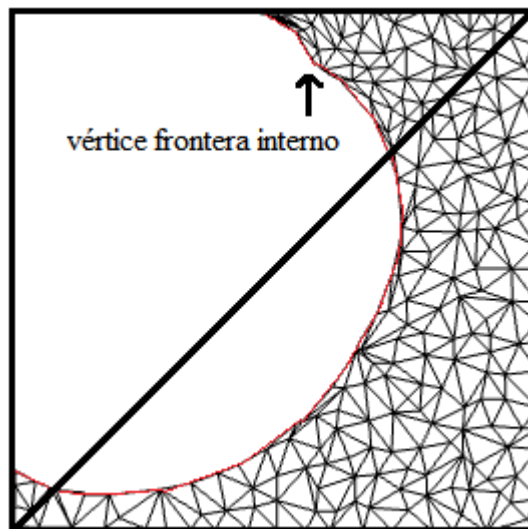


Figura 2. Dos parches que conforman un diamante, la región no triangulada dentro de los parches representa los valores nulos, los vértices con datos no nulos adyacentes a la región no triangulada se consideran vértices frontera.

Para lograr que los vértices frontera internos se simplifiquen en conjunto con el resto de los vértices se utilizó el algoritmo SESIMP (Hussain, 2013) con el fin de dirigir la simplificación de los parches en vez del algoritmo basado

en matrices cuadráticas (QEM<sup>3</sup>) que se utilizó en (Cignoni et.al, 2003). SESIMP hace uso de la variación del campo normal como métrica para asignarle un valor de importancia a cada vértice y realiza colapsos media-arista a través de un algoritmo ávido para simplificar la geometría de la TIN. La principal utilidad de este algoritmo para el presente trabajo es la definición de un parámetro escalar  $\lambda$  que permite controlar el nivel de preservación de las fronteras. Otra de las ventajas es que el colapso media-arista no modifica la posición de los vértices de la malla triangulada original.

El cálculo del error espacio-objeto en un parche mediante la máxima diferencia en altura con respecto a la malla original solo contempla las diferencias en el eje  $z$  y no en el resto de los ejes, por lo que no refleja de manera total los cambios realizados a la malla a través del colapso de un vértice frontera. Esto se evita sumándole al error espacio-objeto de un parche la distancia máxima entre los puntos fronteras internos del parche y los puntos frontera internos de la malla original correspondiente a la región cubierta por el parche.

### **Visualización del agua usando la técnica de Mapeo de Textura de Proyección.**

La idea principal de esta técnica es proyectar la geometría en 3D a un mapa de textura 2D de igual forma que se proyecta la geometría en 3D al monitor en 2D (de la Fuente, 2011). En el presente trabajo se proponen cuatro fases para la generación de la reflexión y refracción en el agua.

1. Captura de la textura de reflexión y refracción.
2. Creación de los *shaders*<sup>4</sup>.
3. Recopilación y transmisión de los datos al GPU.
4. Visualización.

La **primera fase** del proceso está basada en el algoritmo básico para superficies planas propuesto en (Kilgard, 1999), pero usando una modificación descrita en (Jensen y Goliás, 2001). Esta fase está dividida en dos pasos muy similares que son la captura de las texturas de reflexión y refracción. Para la captura de la textura de reflexión, antes de mostrarse en pantalla un fotograma, se obtiene la geometría de todas las capas excepto las de agua haciendo uso para esto del modelo BDAM. Luego se aplica la técnica *Recorte de cara* (Vega, 2013), recortando de esta forma la cara delantera para que, al reflejarse la imagen, desde el punto de visión de la cámara no se obstruya la vista reflejada que se observa por la parte trasera de la geometría. El mecanismo que se usará para dibujar los objetos de forma reflejada

---

<sup>3</sup> QEM del inglés *quadric error matrices*

<sup>4</sup> Los *shaders* son programas que permiten sustituir el procesamiento estándar o por defecto de los datos de una escena que realiza la GPU.

será el de multiplicar el componente  $z$  de cada vértice por  $-1$  cumpliendo así con la ley de reflexión. Con esto se logra una ubicación exacta de la vista virtual de reflexión. La figura 3 muestra con más claridad lo planteado en la idea anterior, desde el punto de visión de la cámara se observa la vista reflejada (en verde) con respecto a la superficie del agua multiplicando la  $z$  por  $-1$ .

De forma análoga, sin mover la cámara se obtiene la imagen con todos los objetos excepto el agua y se mapean en la textura de refracción.

En la **segunda fase** se crean, compilan y unen los *Vertex*, *Fragment* y *Geometry Shaders* que realizarán sus funciones en la GPU.

En la **tercera fase** se recopila la información a utilizar en la visualización de las capas de agua, como pueden ser, entre otros: posición de cada vértice, matriz de proyección, matriz de vista y modelo, textura de reflexión, textura de refracción.

Los datos recopilados se envían al GPU para utilizarse por los *shaders* en el proceso de visualización del agua.

La **cuarta fase** perteneciente a la visualización del agua se divide en varias etapas ejecutándose una en el *Vertex Shader*, otra en el *Fragment Shader* y de ser posible se utiliza el *Geometry Shader* para mejorar la calidad algunos recortes que son necesarios.

En el *Vertex Shader* se calculará el color inicial del agua antes de tener reflexión y refracción. Este color se calcula teniendo en cuenta la iluminación de la escena. Luego se calculan las coordenadas para acceder más tarde a la textura de reflexión y de refracción en el *Fragment Shader*. Es válido aclarar que como el *Vertex Shader* reemplazará esta funcionalidad en el *Pipeline* de OpenGL este debe realizar todas las operaciones necesarias de transformación de los vértices.

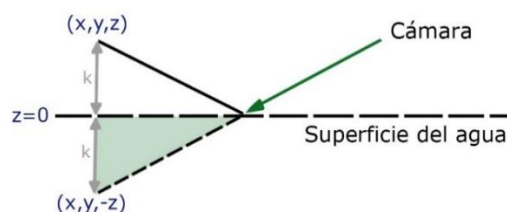


Figura 3. Vista reflejada de la imagen respecto al plano de la superficie del agua.

En el *Fragment Shader* las texturas de reflexión y refracción se usan para calcular el color de cada píxel del agua teniendo en cuenta varias leyes físicas como la de Snell. El color de un píxel será igual al color de reflexión



interpolado linealmente con el resultado de interpolar linealmente también el color del agua y el color de refracción, teniendo en cuenta el coeficiente de reflexividad calculado con las ecuaciones (1) y (2). Estas ecuaciones brindan una aproximación bastante exacta al coeficiente de reflexividad de Fresnel y fueron propuestas en (Schlick, 1994). Esta aproximación ofrece unos resultados razonablemente precisos a un menor costo computacional (de la Fuente, 2011). En las ecuaciones  $\theta_i$  es el ángulo de incidencia de la luz y la variable  $n$  es el índice de refracción del medio, en este caso el del agua.

$$Rf(\theta_i) = Rf(0^\circ) + (1 - Rf(0^\circ)) (1 - \cos \theta_i)^5 \quad (1)$$

$$Rf(0^\circ) = \left(\frac{n-1}{n+1}\right)^2 \quad (2)$$

## Resultados y discusión

Como prueba de conceptos se implementó una biblioteca en C# y se realizaron un conjunto de experimentos en una configuración de hardware con las siguientes características: PC DualCore Intel Core i3, 3.30 GHz, con una memoria RAM de 8GB y una tarjeta gráfica NVIDIA GeForce GTX 750 Ti. La resolución de la pantalla utilizada fue de 1380 x 768 y el umbral de error espacio-pantalla para todos los juegos de datos fue de 1 pixel.

Tabla 1 FPS y MTPS para los distintos juegos de datos utilizados.

MDE	Orden del MDE	Tamaño de la textura (MB)	FPS	MTPS
Puget Sound	18692 x 16406	219	32	83
Monai Valley	Dos capas de 3921 x 2431	escala de colores	75	384

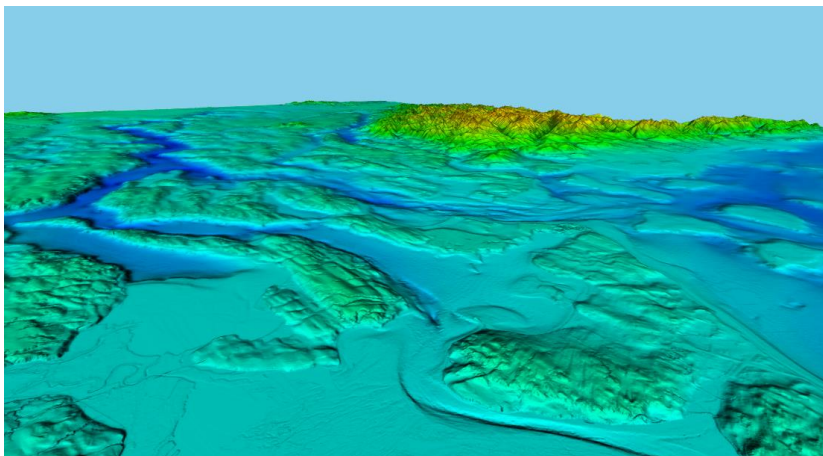


Figura 4. Aproximación (alrededor de 1 millón de triángulos) del juego de datos *Puget Sound* (en el orden de 18692 x 16406) con información batimétrica generada por el modelo BDAM a 1 pixel de error espacio-pantalla.

Se utilizaron los juegos de datos *Puget Sound* (FINLAYSON, 2005) y *Monai Valley* (MATSUYAMA, 2001), este último representando 2 capas, una de agua y otra de terreno, ambas con elevadas dimensiones (alrededor de 9.5 millones de puntos cada una). Se tuvieron en cuenta las variables tramas por segundo (FPS<sup>5</sup>) y millones de triángulos por segundo (MTPS) que brindan un indicador de la velocidad del proceso de visualización. Para extraer estos valores se realizó un recorrido de variada dirección en cada uno de los juegos de datos representados.

En la figura 4 se muestra una vista aérea del juego de datos *Puget Sound* con un umbral de error espacio-pantalla de 1 píxel. Las superficies representadas en estos juegos de datos son de alta variedad y complejidad, debido a que estas incluyen información batimétrica, lo que demuestra la capacidad del modelo BDAM para obtener de manera eficiente (ver tabla 1) aproximaciones de alta calidad en terrenos con superficies complejas y de gran tamaño (306.7 millones de puntos de puntos aproximadamente). Las imágenes presentadas en la figura 5 (a) y (b) muestran los resultados con respecto a la calidad de la representación del agua, donde se evidencia el realismo que proporciona el uso de las propiedades físicas de reflexión y refracción mediante la técnica de Mapeo de Textura de Proyección, otro elemento a tener en cuenta es que la aplicación de esta técnica no tiene altas repercusiones en la velocidad de visualización, tal y como se evidencia en la tabla 1.

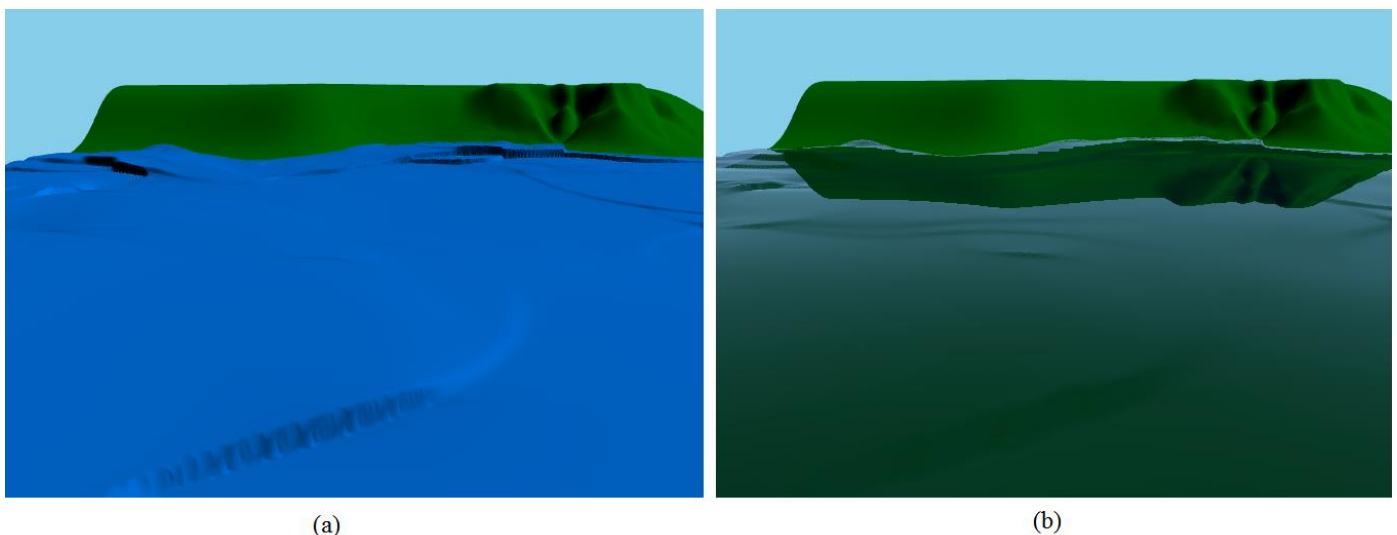


Figura 5. Juego de datos *Monai Valley* de dos capas, una con información de terreno e información batimétrica, y la otra con superficie de agua, (a) usando colores sólidos, (b) aplicando propiedades de reflexión y refracción de la luz.

<sup>5</sup> FPS: del inglés *frames per second*

## Conclusiones

En este trabajo se mostraron un conjunto de técnicas para lograr la visualización eficiente y realista de información geográfica a través de modelos digitales de elevación. La utilización del modelo BDAM para la representación multi-resolución de los MDE permite explotar las capacidades de los GPUs modernos a través del *rendering* por parches de triángulos, además de proveer un control del nivel de detalle a través de la métrica de error espacio-pantalla. Tal y como se evidencia en los experimentos, la integración de BDAM con la técnica de mapeo de textura de proyección permitió dotar de las propiedades de reflexión y refracción de manera eficiente a superficies de agua que interactúan con superficies de terreno, ambas representadas por modelos digitales de elevación con elevadas cantidades de puntos. Los valores nulos de un modelo digital de elevación, además de indicar la ausencia de valor, también permiten representar superficies con formas no necesariamente rectangular, como es el caso de ríos, lagos, fondos marinos, entre otros. El algoritmo de simplificación de mallas trianguladas SESIMP utilizado para la construcción del modelo BDAM sobre un MDE con valores nulos, permitió hacer un tratamiento diferenciado a los vértices frontera internos generados por la ausencia de altura que implican los valores nulos. Los resultados mostrados en los experimentos evidencian la efectividad de las técnicas utilizadas.

## Referencias

- BÖSCH, JONAS; GOSWAMI, PRASHANT; PAJAROLA, RENATO. RASTeR: Simple and efficient terrain rendering on the GPU. Eurographics Areas Papers, 2009, p. 35-42.
- CIGNONI, PAOLO, et al. BDAM—Batched Dynamic Adaptive Meshes for high performance terrain visualization. En Computer Graphics Forum. Blackwell Publishing, Inc, 2003. p. 505-514.
- COHEN, JONATHAN D.; MANOCHA, DINESH. Model simplification. Visualization Handbook, Elsevier Butterworth-Heinemann, Oxford, 2005, p. 393-411.
- DARLES, EMMANUELLE, et al. A survey of ocean simulation and rendering techniques in computer graphics. En Computer Graphics Forum. Blackwell Publishing Ltd, 2011. p. 43-60.
- DE FLORIANI, LEILA; KOBELT, LEIF; PUPPO, ENRICO. A survey on data structures for level-of-detail models. En Advances in multiresolution for geometric modelling. Springer Berlin Heidelberg, 2005. p. 49-74.

DE LA FUENTE, Antonio C. Fluids real-time rendering. Tesis de maestría, Universitat Politècnica de Catalunya, Barcelona, 2011.

DOMINGOS DA SILVA, DIVALDO; ESCARTIN SAULED, EMILIO RICARDO y JIMENEZ SAEZ, GERARDO CRESCENCIO. Modelo digital de elevaciones de una zona urbana para la simulación hidrológica. RIHA [online]. 2014, vol.35, n.2 [citado 2016-06-10], pp. 123-137. Disponible en: <[http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1680-03382014000200009&lng=es&nrm=iso](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1680-03382014000200009&lng=es&nrm=iso)>. ISSN 1680-0338.

DUCHAINÉAU, MARK, et al. ROAMing terrain: real-time optimally adapting meshes. En Proceedings of the 8th Conference on Visualization '97. IEEE Computer Society Press, 1997. p. 81-88.

EASSON, GREG; ROBINSON, HAROLD. Assembling DEM Files for Watershed Analysis. *ArcUser July-September 2001*, 2001.

EVANS, WILLIAM; KIRKPATRICK, DAVID; TOWNSEND, GREGG. Right-triangulated irregular networks. *Algorithmica*, 2001, vol. 30, no 2, p. 264-286.

FINLAYSON D.P. Combined bathymetry and topography of the Puget Lowland, Washington State. University of Washington, 2005 (<http://www.ocean.washington.edu/data/pugetsound/>)

HOPPE, HUGUES. View-dependent refinement of progressive meshes. En Proceedings of the 24<sup>th</sup> annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1997. p. 189-198.

HUSSAIN, MOHAMMAD. Efficient Decimation of Polygonal Models Using Normal Field Deviation. *Computing and Informatics*, 2013, vol. 32, no 3, p. 527-547.

JENSEN, LASSE S.; GOLIAS, ROBERT. Deep-water animation and rendering. En: Game Developer's Conference. Proceedings of the Game Developer's Conference Europe. Disponible en: [[http://www.gamasutra.com/gdce/2001/jensen/jensen\\_01.htm](http://www.gamasutra.com/gdce/2001/jensen/jensen_01.htm)]. 2001.

KILGARD, MARK J. Improving shadows and reflections via the stencil buffer. En: Game Developers Conference. Advanced OpenGL Game Development. <https://developer.nvidia.com/sites/default/files/akamai/gamedev/docs/stencil.pdf>. 1999.

LEVENBERG, JOSHUA. Fast view-dependent level-of-detail rendering using cached geometry. En Visualization, 2002. VIS 2002. IEEE. IEEE, 2002. p. 259-265.

MATSUYAMA, MASAFUMI; TANAKA, HIROYOSHI. An experimental study of the highest run-up height in the 1993 Hokkaido Nansei-oki earthquake tsunami. En Proceedings of the US National Tsunami Hazard Mitigation Program Review and International Tsunami Symposium (ITS). 2001. p. 879-889.

PAJAROLA, RENATO. Large scale terrain visualization using the restricted quadtree triangulation. En Visualization'98. Proceedings. IEEE, 1998. p. 19-26.

PAJAROLA, RENATO; GOBBETTI, ENRICO. Survey of semi-regular multiresolution models for interactive terrain rendering. The Visual Computer, 2007, vol. 23, no 8, p. 583-605.

RODGERS, ALEXANDER P. CUDA RAY TRACER. Tesis de B.Sc. (Hons), Universidad de Brighton, Brighton, 2014.

SANDER, PEDRO V., et al. Texture mapping progressive meshes. En Proceedings of the 28<sup>th</sup> annual conference on Computer graphics and interactive techniques. ACM, 2001. p. 409-416.

VEGA, GENDRYS E. Algoritmo de recortes y de niveles de detalles para el incremento de la velocidad de visualización de modelos 3D en dispositivos de bajo coste. 3C TIC, 2013, vol. 2, no 4.

WHITTED, T. An improved illumination model for shaded display. ACM, 1980, 23: p. 343-349.