

Interacción en Tiempo Real para un Sistema de Escultura Virtual

A. León, F. Velasco y F. Soler

Grupo de Investigación en Informática Gráfica (<http://giig.ugr.es>), Dpto. de Lenguajes y Sistemas Informáticos,
E.T.S. de Ingenierías Informática y de Telecomunicación, Periodista Daniel Saucedo Aranda, s/n 18071 Granada (Spain)

Abstract

La escultura virtual permite al usuario crear objetos de forma que la representación interna de éstos sea transparente. Se pretende simular un proceso de modelado de objetos que imite la escultura de materiales que se realiza en el mundo real. Dicha imitación precisa que la representación permita al usuario realizar operaciones de edición en tiempo real de forma interactiva.

En este trabajo presentamos la aplicación de una representación basada en un octree al proceso de escultura virtual. El octree permite representar regiones con mayor y menor nivel de detalle y, con respecto al requisito de abstraer al usuario de la representación subyacente, nuestra representación presenta una isosuperficie extraída a partir del octree mediante una rejilla dual de celdas definidas implícitamente. Esta superficie representa el objeto sobre el que se realizan operaciones de edición mediante herramientas de modelado de material, dando la impresión al usuario de interactuar sobre la superficie percibida.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques: *Interaction techniques*

1. Introducción

La escultura virtual es una técnica de creación/modificación de objetos que fue presentada por Galyean [GH91]. Esta técnica requiere una representación computacional de los objetos que permita al usuario abstraerse de los detalles de dicha representación. Se pretende que éste lleve a cabo un proceso de modelado de objetos que imite la forma en la que un escultor realiza su trabajo sobre los materiales del mundo real. Imitar dicho proceso lleva aparejado que la representación permita al usuario realizar operaciones de edición de forma interactiva.

Debido al proceso de escultura con distintos tipos de herramientas y a las posibles variaciones en sus tamaños, es necesario que el modelo computacional permita representar partes del objeto con un alto nivel de detalle, junto con otras con un nivel de detalle menor. Por tanto, otro requisito importante de la representación es que ésta tenga una resolución variable, con el objetivo de no tener que almacenar información innecesaria sobre el objeto que se está mode-

lando. Este sería el caso, por ejemplo, de usar una rejilla regular como representación.

Las primeras representaciones que permitían aplicar la técnica de escultura virtual sobre objetos volumétricos utilizaban una rejilla regular como estructura de datos subyacente [GH91, WK95, AS96]. Como acabamos de comentar, esta representación limita el dominio de objetos a modelar ya que se utiliza demasiada información en zonas en donde no es necesario. La rejilla regular es una representación demasiado estática. Una representación adecuada debe poder adaptarse para ocupar el mínimo espacio posible en las regiones del objeto a modelar en las que no es necesaria una resolución alta. Además, en las regiones donde se están realizando operaciones, es necesario que la resolución se adapte a la forma de la herramienta que se está aplicando al objeto.

Los *voxels* (elementos de volumen) de la mayoría de las representaciones que permiten modelar objetos con superficies suaves siguiendo esta técnica, toman valores escalares en un dominio discreto [GH91, WK95, AS96, Bae98, RE00, FCG00]. Se han propuesto varias soluciones que permiten

que la representación se adapte a las necesidades de resolución en cada zona del objeto [FPRJ00, PF01, JLSW02, FCG02]. En todas ellas, la solución pasa por utilizar algún tipo de estructura jerárquica.

Friskén y otros [FPRJ00, PF01] proponen el uso de campos de distancias muestreados adaptativamente (ADFs) con el objetivo de almacenar un ratio de muestreo alto en las regiones de los objetos donde exista gran nivel de detalle a la vez que se almacena un ratio bajo de muestreo en las áreas donde el campo varía suavemente. A fin de procesar de forma eficiente los datos muestreados adaptativamente, almacena las celdas en los nodos de una estructura de datos *octree*.

Ju y otros [JLSW02] proponen un *octree* cuyos nodos hoja contienen celdas con signos en sus esquinas en función del umbral de propiedad escogido. Las aristas de dichas celdas que presentan cambios de signo almacenan puntos de intersección y normales exactas.

Ferley y otros [FCG02] utilizan una estructura jerárquica de rejillas para abordar tanto el problema de la adaptación de las muestras requeridas como el problema de la modificación local de la estructura cuando se aplica una operación de modelado al objeto.

En este trabajo mostramos la aplicación de un *octree* con celdas duales definidas implícitamente a la escultura virtual utilizando la capacidad de la representación para modificar sobre la marcha, y sólo en la zona afectada, su configuración de celdas duales, en función de los elementos de volumen sobre los que se aplica una operación determinada.

En nuestro caso, hemos optado por un *octree* que mantiene implícitamente la topología de una rejilla dual. La rejilla permite extraer la iso-superficie que representa para el usuario el objeto que está creando o modificando. Cuando se aplica una operación de modelado al objeto, se localizan en el *octree* las regiones que van a ver modificados sus valores y se recalcula la topología de la rejilla dual localmente, si es necesario. La topología de la rejilla dual cambia como consecuencia de que los nodos hoja del *octree*, que representan regiones del espacio con un valor de propiedad determinado, se compacten o se subdividan. Este hecho viene determinado por el cambio en los valores de propiedad que provoca la operación sobre cada una de las regiones en las que influye.

Una vez actualizada la topología de las celdas en las que influye la operación, se extrae la iso-superficie que atraviesa dichas celdas para que el usuario pueda percibir el cambio que ha provocado la operación sobre el objeto. Nuestra representación permite que la determinación del alcance de la operación, la modificación de la porción de la rejilla dual correspondiente y la extracción de la iso-superficie asociada se realice en tiempo interactivo. De esta forma nuestra representación cumple el requisito de ocultación de la representación subyacente de cara al usuario, así como la respuesta interactiva que requiere la técnica de escultura virtual.

La sección 2 presenta resumidamente el *octree* con celdas duales implícitas, la sección 3 describe el algoritmo de modificación local de la estructura de datos y las herramientas que se han implementado para probarlo. La sección 3.2 muestra la utilización de los *Vertex Buffer Objects (VBO)* de la extensión de OpenGL para permitir descartar la geometría local previa a la aplicación de la operación y, finalmente presentamos las conclusiones.

2. El octree de volumen

A partir de una rejilla regular de valores de propiedad, construimos un *octree* simplificado que almacena regiones del espacio con un valor de propiedad homogéneo. Consideramos cada vértice de la rejilla como una región de máxima resolución. Estas regiones se corresponden con los nodos hoja del *octree*, y su tamaño y posición vienen determinados por su nivel asociado y la rama del árbol en donde se encuentran respectivamente. El proceso de simplificación compacta ocho nodos hoja en su correspondiente nodo padre cuando cumplen un criterio de homogeneidad definido por el usuario.

A la hora de calcular los valores de propiedad en todo el espacio ocupado por el volumen, se utiliza una rejilla de celdas duales a las regiones representadas por el *octree*. Esta rejilla dual se encuentra definida implícitamente en los nodos hoja. Para generar la representación implícita de la rejilla dual, se realiza un paso de preprocesamiento que asigna las responsabilidades de generación de celdas a cada una de las regiones representadas por los nodos hoja. Este proceso se describe con detalle en [LTV07]. La estructura final se puede ver representada mediante la simplificación de un *quadtree* en la figura 1.

A la hora de visualizar el volumen, utilizaremos un método de extracción de superficies basado en *Marching Cubes* [LC87], que se aplicará sobre la rejilla dual. El procedimiento requiere visitar cada uno de los nodos hoja. Para cada nodo hoja, se exploran las celdas duales cuya responsabilidad recae en dicho nodo y se determina si son atravesadas por la iso-superficie. Si es así, se genera la celda dual geométrica, cuyos vértices se corresponden con los centros de las regiones que la forman y cuyos valores de propiedad, en dichos vértices, coinciden con los de sus regiones asociadas.

El *octree* de volumen permite realizar operaciones de escultura virtual, de forma que una operación se aplica sobre un conjunto de regiones, y los cambios en la representación permanecen acotados en la vecindad de dicho conjunto. Por consiguiente, sólo necesitamos modificar la parte de volumen que se ve alterada por la operación y extraer la parte de iso-superficie correspondiente.

3. Operaciones de modelado

Una herramienta para realizar escultura virtual sobre el *octree* se define mediante dos parámetros: el alcance o vo-

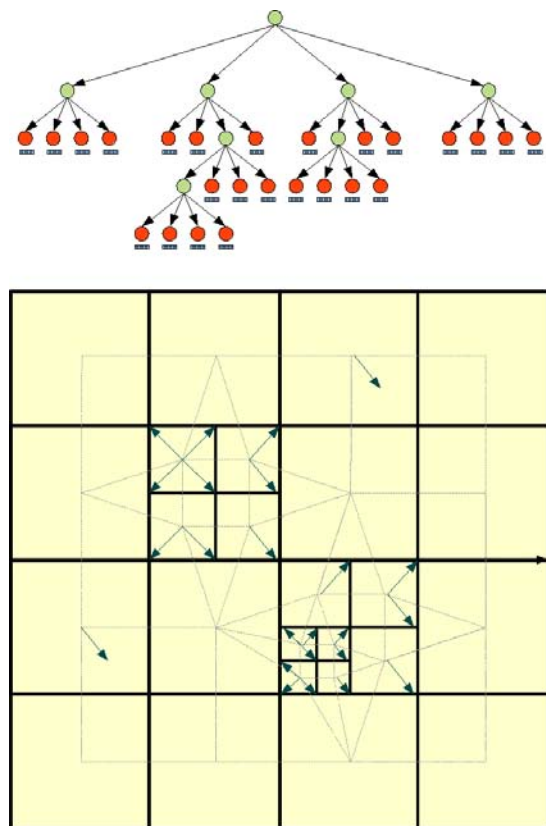


Figure 1: Octree con la estructura dual definida implícitamente. Las líneas discontinuas muestran la estructura de celdas dual mientras que las flechas asignadas en cada región o nodo indican una responsabilidad de procesamiento de celda en una dirección determinada. En esta simplificación con un quadtree hay 4 posibles direcciones de procesamiento; en el caso del octree tenemos 8 direcciones.

lumen y la función que asigna valores de propiedad dentro de dicho volumen. El *alcance* de la herramienta determina el conjunto de regiones del *octree* que van a ser modificadas y la *función de asignación* determina el nuevo valor que se asignará a cada una de las regiones involucradas en la aplicación de la operación.

Hemos elegido una esfera de radio R , definida implícitamente, como el parámetro de alcance de nuestras herramientas. La herramienta esférica actuará sobre un mayor o menor número de regiones del *octree* en función del valor de su radio. Las funciones de asignación que hemos implementado hasta el momento presentan un rango discreto de valores y utilizamos un umbral de propiedad para determinar el interior y el exterior del objeto modelado. El nuevo valor de propiedad calculado por esta función de asignación tendrá en cuenta la posición relativa de la región con respecto al centro de la herramienta y el valor actual de propiedad de la región.

El hecho de que la asignación de valores de propiedad a regiones sea discreta provoca un efecto de “aliasing” en el objeto modelado. Sin embargo, nuestro algoritmo de modificación local del *octree* es independiente de la función de asignación utilizada, por lo que podemos utilizar otras funciones de asignación de valores de propiedad basadas en la aplicación de *kernels* sobre las regiones del volumen.

Utilizamos un dispositivo háptico para realizar la interacción sobre nuestro sistema de escultura virtual. Disponemos de un interfaz de inmersión a través de un espejo polarizado y un dispositivo de fuerza que será el que dirigirá el proceso de escultura mediante las herramientas virtuales que permiten operar sobre el objeto.

Utilizamos algoritmos de detección de colisiones que impiden que la herramienta atraviese la malla de triángulos que representa la superficie del objeto sobre el que trabajamos. Una vez que la herramienta se encuentra sobre la superficie del objeto, se aplica la operación deseada. Para ilustrar nuestro procedimiento de modificación local con tiempos de respuesta interactivos vamos a utilizar el caso de la operación sustraer material. La operación añadir material se implementa de forma similar con pequeñas variaciones.

3.1. Operación sustraer material

La operación sustraer material asigna un valor de propiedad, dentro del rango permitido, en función de la distancia a su centro, a cada una de las regiones sobre las que tenga influencia. De esta forma se evitan asignaciones de valores perniciosos que podrían prevenir el añadido de material en las zonas vacías en las que, previamente, se hubiese realizado una sustracción de material.

La detección de colisiones sobre la superficie de triángulos extraída a partir de la representación no permite que la herramienta profundice en el objeto una distancia superior al radio de alcance. El proceso de modificación de la estructura subyacente conlleva una serie de compactaciones y subdivisiones en función de varios casos que detallaremos a continuación. En primer lugar describiremos a alto nivel los pasos que se llevan a cabo cuando se realiza la operación sobre el modelo volumétrico.

- Determinar el conjunto de regiones, S , que se encuentran bajo la influencia del volumen de la herramienta. Para esto, se realiza un recorrido a través del *octree* determinando las regiones solapadas con el alcance de la herramienta.
- A partir de este conjunto S , efectuar los cambios necesarios en el *octree* de acuerdo con la modificación del valor de propiedad que efectuará la función de asignación de la herramienta. Esta modificación es la que puede implicar la necesidad de realizar operaciones de subdivisión y compactación en los nodos del *octree*.
- Realizar una reasignación de responsabilidades en las regiones del conjunto S cuya topología haya cambiado. Para ello, en cada ejecución de la herramienta, obtendremos un

conjunto de regiones subdivididas R_S y otro de regiones compactadas R_C . La reasignación de responsabilidades se realizará sobre cada región de estos conjuntos. Se lleva a cabo una separación en dos conjuntos de regiones ya que cada tipo de región permite realizar optimizaciones distintas a la hora del traslado de responsabilidades. Las regiones que no modifican su topología no cambian sus responsabilidades, salvo que sean modificadas por otras regiones.

- Encontrar el conjunto de celdas, C , tal que una celda pertenecerá a dicho conjunto si al menos una de las 8 regiones que forman su topología pertenece a S . Como las celdas forman la rejilla dual al *octree* que se usa para la extracción, es necesario calcular la nueva topología resultado del cambio que podrán sufrir las regiones del conjunto S .
- Extraer la geometría de todas las celdas activas contenidas en el conjunto C . Las celdas activas serán las únicas celdas que realmente es necesario procesar para extraer el área de superficie que cambia tras la aplicación de la operación.
- Eliminar de la iso-superficie la geometría de las celdas que han dejado de ser activas. Las celdas que han dejado de existir, debido a los cambios topológicos producidos tras la aplicación de la operación, deben descartarse de cara a la nueva visualización del objeto modificado.

Para llevar a cabo estas tareas, el primer paso es determinar el conjunto de regiones que van a ser modificadas, bien cambiando su valor de propiedad, bien sufriendo una operación de subdivisión o compactación. Para determinar este conjunto, se realiza un test de intersección de cada nodo hoja r con el volumen de la herramienta. Como resultado de ese test pueden darse cuatro situaciones:

- La región r no es intersectada por la herramienta, por lo que no sufrirá modificación alguna en su valor de propiedad.
- La región r es intersectada por la herramienta y tiene un nivel en el *octree* menor que el nivel máximo. En este caso, la región debe subdividirse y proceder recursivamente con la aplicación del test de intersección de la herramienta con cada una de sus hijas. Cuando un nodo es subdividido se elimina la geometría (triángulos generados) de cualquiera de las celdas de las que forma parte, independientemente de qué región tenga la responsabilidad de procesamiento. La operación de eliminación de la geometría extraída a partir de una celda se expone con detalle en la sección dedicada al *Vertex Buffer Object*. Cuando una región sufre esta subdivisión recursiva, sólo se almacena en el conjunto R_S el nodo que es raíz del subárbol que se ha producido. Esta optimización permite evitar la búsqueda de responsabilidades redundantes.
- La región r tiene el nivel más bajo del *octree* y ha sido generada por subdivisión de un nodo de nivel superior. En este caso se realiza el reajuste del valor de propiedad según el valor devuelto por la función de asignación de la herramienta.
- La región r tiene el nivel más bajo del *octree* y no ha sido

generada durante la ejecución de la herramienta. En este caso se realiza igualmente un reajuste del valor de propiedad de acuerdo con la función de asignación de la herramienta. Al cambiar el valor de propiedad de la región, la geometría de las celdas de las que forma parte deja de ser válida por lo que es necesario regenerarla en el *VBO* como veremos en la sección siguiente. Puede ocurrir además que, como se observa en la ilustración de la izquierda de la figura 2, la responsabilidad pertenezca a otras regiones fuera del alcance de la herramienta. Sin embargo, se puede tener en cuenta que esta región r se encuentra al más bajo nivel en el *octree* por lo que, independientemente de las operaciones que ocurran en las regiones vecinas, la región r siempre puede obtener la responsabilidad de procesamiento en las celdas de las que forma parte. Las responsabilidades de procesamiento que adquiera la región r después de este proceso se incluyen en el conjunto de celdas C . Este es el caso de la responsabilidad adquirida por la región influida por la herramienta que se muestra en la ilustración de la derecha en la figura 2.

Así podemos proceder a obtener qué regiones vecinas de r tienen la responsabilidad de procesamiento de alguna celda adyacente a r , quitarles la responsabilidad y dársela a r .

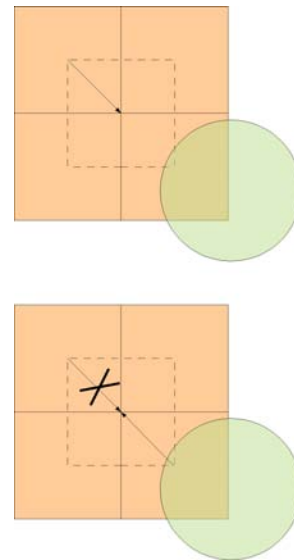


Figure 2: Situación en la que una región de máxima resolución bajo el campo de acción de la herramienta toma la responsabilidad de una celda que estaba asignada a una región que no se ve afectada por el radio de acción de la herramienta.

La asignación de valores de propiedad de la herramienta provoca operaciones de compactación en regiones que han sufrido subdivisión, siempre que se cumpla el criterio de homogeneidad establecido por el usuario. Al igual que las subdivisiones, las operaciones de compactación implican la mo-

dificación de la topología de celdas previa a la reducción del número de regiones, por lo que también es necesario eliminar la geometría de cualquier celda que incluya uno de los nodos a compactar. Cuando se realiza la compactación de un nodo, cualquier otro elemento por debajo del nodo compactado que haya sido añadido a los conjuntos R_S o R_C (regiones) o C (celdas) puede eliminarse para no realizar operaciones innecesarias.

Una vez que finaliza el recorrido del *octree*, se realiza el proceso de reasignación de responsabilidades en los conjuntos R_C y R_S . Dicho proceso consta de dos fases: búsqueda de regiones vecinas en la dirección de procesamiento correspondiente y ejecución del algoritmo de asignación de responsabilidades entre las vecinas y la región considerada. Esta asignación genera celdas que deben añadirse al conjunto C para su procesamiento posterior.

De esta forma, se consigue una modificación local de los valores de propiedad de la regiones influidas por la herramienta, así como una adaptación de la representación a la forma de dicha herramienta solamente en el área necesaria. Sin embargo, nuestra representación volumétrica no almacena explícitamente la geometría de la superficie extraída. Estos triángulos, se extraen y se utilizan para que el usuario se abstraiga de la representación interna. Cuando se aplica la operación, es necesario descartar selectivamente los triángulos que no deben aparecer tras la modificación de la estructura. Para solucionar este problema, manteniendo la respuesta interactiva, hemos utilizado una estructura que pertenece a las extensiones de OpenGL: los *Vertex Buffer Objects*.

3.2. Vertex Buffer Objects

En nuestra representación volumétrica no existe una relación entre la representación del volumen (regiones y rejilla dual de celdas) y la geometría de la iso-superficie (triángulos que se extraen de cada celda). Para modificar, eliminar o añadir nuevos fragmentos de iso-superficie, es necesario añadir a la representación volumétrica la relación entre las celdas duales y la geometría extraída. La solución que proponemos a este problema hace uso de una estructura de datos geométricos proporcionada por la extensión de OpenGL: el *Vertex Buffer Object (VBO)*. Concretamente, el *VBO* nos va a permitir:

- Almacenar la geometría del modelo en la memoria de la tarjeta gráfica con lo que se aumenta la velocidad de visualización de la iso-superficie.
- Almacenar información geométrica de distintos tipos: vértices, normales, colores,...
- Dibujar de forma independiente partes separadas de la geometría del modelo, mediante llamadas a la función *glDrawArrays*.
- Modificar la información geométrica de partes concretas del modelo, mediante llamadas a la función *glBufferData*.

De esta forma podremos asociar a cada celda dual una parte del *buffer*, la cual almacena el conjunto de triángulos extraídos a partir de dicha celda. A la hora de descartar geometría durante las operaciones de compactación y subdivisión, simplemente se dejarán huecos libres en el *buffer* cuyo tamaño depende del número de triángulos generados a partir de la celda descartada.

En el *VBO* se irá almacenando, de forma consecutiva, la geometría generada para cada celda dual atravesada por la iso-superficie. De esta forma, la geometría de una celda dual ocupa en el *VBO* un espacio que se representa mediante un par $\langle \text{elemento inicial, longitud} \rangle$. De esta forma, cada región tendrá asociado un par de este tipo por cada responsabilidad de celda que realmente sea atravesada por la iso-superficie. La geometría correspondiente a dicha celda estará almacenada en el *VBO* entre las posiciones $\langle \text{elemento inicial} \rangle$ y $\langle \text{elemento inicial} + \text{longitud} \rangle$. En la figura 3 (a) se representa la estructura del *VBO*. En la primera mitad del *buffer* se almacenan los vértices de los triángulos procedentes de cada celda y en la segunda mitad las normales asociadas a dichos vértices.

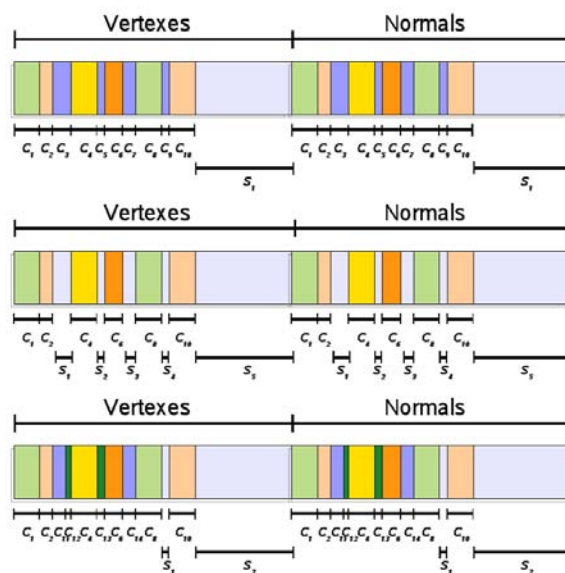


Figure 3: (a) Situación del VBO después de la extracción inicial de la iso-superficie. Cada segmento C_n marca una zona ocupada por la geometría de una celda. (b) Huecos marcados como S_n sobre el VBO después de eliminar las celdas modificadas. (c) Estado del VBO después de añadir la geometría de las nuevas celdas procesadas.

Durante la extracción de la iso-superficie, el *VBO* va almacenando toda la geometría sin dejar ningún hueco en la estructura, tal y como se aprecia en la figura 3 (a). No obstante, como ya se ha descrito, al aplicar la herramienta al modelo

algunas celdas dejan de ser atravesadas por la iso-superficie debido a la asignación de valores de la herramienta, o simplemente se modifican y su geometría deja de ser válida. El espacio ocupado por esta información se considera a partir de ese momento como huecos libres. En la figura 3 (b) se observa cómo después de realizar el recorrido del *octree* la estructura contiene huecos sin utilizar.

Posteriormente, cuando se procede a procesar las celdas del conjunto *C*, se utiliza una búsqueda de huecos candidatos mediante un algoritmo del *primer ajuste* para insertar la geometría extraída a partir de las nuevas celdas en los huecos. Esta estrategia es produce poca o nula fragmentación en el *VBO* debido a que el número de triángulos de las celdas es muy similar. No obstante, como se muestra en la figura 3 (c), es muy probable que después de procesar las celdas del conjunto *C* todavía queden huecos en la estructura. Esto no es problema ya que la biblioteca de *OpenGL* proporciona funciones para visualizar por secciones el *VBO*. No obstante, es necesario manejar tanto los huecos de espacio libre como las secciones de información, por lo que hemos optado por usar un mapa de espacio libre.

4. Resultados y conclusiones

A continuación mostramos algunas imágenes capturadas durante una sesión de modelado sobre objetos volumétricos. En los ejemplos mostrados se utilizan modelos con una resolución de 256^3 muestras. En la figura 4 se puede ver el resultado de una operación de sustraer material y otra de añadir material sobre un modelo inicial formado por un cubo de las dimensiones antes mencionadas.

También en la figura 5 se puede ver el resultado de una sesión de modelado en la que se pretende obtener un objeto parecido a una granada.

Hemos presentado una representación que permite aplicar la técnica de escultura virtual cumpliendo los requisitos de abstracción de la representación subyacente y tiempos interactivos de respuesta frente a la interacción con herramientas virtuales de modelado.

La utilización combinada de la capacidad de regeneración adaptativa de la representación solamente en las zonas en las que se aplican las herramientas, junto con el mantenimiento en la memoria de la tarjeta gráfica de la geometría extraída permite unos tiempos de respuesta interactivos con volúmenes de información elevados y operaciones de edición que conllevan zonas de modificación con gran cantidad de muestras.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnología y los fondos FEDER, a través del proyecto TIN2007-67474-C03-02 y por la Consejería de innovación Ciencia y Empresa de la Junta de Andalucía a través del proyecto de excelencia TIC-401.

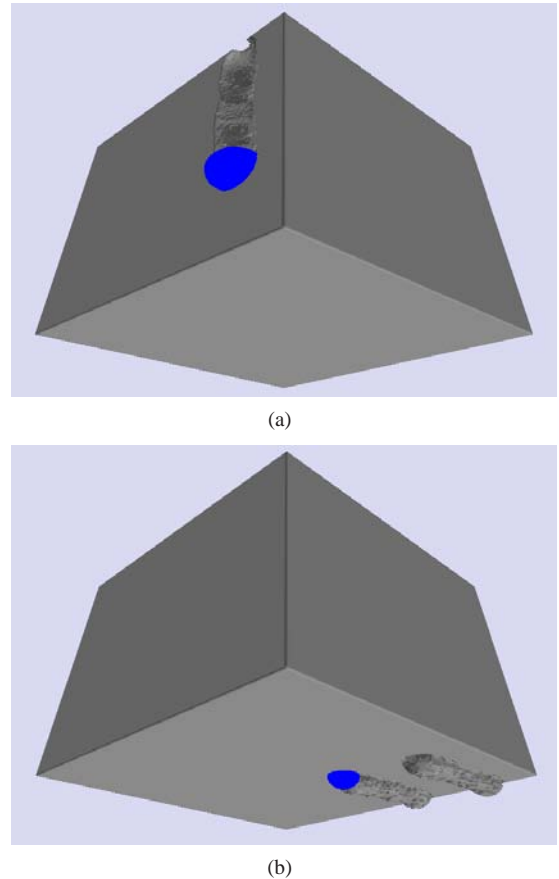


Figure 4: Las imágenes muestran el resultado de aplicar sucesivamente operaciones de sustraer material (arriba) y añadir material (abajo).

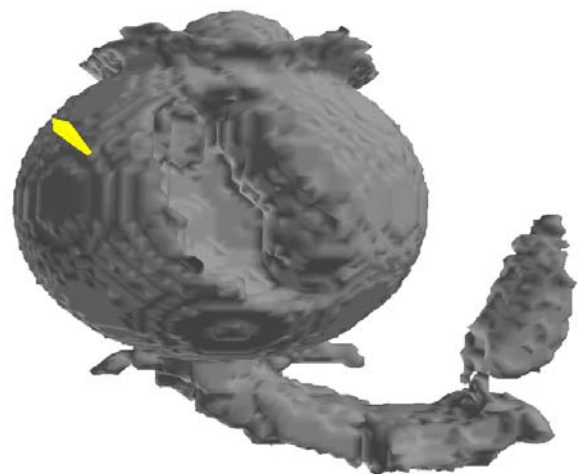


Figure 5: Resultado de modelar una granada a partir de una esfera inicial.

References

- [AS96] AVILA R. S., SOBIERAJSKI L. M.: A haptic interaction method for volume visualization. In *Proc. Visualization'96* (October 1996), pp. 197–204.
- [Bae98] BAERENTZEN A.: Octree-based volume sculpting. In *Proc. Visualization'98* (1998).
- [FCG00] FERLEY E., CANI M.-P., GASCUEL J.-D.: Practical volumetric sculpting. *The Visual Computer* 16, 8 (December 2000), 211–221.
- [FCG02] FERLEY E., CANI M.-P., GASCUEL J.-D.: Resolution adaptive volume sculpting. *Graphical Models* 63, 6 (March 2002), 459–478.
- [FPRJ00] FRISKEN S., PERRY R., ROCKWOOD A., JONES T.: Adaptively sampled distance fields: A general representation of shape for computer graphics. In *SIGGRAPH 2000 ACM Press/ACM SIGGRAPH* (July 2000), pp. 249–254.
- [GH91] GALYEAN T. A., HUGHES J. F.: Sculpting: An interactive volumetric modeling technique. *ACM Computer Graphics* 25, 4 (July 1991), 267–274.
- [JLSW02] JU T., LOSASSO F., SCHAEFER S., WARREN J.: Dual contouring of hermite data. *ACM Transactions on Graphics* 21, 3 (July 2002), 339–346.
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH'87–ACM Computer Graphics* 21, 4 (July 1987), 163–169.
- [LTV07] LEÓN A. J., TORRES J. C., VELASCO F.: Octree de volumen con celdas duales implícitas. In *XVII Congreso Español de Informática Gráfica CEIG'07* (September 2007), Cerezo E., Feito F. R., (Eds.), pp. 189–198.
- [PF01] PERRY R., FRISKEN S.: Kizamu: A system for sculpting digital characters. In *SIGGRAPH 2001 ACM Press/ACM SIGGRAPH* (August 2001), pp. 47–56.
- [RE00] RAVIV A., ELBER G.: Three-dimensional free-form sculpting via zero sets of scalar trivariate functions. *Computer-Aided Design* 32 (August 2000), 513–526.
- [WK95] WANG S. W., KAUFMAN A. E.: Volume sculpting. In *Symposium on Interactive 3D Graphics* (April 1995), pp. 151–156.