CEIG'08, Barcelona, Sept. 3–5 (2008)
L. Matey and J. C. Torres (Editors)

Simplificación de Mallas con Tetra-trees aplicado a Entornos de Interacción con Dispositivos Hápticos

A. Martínez, J. J. Jiménez, F. Feito, R. Segura [1]

[1]Departamento de Informática, Universidad de Jaén, Spain

Abstract

La complejidad inherente al proceso de interacción entre objetos hace necesaria la búsqueda de soluciones eficientes que reduzcan su complejidad. Un enfoque es la utilización de estructuras jerárquicas de niveles de detalle para pre-procesar el modelo y optimizar la detección de colisión, dentro de este campo una alternativa es la utilización de tetra-trees. Los tetra-trees son árboles de tetra-conos que actúan como superficie envolvente simplista de la malla original y permiten una disminución de la carga computacional en la detección de la colisión. Este enfoque de estructura jerárquica simplista con niveles de detalle se ha aplicado en la interacción humano-escena 3D con dispositivos específicos como son los hápticos. La utilización de hápticos permite evaluar en condiciones reales las características de la estructura desarrollada.

Keywords: Estructuras jerárquicas de detalle, Hápticos, Interacción, Tetra-Tree.

1. Introducción y Trabajos previos

En este artículo se presenta la aplicación de una nueva estructura jerárquica con niveles de detalle, los tetra-trees, en la interacción en tiempo real con dispositivos interactivos humano-escena 3D como son los hápticos. Se ha utilizado una descomposición espacial basada en tetra-conos para clasificar los triángulos de las mallas de objetos 3D. Esta estructura permite una simplificación de la escena y una mejora de la interacción en tiempo real entre dispositivos hápticos y la escena 3D.

La interacción entre objetos es un problema bien estudiado [GASF, JTT01, Eri05]. Habitualmente los sistemas de interacción dependen de la representación de los objetos, siendo este un factor importante a la hora de aplicar distintas técnicas [LG98]. La mayor parte de los algoritmos tratan con objetos convexos formados por polígonos convexos o mallas de triángulos. Entre otros, se pueden destacar el algoritmo de las características más cercanas [LC91, Mir98], el algoritmo GJK [GJK88] y el algoritmo del vector de separación [CW96, Chu96]. En el caso de objetos no convexos es habitual realizar una descomposición previa en partes convexas.

En la mayor parte de los métodos de interacción se utiliza adicionalmente algún tipo de descomposición espacial como Octrees [ABO*97] o BSP-Trees [NAT90] entre otros para clasificar las características de los objetos y reducir la complejidad de los mismos. Para el mismo fin se suelen utilizar diversos tipos de jerarquías de volúmenes envolventes como AABB-Trees [Ber97], OBB-Trees [GLM96], Box-Trees [BCG*96, Zac98], Sphere-Trees [Hub96, BO04] o k-DOP-Trees [KHSZ98].

En nuestro caso el háptico, aparte de permitir la interacción con la escena 3D, permite la definición y asignación de vectores de fuerza a los movimientos y colisiones lo que conlleva una mayor naturalidad a la hora de interactuar con la escena.

2. Metodología. Tetra-Cono y Tetra-Tree

El concepto de tetra-cono y tetra-tree queda formalmente definido tal y como sigue:

Sea $T = V_0V_1V_2V_3$ un tetraedro con orientación positiva. Se define como tetra-cono asociado al tetraedro T, a la región del espacio delimitada por los planos soporte de $V_0V_1V_2$, $V_0V_2V_3$ y $V_0V_3V_1$, de manera que T es interior a dicha región [Figura 1].

Se define un árbol de tetra-conos o Tetra-Tree como

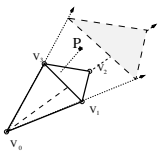

Figure 1: *Estructura general de un tetra-cono.*

una estructura de datos jerárquica en forma de árbol de cuyo nodo raíz parten ocho tetra-conos hijos (T_0, T_1, T_2, T_3, T_4, T_5, T_6, T_7), todos con origen común y que cubren la totalidad del espacio sin solapamientos entre ellos, salvo en la frontera. Cada uno de estos ocho tetra-conos de primer nivel se descompone en cuatro tetra-conos hijos, de manera que el tetra-cono T_i tendrá cuatro tetra-conos de siguiente nivel (T_{i_0},T_{i_1},T_{i_2},T_{i_3}). Del mismo modo, cada tetra-cono se subdivide recursivamente en cuatro sub-tetra-conos, hasta alcanzar un criterio de parada predefinido. Esta estructura es rápida en tiempos de construcción en comparación con algunas de las construcciones simplístas clásicas como los Octrees [[Jim06]] y permite un tratamiento eficiente de la detección de la colisión [[JFSO06]].

Para clasificar los triángulos de la malla en cada uno de los tetra-conos se utilizará como origen de los tetra-conos el centroide del poliedro. La inclusión de un triángulo en un tetra-cono se ha resuelto mediante el cálculo del signo de las coordenadas baricéntricas [Eri05] de los vértices del triángulo respecto al tetraedro asociado al tetra-cono.

2.1. Tetraedro envolvente asociado a un Tetra-Cono

Asociado a cada tetra-cono se define un tetraedro envolvente [Figura 2] que pondrá límite a la geometría del poliedro contenida en el tetra-cono [Figura 3].

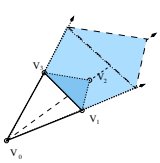

Figure 2: *Tetra-cono envolvente y zona exterior a un tetra-cono.*

Dada F una malla de triángulos con tetra-tree asociado TTF. Sea $T = V_0V_1V_2V_3$ un tetraedro tal que $T \in TTF$. Se define el tetraedro envolvente de la geometría de F contenida en T como el tetraedro $T_{env} = V_0V_1'V_2'V_3'$ tal que:

1. V_1' está sobre la semi-recta definida por V_0V_1.
2. V_2' está sobre la semi-recta definida por V_0V_2.
3. V_3' está sobre la semi-recta definida por V_0V_3.
4. Todo vértice V_i de los triángulos de F clasificados en T se encuentran en el mismo lado que V_0 respecto al plano soporte de $V_1'V_2'V_3'$.

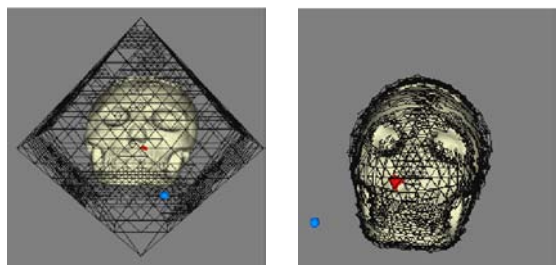

Figure 3: *Tetra-tree no envolvente y tetra-tree envolvente.*

Dado un tetra-cono, existe más de un posible tetraedro envolvente, por lo que se utiliza el tetraedro envolvente que se ajuste a los triángulos de la malla clasificados en su tetra-cono asociado. Se pueden aplicar distintos criterios y funciones a la hora de ajustar los tetra-conos a la malla para generara el tetraedro envolvente, en este caso se ha optado por utilizar una función que determina el plano más cercano a la submalla contenida en cada tetracono basado en las proyecciones de los puntos de los triángulos en el eje formado por el baricentro del triangulo exterior del tetraconos y el centroide del tetra-tree.

3. Resultados obtenidos

Se ha probado la generación de tetra-trees y clasificación de triángulos en distintas mallas. Sobre estas mallas se ha estudiado la simplificación aplicando los tetra-trees, con varios niveles de detalle [Figura 4].

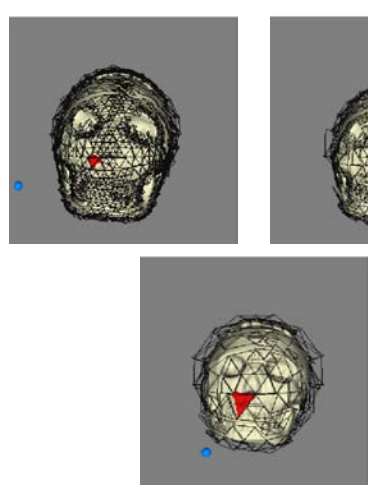

Figure 4: *Tetra-tree asociado a una malla de 40000 triángulos. Umbrales de de 64, 128 y 512 triangulos contenidos en cada tetra-cono respectivamente (de arriba a abajo y de izquierda a derecha.*

Estas mallas se han introducidas en aplicaciones OpenGL conjuntamente con la estructura del tetra-tree obtenido para aplicarlo en la detección de colisión del actuador del háptico con la malla. Inicialmente se presenta la malla (objeto blanco), el actuador (objeto azul) y el tretra-tree generado con una condición de parada tal que el número máximo de triángulos contenidos en cada tetra-cono es de 64 triángulos. Se ha dibujado en rojo los triangulos asociados al tetra-cono que contiene al actuador del háptico [Figura 5]. Cuando ocurre la colisión entre el actuador y la submalla, los triángulos cambian a verde. Destacar que el estudio de la colisión primero se realiza sólo entre el actuador y el triangulo exterior del tetra-cono envolvente activo y cuando el actuador atraviesa este triangulo exterior, se realizar la detección de colisión sólamente entre el actuador y los triángulos contenidos en el tetra-cono. Para generar la respuesta de fuerza al usuario se ha utilizado la definición de fuerzas de la librería del dispositivo háptico.

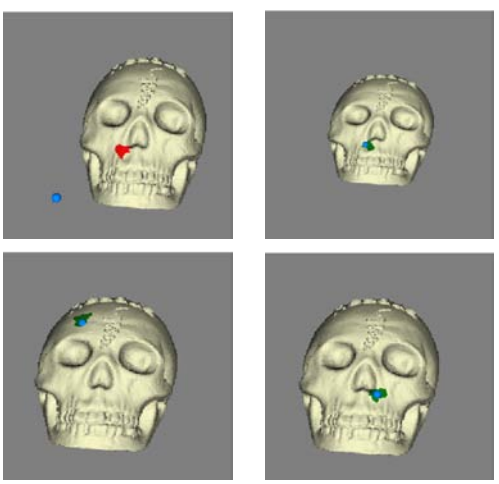

Figure 5: *Detección del tetra-cono activo y la submalla incluida en eventos de colisión y no colisión.*

Se ha realizado un estudio de los tiempos de construcción y el número de tretra-conos generados en relación con el umbral de triángulos máximos contenidos en cada tetracono [Tabla 1] dentro de la aplicación de interacción. También se ha estudiado la tasa de frames por segundo obtenido para cada tetra-tree al interactuar con el háptico para conseguir escenas de interacción en tiempo real, en nuestro caso la tasa es siempre mayor de 100 fps [Tabla 2]. Las pruebas has sido realizadas en un ordenador Intel *Core2 Quad* a 2.4 Mhz con una GPU compuesta por dos tarjetas Nvidia GForce 8800GT unidas en arquitectura SLI, 4GB de RAM, visualizando el resultado a una resolución de 1680 x 1050 ppp. El dipositivo háptico utilizado es *PHAMTOM Omni Haptic Device* ®de la empresa *Sensable* ™conectado al equipo por puerto *firewire*.

Umbral	Tiempo Construcción	Tetra-conos generados
2048	0.7 s.	39
1024	1.2 s.	87
512	1.01 s.	207
256	1.27 s.	489
128	1.4 s.	996
64	1.58 s.	2268
32	1.84 s.	7456
16	3.12 s.	20324

Table 1: *Tiempos de construcción y tretra-conos generados para una malla de 40000 triángulos y 20002 puntos. El umbral corresponde al número máximo de triángulos contenidos en cada tetra-cono*

Tetra-conos generados	FPS
2048	101
1024	105
512	110
256	110
128	110
64	111
32	107
16	101

Table 2: *Tetra-conos generados y fps para una malla de 40000 triángulos y 20002 puntos.*

4. Conclusiones y Trabajos Futuros

En este proyecto se ha aplicado una forma novedosa de simplifiar las mallas de objetos aplicando una superficie simplista basada en tetra-conos. Esta superficie simplista se estructura en una forma arborescente para opitmizar la búsqueda y acceso a los elementos contenidos así como la posibilidad de definir niveles de detalle en relación con la profundidad. Esta estructura se ha probado en entornos de trabajo reales, utilizando dispositivos de interacción humano - escena 3D, cómo son los hápticos. Los hápticos son herramientas de interacción muy versátiles que permiten aplicar la interacción entre humano y escena3D a multitud de campos, tanto en la simulación de condiciones cómo en la interacciones realista con la aplicación de fuerzas. La combinación de nuestra estructura con el dispositivo de interacción permite una eficiente gestión de la detección de la colisión en situaciones reales.

El desarrollo de mallas específicas y el estudio de las condiciones de los materiales con los que interactuar, mejorarán la experiencia de interacción. La comunicación entre las aplicaciones desarrolladas con entornos reales permite la utilización de escenas 3D que sean reproducciones reales del escenario de trabajo. La optimización de los algorirmos de detección de la colisión y su desarrollo a nivel de GPU es otro de las lineas de continuación de estre trabajo.

5. Agradecimientos

Este trabajo ha sido financiado parcialmente por el Ministerio de Educación y Ciencia y la Unión Europea (a través de los fondos FEDER) por medio del proyecto de investigación TIN2007-67474-C03-03, y por la Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía por medio de los proyectos de investigación P06-TIC-01403 y P07-TIC-02773.

References

[ABO*97] AVANAIM F., BOISSONNAT J., O.DEVILLERS, PREPARATA F., YVINEC M.: Evaluating sings of determinants using simple-precision arithmetic. *Algorithmica* (1997).

[BCG*96] BAREQUET G., CHAZELLE B., GUIBAS L., MITCHELL J., TAL A.: Boxtree: A hierarchical representation of surfaces in 3d. *Proceedings of Eurographics'96.* (1996).

[Ber97] BERGEN G.: Efficient collision detection of complex deformable models using aabb trees. *Journal of Graphics Tools* (1997).

[BO04] BRADSHAW G., O'SULLIVAN C.: Adaptative medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics* (2004).

[Chu96] CHUNG K.: *An efficient collision detection algorithm for polytopes in virtual environments*. PhD thesis, University of Hong Kong, 1996.

[CW96] CHUNG K., WANG W.: Quick collision detection of polytopes in virtual environments. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST 96)* (1996).

[Eri05] ERICSON C.: *Real-Time Collision Detection*. Morgan Kaufmann, 2005.

[GASF] GARCÍA-ALONSO A., SERRANO N., FLAQUER J.: Solving the collision detection problem. In *IEEE Computer Graphics and Applications*.

[GJK88] GILBERT E., JOHNSON D., KEERTHI S.: A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics and Automation* (1988).

[GLM96] GOTTCSHALK S., LIN M., MANOCHA D.: Obb-tree: A hierarchical structure for rapid interference detection. *Proceedings of the ACM SIGGRAPH 96* (1996).

[Hub96] HUBBARD P.: Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics* (1996).

[JFSO06] JIMENEZ J., FEITO F., SEGURA R., OGAYAR C.: Particle oriented collision detection using simplicial coverings and tetra-trees. *Computer Graphics Forum* (2006).

[Jim06] JIMENEZ J.: *Detección de Colisiones Mediante Recubrimientos Simpliciales*. PhD thesis, Dpto. de Informática - Universidad de Jaén, 2006.

[JTT01] JIMÉNEZ P., THOMAS F., TORRAS. C.: 3d collision detection: a survey. *Computer and Graphics* (2001).

[KHSZ98] KLOSOWSKI J., HELD M., SOWIZRAL J. M. H., ZIKAN K.: fficient collision detection using bounding volume hierarchies of k-dops. *EEE Transactions on Visualization and Computer Graphics* (1998).

[LC91] LIN M., CANNY J.: Efficient algorithms for incremental distance computation. *IEEE Conference on Robotics and Automation* (1991).

[LG98] LARSE E., GOTTSCHALK S.: Collision detection between geometric models: A survey. *IMA conference on Mathematics of Surface* (1998).

[Mir98] MIRTICH B.: V-clip: Fast and robust polyhedral collision detection. *ACM Transactions on Graphics* (1998).

[NAT90] NAYLOR B., AMANATIDES J., THIBAULT W.: Merging bsp trees yield polyhedral modeling results. *Proceedings of ACM Siggraph* (1990).

[Zac98] ZACHMANN G.: Rapid collision detection by dynamically aligned dop-trees. *Proceedings of IEEE Virtual Reality Annual International Symposium VRAIS'98* (1998).