2010 International Forum on Information Technology and Applications

Efficient Collision Detection Using a Dual K-DOP-Sphere Bounding Volume Hierarchy

Fang Zhigang[1,2], Jiang Jianxun[1], Xu Jie[2]
[1] Department of Information Science& Electronic Engineering , Zhejiang University (ZJU), Hangzhou 310015, China
[2] Zhejiang University City College(ZUCC), Hangzhou 310015, China
jiangjianxun72@gmail.com, fangzg@zucc.edu.cn, xujie@zucc.edu.cn

ABSTRACT: **Collision detection is of paramount importance for many applications in computer graphics and visualization. In this research, we present an efficient algorithm for collision detection using a dual bounding hierarchy which consists of an discrete orientation polytopes (k-DOPs) tree enhanced with bounding sphere. The algorithm combines the compactness of the k-DOP and the efficient overlap test for spheres. The more efficient sphere test is applied firstly to eliminate distant objects. The remaining objects are in close proximity are tested using separation axis, where some separation axis are more effective and should be chosen first. We apply the efficient approach to the virtual acupuncture medical treatment systems, and the experimental results show that the new algorithm effectively reduces the query time and enhances the reality character with respect to the existing collection detection algorithms.**

KEYWORDS: collision detection; sphere; k-DOP; bounding volume hierarchy; Minkowski sum

I. INTRODUCTION

Virtual reality refers to the use of computer graphics to simulate physical worlds or to generate synthetic ones, where a user is to feel immersed in the environment to the extent that the user feels as if "objects" seen are really there. For example, "objects" should move according to force exerted by the user and they should not go through each other. In order to achieve this feeling of presence, the key is that of collision detection, which is the fundamental in many other application as well [1]。

Collision detection algorithms have been studied for many years. Among many different approaches, the bounding volume has proved to be the most successful in contemporary system. The computation time of this approach can be formulated as follows [2]:

$$T=Nv\times Cv+Np\times Cp+ Nu\times Cu \qquad (1)$$

where

T:the total cost function for collision detection,

Nv: the number of pairs of bounding volumes tested for overlap,

Cv: the cost of testing a pair of bounding volumes for overlap,

Np: the number of pairs of primitives tested for contact,

Cp: the cost of testing a pair of primitives for contact,

Nu: the number of nodes of the flying hierarchy that must be updated,

Cu: the cost of updating each node.

This formula clearly shows that the performance mainly depends on three factors: the tightness of the bounding volumes, the simplicity of the overlap test for a pair of bounding volumes, and the cost of updating the flying hierarchy as the flying object rotates. The first factor is related to Nv and Np, the second factor is related to Cv, whereas the third factor is related to Nv.

However, the all three factors often conflict with each other, and therefore there must be a compromise between them. The construction difficulty, simplicity of the overlap test for a pair of bounding volumes, tightness of the bounding volumes and cost of updating the flying hierarchy as the flying object rotates of several typical bounding box are compared in Table 1, where 1 means the best, 4 equals worst, and so forth.

Table 1 presents that spheres and axially aligned bounding boxes (AABBs) allow the simplest overlap tests. On the other hand, oriented bounding boxes (OBBs) and discrete orientation polytopes (k-DOPs) fit against the surface of the object it encloses more tightly. In this paper, we propose a dual scheme that combines the simplicity of spheres and the compactness of k-DOP to produce an efficient algorithm.

TABLE I. COMPARISON OF TYPICAL BOUNDING BOX（1 MEANS BEST）

Algorithms	Construction difficulty	Simplicity of the overlap test	Tightness	Cost of updating
AABBs	1	2	3	3
Spheres	2	1	4	1
OBBs	4	4	2	2
k-DOPs	3	3	1	4

James T. Klosowski [2] proposed Longest Side guideline on choice of axis: Choose the axis along which the k-DOP, bounding volume, is longest. In this research, we propose a new principle choice of axis after a preceding sphere test. We point out the geometric intuition behind the selection of effective separation axes. Besides, we present a theoretical analysis of the principle.

The remainder of the paper is organized as follows: Prior and related work is reviewed in Section 2. Section 3 provides an introduction to our collection algorithm. Implementation details and experimental results are reported in Section 4.

978-0-7695-4115-0/10 $26.00 © 2010 IEEE
DOI 10.1109/IFITA.2010.323

The conclusion, Section 5, includes a discussion of extensions and future work.

II. RELATED WORK

In this section, we will briefly review related work on collision detection. Due to its widespread importance, there has been an abundance of work on the problem of collision detection. Many of the approaches use hierarchies of bounding volumes or spatial decompositions to solve the problem. The idea behind these approaches is to approximate the objects (with bounding volumes) or to decompose the space they occupy (using decompositions), to reduce the number of pairs of objects or primitives that need to be checked for contact [2].

Octrees [3, 4], k-d trees [5], BSP-trees [6], brep-indices [7, 8], tetrahedral meshes [5], and (regular) grids [5,9] are all examples of spatial decomposition techniques. By dividing the space occupied by the objects, one needs to check for contact between only those pairs of objects (or parts of objects) that are in the same or nearby cells of the decomposition. Using such decompositions in a hierarchical manner (as in octrees, BSP-trees, etc.) can further speed up the collision detection process.

Hierarchies of bounding volumes have been proved to be the most successful in contemporary system [10]. In building hierarchies on objects, one can obtain increasingly more accurate approximations of the objects, until the exact geometry of the object is reached. The choice of bounding volume has often been to use spheres [12,13] or axis-aligned bounding boxes (AABBs) [5,11], due to the simplicity in checking two such volumes for overlap (intersection). In addition, it is simple to transform these volumes as an object rotates and translates.

Another bounding volume that has become popular recently is the oriented bounding box, which surrounds an object with a bounding box (hexahedron with rectangular facets) whose orientation is arbitrary with respect to the coordinate axes. This volume has the advantage that it can, in general, yield a better (tighter) outer approximation of an object, as its orientation can be chosen in order to make the volume as small as possible. In 1981, Ballard [14] created a two-dimensional hierarchical structure, known as a “strip tree,” for approximating curves, based on oriented bounding boxes in the plane. Barequet et al. [15] have recently generalized this work to three dimensions (resulting in a hierarchy of k-DOPs known as a “BOXTREE”), for applications of oriented bounding boxes for fast ray tracing and collision detection.

One leading system publicly available for performing collision detection among arbitrary polygonal models is the “RAPID” system, which is also based on a hierarchy of oriented bounding boxes, called “k-DOP Trees”, implemented by Gottschalk, Lin, and Manocha [16]. The efficiency of this method is due in part to an algorithm for determining whether two oriented bounding boxes overlap. This algorithm is based on examining projections along a small set of “separating axes” and is claimed to be an order of magnitude faster than previous algorithms. (We note that Greene [17] previously published a similar algorithm; however, we are not aware of any empirical comparisons between the two algorithms.)

Other approaches to collision detection have included using four-dimensional geometry [18,19] and space-time bounds [12] to bound the positions of objects within the near future. By using a fourth dimension to represent the simulation time, contacts can be pin-pointed exactly; however, these methods are restrictive in that they require the motion to be pre-specified as a closed-form function of time. Hubbard’s space-time bounds [12] do not have such a requirement; by assuming a bound on the acceleration of objects, he is able to avoid missing collisions between fast-moving objects.

The most basic type of collision detection algorithm deals with rigid bodies in static poses. However, many recent studies have looked at more complicated problems, including detecting collisions between deformable models rather than rigid models [20,21,22], collision detection in the continuous time domain rather than static pose [23,24] and collision detection algorithms which can run on graphics hardware [21,25]. Even though these complicated problems deal with more general cases, the discrete collision detection between rigid bodies is still quite important because the algorithms for the basic type problem are much more efficient than the algorithms for complicated problems. Moreover, the basic type algorithm serves as the basis for the development of efficient algorithms for general problems. For example, the continuous collision detection algorithms of Redon et al. [23,24] use the OBB bounding volume hierarchy [16]. Performance improvement in the static case can also lead to similar improvement in the continuous case. In particular, the simple sphere pretest of our approach has great potential in the continuous collision detection environments based on the OBB trees. Recently, Choi Y-K [25] work on the continuous collision detection for moving ellipsoids, an efficient algorithm for testing the separation of two static ellipsoids plays an important role in the design of the overall algorithm. In this paper, we focus on the development of an efficient algorithm for the static case.

III. COLLECTION DETECTION ALGORITHM

We will now present our efficient collision detection algorithm using dual k-DOP-sphere bounding volume hierarchy. Firstly, we describe the construction of a dual k-DOP-sphere tree and show how the problem can be reduced to a situation of close proximity using a sphere test. Then we will go on to describe the choice of effective separating axes.

A. Dual k-DOP bounding volume tree

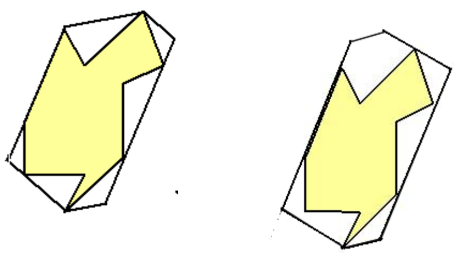
The basic structure of the dual k-DOP tree is the same as the k-DOP tree proposed by James T. Klosowski et al [2]. For every node of a k-DOP tree, the dual k-DOP tree also contains a sphere which bounds the elements of the polygonal mesh that are at that node. There are two ways commonly used for the construction of a bounding sphere. The first method is to construct the smallest enclosing sphere. The second is to fix the center of the bounding sphere at the center of the k-DOP. In the latter case the centers of the two bounding spheres under a separation test can be reused for the subsequent k-DOP separation test, so the second method provides a simpler overlap test. However, the first method naturally guarantees a tighter bounding sphere. We choose the first method in our collision detection.

The construction of k-DOP in our method is a little different from the k-DOP proposed by James T. Klosowski et al [2]. Our K-DOPs is revised, it must satisfy the centrally-symmetric condition (see Remark in Section 3.2), as shown in Fig.1. The left image of Fig.1 is a non-centrally-symmetric 6-DOP. The right image of Fig.1 is a centrally-symmetric 6-DOP. By following simple steps, we can construct our K-DOPs:

1) Choose any two in all k/2 axes, and calculate the maximum and minimum in both directions. According to the above results, we construct a 4-DOP, as shown in Fig.2 (b).

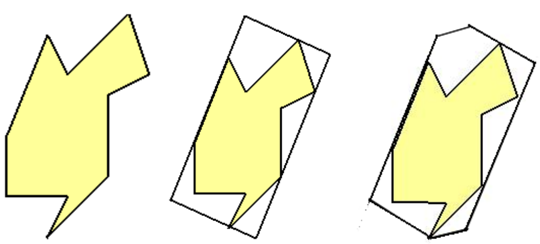
2) Calculate the symmetry of 4-DOP, set it as the center of projection, and then choose any one direction from the other directions, which we did not use in the 4-DOP, as the projection direction. Finally, calculates the maximum orthographic projection of object being detected in the direction. Using the maximum value as the distance of the symmetry to the fifth and sixth edges, we establish a 6-DOP bounding box, shown in Fig.2 (c).

3) To construct k-DOPs with a larger k, such as 8-DOPs, repeat the second step.



a b

Fig.1 Comparison of two different 6-DOPs: non-centrally-symmetric and centrally-symmetric

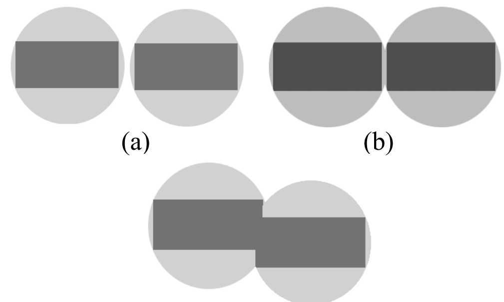


a b c

Fig.2 Construction of a 6-DOP

B. Choice of separating axes

The k/2 potential separating axes for all the k-DOPs are a1, a2,… ak/2. Based on the geometric intuition, we choose the effective separating axes. As shown in Fig.3(a), the two spheres separate, so we need not process a further collision detection between two k-DOPs. When the two spheres intersect, there are two cases. Fig.3 (b) shows two spheres intersect, but the two k-DOP separate in a separating axis direction. Once we confirm that the two k-DOPs separate in the above separating axis direction, the further collision detection is no longer needed. Fig.3 (c) shows that the two spheres intersect and the two k-DOPs also intersect. In this case, we need a further collection detection using another separating axis. The key to this question becomes how quickly distinguish the case Fig.3 (b) and the case Fig.3(c).



(c)

Fig.3 Spheres separate and comparison of two different cases of spheres intersect

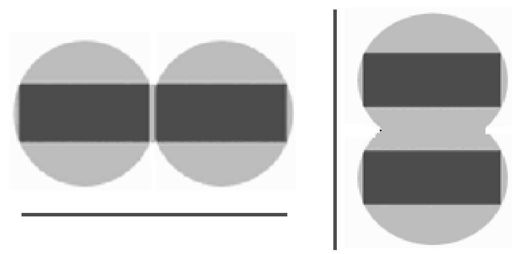


Fig.4 Select an effective separating axes

In order to analyze the principle how to select the effective separating axes, we propose a concept called invalid volume. Fig.4 shows two spheres intersect and the two k-DOPs separate. The invalid volume in Fig.4 (a) is smaller than Fig.4 (b). After the sphere intersection test, we choose such a separating axis, along which the invalid volume is small, to make a collision detection query. It is inefficient to do that query, because we can almost exclude that the two k-DOPs separate in sphere intersection test.

Our selection of the separating axes is based on the geometric intuition that we explain using examples of Figs. 3-4. For a more rigorous justification of the selection of axes, we will also consider the Minkowski sum of bounding volumes. Given two k-DOPs A and B with centers located at the origin, the Minkowski sum $A \oplus B = \{a + b \mid a \in A, b \in B\}$, is a centrally-symmetric convex polyhedron. The interference between two translated copies A+ p and B+ q is equivalent to a set containment problem: $(A + p) \cap (B+ q) \neq \Phi$ if and only if $p \in (A \oplus B)$. (see theorem 1). We also call p, the center of $A \oplus p$, or the reference point of $A \oplus p$.

Theorem 1: For two centrally-symmetric objects A and B with centers at the origin, we have $A = -A = \{-a \mid a \in A\}$ and $B=-B$. Now we can show the equivalence:

$(A+p) \cap (B+q) \neq \varnothing$
$\Leftrightarrow a+p=b+q$, $a \in A, b \in B$
$\Leftrightarrow p=b-a+q$
$\Leftrightarrow p \in A \oplus B+q$

Since it is rather difficult to visualize the Minkowski sums in 3D, we illustrate the main idea using a 2D example, where the Minkowski sum of two centrally-symmetric convex hexagons is also a centrally-symmetric convex hexagons. The separation test is then reduced to the containment test for a reference point with respect to 3 slabs, each determined by two parallel edges of the Minkowski sum. This relation is shown in Fig. 5. The left images of Fig.5 denote the objects themselves and the right images denote their configuration spaces defined by the Minkowski sum of two hexagons. (Note that the upper hexagon is reduced to its center and the lower hexagon is simultaneously grown to the Minkowski sum of two hexagons.) In Fig.5, two hexagons are separated with respect to the y axis, and the reference point is placed outside the slab, which is orthogonal to the y axis.

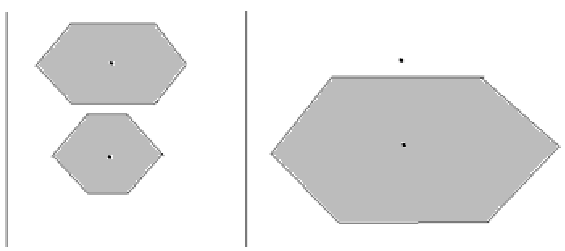


Fig.5 The relation between the overlap test for an axis and the containment test for a slab

Fig.6 shows the relationship between the extent corresponding to an axis and the compactness achieved by that axis when the objects are in close proximity. In the case of 2D hexagon, the potential separating axes are defined by the orientation of each hexagon. The 3 potential separating axes are a1, a2, a3. We assume that the corresponding extent of a1 in Minkowski sum of two centrally-symmetric convex hexagons is smaller than a2, and a3 correspond to the largest extents. Figs.6 shows the importance of the extent corresponding to an axis under the condition of that the objects are in close proximity. If the reference point of the reduced object is contained in the Minkowski sum of two bounding circles, which is also a circle, the two bounding hexagons must be tested. Regarding the selection of separating axes, the Minkowski sum of two bounding circles can be subdivided to 2 regions---the white region and shaded region. Different separating axis responds to different shaded region. Here a1 responds to light gray region, a2 responds to dark gray region and a3 responds to black region. If we choose any of them as the separating axis, the separation test is then reduced to the containment test for the reference point with respect to the white region. The white region shows that the two rectangles overlap for the separating axis. In the light gray region the two rectangles are separated with respect to a1 axes, in the dark gray region the two rectangles are separated with respect to a2 axes and in the black region the two rectangles are separated with respect to a2 axes. Since a1≤a2≤a3, the area of light gray region is greater than the area of light dark gray region, and the area of black region is the smallest. We can exclude the largest invalid area if we choose a1 as the separating axis. That is to say, a1 is the best separating axis.

The foregoing discussion can be summarized as follows:

- In all k/2 potential separating axes for k-DOP, the axis with the smallest extent in Minkowski sum of two k-DOPs is preferred and should be test as a priority.

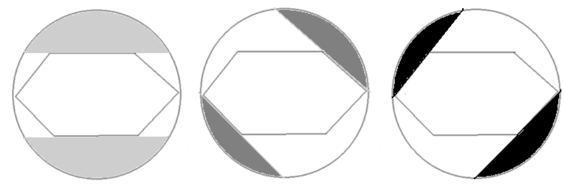


Fig.6 The invalid area be excluded using different separating axes

IV. RESULTS

We design and implement a virtual acupuncture training system based on OpenGL and VC++, the system use spatial sensors and data gloves as input device. Through motion tracking, we establish the link between hands and the virtual hands.

In the virtual acupuncture training system, virtual hand can grab the silver needle. However, the silver needle is small and it is difficult to grab the silver needle. In fact, we achieve the silver needle grabbing through collision detection, so the collision detection requires higher accuracy. In addition, the human body model is complex. In order to ensure collision detection real-time, the efficiency of algorithm must be high enough. If you use regular OBB collision detection algorithms, such as V-Collide collision detection which is an open-source library, it is difficult to meet the real-time requirements and the accuracy requirements. We apply our efficient algorithm to virtual acupuncture training system and the results show that our algorithms work well.

We have implemented our collision detection algorithm in C++ on an Intel Core（TM）2 CPU with a 4.0 GB main memory and NVIDIA GeForce 9800GT graphics card with a 1.0 GB memory. The experiment concerned about the refresh rate. We apply the collision detection algorithm to virtual acupuncture training system. In the Scenario, there are a human body model with 53,212 triangles, a virtual hand with 12884 triangles and a needle with 132 triangles. We apply the improved collision detection algorithm to the virtual acupuncture training system, it fully met the real-time requirements (refresh rate reach 40 images per second) and the accuracy requirements can also be well met. As shown in Figure 7, the virtual hand color turns red when the needles been inserted into the body.

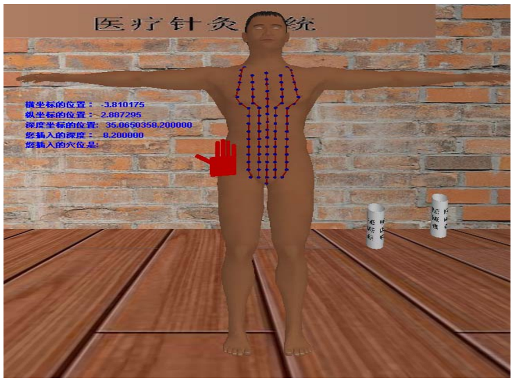


Fig.7 Virtual hand grab needles into the body

V. Conclusion

We have presented a fast k-DOP-based collision detection algorithm that uses both k-DOPs and spherical bounding volumes. The collision detection algorithm combines the compactness of k-DOP and the efficient overlap test for spheres. When it comes to the choice of axis, we propose a new algorithm. Experimental results show that our scheme makes favorable speed up with respect to existing algorithms based on k-DOP.

In future work, we plan to introduce a general selection scheme for separating axes so that the axes can dynamically be selected depending on the relative shapes of two k-DOPs under consideration. Furthermore, we would like to extend our algorithm to the general case of continuous collision detection and to an algorithm for deformable objects.

Acknowledgment

The authors would like to thank anonymous reviewers for their invaluable comments. The work of XU Jie was partially supported by the Provincial Natural Science Foundation of Zhejiang under Grant No. Y1080338. This research was also supported in part by Education of Zhejiang Province under Grant No.Y200804071.

References

[1] Sigal Ar, Bernard Chazelle, "Self-customized BSP trees for collision detection. "Computional Geometry.91-102 (15)2000

[2] James T. Klosowski, Martin Held, Joseph S.B. Mitchell, Henry Sowizral, Karel Zikan, "Efficient collision detection using bounding volume hierarchies of k-DOP." IEEE Transactions on Visualization and Computer Graphics. 4(1)1998

[3] M. Moore, J. Wilhelms, "Collision detection and response for computer animation." In Comput. Graphics(SIGGRAPH '88 Proc.), volume 22, pages 289–298, Aug 1988.

[4] H. Noborio, S. Fukuda, S. Arimoto, "Fast interference check method using octree representation." Advanced robotics, 3(3):193–212, 1989.

[5] M. Held, J.T. Klosowski, and J.S.B. Mitchell, "Evaluation of collision detection methods for virtual reality flythroughs." In Proc. 7th Canad. Conf. Comput. Geom., pages 205–210, 1995.

[6] B. Naylor, J.A. Amatodes, and W. Thibault, "Merging BSP trees yields polyhedral set operations." In Comput. Graphics (SIGGRAPH '90 Proc.), volume 24, pages 115–124, Dallas, TX, USA, Aug 1990.

[7] W. Bouma and G. Vanˇeˇcek, Jr, "Collision detection and analysis in a physical based simulation." In Eurographics Workshop on Animation and Simulation, pages 191–203, Vienna, Austria, Sep 19.

[8] G. Vanˇeˇcek, Jr, "Brep-index: A multidimensional space partitioning Tree." Internat. J. Comput. Geom. Appl., 1(3):243–261, 1991.

[9] A. Garcia-Alonso, N. Serrano, and J. Flaquer, "Solving the collision detection problem." IEEE Comput. Graph. Appl., 14:36–43, May 1994.

[10] Akenine-Moller T, Hains E. Real-time rendering. A.K. Peters; 2002.

[11] D. Baraff, "Fast contact force computation for nonpenetrating rigid bodies." In Comput. Graphics (SIGGRAPH '94Proc.), volume 28, pages 23–34, Orlando, FL, USA, Jul 1994.

[12] P.M. Hubbard, "Collision detection for interactive graphics applications." IEEE Trans. Visual. Comput. Graph.,1(3):218–230, Sep 1995.

[13] P.M. Hubbard, "Approximating polyhedra with spheres for time-critical collision detection." ACM Trans. Graph., 15(3):179–210, July 1996.

[14] D.H. Ballard, "Strip trees: A hierarchical representation for curves. Comm." ACM, 24(5):310-321, May 1981.

[15] G. Barequet, B. Chazelle, L.J. Guibas, J.S.B. Mitchell, A. Tal, "BOXTREE: A Hierarchical Representation for Surfaces in 3D." EuroGraphics'96, J. Rossignac and F. Sillion, eds., Blackwell Publishers, Eurographics Association, Volume 15, (1996), Number 3, pages C-387–C-484.

[16] S. Gottschalk, M.C. Lin, and D. Manocha. "OBBTree: A hierarchical structure for rapid interference detection." In Comput. Graphics (SIGGRAPH '96 Proc.), volume 30, pages 171–180, New Orleans, LA, USA, Aug 1996.

[17] N. Greene. "Detecting intersection of a rectangular solid and a convex polyhedron." In P.S. Heckbert, editor, Graphics Gems IV, pages 74–82. Academic Press, 1994. ISBN 0-12-336155-9.

[18] S. Cameron. "Collision detection by four-dimensional intersection testing." IEEE Trans. Robot. Autom., 6(3):291–302, 1990.

[19] J. Canny, "Collision detection for moving polyhedra." IEEE Trans. Pattern Anal. Mach. Intell., PAMI-8(2):200-209, March 1986.

[20] James D, Pai D, "BD-tree: Output-sensitive collision detection for reduced deformable models." ACM Transactions on Graphics 2004;23(3):393-8.

[21] Zhang X, Kim Y, "Interactive collision detection for deformable models using streaming AABBs." IEEE Transactions on Visualization and Computer Graphics 2007;13(2):318-29.

[22] Larsson T, Akenine-Moller T, "Efficient collision detection for models deformed by morphing." The Visual Computer 2003; 19(3): 164 74.LINMC, GOTTSCHALKS. Collision detection between geometric models: a survey.Proc of IMA Conference on Mathematics of Surfaces. 1998

[23] Redon S, Kheddar A, Coquillart S, "Fast continuous collision detection between rigid bodies." Computer Graphics Forum 2002;21(3):279-87.

[24] Redon S, Kim Y, Lin M, Manocha D, "Fast continuous collision detection for articulated models." In: Proc. ACM symp. on solid modeling and applications.2004. p. 145-56.

[25] Choi Y-K, Chang J-W, Wang W, Kim M-S, Elber G, "Continuous collision detection for ellipsoids." IEEE Transactions on visualization and Computer Graphics 2009;15(2):311-24.