Un Modelo Multirresolución para Visualización Interactiva

José Ribelles, Miguel Chover, Joaquín Huerta, Ricardo Quirós
Departamento de Informática. Universidad Jaume I.
Campus de Penyeta Roja. 12071. Castellón.
Tel: (964) 345771 / Fax: (964) 345848
e-mail: ribelles@inf.uji.es

Resumen

Los modelos geométricos con un alto grado de detalle aparecen, cada vez con más frecuencia, en las aplicaciones. Estos modelos se representan, normalmente, mediante complejas mallas de triángulos y conllevan tres problemas importantes: el almacenamiento, la transmisión y la visualización. Se propone un nuevo modelo multirresolución para representar mallas arbitrarias de triángulos, con bajo coste espacial, eficiente en la extracción de un nivel de detalle y que permite la transmisión progresiva. El modelo se construye almacenando las representaciones de forma ordenada dependiendo del proceso de simplificación. La extracción de un determinado nivel se realiza con un coste lineal respecto al número de triángulos que pertenecen a dicho nivel. La implementación es sencilla ya que la estructura de datos está basada en listas, fácilmente implementables mediante vectores.

Palabras Clave: multirresolución, nivel de detalle, transmisión progresiva, compresión de geometría, simplificación de superficies.

1 Introducción

Muchas aplicaciones utilizan las mallas poligonales para una visualización eficiente de superficies complejas. Los avances tecnológicos han propiciado que estas mallas sean cada vez de un mayor número de vértices y polígonos. Por ejemplo, la adquisición de datos vía satélite para representación de terreno, o los sistemas de escáner por láser utilizados en diversas áreas, son métodos que gracias a su alta precisión son capaces de generar modelos de gran complejidad. Campos como la cartografía, la ingeniería, la informática gráfica o la visión por ordenador hacen uso de este tipo de mallas en aplicaciones como la representación de terreno, tareas de reconocimiento, realidad virtual, CAD, visualización científica, simuladores de vuelo, análisis de elementos finitos, etc.

Independientemente de cómo se haya generado el modelo, de la aplicación que lo utilice y del campo en el que se trabaje, visualizar tal enorme cantidad de datos se hace muy difícil

o en algunos casos imposible ya que exceden la capacidad del hardware gráfico actual. El uso de métodos de simplificación [1] de modelos poligonales se plantea como solución a este problema consiguiendo una representación aproximada del modelo original utilizando un número de polígonos muy inferior y cumpliendo además cotas de error muy bajas.

Frente a tener una única representación se han hecho implementaciones (Vrml) que permiten disponer de varias representaciones o niveles de detalle (LOD) [2]. El objetivo es utilizar la más adecuada según la distancia al observador, visualizando la mejor representación del modelo si se encuentra cerca, o la peor si se encuentra lejos. Sin embargo, esta solución no resuelve los problemas de almacenamiento y transmisión, ya que la cantidad de datos se ha aumentado de forma considerable respecto a cuando hay una única representación. Con el fin de solucionar estos problemas aparece el modelado multirresolución [3]. Este tipo de representación es capaz de albergar de forma compacta un numero n de niveles de detalle, donde n es proporcional al tamaño del modelo [4].

La representación de mallas mediante un modelo multirresolución es especialmente útil para las siguientes aplicaciones:

- Control del nivel de detalle: la posibilidad de ajustar mejor el LOD pudiendo seleccionar el mínimo que mantenga una calidad suficiente en la visualización y además realizarlo de forma automática.

- Transmisión progresiva: cuando un modelo se transmite a través de una línea de comunicación, es interesante poder mostrarlo a medida que se vayan recibiendo los datos que van añadiendo el detalle.

- Compresión de la malla: minimizar el espacio necesario para almacenar la malla

- Refinamiento selectivo: posibilidad de representar mayor detalle en algunas regiones del modelo, con el fin de obtener mejores aproximaciones en las zonas de interés.

El objetivo es construir un modelo multirresolución que resuelva estos problemas. Además deberá cumplir que:

- la extracción de una malla dada una resolución se realice en tiempo real;

- la malla extraída sea continua, es decir, que no presente ningún tipo de rotura;

- la transición de un nivel a otro se realice de forma suave;

- no haya pérdida de información.

Se propone una nueva estructura de datos y los algoritmos de acceso necesarios para representar una malla arbitraria de triángulos con n niveles de detalle, permitiendo una visualización interactiva del modelo y su transmisión de forma progresiva.

Este artículo está organizado de la siguiente forma. En la sección 2 se hace una breve revisión de los modelos multirresolución que han sido presentados en trabajos anteriores. En la sección 3 se presenta el nuevo modelo multirresolución, un análisis del coste espacial, los

algoritmos necesarios para extraer un LOD, y la solución a los problemas de la transmisión progresiva y la compresión de la malla. Además se compara con otros modelos multi-rresolución propuestos en la literatura. En la sección 4 se muestran los resultados obtenidos y, finalmente, en la sección 5 las conclusiones y algunas ideas para el trabajo futuro.

2 Trabajo previo

Para construir un modelo multirresolución es necesario utilizar algún método de simplificación que genere las distintas aproximaciones. Algunos resúmenes del estado del arte son [1][4][5]. En este trabajo, el criterio utilizado para clasificar los métodos de simplificación es el del dominio de los vértices de salida, es decir, si el conjunto de los vértices del modelo simplificado es un subconjunto o no de los de la malla original.

Entre los métodos que producen un subconjunto de vértices del conjunto original se encuentran, por ejemplo, los siguientes: Schroeder et al. [6] proponen un método basado en eliminación de vértices utilizando un criterio de error local, es decir, evaluar el error a partir de la malla parcialmente simplificada; Hinker y Hansen [7] hacen un recorrido de la malla buscando zonas donde los triángulos tienen normales casi paralelas, eliminándolos y retriangulizando; Klein et al. [8] también basan su método en la eliminación de vértices pero utilizan un criterio de error global. Entre los que si producen cambios en el subconjunto de vértices se encuentran, por ejemplo: Hoppe [9] se basa en la eliminación de aristas uniendo parejas de vértices, generando vértices nuevos; Rossignac y Borrel [10] proponen subdividir el objeto utilizando una rejilla regular de cajas, mezclando los vértices que hay en cada caja y sustituyéndolos por uno solo.

Los modelos multirresolución más simples constan de entre 5 y 10 niveles de detalle [2]. Sin embargo, el hecho que los métodos de simplificación eliminen de forma iterativa un vértice, una arista o un triángulo, produciendo que la diferencia entre dos LOD consecutivos difieran sólo en eso, anima a formalizar modelos multirresolución con n niveles de detalle.

Lounsbery et al. [11] generaliza el concepto de análisis multirresolución, MRA, para superficies de topología arbitraria. Eck et al. [12] amplía MRA de forma que pueda ser aplicado a mallas arbitrarias, mediante un remallado de la superficie original. Certain et al. [13] extienden MRA para poder capturar color analizándolo separadamente de la forma.

Hoppe [9] propone un nuevo esquema para representar modelos multirresolución de mallas arbitrarias de triángulos, dando solución a los problemas de transmisión progresiva, transición suave, compresión de la malla y refinamiento selectivo. Consigue representar una malla M como una malla M^0 burda más una secuencia de n registros que indican como refinar M^0 de forma incremental hasta llegar a M. La construcción del modelo está basado en un método de simplificación por unión de vértices y eliminación de aristas como se comentó anteriormente. Xia y Varshney [14] proponen un método similar al de Hoppe, en el que en lugar de tener una secuencia lineal de uniones, construyen un árbol mejorando el coste de acceso a los vértices del modelo.

Ciampalini et al. [15], proponen un modelo basado en una estructura de árbol de intervalos. A cada triángulo que pueda aparecer durante el proceso de simplificación se le asigna un par de valores: el momento en el que nace y el momento en el que muere. Después organiza todos los triángulos en el árbol en base a estos valores. Consiguen extraer un

determinado LOD con un coste O(m log k) donde *m* es el número de triángulos y *k* el número de simplificaciones realizadas.

Klein et al. [16] proponen utilizar un modelo ya desarrollado por Puppo [17], que mejora el algoritmo de extracción de un LOD. La estructura almacena un conjunto de fragmentos, que representan todas las retriangulizaciones realizadas en el proceso de simplificación.

Otros modelos multirresolución, utilizando técnicas de partición binaria del espacio (BSP Trees) están todavía en desarrollo, habiendo solucionado el problema sólo para 2D½[18][19]. Sin embargo, el punto de vista de este tipo de técnica es muy interesante debido a que no necesita de un algoritmo de ocultación a la hora de visualizar el LOD extraído.

Generar un LOD mediante un método de simplificación en tiempo real es algo realmente difícil, caro e ineficiente. Sin embargo, una vez aplicado el algoritmo de simplificación, como un preproceso fuera de línea, conseguir extraer un LOD de forma suficientemente rápida como para que permita una visualización interactiva es perfectamente posible.

Para construir el modelo, de entre los algoritmos de simplificación existentes, es necesario aplicar alguno de los que restringen el domino de los vértices de salida (no incluyen nuevos vértices, ni modifican su posición), como por ejemplo, el algoritmo basado en eliminación de vértices que proponen Schroeder et al. [6] que es el que se ha utilizado.

3 MOM: El Modelo Multirresolución

A continuación se describe la estructura de datos utilizada para construir el modelo multirresolución. Consta de los tres conjuntos de datos siguientes.

- Una lista con todos los vértices de la malla. Estos aparecen de forma ordenada según se han ido eliminando en el proceso de simplificación. Cada uno tiene asociada una lista de triángulos correspondientes a los que desaparecen cuando él es eliminado.

- Una lista con todos los triángulos que puedan ser utilizados para representar cualquier LOD. Estos se encuentran ordenados según han sido generados como consecuencia del proceso de simplificación.

- Una lista de control donde para cada nivel de detalle figure cuantos vértices no pertenecen a ese nivel y qué triángulos son candidatos a ser visualizados.

Para explicar el proceso de construcción del modelo multirresolución se va a utilizar una malla formada por nueve vértices y nueve triángulos (figura 1). A partir de esta malla se van a obtener dos nuevos niveles de detalle que se añadirán al modelo.

3.1 Construcción del modelo

Se parte de una estructura formada por una lista de triángulos y una lista de vértices. Los vértices de la malla aparecen sólo una vez en la lista de vértices y los triángulos se definen como tres índices (o punteros) de la lista de vértices [20]. La lista de control indicará que no

se ha de eliminar vértice alguno y que todos los triángulos se visualizarán. La estructura contiene la representación de mayor detalle (figura 1).

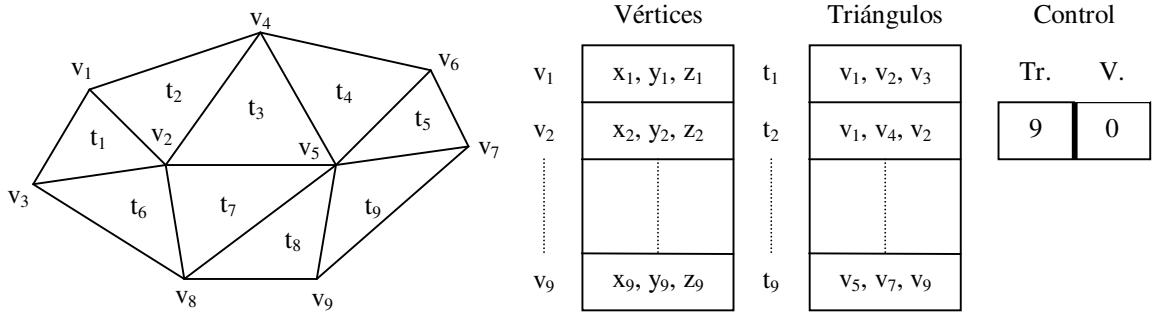


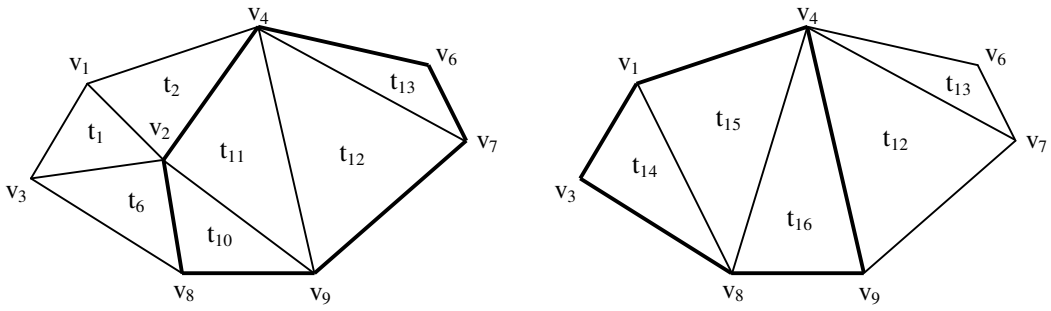
Figura 1: Representación de la malla original.

El proceso de simplificación asegura que no aparezcan nuevos vértices ni cambien de posición, por lo que la lista de vértices permanecerá constante. La lista de triángulos, en cambio, crecerá cada vez que se elimine un vértice con los triángulos que se generen al retriangulizar el agujero producido.

Para añadir un nuevo LOD hay que determinar:

- el vértice que produce la simplificación, por ejemplo, v_5;
- los triángulos que utilizan ese vértice y que como consecuencia de eliminarlo van a desaparecer: $t_3,\ t_4,\ t_5,\ t_7,\ t_8,\ t_9$;
- y los triángulos nuevos generados al retriangulizar el agujero producido: $t_{10},\ t_{11},\ t_{12},\ t_{13}$;

quedando después de retriangulizar, por ejemplo, la malla de la figura 2a.



a) b)

Figura 2: a) Malla resultado de la simplificación por eliminación de v_5.
b) Malla resultado de la simplificación por eliminación de v_2.

Como consecuencia del proceso de simplificación, la nueva malla ha de quedar reflejada en la estructura, para ello hay que seguir los siguientes pasos:

1. Colocar v_5 ordenado, es decir, detrás del último vértice eliminado, que en este caso, al no haberse eliminado antes ninguno será el primero.

2. Asociar a v_5 los triángulos que ha eliminado: t_3, t_4, t_5, t_7, t_8, t_9.

3. Añadir a la lista de triángulos los triángulos nuevos: t_{10}, t_{11}, t_{12}, t_{13}.

4. Actualizar la estructura de control, añadiendo cuántos vértices hay que eliminar y cuál es número máximo de triángulos a visualizar, que en este caso será un vértice y un total de 13 triángulos.

En la figura 3 se puede observar estos cambios, quedando representados los dos primeros niveles de detalle. Suponiendo que para producir el siguiente LOD haya que eliminar v_2 y procediendo de la misma forma, se obtendría la malla de la figura 2b.

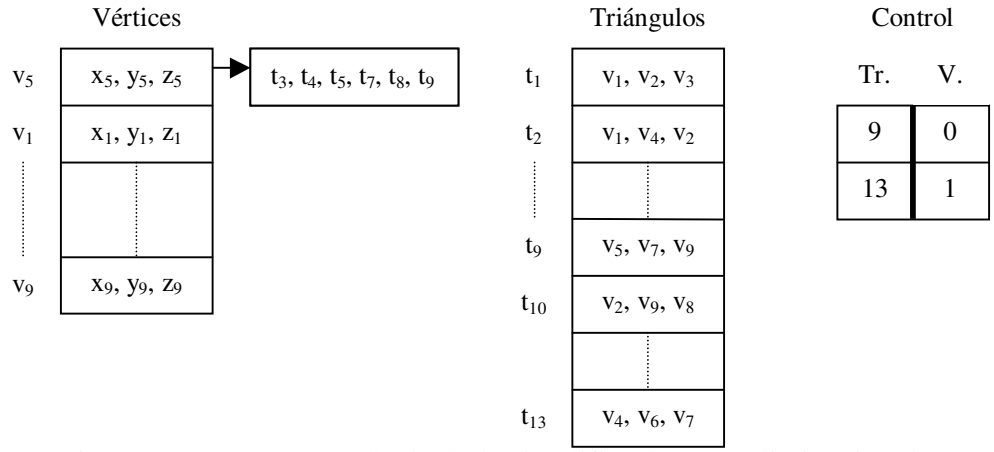


Figura 3: Estructura resultado de la simplificación por eliminación de v_5.

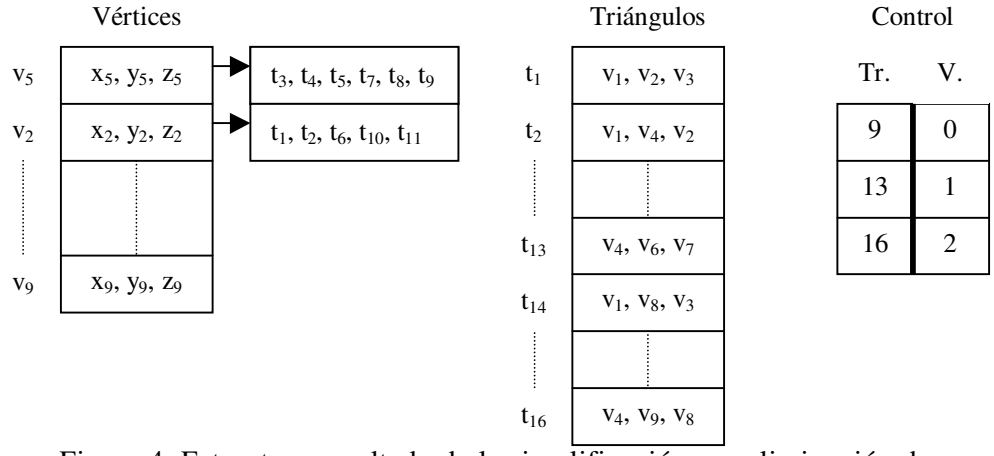


Figura 4: Estructura resultado de la simplificación por eliminación de v_2.

De nuevo, para actualizar la estructura de datos se utilizan los mismos pasos que antes. En la figura 4 puede verse la estructura resultado. Obsérvese que la lista de triángulos

asociada a v_2 contiene triángulos que pertenecen a distintos niveles de detalle. En este caso es necesario que estén ordenados según aparecen en la lista de triángulos.

En este momento se da por terminado el proceso de simplificación. Sin embargo, la estructura aún no está generada en su totalidad, quedando por hacer el siguiente paso:

- Repartir los triángulos que quedan (t_{12}, t_{13}, t_{14}, t_{15}, t_{16}) entre los vértices que no han sido eliminados (v_1, v_3, v_4, v_6, v_7, v_8, v_9).

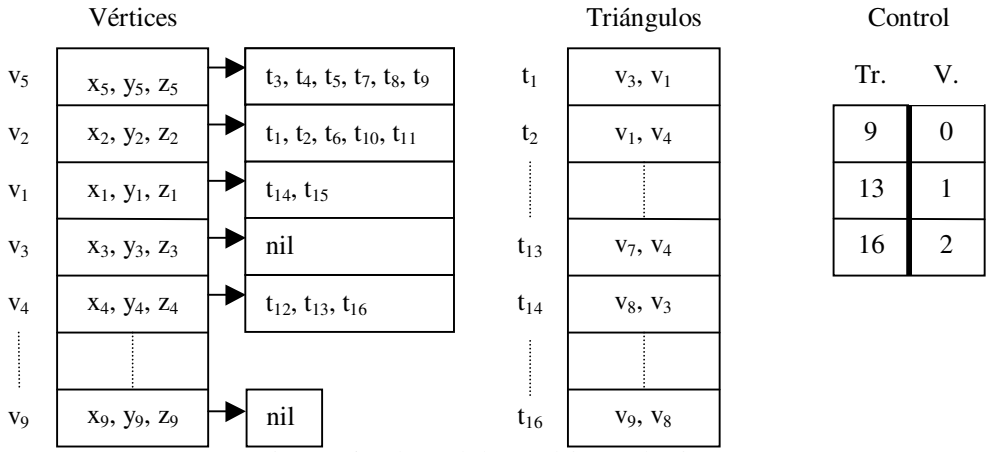


Figura 5: El modelo multirresolución.

Aún es posible simplificar el modelo aprovechando que los vértices apuntan a los triángulos y que un triángulo es apuntado sólo una vez por cada vértice, siendo suficiente almacenar dos índices por triángulo. En la figura 5 se muestra como quedaría finalmente el modelo multirresolución.

Como se puede observar, el triángulo t_1 (v_1, v_2, v_3) se encuentra en la lista de triángulos asociada al vértice v_2, por lo que sólo es necesario almacenar en la lista de triángulos los otros dos vértices v_3 y v_1.

3.2 Coste Espacial

El coste espacial del modelo puede ser evaluado como sigue. Sea V el número de vértices, T el número de triángulos de la malla inicial, T' el número de triángulos generados en el proceso de simplificación y N el número de niveles de detalle.

- Lista de vértices. El número de vértices ha permanecido constante. Sin embargo, por cada uno es necesario un índice (o puntero) al primer triángulo de la lista asociada.

- Lista de triángulos. El número de triángulos ha aumentado. La malla original consta de T triángulos y durante el proceso de simplificación se han generado T' triángulos nuevos. En general, no se puede establecer una cota lineal para T'. Por otro lado, sólo es necesario almacenar dos índices por triángulo.

- Lista de triángulos asociada a cada vértice. Cada triángulo es apuntado por sólo un vértice, por lo que el coste es proporcional al número total de triángulos. Se puede ver que este espacio se compensa con el que se ahorra al eliminar un vértice de cada triángulo en la lista de triángulos.

- Lista de control. Depende del número de niveles de detalle N.

En la tabla 1 se muestran los costes espaciales de ambas estructuras.

	Lista de Vértices	Lista de Triángulos	Listas de triángulos de los vértices	Lista de Control
Monorresolución	3V reales	3T enteros	No existe	No existe
Multirresolución	3V reales + V enteros	2 (T + T') enteros	T + T' enteros	2N enteros

Tabla 1: Relación de costes espaciales.

En la sección 4 se muestran los datos relativos al modelo con el que se han realizado los experimentos.

3.3 Visualización de un LOD

Gracias a la ordenación que presentan los vértices y triángulos en el modelo (ordenación generada en el proceso de construcción), y junto con la información de control la extracción del conjunto de triángulos asociados a ese LOD se realiza con un coste lineal respecto al número de triángulos pertenecientes a él. El algoritmo para visualizar la malla se muestra en la figura 6.

```
De la lista de control, se extrae:
T el número máximo de triángulos a visualizar
V el número de vértices a eliminar (simplificados)

/* recorrer la lista de vértices */
Para cada vértice desde V hasta el final de la lista de vértices
        /* recorrer la lista de triángulos asociada a ese vértice */
        Mientras que el índice del triángulo sea ≤ T
                Visualizar el triángulo
        Fin Mientras
Fin Para
```

Figura 6: Algoritmo para visualizar un LOD.

1er LOD		2º LOD		3 er LOD	
V=0, T= 9		V=1, T= 13		V=2, T= 16	
v_5	t_3, t_4, t_5, t_7, t_8, t_9	v_2	t_1, t_2, t_6, t_{10}, t_{11}	v_1	t_{14}, t_{15}
v_2	t_1, t_2, t_6	v_4	t_{12}, t_{13}	v_4	t_{12}, t_{13}, t_{16}

Tabla 2: Relación de vértices y triángulos a visualizar por LOD.

En el ejemplo que se está utilizando se ha representado una malla con tres niveles de detalle. En la tabla 2 se muestra para cada LOD los triángulos que se visualizan y el vértice relacionado.

En la sección 4 se muestran los datos relativos al modelo con el que se han realizado los experimentos.

3.4 Transmisión progresiva

Para realizar una transmisión progresiva del modelo multirresolución es necesario enviar en primer lugar una malla inicial M^0 correspondiente con el peor nivel de detalle y, a continuación, un conjunto de registros que contienen la información necesaria para ir añadiendo el detalle hasta llegar a reconstruir la malla original sin pérdida de información. Una vez recibida M^0, ésta se visualiza y actualiza a medida que se reciben los detalles. Cada uno de estos registros se puede representar como el trío (v, n, t) donde:

- v es el vértice (x, y, z) que se añade a la malla.
- n es el número de datos que se envían a continuación, la longitud de t.
- t son los índices de los triángulos a los que afecta el vértice v.

Con esta información se puede reconstruir la malla original. Tomando la malla de la figura 4 como M^0, la secuencia de operaciones sería la siguiente:

- Enviar la malla inicial formada por los vértices (v_1, v_3, v_4, v_6, v_7, v_8, v_9) y los triángulos (t_{12}, t_{13}, t_{14}, t_{15}, t_{16}).
- Enviar el registro formado por (v_2, *3, (14, 15, 16)*). Con esta información recibida, es fácil generar los triángulos (t_1, t_2, t_6, t_{10}, t_{11}).
- Enviar el registro formado por (v_5, *4, (10, 11, 12, 13)*), que permitiría generar los triángulos (t_3, t_4, t_5, t_7, t_8, t_9).

3.5 Compresión de la malla

La misma representación utilizada para realizar la transmisión progresiva, se propone para almacenar de forma más compacta el modelo multirresolución. Este tipo de representación está formado por una malla inicial M^0, que se corresponde con la representación más burda, y un conjunto de registros que aportan la información necesaria para a partir de M^0 ir refinando sucesivamente hasta llegar a M, la representación con máximo detalle. La cantidad de memoria requerida para almacenar el modelo será la utilizada por M^0 más el espacio necesario para almacenar todos los registros.

Un registro está formado por el trío (v, n, t). Tanto *v* como *n* son de tamaño constante, pero *t* es variable, dependiendo de a cuantos triángulos haga referencia el vértice *v*. Este es el análisis:

- *v*, los datos del vértice, son tres reales
- *n*, la longitud de *t*, es un entero
- *t*, el conjunto de triángulos afectados por *v*, es *n* enteros

En la sección 4 se muestran los datos relativos al modelo con el que se han realizado los experimentos.

3.6 Comparación con otros modelos multirresolución

En este apartado se realiza una comparación teórica con otros modelos multirresolución remarcando las principales diferencias con el modelo propuesto MOM.

Multiresolution Analysis, MRA [11]*:* teniendo en cuenta la evaluación realizada por Hoppe [9], la principal diferencia es que MOM mantiene una representación sin pérdida de información ya que en MRA la malla original sólo puede ser alcanzada a partir de una malla inicial con un error ε. Además, las distintas aproximaciones de MOM pueden ser mejores al no tener la restricción de que la malla se obtenga por subdivisión conectada, pudiendo representar con más precisión aristas o discontinuidades. Sin embargo, MRA permite la edición multirresolución y que la geometría y el color se compriman de forma independiente.

Progressive Meshes, PM [9]: este modelo es preciso en términos de almacenamiento, pero requiere un esfuerzo computacional alto para extraer los triángulos que pertenecen a un LOD, realizando *n* operaciones de unión o separación de vértices para ir de una malla a otra.

Merge Tree, MT [14]: basado en PM, lo mejora construyendo un árbol sobre los vértices de la malla, de forma que en lugar del coste lineal de PM, se obtiene un coste O(log n).

JADE [15]: el coste espacial de este modelo es mayor que el de MOM debido a que necesita almacenar dos valores por triángulo correspondientes a los momentos de creación y eliminación en el proceso de simplificación. Además utiliza dos listas de triángulos por nodo para lograr extraer un LOD de forma eficiente [21]. El coste de extracción O (m log k) depende del número de simplificaciones, *k*, y del número de triángulos, *m*. En MOM, este coste sólo depende del número de triángulos *m*. Por contra, JADE permite utilizar cualquier método de simplificación.

Multiresolution Model, MRM [16]: el coste espacial del modelo es mayor ya que almacena un conjunto de fragmentos que representan la multi-triangulización. Un fragmento se corresponde con el agujero que se produce en el proceso de simplificación, y contiene los índices de los triángulos del suelo y el techo. Además, cada triángulo almacena dos punteros a los fragmentos que lo contienen. La transmisión progresiva es posible [17] pero necesita transmitir todos los fragmentos. MOM sólo almacena los índices de los triángulos correspondientes al techo y sólo transmite los correspondientes al suelo.

Señalar por último, que MOM tiene una característica que lo hace mejor frente a los otros modelos y es su simplicidad de construcción, ya que frente a estructuras de árboles se propone una estructura basada en listas, fácilmente implementable mediante vectores.

4 Resultados

Los experimentos se han realizado utilizando el modelo del conejo, formado por 69473 caras (triángulos) y un total de 34835 vértices. Cuatro representaciones distintas del modelo pueden verse en la figura 7. Asimismo, en la tabla 3 se muestran los tiempos de acceso a la estructura de datos, es decir, el tiempo invertido en extraer los triángulos que pertenecen a un determinado LOD. El hardware utilizado ha sido una estación de trabajo basada en un Pentium II (266Mhz, 64Mb SDRAM), y un acelerador gráfico para OpenGL basado en el chip Permedia 2. El método de simplificación utilizado está basado en eliminación de vértices y fue propuesto por Schroeder et al. [6].

De los resultados expuestos en la tabla 3, se observa que el tiempo de acceso al modelo con una única representación es un poco más bajo que cuando se accede al LOD equivalente en MOM. Sin embargo, el uso de un LOD peor, pero que mantiene la calidad visual requerida, rebaja considerablemente los tiempos de acceso.

	Monorres.	LOD 1	LOD 2	LOD 3	LOD 4
Vértices	34.835	34.835	7.267	3.548	778
Triángulos	69.473	69.473	14.338	6.900	1.360
Vértices eliminados	-	0	27.548	31.287	33.987
Tiempo de acceso (ms.)	50'000	61'589	11'510	5'510	1'499
Tiempo de acceso + Visualización (ms.)	200'000	235'300	48'099	23'000	5'000

Tabla 3: Relación de tiempos.

Los costes de almacenamiento del modelo, según las fórmulas indicadas en los puntos 3.2 y 3.5, quedan reflejados en la tabla 4. El número de niveles de detalle es de 33988. Si no hay necesidad de disponer de tal cantidad de niveles, el coste total del modelo multirresolución disminuiría ligeramente, ya que MOM no obliga a que entre dos LOD consecutivos haya únicamente un vértice de diferencia.

Monorresolución	Multirresolución	Multirresolución comprimido
102k reales + 204k enteros	102k reales + 685k enteros	2.28k reales + 4k enteros + 99.72k reales + 159k enteros

Tabla 4: Coste espacial.

De los resultados expuestos en la tabla 4, se observa que el coste de almacenamiento del modelo multirresolución es entre dos y tres veces mayor que el monorresolución, sin

embargo MOM está albergando casi 34000 niveles de detalle. Además, el modelo comprimido requiere menos espacio que ambos modelos.

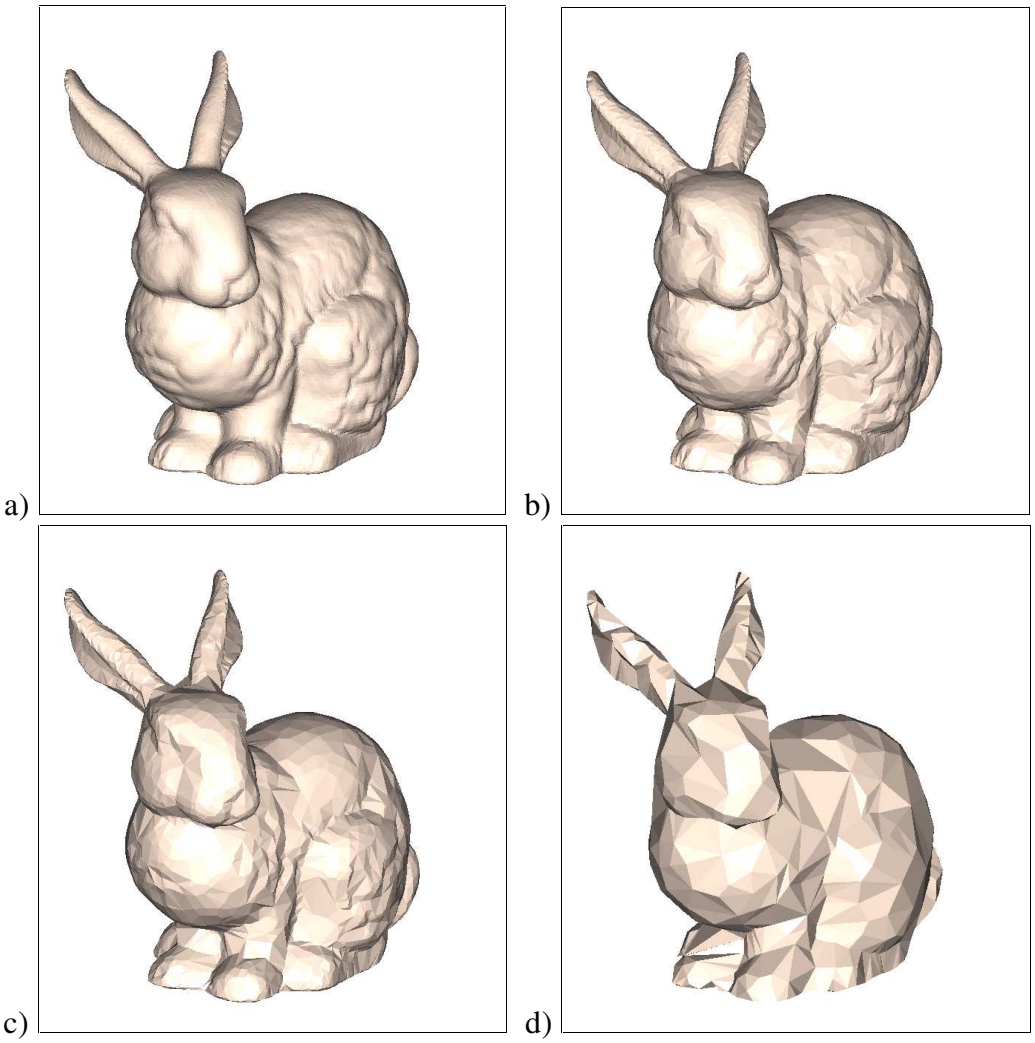


Figura 7: a) LOD 1; b) LOD 2; c) LOD 3; d) LOD 4.

5 Conclusiones y Trabajo Futuro

Se ha presentado un nuevo modelo multirresolución que permite representar mallas arbitrarias de triángulos con *n* niveles de detalle de forma compacta y simple, y extraer un determinado LOD con un coste lineal respecto al número de triángulos que pertenecen a ese LOD, permitiendo una visualización interactiva. Además se da solución a los problemas de la transmisión progresiva y almacenamiento del modelo.

En los resultados experimentales se puede observar que el modelo multirresolución requiere más espacio que el monorresolución, pero la posibilidad de realizar compresión de la malla rebaja el espacio requerido dejándolo por debajo de ambos.

La principal tarea que se ha quedado por resolver es el llamado refinamiento selectivo, o dependencia de la vista, que permitiría representaciones cuya resolución no sea constante en todo el modelo, sino que pueda variar, siendo mayor en las regiones de interés y menor en el resto. Esta zona de mayor detalle puede venir indicada tanto por el usuario, como por otros motivos, por ejemplo, la vista, la iluminación, etc.

MOM no puede ser utilizado junto con cualquier método de simplificación. Sólo aquellos que restrigen el dominio de los vértices de salida pueden aplicarse. Otro de los objetivos es estudiar la posibilidad de eliminar esta restricción.

Por otra parte, la estructura tampoco almacena datos como normales o color, que de utilizarlos mejorarían notablemente la calidad de las visualizaciones.

Referencias

[1] P. Heckbert, M. Garland, "Survey of surface simplification algorithms", Technical Report, Carnegie Mellon University - Dept. of Computer Science, 1997.

[2] Thomas A. Funkhouser, Carlo H. Sequin, "Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments", Computer Graphics (SIGGRAPH '93 Proceedings), 1993, aug, vol. 27, pag. 247--254

[3] Paul Heckbert, Michael Garland, "Multiresolution Modeling for Fast Rendering", Proceedings of Graphics Interface '94, Banff Alberta Canada, 1994, may, pag. 43--50

[4] Enrico Puppo, Roberto Scopigno, "Simplificacion, LOD and Multiresolution Principles and Applications", EUROGRAPHICS'97, 1997, vol. 16, no. 3.

[5] C. Erikson, "Polygonal Simplification: An Overview", Technical Report TR96-016, Department of Computer Sciencie, UNC-Chapel Hill, 1996.

[6] William J. Schroeder, Jonathan A. Zarge, William E. Lorensen, "Decimation of triangle meshes", Computer Graphics (SIGGRAPH '92 Proc.), 1992, jul, vol. 26, no. 2, pag. 65--70.

[7] Paul Hinker, Charles Hansen, "Geometric Optimization", Proc. Visualization '93, San Jose CA, 1993, oct, pag. 189--195.

[8] R. Klein, G. Liebich, W. Straber, "Mesh reduction with error control", Proceedings of Visualization'96, Ed. R. Yagel, G. Nielson, 1996, pag. 311-318.

[9] Hugues Hoppe, "Progressive Meshes", SIGGRAPH '96 Proc., 1996, aug, pag. 99--108.

[10] Jarek Rossignac, Paul Borrel, "Multi-resolution 3D approximations for rendering complex scenes", Geometric Modeling in Computer Graphics, Genova, Italy, Ed. Springer-Verlag, 1993, pag. 455--465.

[11] Michael Lounsbery, Tony DeRose, Joe Warren, "Multiresolution analysis for surfaces of arbitrary topological type", ACM Transactions on Graphics, 1994, vol. 16, no. 1, pag. 34-73.

[12] Matthias Eck , Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, Werner Stuetzle, "Multiresolution Analysis of Arbitrary Meshes ", SIGGRAPH '95 Proc., Ed. ACM, 1995, aug, pag. 173--182.

[13] Andrew Certain, Jovan Popovic, Tony DeRose, Tom Duchamp, David Salesin, Werner Stuetzle, "Interactive Multiresolution Surface Viewing", SIGGRAPH 96 Conference Proceedings, Ed. Addison Wesley, 1996, aug, pag. 91--98.

[14] J. Xia , A. Varshney, "Dynamic view-dependent simplification for polygonal models", IEEE Visualization'96 Proc., Ed. R. Yagel, G. Nielson, 1996, pag. 327-334.

[15] A. Ciampalini, P. Cignoni, C. Montani, R. Scopigno, "Multiresolution decimation based on global error", Technical Report C96-021, CNUCE-C.N.R. (To apperar on The Visual Computer, 1997), Pisa, Italy, 1996, Jul.

[16] Reinhard Klein, Jörg Krämer, "Building multiresolution models for fast interactive visualization", SCCG'97, 1997.

[17] E. Puppo, "Variable resolution triangulations", Technical Report. 12/96. Institute for Applied Mathematics, C.N.R., Genova (Italy), 1996, Nov.

[18] J. Huerta, M. Chover, R. Quirós, R. Vivó, J. Ribelles, "Binary Spaces Partitioning Trees: A Multiresolution Aproach", Proceedings Information Visualization, London, 1997, August, pag. 148-154.

[19] J. Huerta, M. Chover, J. Ribelles, R. Quirós , "Construction and Rendering of Multiresolution Binary Space Partitioning Trees", Compugraphics'97, Algarve, Portugal, 1997, pag. 212-221.

[20] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Richard L. Philips, "Introduction to Computer Graphics", Reading MA, Ed. Addison-Wesley, 1994.

[21] Paolo Cignoni, Paola Marino, Claudio Montani, Enrico Puppo, Roberto Scopigno, "Speeding Up Isosurface Extraction using Interval Trees", IEEE Transactions on Visualization and Computer Graphics, 1997, Marzo, vol. 3, no. 4, pag. 352-369.