

Un Modelo de Detector de Colisiones Basado en Cajas Rectangulares Externas e Internas

Olmedo Arcila, Simena Dinás, and José María Bañón

Universidad del Valle, Facultad de Ingeniería,
Cali, colombia

{olarcila, simdinas, banon}@eisc.univalle.edu.co
<http://www.univalle.edu.co>

Abstract A hierarchical model is presented for collision detection with convex polyhedra. This model is based on double rectangular representation: the exterior composed of outer boxes and the interior composed of inner boxes. The use of inner boxes allows testing faster overlap detection. The collision detection algorithm has been implemented using C++. Experimental results shown inner boxes improve the collision detection process.

Key words: Collision Detection, Rectangular Representation, Convex Polyhedra

1. Introducción

La detección de colisiones es un problema importante por su alta complejidad computacional en situaciones normales. La búsqueda de colisiones resulta un problema crítico para resolver intersecciones entre un objeto y su entorno así como entre varios objetos. En aplicaciones de tiempo real es deseable y necesario obtener respuestas rápidas [4], [17], [10], [14], [6]. La mayor parte de las librerías de detección de colisiones entre objetos rígidos están basadas en jerarquías de volúmenes limitantes [10], [14], [8].

El objetivo de cualquier detector de colisiones jerárquico es rechazar rápidamente cualquier par de objetos que no se intersectan. Si dos objetos se solapan, se busca una intersección entre ellos solamente mirando si las caras y aristas de un objeto intersectan las caras y aristas del otro objeto [4], [17].

En este artículo, se presenta un detector de colisiones para poliedros convexos basado en una doble jerarquía de volúmenes limitantes: *i*) una está compuesta por cajas rectangulares externas al poliedro y *ii*) la otra por cajas rectangulares internas al poliedro. Con la jerarquía de volúmenes externos se evitan pruebas de intersección complejas entre objetos y con la jerarquía de volúmenes internos se detectan muchas de las colisiones que ocurren. Se presentan pruebas experimentales que muestran la eficiencia y rapidez en la intersección entre cajas rectangulares internas para la detección de colisiones.

La organización del artículo es la siguiente. La sección 2 describe los antecedentes del trabajo. La jerarquía propuesta se presenta en la sección 3. En la

sección 4 se describe el cálculo de la caja máxima inscrita en un poliedro convexo. La sección 5 describe el cálculo de la orientación y del centroide de un poliedro convexo. El algoritmo del detector de colisiones se describe en la sección 6. En la sección 7 se muestran varias pruebas realizadas con el detector de colisiones y se dan los resultados. Finalmente, en la sección 8 y en la sección 9 se da la discusión y las principales conclusiones del trabajo.

2. Antecedentes

En las aplicaciones de tiempo real es necesario y deseable obtener respuestas rápidas [4], [17], [10], [13] y [6]. La mayoría de las librerías de detección de colisiones están basadas en jerarquías de volúmenes limitantes [10], [13] y [8].

Las representaciones jerárquicas más utilizadas por los detectores de colisión utilizan volúmenes limitantes que envuelven un objeto ó a sus partes permitiendo que rápidamente se puedan saber que pares de primitivas no se interseccionan. Existen varios trabajos que utilizan e incorporan representaciones interiores como parte de la jerarquía de volúmenes limitantes [15], [7], [16], [12], [1].

Arcila y Arcila et al. [3] y [2] demuestran la importancia de la intersección entre los volúmenes interiores (Cajas rectangulares, Esferas y Elipsoides) y se enfocan en acelerar el proceso de la detección de colisiones.

3. La Representación Exterior e Interior Basada en Cajas Rectangulares

La caja rectangular es muy usada como volumen limitante externo [6], [9], es fácil de implementar y su almacenamiento no es costoso [9].

En este trabajo se propone la utilización de las cajas rectangulares como volúmenes limitantes externos e internos. Como volumen limitante externo se considera la caja rectangular mínima que envuelve al poliedro. Como volumen limitante interno se considera la caja rectangular máxima inscrita en el poliedro. La jerarquía de volúmenes externos evita pruebas de intersección complejas entre objetos; si los volúmenes limitantes externos no colisionan los objetos que encierran tampoco lo hacen. La jerarquía de volúmenes internos detecta colisiones; si los volúmenes limitantes internos se solapan, los objetos colisionan.

La representación jerárquica considerada se almacena en un árbol binario. En la raíz ó nivel cero del árbol binario se almacena el poliedro original, también se almacena la caja rectangular exterior mínima que envuelve al poliedro original y la caja rectangular interior máxima contenida en el poliedro original.

El poliedro original se subdivide recursivamente en dos subpoliedros hasta una profundidad predeterminada. Los subpoliedros resultantes se almacenan en los nodos del árbol binario junto con las cajas exterior mínima e interior máxima. En general, cada nodo del árbol binario corresponde a un subpoliedro del poliedro original. Los subpoliedros cuyas caras no son parte de las caras del poliedro original y se les etiqueta como INTERIOR, de lo contrario se les etiqueta como

EXTERIOR. Los subpoliedros etiquetados como INTERIOR están contenidos en el interior del poliedro original y no se subdividen. Los subpoliedros etiquetados como EXTERIOR están en los bordes del poliedro original y se subdividen. En las hojas del árbol se almacenan aquellas caras del subpoliedro que corresponden a partes de las caras exteriores del poliedro original. Todas las caras de los poliedros colisionantes son triangularizadas previamente [11].

La subdivisión binaria del poliedro original se realiza mediante un plano tal que: *i*) es perpendicular a la orientación del poliedro y *ii*) pasa por el centroide del mismo. En la sección 5 se explica como se calcula la orientación y el centroide de un poliedro. La caja interior de un poliedro convexo es la caja de volumen máximo inscrita en el poliedro, la cual se calcula mediante un programa de optimización. En la sección 4 se explica el cálculo de la caja máxima inscrita en un poliedro convexo cualquiera.

Inicialmente, la caja exterior de un poliedro es la caja *AABB* que la envuelve. Esta se calcula mediante los valores máximos y mínimos de sus vértices. Después, en el proceso de subdivisión, los subpoliedros hijos adquieren una orientación determinada y la caja exterior se calcula teniendo en cuenta dicha orientación.

La construcción de las jerarquías externas e internas basadas en cajas se realiza en una etapa de preprocesamiento. Las pruebas de intersección entre dos cajas rectangulares arbitrarias han sido calculadas mediante el Teorema del Eje Separador para poliedros convexos [6].

La figura 1(a) muestra, en el plano, un polígono acotado por su caja alineada a los ejes -*AABB*. La figura 1(b) muestra un polígono acotado tanto por el exterior como por el interior con cajas *AABB*.

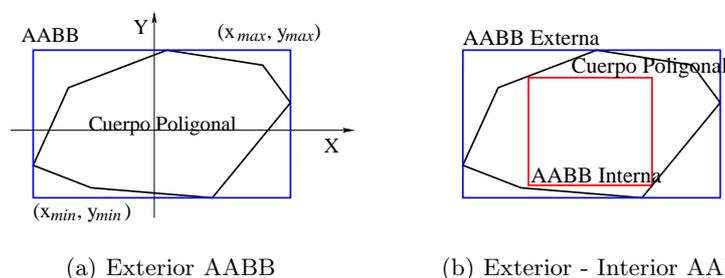


Figura 1. (a) Representación Exterior y (b) Representación Exterior e Interior.

4. Cálculo de la Caja Máxima Inscrita en un Poliedro Convexo

La construcción de una caja interna, de volumen máximo, de un poliedro convexo cualquiera, y alineada con los ejes principales del sistema es un problema

clásico de optimización. Un poliedro \mathcal{P} cualquiera puede definirse como la intersección de un número m finito de semi-espacios, es decir como el subconjunto de \mathbb{R}^3 definido por el subconjunto de las desigualdades lineales:

$$\mathcal{P} = \{\mathbf{v} \in \mathbb{R}^3 : A\mathbf{v} \leq b\}$$

donde $A \in \mathbb{R}^{m \times 3}$ es una matriz de rango 3. Se busca la caja rectangular $\mathcal{R}(\underline{x}, \bar{x})$ de volumen máximo contenida en \mathcal{P} . Donde $\underline{x} = (x_{min}, y_{min}, z_{min})$ y $\bar{x} = (x_{max}, y_{max}, z_{max})$ son las coordenadas de las esquinas diagonalmente opuestas de la caja. El volumen de dicha caja es $\prod_i (\bar{x}_i - \underline{x}_i)$. Entonces bajo estas circunstancias, el problema de optimización para calcular la caja de volumen máximo contenida en el poliedro convexo se plantea de la forma siguiente:

$$\begin{aligned} & \text{maximizar } \prod_i (\bar{x}_i - \underline{x}_i) \\ & \text{sujeto a } \mathcal{R}(\underline{x}, \bar{x}) \subset \mathcal{P} \\ & \quad \underline{x} < \bar{x}. \end{aligned}$$

Teniendo en cuenta que la expresión $\mathcal{R}(\underline{x}, \bar{x}) \subset \mathcal{P}$ se puede escribir como el conjunto de m restricciones

$$A^+ \bar{x} - A^- \underline{x} \leq b$$

donde $A^+ = \max\{0, A_{i,j}\}$ y $A^- = \max\{0, -A_{i,j}\}$. El planteamiento queda de la siguiente forma:

$$\begin{aligned} & \text{maximizar } \prod_i (\bar{x}_i - \underline{x}_i) \\ & \text{sujeto a } A^+ \bar{x} - A^- \underline{x} \leq b \\ & \quad \underline{x} < \bar{x}. \end{aligned}$$

donde \underline{x} y \bar{x} son las incógnitas a calcular.

Para resolver este problema se ha utilizado una librería de optimización llamada **knitro** desarrollada por Richard Waltz en Optimization Technology Center, Northwestern University.

5. Cálculo de la orientación y del centroide de un Poliedro Convexo

Se supone, que el poliedro convexo tiene todas sus caras compuestas por triángulos. Por definición, el centroide de un poliedro es el vector media μ .

$$\mu = \frac{1}{3n} \sum_{i=0}^n (\mathbf{p}^i + \mathbf{q}^i + \mathbf{r}^i)$$

donde n es el número de triángulos que forman las caras del poliedro y los vértices del i -ésimo triángulo son los puntos \mathbf{p}^i , \mathbf{q}^i y \mathbf{r}^i . La matriz de covarianza del poliedro se escribe como la matriz 3×3 con elementos \mathbf{C}_{jk} :

$$\mathbf{C}_{jk} = \frac{1}{3n} \sum_{i=0}^n (\bar{\mathbf{p}}_j^i \bar{\mathbf{p}}_k^i + \bar{\mathbf{q}}_j^i \bar{\mathbf{q}}_k^i + \bar{\mathbf{r}}_j^i \bar{\mathbf{r}}_k^i), \quad 1 \leq j, k \leq 3$$

donde, $\bar{\mathbf{p}}^i = \mathbf{p}^i - \mu$, $\bar{\mathbf{q}}^i = \mathbf{q}^i - \mu$ y $\bar{\mathbf{r}}^i = \mathbf{r}^i - \mu$. Cada uno es un vector 3×1 .

Puesto que la matriz de covarianza \mathbf{C} es simétrica, los vectores propios de \mathbf{C} son ortogonales. Una vez normalizados estos ejes constituyen un nuevo sistema de referencia en el cual la orientación del poliedro está alineada con estos nuevos ejes. En este nuevo sistema coordenado se determina la caja exterior mínima que envuelve al poliedro y alineada con estos ejes [6].

6. El Algoritmo Propuesto

6.1. Esquema General del Algoritmo

Los algoritmos de detección de colisiones basados en jerarquías de volúmenes limitantes externos consideran las representaciones jerárquicas de un par de cuerpos y las recorren simultáneamente, realizando las pruebas de intersección que permitan decidir si hay ausencia de colisión. En el caso que no se pueda decidir con la representación externa si existe ausencia de colisión, se deben realizar las pruebas de intersección entre las caras para determinarlo.

El algoritmo de la detección de colisiones, que se propone en este trabajo, verifica las intersecciones entre las representaciones externas de los poliedros para realizar las pruebas de rechazo rápido. Además verifica las intersecciones entre las representaciones internas para realizar las pruebas de detección de colisiones. En el caso que no se pueda decidir con las representaciones externa e interna, si el cuerpo está en contacto ó no se deben realizar las pruebas de intersección entre las caras. El algoritmo de detección de colisiones, utilizado en este trabajo y basado en las jerarquías dobles [3], consideradas, descritas a continuación.

Sean dos cuerpos poliédricos convexos \mathcal{C}_1 y \mathcal{C}_2 . Sean $\mathcal{A}_{\mathcal{C}_1}$ y $\mathcal{A}_{\mathcal{C}_2}$ los árboles de la representación de cada cuerpo. Sean $n_{\mathcal{C}_1}$ y $n_{\mathcal{C}_2}$ nodos de $\mathcal{A}_{\mathcal{C}_1}$ y $\mathcal{A}_{\mathcal{C}_2}$, respectivamente. Sean x e y dos nodos cualesquiera, y sean las funciones $en_contacto(x, y)$ para determinar si x y y se tocan, $rec_ext(x)$ que obtiene la caja exterior del nodo x , $rec_int(x)$ que obtiene la caja interna del nodo x , $es_hoja(x)$ determina si el nodo x es una hoja del árbol e $hijos(x)$ retorna los hijos del nodo x . La función $interno(n_{\mathcal{C}})$ determina si el nodo $n_{\mathcal{C}}$ está etiquetado como INTERIOR, mientras que la función $externo(n_{\mathcal{C}})$ determina si el nodo $n_{\mathcal{C}}$ está etiquetado como EXTERIOR. Las etiquetas INTERIOR y EXTERIOR asignadas a los nodos en el proceso de construcción de la representación sirven para acelerar el proceso de detección de colisiones.

Algoritmo Recursivo de Detección de Intersecciones

```

interseccion( n_c1:nodo, n_c2:nodo ): string
begin
  if interno( n_c1 ) and interno( n_c2 ) then
    if en_contacto( rec_int( n_c1, n_c2 ) ) then
      return "Hay Contacto"
    if externo( n_c1 ) and externo(n_c2) then
      if es_hoja( n_c1 ) and es_hoja( n_c2 ) then

```

```

    deteccion_exacta( n_c1, n_c2 )
    if es_hoja( n_c1 ) and not es_hoja( n_c2 ) then
        interseccion( n_c1, hijos( n_c2 ) )
    if not es_hoja( n_c1 ) and es_hoja( n_c2 ) then
        interseccion( hijos( n_c1 ), n_c2 )
    else
        interseccion( hijos( n_c1 ), n_c2 )
        interseccion( n_c1, hijos( n_c2 ) )
    if interno( n_c1 ) and externo( n_c2 ) then
        if es_hoja( n_c2 ) then
            deteccion_exacta( n_c1, n_c2 )
        else
            interseccion( n_c1, hijos( n_c2 ) )
    if externo(n_c1) and interno( n_c2 ) then
        if es_hoja( n_c1 ) then
            deteccion_caras_poligonales( n_c1, n_c2 )
        else
            interseccion( hijos( n_c1 ), n_c2 )
    return "No Hay Contacto"
end.

```

Donde la función $deteccion_caras_poligonales(n_{c_1}, n_{c_2})$ calcula la intersección entre las caras de los subpoliedros asociados a los nodos. Si con la jerarquía doble no se puede decidir si hay ó no colisión, las pruebas de intersección entre los triángulos de las caras de los poliedros originales permitan decidirlo. Esto se realiza en las hojas de los árboles de las representaciones. Más adelante se muestra que la introducción de la doble jerarquía basada en volúmenes limitantes externos e internos es un mecanismo de detección de colisiones que permite por sí mismo, rápidamente, decidir la existencia de una colisión entre los cuerpos sin necesidad de llegar a las hojas de los árboles de las representaciones.

El cálculo de la intersección triángulo-triángulo ha sido realizado mediante el método de Moller [11], que consiste en un algoritmo rápido y eficiente basado en las proyecciones escalares de los vértices de los triángulos en la línea de intersección entre los planos de los mismos.

6.2. Esquema del algoritmo de Intersección entre las Caras Poligonales de los Poliedros

Cuando la búsqueda de colisiones no se ha podido realizar por medio de las jerarquías y se llega a un par de hojas, el algoritmo debe buscar una intersección entre las caras externas de los cuerpos contenidos en dichas hojas.

Como se indicó anteriormente, las caras externas de cada subpoliedro están triangularizadas, por lo que la intersección entre cada par de caras se reduce a la la intersección entre los triángulos que conforman las respectivas caras.

Sea $caras_externas(x)$ la función que retorna las caras externas del poliedro subdividido almacenado en el nodo $x = n_{c_j}$ con $j = 1, 2$ de alguno de los árboles que intervienen en el proceso de detección de intersecciones. Sea la función

$triangulos(x)$ la función que retorna los triángulos en los cuales se descompuso la cara x . Sea la función $interseccion_triangulo_triangulo(x,y)$ la función que determina si los triángulos x y y se intersectan o no.

Algoritmo de Intersección entre las Caras de los Poliedros

```

deteccion_caras_poligonales( n_c1:nodo, n_c2:nodo ): string
begin
  c_1 = caras_externas( n_c1 )
  c_2 = caras_externas( n_c2 )
  for i to c_1
    t_1 = triangulos( c_1[i] )
    for j to c_2
      t_2 = triangulos( c_2[j] )
      for k to t_1
        for l to t_2
          if interseccion_triangulo_triangulo( t_1[k], t_2[l] )
            return "Hay Contacto"
          return "No Hay Contacto"
        end
      end
    end
  end
end

```

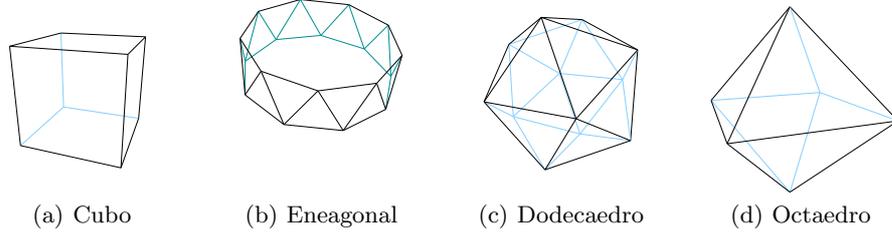
7. Resultados

La implementación y experimentación han sido realizadas en $C++$ usando el compilador $g++(GCC)4,0,3$. Los experimentos se han realizado sobre un computador con 4 procesadores Intel(R) Xeon(TM) $CPU2,80GHz$ sobre Linux Ubuntu 7,0,4 (dapper) y 4GB en la memoria RAM.

Tabla 1. Poliedros

Poliedros	Vértices	Aristas	Caras
Cubo	8	12	6
Octaedro	6	12	8
Dodecaedro	12	30	20
Eneagonal	18	36	20

Se han considerado cuatro poliedros convexos: *i*) el Cubo, *ii*) el Eneagonal, *iii*) el Octaedro, y *vi*) el Dodecaedro, cuyas formas se muestran en la figura 2 y sus características geométricas se indican en la tabla 1. Las colisiones consideradas son: *i*) Cubo - Cubo, *ii*) Eneagonal - Eneagonal, *iii*) Cubo - Eneagonal y *vi*) Dodecaedro - Octaedro. Se han colocado los pares de cuerpos en un espacio de experimentación circular y se han movido aleatoriamente mediante traslaciones y rotaciones. Se ha repetido 100000 veces para obtener tiempos y número de colisiones promedios. Se han considerado dos tamaños diferentes del espacio de

**Figura 2.** Poliedros Convexos**Tabla 2.** Número y Tiempos de las Colisiones

<i>Colisión</i>	<i>Espacio Grande 25 %</i>				<i>Espacio Pequeño 75 %</i>			
	I_{CI}	t_{CI}	I_{CP}	t_{CP}	I_{CI}	t_{CI}	I_{CP}	t_{CP}
Cubo - Cubo	28046	5.8	0	0	76521	4.9	0	0
Cubo - Eneagonal	24749	5.69	654	375.28	73816	3.86	1246	379.59
Eneagonal - Eneagonal	15812	38.56	9259	121.38	54138	27.86	20740	111.17
Dodecaedro - Octaedro	16401	0.66	8503	46.29	57783	0.65	17171	49.01

experimentación que corresponden a probabilidades de colisión de: 25 % y 75 %.

Los resultados obtenidos se resumen en las tablas 2, donde I_{CI} es el número de intersecciones ocurridas entre caras internas e I_{CP} es el número de intersecciones ocurridas entre caras poligonales. t_{CI} y t_{CP} son el tiempo de intersecciones ocurridas entre caras internas y caras poligonales respectivamente.

En la tabla 2 se compara el número de colisiones y el tiempo, en μseg empleado en detectar entre las cajas internas frente al número de colisiones y el tiempo entre las caras poligonales de los poliedros, para los dos tamaños del espacio de experimentación considerados y para las 4 colisiones que se tienen en cuenta. Se ha considerado hasta el nivel 4 de los árboles en cuanto a su recorrido simultáneo.

Tabla 3. Tiempos (μseg) de Detección para la pareja *dodecaedro - octaedro*

Detector	<i>Dodecaedro - Octaedro</i>				<i>Cubo - Eneagonal</i>			
	25 %		75 %		25 %		75 %	
	Υ_i	Υ_{ni}	Υ_i	Υ_{ni}	Υ_i	Υ_{ni}	Υ_i	Υ_{ni}
PQP	56.75	29.21	48.03	78.46	48.70	5.31	29.56	19.55
RAPID	45.73	14.70	28.64	39.10	27.01	5.30	28.74	19.55
SOLID	40.74	26.52	40.81	70.31	51.02	22.81	50.81	99.17
I-COLLIDE	63.50	49.14	51.22	80.21	73.12	45.60	60.09	100.09
QUICKCD	27.83	10.69	29.05	44.84	23.83	12.03	37.67	43.44
PROPUESTO	8,86	22,77	5,93	61,50	5,61	2,39	4,80	8,48

Se han comparado los tiempos promedio de respuesta del detector de colisiones propuesto con los obtenidos por los detectores de dominio público, tanto para la detección de colisión (\mathcal{Y}_i) como para la detección de no-colisión (\mathcal{Y}_{ni}). Los resultados se muestran en la tabla 3.

8. Discusión

En las diferentes colisiones consideradas se observa (ver tabla 2) que el número de colisiones detectadas debido a las intersecciones caja interna - caja interna es superior al número de colisiones detectadas debido a la intersección entre las caras poligonales de los poliedros. Además, el tiempo empleado en detectar las colisiones con caja interna - caja interna es muy inferior al tiempo empleado con caras las poligonales de los poliedros.

La intersección caja interna - caja interna no solo permite detectar más colisiones que las intersecciones entre las caras poligonales de los poliedros sino que las detecta más rápidamente. Para la misma colisión, a medida que aumenta la probabilidad de colisión el número de intersecciones caja interna - caja interna aumenta.

Se señala que en la tabla 2 y en la colisión cubo - cubo no hay colisiones debido a la intersección entre las caras poligonales de los prismas. Ello es normal debido a que las primitivas son cajas rectangulares y el cubo es una caja rectangular. Es decir, el cubo está perfectamente descrito por cajas rectangulares y la colisión cubo - cubo es detectada mediante caja interna - caja interna.

Como se muestra en la tabla 3, el detector de colisiones propuesto mejora los tiempos de los detectores de domino público, excepto para la comparación de los \mathcal{Y}_{ni} con Docecaedro-Octaedro. Obsérvese que en todos los casos los tiempos aumentan con la probabilidad de colisión.

9. Conclusiones

Las intersecciones caja interna - caja interna tienen gran impacto en la detección de colisiones. Además cumplen con la filosofía de la utilización de los volúmenes limitantes consistente en detectar tempranamente las colisiones. Estas intersecciones son muy rápidas por lo que contribuyen a aumentar el desempeño del detector.

Los resultados de las intersecciones caja interna - caja interna mejoran cuando el entorno esté caracterizado por una congestión geométrica. Se señala la importancia de considerar volúmenes limitantes tanto externos como internos pues las pruebas de intersección entre volúmenes internos son pruebas de aceptación de la colisión y contribuyen a mejorar el desempeño de los detectores de colisión jerárquicos.

Los tiempos obtenidos por el detector de colisión basado en cajas externas e internas mejoran el desempeño de los detectores públicos considerados. Esto es debido al gran impacto que tienen las intersecciones caja interna -caja interna en

la detección temprana de colisiones. Las intersecciones entre las cajas internas aparece como un nuevo mecanismo para detectar colisiones, el cual no actúa en los detectores públicos considerados.

Referencias

1. Arcila O., Muñoz C. A. and Bañón J. M.: Un Algoritmo Eficiente para la Detección de Colisiones en Tiempo Real, In: Memorias de CLEI XXXIV. Montevideo, Uruguay (2002)
2. Arcila, O. and Bañón, J. M.: Un Nuevo Detector de Colisiones basado en Elipsoides. In: Memorias del CLEI XXXVII. Cali, Colombia (2005)
3. Arcila, O.: Nuevas Representaciones Dobles (Externas e Internas) en Detectores de Colisión Jerárquicos. Tesis Doctoral. Universidad del Valle, Cali, Colombia (2008)
4. Ericson Ch.: Real Time Collision Detection. Morgan Kaufmann Publishers, San Francisco (2005)
5. Faverjon, B.: Hierarchical Object Models for Efficient Anti-Collision Algorithms. Proc. IEEE Int. Conf. Robotics and Automation, pp. 333-340. Scottsdale, AZ (1989)
6. Gottschalk, S., Lin, M. C. and Manocha, D.: OBBTree: A Hierarchical Structure for Rapid Interference Detection. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 171-180. ACM, New York (1996)
7. Gupta, K. and del Pobil A. P.: Practical Motion Planning in Robotics. John Wiley & Sons, New York (1998)
8. Hubbard, P. M.: Approximating Polyhedra with Spheres for Time-Critical Collision Detection, ACM Trans. Graphics, vol. 15, no. 3, pp. 179-210. ACM, New York (1996)
9. Klosowski, J.T., Held M., Mitchell J., Sowizrai H. and Zikan K.: Efficient collision detection using bounding volume hierarchies of k-DOPs, IEEE Transaction on Visualization and Computer Graphics, Vol. 4 No. 1, pp. 21-36. Piscataway, NJ (1998)
10. Lin, M.C., Manosha D., Cohen, J. and Gottschalk S.: Collision detection: algorithms and applications In 2nd Workshop on Algorithmic Foundations of Robotics, Toulouse, France, (1996)
11. Moller, T.: A Fast Triangle-Triangle Intersection Test, Journal of Graphics Tools, 2(2), p 25-30. A. K. Peters, Ltd. Natick, MA (1997)
12. Pitt-Francis, J. and Featherstone R.: Automatic generation of sphere hierarchies from CAD data, Proceeding of the 1998 IEEE International Conference on Robotics and Automation, pp. 324-329. Leuven, Belgium (1998)
13. del Pobil, A. P., Martínez, B., Cervera E. and Pérez-Francisco, M. : Representations for Contact and Collision Detection with Complex Geometry, Workshop Practical Motion Planning in Robotics: current Approaches and Future Directions. IEEE Intl. Conference of Robotics and Automation. Minneapolis, MN (1996)
14. del Pobil, A. P., Pérez-Francisco, M. and Martínez, B.: A Practical Approach to Collision Detection between General Objects, Proc. IEEE Int. Conf. Robotics and Automation, pp. 779-784. Minneapolis, MN, (1996)
15. del Pobil, A.P. and Serna, M. A.: Spatial representation and motion planning (Lecture notes in Computer Science). Springer, New York, (1995)
16. del Pobil, A.P., Serna, M. A. and Llovet, J.: A New Representation for Collision Avoidance and Detection. In Robotics and Automation, IEEE International Conference, Nice, France (1992)
17. Van den Bergen, G.: Collision Detection in Interactive 3D Environments. Morgan Kaufmann Publishers, San Francisco (2003)