

# CAPITULO 2. ALGORITMOS DE DETECCIÓN DE COLISION BASADOS EN ESTRUCTURAS JERARQUÍCAS

## 2.1 ESTRUCTURAS JERARQUICAS

El objetivo principal de los algoritmos de detección de colisión es calcular las interacciones geométricas entre los objetos, sin importar el número y complejidad que los objetos puedan tener.

Los algoritmos de detección de colisión tradicionales han requerido de una gran cantidad de pruebas de intersección geométrica, verificando si todos los polígonos que modelan la superficie de un objeto intersectan a algún polígono del otro objeto, determinando de esta manera si dos objetos colisionan [O'S99].

La mayoría de los investigadores en el área proponen algoritmos que reduzcan el número de llamados para verificar la intersección entre dos primitivas geométricas, estas técnicas plantean un tipo de volúmenes envolventes, organizados en una estructura jerárquica y con ello evitar lo más posible la verificación de los pares de primitivas geométricas.

Afortunadamente, estos algoritmos han demostrado ser efectivos y en la mayoría de las aplicaciones solamente se verifica una pequeña fracción de pares de primitivas [YEN98]. Donde dado dos o mas modelos geométricos con estas representaciones, la meta es verificar si existe una intersección entre ellas, si los sistemas jerárquicos presentan intersección, entonces se aplica una prueba de verificación más precisa. [LIN96, YEN98]

Las estructuras jerárquicas usadas en la detección de colisión incluyen árboles de conos, árboles k-d y octrees, árboles de esferas[PHI95, PHI96], árboles R y sus variantes, árboles basados en S-Bounds, y representaciones basadas en BSP, etc., estos métodos llevan a cabo muy bien las pruebas de rechazo, cuando dos objetos están suficientemente separados [LIN96].

Las estructuras jerárquicas han demostrado ser herramientas eficientes en el cálculo de aproximación de objetos. La propiedad que satisfacen es que cada nodo envuelve a toda la geometría de sus subárboles.

Las estructuras jerárquicas se dividen en dos grupos: La jerarquía de la subdivisión de espacio y la jerarquía de la subdivisión del objeto.

En la jerarquía de la subdivisión del espacio, el ambiente se subdivide en una jerarquía espacial. Los objetos en el ambiente son agrupados jerárquicamente según las regiones en que ellos se encuentran. Cuando un objeto en el ambiente cambia su posición y se mueve a otra región, sólo los objetos dentro de la nueva región serán los que se verifiquen por colisión con el objeto entrante. Hay muchas maneras de subdividir el ambiente: En octrees, en un quadtree para la determinación rápida de objetos visibles, en un árbol K-d (árbol binario equilibrado), en un árbol BSP [LIN96].

En la jerarquía de la subdivisión del objeto, cada objeto se subdivide en una jerarquía envolvente, es decir, en una jerarquía de volúmenes envolventes. Para verificar la colisión entre dos objetos utilizando este tipos de jerarquías, es necesario determinar si las jerarquías envolventes de los objetos se intersectan (o traslapan) unas con otras. Los métodos de este tipo de acercamiento incluyen el uso de las esferas [PHI95,QUI94], AABB (axis-aligned bounding boxes) [COH95]. Aunque ambos métodos son eficaces en determinar si dos jerarquías envolventes se intersectan, ellos no producen volúmenes envolventes ajustados. En el caso de AABB, los volúmenes envolventes pueden necesitar ser recalculados, aún si el objeto sólo realiza una transformación rígida como la rotación. Otro método son las cajas envolventes orientadas (OBB)[GOT96], la idea de este método es encontrar la mejor orientación apropiada de cada caja envolvente para minimizar el volumen envolvente, y no requiere que las jerarquías envolventes sean reconstruidas cuando el objeto realiza una transformación rígida. Otros métodos de la jerarquía de la subdivisión del objeto incluyen el uso de DOP [KLO98] y la combinación de conchas esféricas (spherical shells) [KRI98].

## **2.2 ESTRUCTURAS JERARQUICAS BASADAS EN ESFERAS**

El enfoque para aproximar la detección de colisiones entre dos objetos, es la localización de intersecciones entre representaciones simples de la superficie de los objetos

Una de las formas simples para comprobar la intersección es la esfera. Un par de esferas se intersectan si la distancia entre sus centros es menor que la suma de sus radios. Las operaciones que involucran la intersección entre esferas son menos costosas que las utilizadas por el método de Fuerza Bruta (Triángulo VS Triángulo).

Los algoritmos de detección de colisión basados en jerarquías de esferas, no cambian la modelación de los objetos sino utilizan las esferas para cubrir la superficie de los objetos, y con ellas resolver el problema de detección de colisiones más rápidamente.

La jerarquía de esferas se utiliza para dividir la superficie del objeto en varias regiones y a través de ellas poder dirigir la detección de la colisión hacia el área de la superficie del objeto donde probablemente existe intersección con el otro objeto.

La construcción de un árbol de esferas requiere de generar múltiples niveles de esferas que se aproximan a la superficie del objeto. El reto particular de la utilización de esferas es hacer que cada nivel se ajuste adecuadamente al objeto, este ajuste es necesario porque de no ser así, la aplicación podría generar una respuesta de colisión.

Philip M. Hubbard dirige su trabajo en esta dirección, proponiendo algoritmos basados en la construcción de esferas, en uno de ellos [PHI96], plantea la construcción de árbol jerárquico de esferas basado en octree para ajustar las esferas alrededor del objeto, el otro algoritmo [PHI96] es la construcción del árbol de esferas usando superficies con ejes centrales. Un área con eje central corresponde al esqueleto del objeto, la construcción exacta de superficies con eje central para objetos en 3D es difícil, pero J. A. Goldak [GOL91], propone un algoritmo que produce aproximaciones satisfactorias, Philip M. Hubbard retoma esta idea y propone algunas extensiones a este algoritmo. Bradshaw, G. y O'Sullivan [BRA04] han trabajado con esta idea y proponen también cambios sobre el algoritmo, ellos basan su trabajo alrededor de un "*Sphere Reducer*" [BRA03]. Los algoritmos "*Sphere Reducer*" producen un conjunto de esferas para aproximar a una sub-sección de el volumen del objeto El algoritmo *Sphere Tree Generator* controla la construcción de el árbol de esfera y dirige como el objeto es subdividido. Cuando un conjunto de esferas es construido el generador usa las esferas para dividir al objeto en un número e regiones, cada una de las cuales es entonces aproximada usando un algoritmo reductor de esferas. [BRA03].

La meta principal de este tipo de algoritmos basados en árboles de esferas, es descartar rápidamente las posiciones libres de colisión dentro del ambiente. Y concentrar su trabajo en aquellas donde probablemente se presente una colisión y así reducir su tiempo de detección. Podemos concluir que la clave para la detección de colisiones en tiempo real, es el método que permita detectar más rápidamente cuando dos objetos no presentan colisión.

## 2.3 MÉTODO DE SEAN QUINLAN

Sean Quinlan [QUI94] utiliza representaciones envolventes basadas en esferas, estas esferas pueden ser especificadas con un vector de posición y un radio, el cálculo de la distancia entre dos esferas no es muy costoso.

Su estructura jerárquica consiste en una aproximación de árbol binario aproximado, donde cada nodo contiene una esfera y el árbol tiene las siguientes dos propiedades. La unión de todas las esferas contienen completamente la superficie de el objetos y la esfera de cada nodo contiene las esferas de sus nodos descendientes.

El primer paso para construir el árbol es cubrir la superficie con esferas pequeñas. Estas esferas serán los nodos-hojas del árbol. El número de esferas es igual al número de polígonos de la malla que modela al objeto. Cada una de las esferas es etiquetada con el polígono para el que fue creada.

Después de cubrir el objeto con las esferas pequeñas utiliza la estrategia de "Divide y Vencerás" para construir los nodos del árbol. El conjunto de nodos-hojas es dividido en dos subconjuntos aproximadamente iguales, y construye un árbol para cada subconjunto y estas son combinadas dentro de un árbol para crear un nuevo nodo con cada uno de los subárboles como hijos. Los subárboles se construyen de manera recursiva hasta que el conjunto consista en un nodo hoja.

Para dividir el conjunto de nodos en dos conjuntos, se calcula la caja rectangular envolvente que esta alineada a las coordenadas del objeto y que contienen los centros de todas las esferas hojas. Posteriormente se selecciona el eje a lo largo de la parte más larga de la caja, y se dividen los nodos usando el valor promedio a lo largo de este eje como discriminante. Cada uno de los dos subárboles resultantes deberá ser lo bastante compacto y contendrá aproximadamente el mismo número de elementos.

Después de dividir el conjunto en dos subconjuntos, se construyen los árboles para cada subconjunto, llamando de manera recursiva al algoritmo que creará un nuevo nodo con los dos subárboles como hijos.

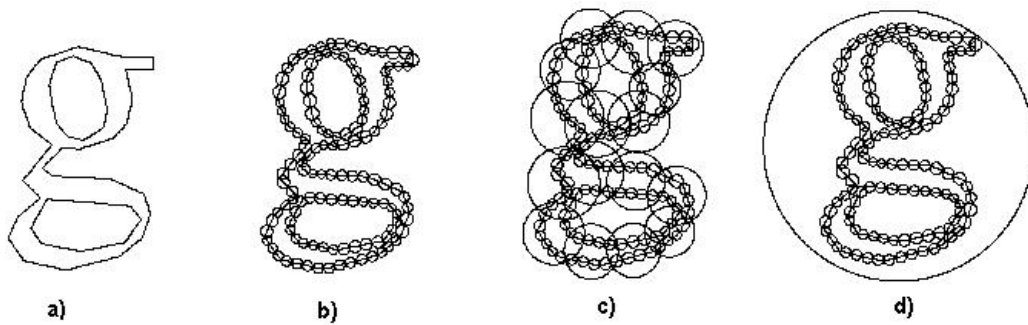


figura 2.1 Algunos niveles de detalle del árbol de esferas<sup>1</sup>

La figura 2.1 en el inciso a) muestra la forma del objeto como una letra “g” al cual se le calcula su estructura, el inciso b) muestra las 130 esferas hojas que cubren la superficie del objeto, el inciso c) muestra el quinto nivel y el inciso d) muestra la esfera raíz.

Para determinar la esfera que envuelve a dos esferas, utiliza dos métodos heurísticos y usa la menor de las dos esferas generadas por estos métodos. El primer método consiste en encontrar una esfera envolvente que contenga las esferas de los 2 nodos hijos y así por inducción todas las esferas-nodos descendentes. El segundo método primero selecciona un centro para la esfera envolvente y entonces examina cada una de las esferas descendentes para determinar la que requiere menor radio. La selección del centro de la esfera envolvente se hace obteniendo la posición promedio de los centro de las esferas hojas. [QUI94]

## 2.4 MÉTODO DE PHILIP M. HUBBARD

El algoritmo de Philip M. Hubbard, es un método para aproximar objetos que apoyen a algoritmos de detección de colisión en tiempo-critico. Las aproximaciones son jerarquías de esferas, las cuales permiten que un algoritmo de tiempo crítico refine progresivamente la exactitud de su detección, deteniéndose cuando sea necesario para mantener el funcionamiento esencial para aplicaciones interactivas en tiempo real.

La clave para la detección de colisión en tiempo-critico se basa en un método que aproxime automáticamente la superficie de un objeto. Esta aproximación consiste

<sup>1</sup> Imagen Sean Quinlan

en construir las estructuras jerarquías que modelen en forma ajustada la superficie e los objetos durante una etapa de preprocesamiento [PHI96].

El preprocesamiento aproxima a los objetos con un conjunto de esferas, este produce múltiples niveles de detalle arreglados en una jerarquía de esferas, como lo muestra la figura 2.2.

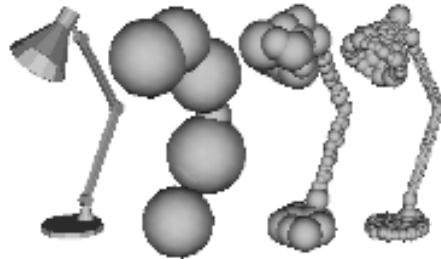


Figura 2.2 Una lámpara con 626 Triángulos y tres niveles de detalle usando esferas <sup>2</sup>

Para acomodar las esferas, el preprocesamiento utiliza una superficie de eje central, la cual representa a un objeto en forma esquelética. La superficie de eje central guía a un proceso de optimización que asocie las esferas a la forma del objeto.

El algoritmo que construye la jerarquía de esferas a partir de una superficie de eje central debe primero construir la superficie de eje-central. Una forma de aproximar superficies de eje-central es utilizando un diagrama de Voronoi [FRA85, PHI96]. El diagrama de Voronoi para un conjunto discreto de puntos identifica para cada punto, la región más cercana a él que a cualquiera de los otros puntos. Las regiones son llamadas celdas de Voronoi. Para puntos en 2D las celdas son polígonos convexos y para puntos en 3D son poliedros convexos. Cada cara de una celda es equidistante entre dos puntos en el conjunto discreto. Así que para un conjunto de puntos  $P$  en la superficie de un poliedro, las celdas del diagrama de Voronoi tienen caras que están aproximadamente en el eje central. Como lo muestra la figura 2.3.

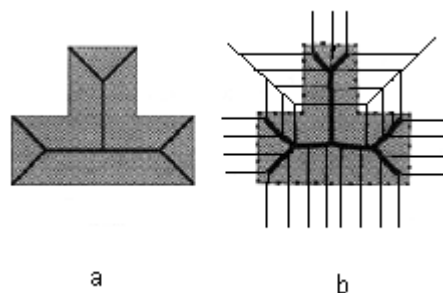


figura 2.3. a) Las líneas gruesas son el eje central en 2D  
b) Un diagrama de Voronoi aproximado al eje central<sup>3</sup>

<sup>2</sup> Imagen de Philip M. Hubbard

<sup>3</sup> Imagen de Philip M. Hubbard

El primer paso es colocar el conjunto de puntos  $P$  en la superficie del poliedro y el segundo paso es construir un diagrama de Voronoi para los puntos. Cada vértice de Voronoi es el centro de una esfera en la cual están 4 puntos del conjunto  $P$ . Los cuatro puntos que están asociados con un vértice son sus puntos formantes. Las esferas centradas en los vértices de Voronoi y sus puntos asociados son la base de un algoritmo para construir la jerarquía a partir de superficies de ejes centrales.

La unión de dos esferas  $e_1$  y  $e_2$  involucra reemplazarlas con una nueva esfera  $e_{12}$ , que cubra las partes del poliedro que ellas cubren. El preprocesamiento aproxima lo que una esfera cubre en términos de sus puntos formantes. Así  $e_{12}$  será la esfera envolvente para los puntos formantes asociados a  $e_1$  y  $e_2$ . Producir un nivel de detalle requiere repetidamente unir un par de esferas.

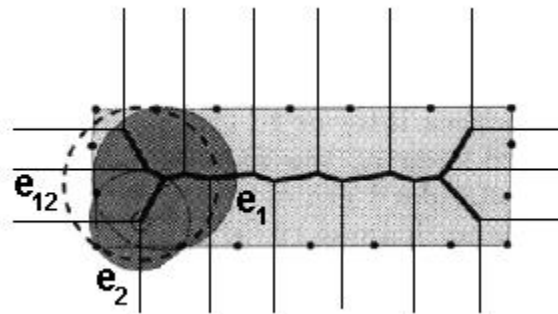


figura 2.4. La unión de las esferas  $e_1$  y  $e_2$  forman la esfera  $e_{12}$ .<sup>4</sup>

Para construir toda la jerarquía se requiere que el preproceso aplique repetidamente la técnica de unión de esferas. Para un árbol de esferas, el preproceso primero une todas las esferas centradas en los vértices del Voronoi para producir un número fijo de hijos para la raíz. Subsecuentemente se construirán los nodos-hijos a partir de un nodo-padre. [PHI95, PHI96]

Durante la ejecución de una aplicación, estos niveles de detalle toman el espacio de la superficie del objeto para aproximar más rápidamente la respuesta y detección de colisión.

---

<sup>4</sup> Imagen de Philip M. Hubbard

## 2.5 CONCLUSIONES DEL CAPITULO

El estudio e investigación de los algoritmos de detección de colisión en especial aquellos métodos que trabajan con estructuras jerárquicas, nos ayudó a dirigir el desarrollo de nuestro trabajo.

El utilizar las estructuras jerárquicas y los volúmenes envolventes, basados en figuras geométricas, provocan que el cálculo de la intersección entre ellas sea menos costoso computacionalmente.

La verificación de la intersección entre dos esferas es una operación rápida de realizar, por esta razón se utilizó para modelar las primitivas, no consideremos sustituir con ellas las primitivas, sino sólo nos permitirán agilizar el proceso de verificación de colisión.

El utilizar las estructuras jerárquicas, nos permite aislar las primitivas, donde existe más posibilidad de intersección.

Por ello consideramos los métodos de Sean Quinlan y Philip M. Hubbard, y tomamos algunas de sus características para el desarrollo de nuestro algoritmo, las cuales explicamos más ampliamente en el capítulo 3.