

# Una aplicación para la navegación en tiempo real sobre grandes modelos topográficos.

L. Hernández, J. Taibo, A. Seoane  
Universidad de A Coruña

e-mail: [lahernandez@iccp.udc.es](mailto:lahernandez@iccp.udc.es), [jtaibo@udc.es](mailto:jtaibo@udc.es), [ynot@udc.es](mailto:ynot@udc.es)

## Resumen

En esta ponencia se describe una aplicación informática que permite el sobrevuelo sobre grandes modelos topográficos, particularizada para una base de datos correspondiente a la Comunidad Autónoma Gallega.

El sistema ha sido diseñado para permitir una representación en tiempo real independientemente del tamaño de la base de datos, desligando el rendimiento gráfico del sistema del tamaño de la malla a visualizar. Para ello, el sistema desarrolla un proceso asíncrono de acceso a la base de datos respecto a su visualización, y utiliza técnicas de mallado con nivel de detalle variable y texturizado en multiresolución.

**Palabras clave:** tiempo real, simulador, topografía, *clip-mapping*.

## 1 Introducción

Los sistemas de visualización tridimensional de modelos topográficos han sido usados en los años precedentes fundamentalmente para dos finalidades:

- Análisis topográficos, interpretación del terreno o de información territorial, sistemas GIS, o teledetección.
- Soporte contextual para sistemas de simulación o Simuladores de Vuelo, Guiado de Misiles, etc.

En el primero de los casos, la visualización del territorio constituye una herramienta para la interpretación de los datos. Estos son los auténticos protagonistas, por lo que la calidad, inmediatez e interactividad del interface se presentan en un papel secundario frente a la rigurosidad de los datos mostrados. La visualización no tiene por qué ser en tiempo real y el acceso a la base de datos no tiene por qué ser instantáneo. Este tipo de sistemas se presenta como una forma de estudio de datos relacionados con la geografía, más que como una experiencia de sobrevuelo.

En el segundo de los casos, la atención se focaliza en el correcto funcionamiento del simulador, en su adecuada correspondencia con el dispositivo móvil que se pretende emular, tal como una aeronave o un proyectil. La base de datos sobre la que se produce la simulación, puede ser reducida o simplificada en beneficio de la interactividad en la simulación. El tiempo real ha de estar necesariamente presente, la simulación del funcionamiento del dispositivo de vuelo ha de ser perfecta, y la calidad de la base de datos se reduce si es necesario para este fin.

Unificar en un sistema las características de uno y otro caso – gran volumen de base de datos a visualizar, y tiempo real en la simulación -, presenta serios problemas, por cuanto se exige del sistema un tiempo de respuesta extremadamente corto para hacer frente al conjunto de operaciones que conducen a la visualización de un gran volumen de información.

El ejemplo que aquí se presenta permite la navegación en tiempo real sobre un área geográfica de más de 40.000 Km<sup>2</sup>, e involucra una malla total de más de 1 millón de polígonos, texturizada utilizando un mapa general de 16384 x 16384 pixels, cantidades que sobrepasan la capacidad de visualización de los sistemas gráficos disponibles.

## **2 Objetivos**

El presente trabajo se llevó a cabo como encargo para la realización de un sistema de representación de la geografía gallega, de cara a obtener aplicaciones de tipo divulgativo y promocional, no obstante, tras su realización surgieron una miríada de otras aplicaciones susceptibles de ser implementadas en el sistema, como el planeamiento territorial, el diseño de carreteras y ferrocarriles, el análisis geológico, o estudios agrícolas y forestales.

## **3. Proceso de desarrollo**

Las diferentes etapas que condujeron a la realización del sistema pueden resumirse en:

- 1- Diseño del Sistema.
- 2- Obtención y tratamiento de datos existentes. Creación de la base de datos.
- 3- Desarrollo del navegador.

### 3.1 Diseño del sistema

A la hora de establecer los requisitos que debería cumplir el sistema se establecieron las siguientes consideraciones:

#### Hardware.

- Debería poder ser implementado en una estación SGI ONYX 2 Reality. Ello condicionó la cantidad máxima de polígonos y número máximo de pixels de textura que el sistema podría visualizar en tiempo real.
- La visualización debería realizarse con un ratio de refresco no inferior a 30 fotogramas/segundo para evitar saltos o temblores en la imagen.
- Para la programación se utilizarían las librerías IRIS Performer de Silicon Graphics.

#### Calidad de la representación.

El sistema debería mostrar una alta calidad en la representación del territorio. Tal calidad vendría del detalle de la geometría y de las texturas utilizadas. El detalle podría ser variable en función de la distancia del observador a la zona observada.

#### Precisión de la geometría.

Debería contarse con una densidad de datos altimétricos no inferior a 1 cota cada 200 m. La experiencia previa indicaba que tal resolución es suficiente para percibir adecuadamente el territorio desde el aire a la distancia prevista, si se cuenta con texturas que ayuden a la definición del terreno.

#### Precisión de las texturas.

Para el texturizado de un área geográfica de tal extensión se estimó como conveniente acudir a información fotográfica procedente de satélite. Los satélites comerciales a los que los autores pudieron tener acceso (LANDSAT y SPOT) ofrecen diferentes prestaciones en sus imágenes. LANDSAT proporciona una resolución de 1 pixel cada 30 m e información de color en la banda del espectro visible. SPOT permite una resolución mayor, 1 pixel cada 10 mts., pero genera imágenes monocromas.

Aprovechando que las imágenes procedentes de ambos satélites se hallan perfectamente georeferenciadas, es posible obtener imágenes mediante tratamiento digital, que posean la información de luminancia del SPOT y de crominancia de LANDSAT, obteniéndose así una resolución mixta de 10 m/pixel en luminancia y 30 m/pixel en crominancia. Esta resolución mixta fue la usada finalmente en el trabajo.

### Tratamiento particularizado de grandes núcleos urbanos.

Dado que un núcleo urbano del tamaño de una gran ciudad, presenta una volumetría perfectamente capaz de ser reconocida dentro de una visión más general del territorio en que se ubica, se decidió, que tanto el volumen de la ciudad, definido al nivel de manzana de forma aproximada, como su aspecto visual deberían ser tomados en cuenta.

Para ello deberían ser modeladas dichas manzanas, ubicadas en su posición geográfica, y la ciudad entera debería ser mapeada mediante el uso de fotografía aérea.

### Navegación sobre la base de datos.

La inclusión de todos los datos referidos anteriormente en un solo modelo se presentaba a todas luces inadecuada para su visualización en tiempo real, por cuanto sobrepasaba ampliamente las capacidades del sistema gráfico disponible, siendo aún este uno de los más potentes del mercado.

La solución pasaba por tanto por la división de la base de datos en piezas más pequeñas, de forma que el sistema presentase la pantalla sólo la zona que el observador tuviese delante hasta cierta distancia máxima, cuya magnitud vendría dada por las prestaciones gráficas del sistema.

Dado que no se consideró la esfericidad del planeta, el horizonte debería ser ocultado mas allá de la distancia máxima visible mediante el uso de niebla.

Así pues, el territorio de Galicia fue dividido en 20 secciones, que a su vez se dividieron en 16 subsecciones cada una, según se muestra en la figura 1.

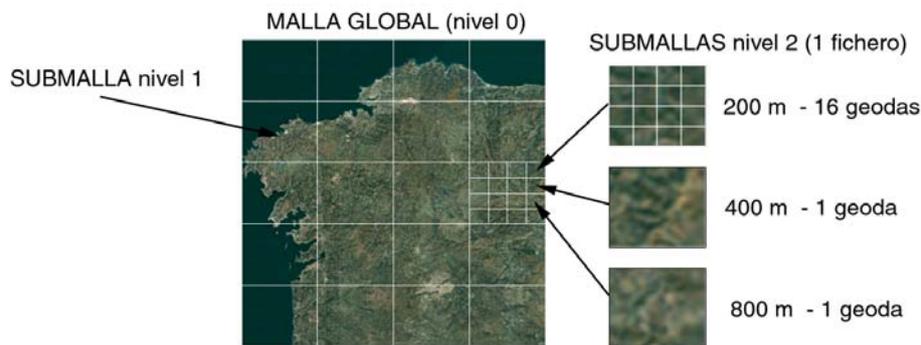


Figura 1. *División de la base de datos topográfica.*

El sistema debería visualizar tan solo un conjunto de zonas simultáneamente. A medida que el observador se moviese, nuevas zonas irían agregándose a la malla, a la vez que el sistema descargaría las que dejaran de ser visibles.

Tal proceso de carga y descarga podría traducirse en paradas que interrumpirían la navegación fluida, por lo que dicha lectura de la base de datos debería hacerse de forma asíncrona a su visualización. Precargando las áreas de inmediata visualización antes de que éstas apareciesen en el frustrum visible.

Se contempló también el uso de geometrías de nivel de detalle variable en función de la distancia al espectador, generándose 3 modelos de diferente resolución geométrica para cada zona del territorio.

En cuanto a las texturas, se contemplarán dos soluciones posibles para su aplicación.

#### 1- Asignación de texturas por malla.

La solución inmediata a la asignación de texturas pasó por asignar un mapa independiente a cada submalla para cada versión de sus diferentes resoluciones geométricas. Ello generaba la limitación de que la suma total de las texturas asignadas a las mallas visibles en un momento dado debería ser inferior a la memoria de texturas disponible en el hardware, incluso aunque dichas mallas ocupasen zonas pequeñas de la pantalla. Igualmente limitaba la resolución máxima de las texturas aplicadas a las mallas que se encontrasen en primer plano.

#### 2- Asignación de texturas por clip-mapping.

La técnica de clip-mapping permite asignar texturas multiresolución a cualquier malla dada, permitiendo asignar mayor o menor definición de la textura aplicada en función de la visual del observador. Esta técnica se explicará con más detalle en los capítulos siguientes, y fue la que se utilizó finalmente.

### **3.2 Obtención y tratamiento de los datos existentes. Creación de la base de datos**

La problemática de los modelos geográficos extensos fue abordada basándose en dos estrategias: la división de la malla topográfica en fragmentos de un tamaño manejable y el uso de niveles de detalle. Jugando con estos dos enfoques se pretendía lograr un equilibrio entre los factores involucrados: el detalle de la imagen y la suavidad del movimiento frente a la carga computacional del sistema, que venía limitada por los recursos disponibles.

Para la representación de la geografía gallega con el nivel de detalle adecuado era necesario contar con datos de diferentes tipos y procedentes de diversas fuentes. Debido a sus características particulares, se optó por tratar la geometría de manera independiente de las texturas, de forma que en cada caso se pudiese

gestionar de la forma mas apropiada sin limitaciones y sin que un aspecto interfiriese en el otro.

## **Datos altimétricos**

El Instituto Geográfico Nacional cuenta con varias colecciones de datos de altimetría del territorio nacional. Para el presente caso se utilizó la colección MDT 200 por dos razones: Es la única colección completa de esta fuente que abarcase toda la geografía gallega, y tiene el nivel de detalle adecuado para la correcta percepción del terreno para la altura de sobrevuelo prevista (en torno a 3.000 m).

Dicha Institución facilitó sus datos en coordenadas UTM para el huso 29, pudiéndose obtener la misma proyección para las imágenes de satélite y las fotografías aéreas. La proyección UTM permite considerar el territorio como plano por cuanto meridianos y paralelos se proyectan como ortogonales. Las coordenadas UTM en la zona de estudio eran del orden de millones de metros. Estas coordenadas fueron reducidas en los modelos finales para trabajar con valores más cercanos al cero debido a problemas que se encontraron al trabajar a escalas pequeñas dentro de la geografía.

Sin embargo, dichas colecciones de datos se encuentran organizadas según una zonificación diferente a la deseada, con áreas de solape variables entre zona y zona. Fue por tanto necesario crear programas específicos para extraer los datos de los ficheros ASCII provenientes del IGN, y generar la colección de submallas a diferentes resoluciones usadas en el sistema que aquí se presenta. Para ello se desarrollaron rutinas específicas sobre programas CAD que aportaron una retroalimentación visual muy importante para el control de las tareas de preproceso de la geometría

Con las cotas contenidas en los ficheros MTD200 se generaron unas mallas de diferentes tamaños y posiciones que formaban una matriz no cuadrada. Esta tarea se realizó con el programa a medida. A partir de ellas se obtuvieron nubes de puntos correspondientes a las diferentes submallas de nivel 1 (ver figura 1) que se reordenaron en X e Y para construir las submallas de 256x256 facetas de 200m x 200m. Se optó por trabajar con mallas cuyos lados tuviesen de tamaños potencias de dos, por su facilidad a la hora de realizar divisiones sucesivas en fragmentos del mismo tamaño. Las submallas generadas en este punto (de nivel 1) se tomaron como elemento básico para la subdivisión y organización de la base de datos.

La división de las mallas es un punto crítico en el rendimiento del sistema. En primer lugar, una malla de gran tamaño supone una elevada carga de proceso para

el hardware de cálculo de geometría, por lo que se debe subdividir en fragmentos más pequeños, tanto para poder gestionar la carga y descarga de forma eficiente como para mejorar el proceso de CULL (descrito más adelante) en el *pipeline* gráfico. Por otra parte, un exceso en el extremo contrario llevaría a trabajar con caras sueltas. Esto supondría elevar la carga en gran medida tanto para el proceso de CULL, que tendría que evaluar cada cara por separado como por incrementar enormemente el volumen de cálculo a la hora de manejar la geometría

En conclusión, para la división de las mallas del terreno que maneja el sistema, se deberían tener en cuenta los aspectos siguientes:

- El tamaño de los fragmentos debe ser el mayor posible para reducir el coste del manejo de geometría.
- El tamaño de los fragmentos no debe ser muy elevado para tener una granularidad mas fina a la hora de gestionar la carga/descarga de modelo en la escena.
- El proceso de CULL se debe equilibrar con el de DRAW, especialmente en sistemas multiprocesador. Un tamaño de fragmento excesivamente pequeño cargaría más el proceso de CULL, mientras que un fragmento muy elevado supondría una carga para el proceso de DRAW.
- Un tamaño de fragmento muy pequeño en ficheros independientes provoca que los accesos a disco sean más ineficientes. Un tamaño muy grande resulta más eficiente en cuestión de velocidad de transferencia de disco, pero la gestión de la escena sufre picos de carga mayores que podrían afectar a la fluidez de la simulación.

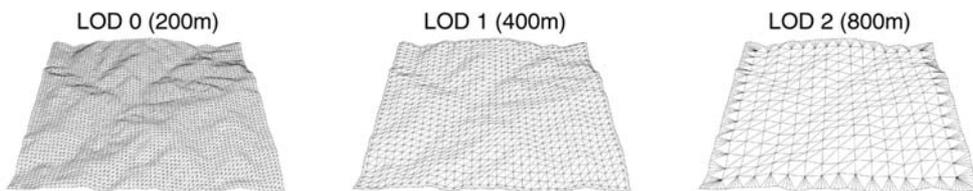


Figura 2. Submallas correspondientes a los diferentes niveles de detalle.

En la implementación realizada en el presente trabajo se tomó una organización según los niveles de detalle manejados. Partiendo de las mallas obtenidas a partir de los ficheros MDT200 del IGN, se tomó ese nivel de detalle como nivel 0, para obtener los niveles de detalle sucesivos correspondientes una cota cada 400 m (Nivel 1)- y a una cota cada 800 m (Nivel 2)

El primer problema al que hubo que enfrentarse al trabajar con estos niveles de detalle fue la unión de mallas de distintos niveles. Cada malla posee la mitad de vértices en los bordes que la del nivel superior, quedando "huecos" en la unión entre ellas. Una solución al problema consiste en realizar una adaptación de las mallas de cada nivel inferior en los bordes, a modo de "costura". De esta forma, las mallas de 400 tienen en los bordes un vértice cada 200 metros, para encajar perfectamente con las mallas de 200. A su vez, las mallas de 800 también tienen un vértice cada 200m puesto que deben encajar correctamente con las de los dos niveles superiores.

Todas las submallas de nivel 1 se subdividieron en 16 (4x4) submallas de nivel 2, que se almacenaron cada una en un fichero independiente. Estos ficheros son cargados por el navegador en el momento adecuado.

En el caso de las submallas de 200, dado su mayor volumen de geometría, se organizaron en varias geodas (mallas de triangulos compartiendo vertices). Concretamente, cada submalla de 200 (64x64 vanos) se dividió en 16 geodas de 16x16 vanos.

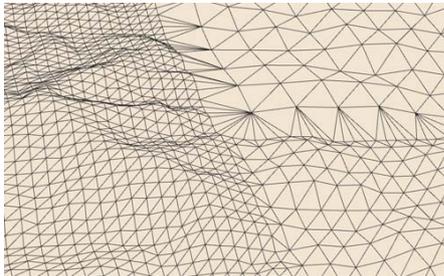


Figura 3. *Detalle de las uniones entre niveles de detalle diferentes.*

Las mallas de 400 y 800, dado su menor volumen de geometría y la complejidad añadida de la unión con las mallas de 200, se organizaron en una única geoda, evitando así una división excesiva de las mallas de geometría más sencilla y se simplifica la tarea de construir la unión con el nivel 200.

El mapeado de texturas se gestiona mediante una técnica independiente, como se describirá más adelante. Las coordenadas de mapeado que se asignan a cada vértice de las mallas se tomaron respecto a la imagen global de terreno correspondiente a la malla de nivel 0.

## **Datos fotográficos**

Para las texturas del terreno se partió de múltiples imágenes de satélite de toda la superficie de Galicia con una resolución de 10 m por pixel además de fotos aéreas adicionales de zonas de especial interés con 1 m por pixel de resolución.

Dado que las imágenes de satélite pueden considerarse como texturas preiluminadas, ello evita llevar a cabo cálculos de iluminación, lo cual incrementa la rapidez en la simulación.

Con las fotografías de satélite se realizó un tratamiento digital conducente a recomponer el mapa completo de Galicia. En algunos casos fue necesario un tratamiento complejo de los bordes para realizar un buen ajuste. Debido a las diferencias cromáticas entre las distintas fotografías de satélite se realizó un ajuste de histogramas para igualar los tonos. Para terminar el proceso se realizaron correcciones en el color de las diferentes imágenes. Como resultado se obtuvo una imagen de 10240x12800 pixels.

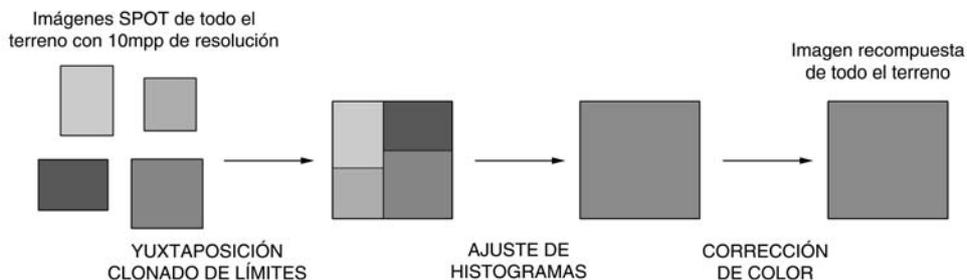


Figura 4. Preproceso de la imagen global del terreno.

Para las zonas de especial interés, se realizó otro tratamiento digital en el que se utilizaron las fotografías aéreas. Para empezar se *resampleó* la imagen global recompuesta hasta hacerla coincidir con la resolución de las imágenes aéreas (1 m por pixel). El proceso continuó difuminando los contornos de las imágenes de gran resolución para adaptarlas a la imagen global del terreno y se ajustó la tonalidad y el color. Finalmente se acoplaron las imágenes de detalle con la imagen global del terreno y se guardaron por separado las zonas de especial interés y la imagen global. Antes de almacenar la imagen global recompuesta y retocada se redujo a la resolución original que tenía antes del retoque.

Partiendo de la imagen global recompuesta y de las imágenes aéreas de gran detalle, ambas retocadas, se utilizó el programa ClipGen para crear la textura definitiva. El resultado fue una textura multiresolución de 16384x16384 texels con 15 niveles de detalle en toda la superficie y con 19 niveles en las zonas de especial interés, de las que se disponía de fotografías aéreas. La textura se almacenó en disco organizada por niveles de detalle y en fragmentos de 512x512 texels ajustándose así a los recursos hardware disponibles y a la técnica de mapeado multiresolución. Para mejorar el rendimiento del sistema las texturas se crearon con una profundidad de pixel de 16 bits, lo que permite mayor velocidad en el *rendering* y menos accesos a disco a la hora de cargarlas.

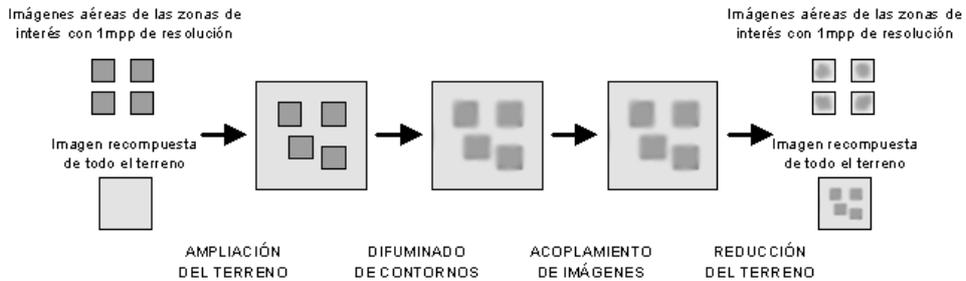


Figura 5. Proceso sobre las imágenes de las zonas de detalle.

### 3.3 Desarrollo del navegador

El sistema de visualización se basa en dos procesos paralelos y asíncronos, que llamaremos respectivamente de *simulación (SIM)* y de *gestión de base de datos (DBASE)*.

El proceso *SIM* es el proceso de simulación clásico, consistente en un bucle que se repite cada *frame*. Este proceso se compone de varias fases, siguiendo el funcionamiento del *pipeline* gráfico. Consideraremos tres fases dentro de dicho *pipeline*: *APP*, *CULL* y *DRAW*. En sistemas multiprocesador, estas fases se pueden realizar en procesos independientes para repartirlos entre las CPUs disponibles. En cualquier caso están fuertemente interrelacionadas.

El proceso *DBASE* se encarga de la gestión de los elementos de la estructura jerárquica de la escena (*scenegraph*), de forma que en cada momento el sistema trabaje únicamente con la zona que se está visualizando. También se encarga de utilizar en cada zona de la escena el nivel de detalle apropiado para lograr un equilibrio entre la calidad y el realismo de la visualización y la carga de proceso del sistema.

El proceso *SIM*, desde un punto de vista secuencial, se compone de tres bloques principales: *pre-frame*, *frame* y *post-frame*. En el *pre-frame* se realizan las operaciones críticas en el tiempo. Tales como la lectura de los dispositivos utilizados para controlar el movimiento, para seguidamente actualizar la posición de la cámara justo antes de entrar en la fase de *frame*, donde se realiza el *rendering* de la escena actual, iniciando el *pipeline* gráfico. El *post-frame* realiza otras tareas sin una importancia crítica en el tiempo. Desde el punto de vista del *pipeline* gráfico, los procesos *pre* y *post frame*, pertenecen a la fase *APP*.

La fase *CULL*, a partir de la salida de la fase *APP* recorre la estructura jerárquica del *scenegraph* y descarta los elementos que queden fuera del *frustum* visible. En este punto es donde cobra importancia la organización de las geodas en las mallas que se manejan, resultado de la fase de preproceso.

Esta fase se debe equilibrar con la siguiente (*DRAW*) para reducir el tiempo de *frame* lo máximo posible. Esto es especialmente importante en sistemas multiprocesador, puesto que dada la naturaleza del sistema, el tiempo de *frame* corresponderá al tiempo de la fase más larga dentro del *pipeline* gráfico.

La fase *DRAW*, generalmente la más larga, procesa las estructuras de la escena que no han sido descartadas previamente en el *CULL*. La velocidad de esta fase puede estar limitada por geometría o por llenado de pixel. En el caso de la geometría, son importantes tanto la optimización de los modelos como la buena gestión de niveles de detalle. En el caso del llenado de pixel, se puede reducir la resolución de la pantalla con la que se trabaja de forma fija o dinámica.

El proceso *DBASE* se basa en los datos de posición actual de la cámara para gestionar la configuración del *scenegraph* adecuándolo a las necesidades de la simulación. Las tareas de carga y descarga de geometría y texturas, así como la gestión de niveles de detalle se realizan en este proceso. Por este motivo, está íntimamente ligado con la organización de la base de datos del terreno, y por consiguiente con la fase de preproceso de la información. Resulta evidente que el diseño de estos aspectos se debe realizar conjuntamente.

Se aplica un proceso relativamente independiente a la geometría y las texturas del terreno. De esta forma se puede gestionar de una forma más inmediata la textura que la geometría, en donde no es necesaria una actualización tan frecuente. Además se pueden generar distinto número de niveles de detalle en uno u otro aspecto según sean necesarios. Estos procesos se describen a continuación.

## **Actualización de la geometría**

De la posición de la cámara, se pueden considerar varios parámetros para calcular la disposición de niveles de detalle en la escena, la posición en el plano x-y, la altura (z), la orientación en el plano x-y (*heading*) y el cabeceo (*pitch*).

El proceso *DBASE* guarda la posición y orientación de la cámara en la última actualización de la escena. En cada *frame* se comprueba la diferencia entre la posición actual y la de la última actualización en dos aspectos: la distancia en el

espacio y el ángulo *heading*. Si alguno de estos dos parámetros supera un umbral preestablecido, se realiza la actualización de las mallas.

La actualización se realiza basándose en la posición de un punto central con nivel de detalle máximo que llamaremos *centro de visión*. Este punto se calcula avanzando una determinada distancia en el plano x-y desde la posición de la cámara en la dirección en la que esta orientada.

El proceso *DBASE* recorre todas las submallas de nivel 2 de la base de datos. Para cada una de ellas, se calcula su distancia al centro de visión y en función de esta distancia, y los “niveles de corte”<sup>1</sup> de cada nivel de detalle, se obtiene el nivel adecuado para esa malla. Si el nivel es distinto del que tenía la malla en ese momento se descarga la que había (en caso de que hubiese alguna cargada) y se carga la nueva malla. Este recorrido secuencial puede llegar a suponer una carga elevada en bases de datos muy extensas; por ello el proceso de actualización se realiza sólo cuando se hace necesario.

## **Actualización de las texturas**

La mayor limitación a la hora de realizar simulaciones sobre terrenos se encuentra en la pequeña cantidad de memoria de texturas disponible en hardware, comparado con el enorme tamaño de las texturas que se requiere manejar, por tanto se necesita una técnica de mapeado que permita aprovechar al máximo este hardware.

En el presente sistema se utilizó la técnica de "*clip-mapping*" implementada en las librerías IRIS Performer de Silicon Graphics. *Clip-mapping* es una técnica de mapeado de texturas que permite el manejo de imágenes de gran tamaño con un bajo consumo de memoria, manteniendo la mayor velocidad y visualizando la máxima resolución disponible. Otra gran ventaja del *clip-mapping* es la posibilidad de poder manejar múltiples grupos de geometría con una sola textura, realizando su tratamiento de forma independiente.

La técnica de *clip-mapping* organiza la textura en niveles de detalle con una estructura piramidal, donde cada nivel tiene el doble de resolución que el anterior, siendo el nivel más alto el nivel de más detalle, y siempre potencia de dos. Esta técnica mantiene, si la carga del sistema lo permite, una zona limitada llamada *Clip Region* a la máxima resolución disponible.

---

<sup>1</sup> Denominamos niveles de corte a las distancias hasta las cuales se utiliza cada nivel de detalle.

La *Clip Region* es la zona de máximo interés de una *cliptextura* (o textura de *clip-mapping*). Todos los niveles de menor tamaño que la *Clip Region* son almacenados por completo en la memoria de texturas, por lo tanto siempre se tiene la textura completa en memoria, aunque sea con muy poco detalle. El tamaño de la *Clip Region* se elige en función de la calidad de imagen requerida, de la memoria de texturas que se desee consumir y del ancho de banda disponible entre los diferentes niveles de memoria.

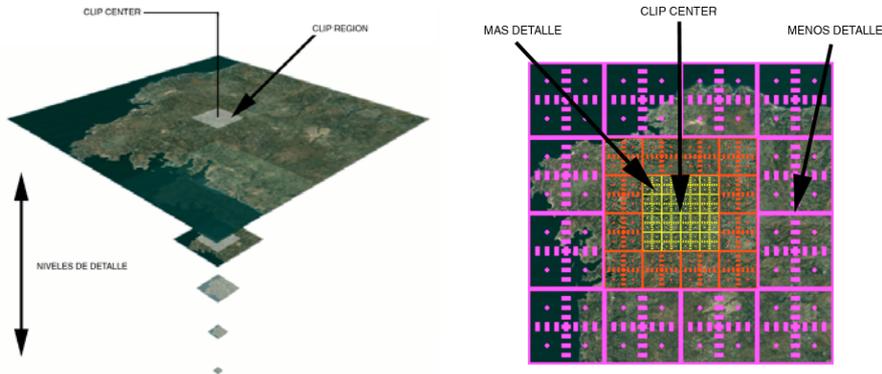


Figura 6. Técnica de *clip-mapping*.

La *Clip Region* se encuentra centrada en el *Clip Center* que, en coordenadas de textura, representa la posición donde se busca la máxima calidad que normalmente coincide con el punto más cercano de la *cliptextura* al punto de vista. Esta posición es actualizada en cada *frame*.

Al alejarse de la *Clip Region*, y por tanto del *Clip Center*, el nivel de detalle disminuye progresivamente. *Clip-mapping* mantiene, tanto en memoria principal como en memoria de texturas, una organización en malla de fragmentos de textura de igual tamaño llamados *Image Caches*, que son actualizados desde disco.

El sistema de navegación cuenta con una textura de 16384x16384 *texels* de toda la superficie de Galicia, por tanto hay un total de 15 niveles de detalle ( $2^{15}=16384$ ). Se emplea una *cliptextura* con una *Clip Region* de 1024x1024 *texels*, y con *Image Caches* de 512x512 *texels*, lo que permite aprovechar bien el ancho de banda en las transferencias de disco que son las que tienen un gasto temporal más elevado.

También se disponía de diversas zonas con mucho más detalle. Una de las ventajas que nos proporciona *clip-mapping* es que además permite tener niveles de

textura incompletos donde se pueden tener ciertas zonas con mucho más detalle que el resto sin necesidad de generar todo el nivel de textura (lo que en *clip-mapping* se denominan *insets*), que permite ahorrar espacio en disco, y mejorar la velocidad de la simulación al no tener que mantener una calidad tan elevada en todo momento. En la textura del terreno se añadieron 4 niveles incompletos con las imágenes aéreas a 1 m/pixel de que se disponía, con lo que el nivel de máximo detalle (aunque incompleto) llega a ser de 256Kx256K *texels* con 19 niveles de detalle. Los *insets* permiten tener un enorme detalle para algunas ciudades gallegas.

Sin embargo no todo son ventajas. Una limitación de *clip-mapping* es que el hardware no puede gestionar mas de 16 niveles de textura simultáneamente. Por tanto hubo que realizar una gestión dinámica de la "ventana" de niveles que en cada momento de la visualización debía utilizar el hardware. Para ello hubo que tener en cuenta el hecho de sobrevolar alguna zona de la que se disponía de gran detalle y ajustar los niveles de textura manejados por el hardware en función de la altura de vuelo y del *pitch* de la cámara. Cuando la cámara se acerca a los *insets* se aumenta el detalle de la simulación hasta alcanzar la máxima calidad posible.

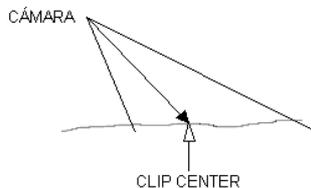


Figura 7. Ubicación del clip center.

Para la ubicación del *Clip Center* se calculó el punto de la textura mapeado sobre la geometría que coincide con el centro del plano de visión. Este calculo depende de la posición y de la orientación de la cámara.

## 5. Equipo de trabajo

En la realización del sistema que aquí se presenta intervinieron las siguientes personas del Grupo de Visualización en Ingeniería y Urbanismo de la ETS de Ingenieros de Caminos, canales y Puertos de la Universidad de A Coruña:

Dirección del proyecto: Luis Hernández

Programación: Javier Taibo, Antonio Seoane, Javier Muñoz, Alfredo Matesanz

Modelado: Manuel Meijide, David Caeiro

Tratamiento de imágenes: Jesús Acuña, David Pardo



Figura 8. - *Imagen de la aplicación sobre el área de A Coruña*

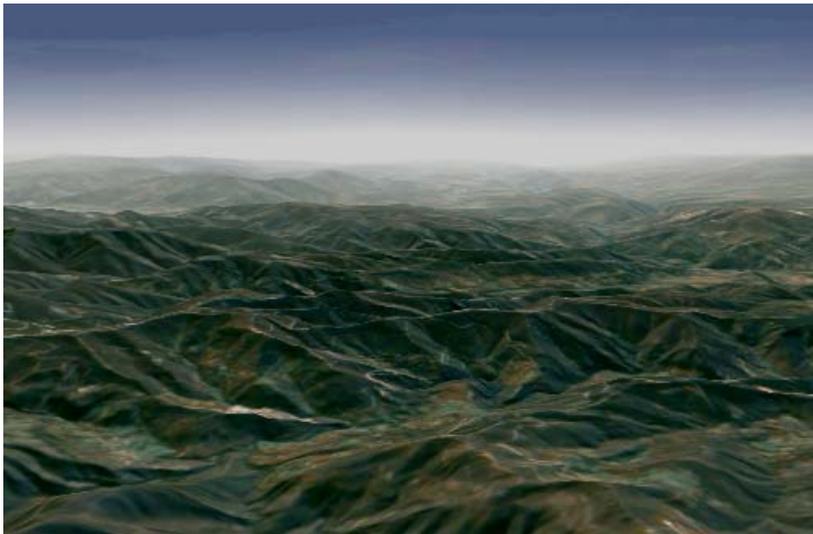


Figura 9.- *Imagen de la aplicación sobre el Macizo Galaico.*

## 6. Conclusiones y futuros desarrollos

La navegación en tiempo sobre grandes modelos topográficos es posible mediante el uso de una adecuada organización de la base de datos gráfica, basada principalmente en la aplicación de niveles de detalle geométricos variables y el uso de texturas multiresolución. Dicha organización presenta tanta importancia como el propio diseño del proceso de simulación, estando ambos aspectos fuertemente relacionados.

De cara a futuros desarrollos sería importante contar con un sistema de generación dinámica de niveles de detalle variables a partir de una malla topográfica global de detalle máximo.

## Referencias bibliográficas

1. Rohlf, J. Helman, J. *"IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time" 3D Graphics. SIGGRAPH'94 Computer Graphics Proceedings*
2. Tanner, C., Migdal, C. and Jones, M., *"The Clipmap: A virtual MipMap"* IRIS Performer Technical Documentation
3. Hatch, D., *"How to do Virtual Clip Textures"* IRIS Performer Technical Documentation
4. Schlauffer, G. Stürzlinger, W. *"Generating Multiple Levels of Detail from Polygonal Geometry Models"*. Virtual Environments'95. Springer Verlag 1995
5. Chevalier, J. *"Images of Utah"*. SIGGRAPH 95 Visual Proceedings
6. Rosignac, J.R. Borrel, P. *"Multi-resolution 3D Approximations for Rendering Complex Scenes"*. Computer Science, IBM Watson Research Center. New York, 1992
7. Lindstrom, P. Hodges, L. *"Real-Time, Continuous Level Of Detail Rendering of Height Fields"*. SIGGRAPH'96 Computer Graphics Proceedings
8. Scarlatos, L. *"A refined Triangulation Hierarchy for Multiple Levels of Terrain Detail"* Proceedings of IMAGE V Conference, Junio 1990
9. Schroeder, F. Rossbach, P. *"Managing the Complexity of Digital Terrain Models"*. Computer & Graphics 18 (6), 1994
10. Hernández, L. *"Application of Digital 3D Models on Urban Planning and Highway Design"*. Actas del III Conference on Urban Transport and the Environment. Computational Mechanics Publications.
11. Berman, D. Bartell, J. Salesin, D. *"Multiresolution Painting and Compositing"* SIGGRAPH'94 Computer Graphics Proceedings