Available online at www.sciencedirect.com

SciVerse ScienceDirect

Physics Procedia 24 (2012) 247 – 252

Physics Procedia

2012 International Conference on Applied Physics and Industrial Engineering

Collision Detection Based on Bounding Box for NC Machining Simulation

Yao Wang [1,2], Yan-juan Hu [1,2], Jiu-chen Fan[1],Yu-feng Zhang[1], Qi-jiu Zhang[1]

1. College of Mechanical Engineering Beihua University
Jilin, China
2. College of Mechanical Science and Engineering, Jilin University
Changchun, China

Abstract

Aiming at the characteristics of NC machining simulation, the bounding box algorithm and detecting step by step are adopted. First, crudity detection is implemented, and many objects which do not obviously intersect are fleetly excluded; then, the further detection is carried out, this step further detects the objects which may be intersected, and check up the latent interference section from the intersecting objects; last, subtlety detection is performed, this step will judge if there is collision from detecting the state of basic elements (triangle element) of objects. The detection examples prove that the developed algorithm in this paper can guarantee the detection precise, improve detection efficiency, and satisfy the need of real-time of NC machining simulation.

*Keywords:*collision detection; bounding box; NC machining; simulation

1. Introduction

Collision detection [1-3] is widely applied in computer graphics, computer cartoon, CAD/CAM, all kinds of virtual realism system of human-computer interaction, etc.. Collision detection algorithms mainly include: collision detection algorithm based on bounding box, collision detection algorithm based on distance calculation, collision detection algorithm based on Voronoi diagram and so on. In those algorithms, collision detection algorithm based on bounding box [4, 5] is the most widely application method. Although bounding box method has defect of imprecise, comparing with the other two methods, it has the advantage of abroad application range and strong adaptability, and it can apply in collision detection of various polyhedrons. The other two algorithms can only detect the collision between convex polyhedrons, although they can use some special methods to realize the collision detection of concave polyhedrons, their computation time is very long, which conflicts with the real-time of simulation system.

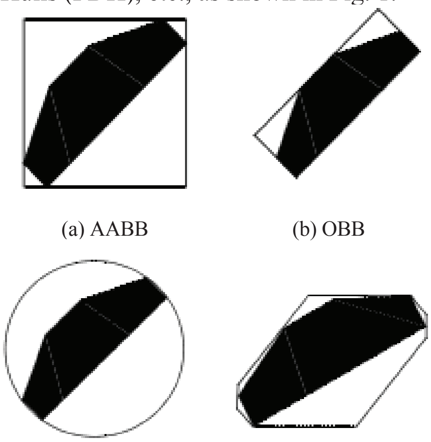
doi:10.1016/j.phpro.2012.02.037

Collision detection is an important content of NC machining simulation [6]. Once the collision happening, the slight consequence is that the tool and workpiece will be damaged, and the serious consequence is that the worktable surface, main shaft and machine tool will be damaged. During the process of NC machining simulation, if the object collision detecting is moving constantly (for example tool), the machining process show and collision detection should be implemented simultaneously. So, collision detection in NC machining simulation need very high detecting speed and strong real-time. In terms of the characteristics of NC machining simulation, the algorithm of collision detection is developed in this paper, and collision detection based on bounding box is realized.

2. Bounding Box Detection Method

The basic idea of bounding box algorithm is making bounding boxes for the objects will be detected respectively, then intersection operation for bounding boxes is implemented. When bounding boxes intersect with each other, the objects enveloped by bounding boxes may intersect; when bounding boxes do not intersect with each other, the objects enveloped by bounding boxes do not intersect. Thus many geometry objects and parts which do not intersect can be excluded, and the intersecting parts can be found fleetly.

Bounding box algorithm include: Axis-Aligned Bounding Boxes (AABB), Oriented Bounding Box (OBB), Sphere, Fixed Directions Hulls (FDH), etc., as shown in Fig. 1.



(a) AABB (b) OBB

(c) Sphere (d) FDH

Figure 1. Bounding box kind

Simplicity and tightness of bounding box directly affect the efficiency of collision detection. Simplicity is that the bounding box should be simple geometrical body, at least simpler than the object enveloped by bounding box, and can realize the calculation of intersecting detection rapidly.

Tightness is that bounding box closes to the object which is enveloped as much as possible. In bounding box algorithms, the simplicity of AABB is good, but its tightness is poor, after the object rotating, AABB should be do the same rotation and update. Either geometrical body or intersecting detection of Sphere is very simple, but its tightness is very poor. The tightness of OBB is best, and it can multiply shorten the number of bounding boxes involved in intersecting detection and basic geometrical elements, after geometrical object rotating, it only need to do the same rotation for the substrate of OBB. The total performance of OBB is better than AABB and Sphere. For the collision detection of rigid bodies, OBB is the better choice, but, after the rigid body deforming, the update of OBB tree is a difficult problem which is not solved so far. Both simplicity and tightness of FDH are very good, because the normal vectors of all surfaces of bounding box come from the fixed direction vector set. After object rotating, calculation of

FDH is not complex, and also, after object deforming, it can regenerate bounding boxes for the deformed object rapidly.

3. Fixed Directions Hulls (FDH)

According to above mentioned, OBB and FDH are good, although, both OBB and FDH can apply to the collision detection of rigid body, FDH is better than OBB. In FDH, the normal vectors of all surfaces of bounding box come from the fixed direction vector set D, and the direction vectors in fixed direction set D are collinear and vector pairs of opposite direction. There are k vectors in fixed direction set D, so there are k/2 vector pairs which are collinear and opposite direction. FDH not only has good tightness, but also has the advantage that bounding box is simple, hence, this paper adopts FDH to realize collision detection.

The k value in fixed direction D is larger, the tightness of bounding box is better, and the needed number of bounding boxes when detecting is smaller, but, with the growing of k value, the needed time of constructing bounding box is adding. In view of tightness and complexity of constructing bounding box, this paper chooses FDH14 (k=14) to ensure the time needed by constructing bounding box and detecting is shortest.

To judge if the objects A and B intersect in space, the core of collision detection algorithm is traversing their respective FDH tree to identify if some section of A collides with some section of B in current position. The basic idea of FDH is that the bounding box of simple geometry characteristic is used to replace the complex geometry object to implement detection, if the bounding boxes at two nodes do not intersect, the subsets of basic geometrical elements of the objects enveloped by them will not intersect, so that the further detection of subset elements is not needed.

Simulating collision detection of bounding boxes is easy: as long as a direction is found, projection intervals of two FDH in this direction do not overlap, they can be judged they do not intersect; if their projection intervals in all direction pairs in D overlap, they can be judged they intersect, namely there collision between them. Fixed direction vector set D is composed of seven vector pairs. When implementing intervals overlapping detection, if detection result is overlapped in a direction, in order to find a not overlapped interval as soon as possible, another direction which has great difference with this direction is chosen to accomplish detection, if the needed direction is found, the querying will be over.

4. Collision Detection Steps

In terms of the workpiece, tool and so on in NC machining simulation system is represented by tri patch model, the collision detection is really judging tri patch of one object if intersect with the tri patch of another object. Virtual machining environment includes: machine tool guideway, headstock, end bracket, chuck and tool holder and so on many objects, directly implementing intersection operation for the tri patches of workpiece and tool, headstock and tool needs many times, which is hard to satisfy the real-time of virtual machining.

When processing the complex scene including many objects, this paper adopts detecting step by step to realize collision detection. First step is crudity detection, in this step, many objects which do not obviously intersect are rapidly excluded; the second step is further detection, this step detect the latent intersection sections by further detecting the crossed object pairs; last is subtlety detection, this step can precisely detect if have collision between basic volumes or polygon patches. As following:

4.1 Crudity detection

If virtual environment includes more objects, in order to improve detection efficiency, some optimization methods should be adopted to exclude the objects which do not collide with each other, and find the crossed object pairs or latent intersection intervals. This paper uses AABB method to realize crudity detection, this method respectively projects AABB of all objects at x, y, z the three coordinate axes, and arranges sequence for boundary value of projection interval of each object at each coordinate axis. If the projection intervals of bounding boxes of two objects at all coordinate axes are overlapped, the bounding boxes of the two objects will be crossed. As shown in Fig. 2.

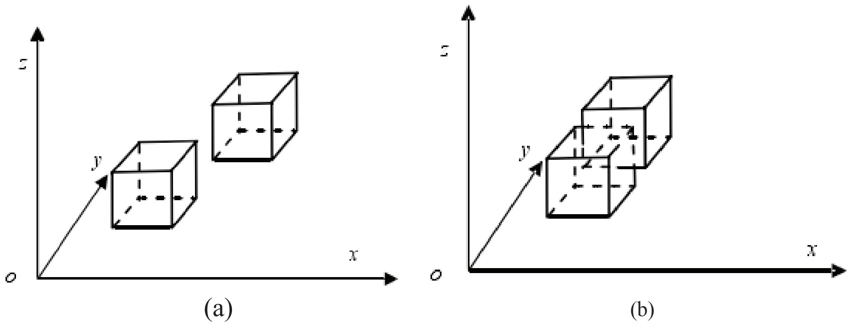


Figure 2. Using AABB method to implement crudity detection

In this algorithm, the projections of AABB of all objects at three coordinate axes form linked list, and the linked list is updated in real time, at last, the object pair which overlap at three axes can be gotten by summarizing, and the object pair can be input object for next subtlety detection. By projecting AABB at one coordinate axis (for example x axis), a set of intervals can be gotten, after arranging intervals sequence, the object which does not intersect with any other objects can be excluded. As shown the object C in Fig. 3. If the overlapped object pairs are only a few, the detection can be implemented in sequence to detect those objects if have collision at other coordinate axes (y axis, z axis). This method not only can save time, but also can avoid unnecessary sequencing and finding. The two-dimensional projection method can be also adopted to realize crudity detection, and the object bounding boxes are projected on coordinate plane (for example x-y plane) using this method, then rectangle sequencing algorithm is used to select, at last, the projections of the overlapped rectangle pair at another coordinate axis (z axis) if intersect can be validated. The two-dimensional method is very effective, and its computation amount is small, for the scene including fixed objects, it can rapidly accomplish preliminary collision detection.

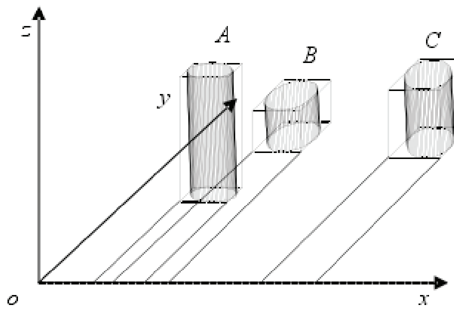


Figure 3. Projection of AABB bounding box of object at x axis

4.2 Further detection

Further detection does the further intersection detection according to the latent crossed object pairs which have been assured, and assure the intersection area (tri patch set) of leaf nodal bounding box including object pair set. This paper uses FDH algorithm to realize subtlety detection.

4.3 Subtlety detection

Subtlety detection stage lastly judges if has collision between objects by implementing intersection detection of basic geometrical elements in bounding box. The geometrical model is composed of tri patches in this paper, so the collision detection is realized by implementing intersection detection of tri patches.

5. Collision detection examples

This paper combines the case of NC vertical milling simulation to demonstrate the effective of this algorithm.

5.1 Crudity detection

Simulation system respectively constructs AABB of tool (including cutter spindle and milling cutter), fixture, block, workpiece and worktable, when moving parts of virtual machine tool moving according to NC code, the system detects if the AABB of tool collide with the AABB of fixture or with others in real time, and the detection result shows the AABB of tool collide with the AABB of right fixture, then system will begin to do the further detection.

5.2 Further detection

Simulation system respectively constructs FDH14 bounding boxes and hierarchical binary trees of tool and fixture, then, the system traverses the two binary trees. The process of traversing as following:

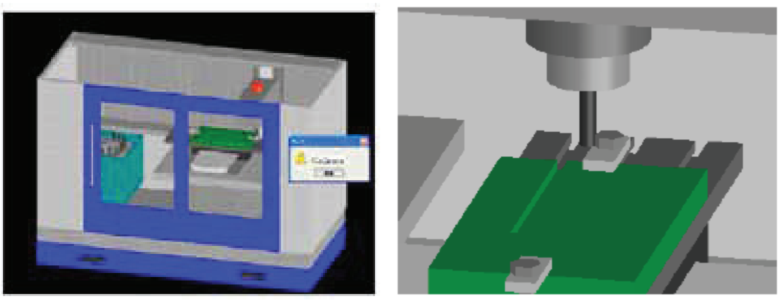
1) The system uses the root nodal bounding box a_1 of fixture bounding box hierarchical tree to do intersection detection with hierarchical tree of tool bounding box in layers, and the detection result shows a_1 intersects with root nodal bounding box b_1, child nodal bounding boxes b_2, b_3, and other lower child nodal bounding boxes, part leaf nodal bounding box and so on of tool bounding boxes.

2) According to above detection, there are five pairs bounding boxes of binary tree leaf nodal intersect.

5.3 Subtlety detection

Based on above detection result, the simulation system respectively detects the tri patches of five pairs crossed bounding boxes of leaf nodal, and detection shows that there are crossed tri patches in the crossed bounding boxes of leaf nodal, thus the simulation system can judge that tool collides with fixture at the crossed tri patches.

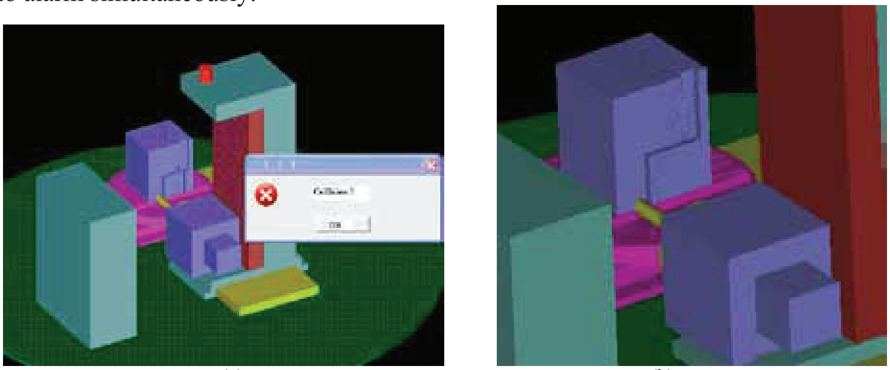
Fig. 4 shows the collision detection result of NC vertical milling, when simulating milling, tool collides with fixture, so that simulation system stops simulation and pops up dialog box to alarm which reminds user to change NC code.



(a) (b)

Figure 4. Collision detection result of NC vertical milling

Fig. 5 shows the collision detection result of NC horizontal milling using same algorithm, when simulating horizontal milling, tool collides with worktable, and also the system stops simulation, pops up dialog box to alarm simultaneously.



(a) (b)

Figure 5. Collision detection result of NC horizontal milling

Conclusion

This paper introduces condition and kind of collision generating, and also analyzes detection algorithms, in which FDH algorithm is studied mainly. Detection includes three steps: crudity detection, further detection and subtlety detection. In crudity detection stage, AABB algorithm and two-dimensional projection are adopted to find latent crossed object pair, which improves detection efficiency. In further detection stage, FDH algorithm is used to judge if collision happens by assuring and traversing hierarchical tree of bounding box, doing intersection detection of bounding box. In subtlety detection stage, the intersection detection of basic volumes is implemented, after space volumes are judged, the volumes are projected on two-dimensional plane to be detected, which can simplify computation and improve detection efficiency. The application examples prove that the presented algorithm is effective and efficiency.

References

[1] S. Ding, M. A. Mannan and A. N. Poo, "Oriented bounding box and octree based global interference detection in 5-axis machining of free-form surfaces ," Computer-Aided Design, vol. 36, Nov. 2004, pp. 1281-1294, doi:10.1016/S0010-4485(03)00109-X.

[2] Kai Xie, Jie Yang and Y.M. Zhu, "Fast collision detection based on no se augmentation virtual surgery ," Computer Methods and Programs in Biomedicine, vol. 88, Oct. 2007, pp. 1-7, doi:10.1016/j.cmpb.2007.06.004.

[3] Juan J. Jiménez and Rafael J. Segura, "Collision detection between complex polyhedra, " Computers & Graphics, vol. 32, Aug. 2008, pp. 402-411, doi:10.1016/j.cag.2008.05.002 .

[4] Gill Barequet and Sariel Har-Peled, "Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions ," Journal of Algorithms, vol. 38, Jan. 2001, pp. 91-109, doi:10.1006/jagm.2000.1127.

[5] Jung-Woo Chang, Wenping Wang and Myung-Soo Kim, "Efficient collision detection using a dual OBB-sphere bounding volume hierarchy," mputer-Aided Design, vol. 42, Jan. 2010, pp. 50-57, doi:10.1016/j.cad.2009.04.010.

[6] Kuan-Chen Lai and Shih-Chung Kang, "Collision detection strategies for virtual construction simulation," Automation in Construction, vol. 18, Oct. 2009, pp. 724-736, doi:10.1016/j.autcon.2009.02.006.