Proceedings of the 6th World Congress on Intelligent Control
and Automation, June 21 - 23, 2006, Dalian, China

An Algorithm on Collision Detection by Computing the Minimum Distance between Two Convex Polyhedra*

Hanjun Jin
Department of Computer Science
Central China Normal University
Wuhan, Hubei Province, China
jinhanjun@163.com

Yanlin Wang, Xiaorong Wang and Jia Fu
Department of Computer Science
Central China Normal University
Wuhan, Hubei Province, China
yanlin_wang@mails.ccnu.edu.cn

***Abstract* – The problem of tracking the distance between two convex polyhedra is finding applications in many areas of robotics, including intersection detection, collision detection, and path planning. An algorithm for computing the minimum distance between two convex polyhedra is presented. The algorithm applies to polyhedral objects that can be represented as convex hulls of its vertex value in three-dimensional space. Nonlinear programming techniques are then employed to compute the minimum distance between two convex polyhedra, and according to the minimum distance, it can conclude whether or not the objects has collided at some time. The simulation results demonstrate the performance of this new method.**

Index Terms - Convex polyhedron, Convex hull, Minimum distance, Nonlinear programming, Optimization.

I. INTRODUCTION

Collision often happens between objects because of users' interaction or the movement of objects in virtual environment (VE). To keep the reality of environment, system should detect collision, re-draw picture or compute the according collision response. Otherwise, penetration may happen, which will destroy the reality sense of VE and users' immersion [1]. Generally speaking, there are three study aims of collision detection: first, detecting collision; second, reporting the happening or happened parts of collision; third, dynamically inquiring the distance between objects.

Tracking distance algorithm is a significant one in detection collision, especially, Lin-Canny (LC) [2] algorithm and Gilbert-Johnson-Keerthi (GJK) algorithm [3]. LC method primarily utilizes the instantaneous correlativity principle [4], specifically; the closest features (vertex, edge or face) change infrequently as the objects move along finely discretized paths. A pair of closest features, which come from two convex polyhedra respectively are determined by the Voronoi region. Then LC algorithm detects whether or not the closest point is in the two features. If it is not, then the near features are detected instead until the closest points are found. GJK algorithm searches and tracks the translational configuration space (TCSO) instead of the polyhedra themselves. In order to realize the complex virtual spectacle, almost all the intersection algorithm use the bounding box [9], that is, some bounding boxes with simple shape are applied to detection algorithms. Nonexistence of inference between bounding boxes makes sure that there is no collision between objects. Obviously, it can decrease the complexity of algorithm greatly, especially in the case that the distance between two objects is not so near. Many bounding box can be chosen, such as axis-aligned bounding box (AABB), oriented bounding box (OBB), sphere, cylinder, k-discrete orientation polytopes(k-dop) [4][5][6]. Though the algorithm with bounding box can greatly decrease the execution time of far distant objects, LC algorithm and GJK algorithm are still necessary when the distance is near.

From optimization angle, we transform the collision detection problem into nonlinear programming problem of computing the minimum distance between objects, and then detect collision by the value of minimum distance. The rest of the paper is arranged as followed: section 2 outlines the mathematical background used in the method; section 3 puts forward the distance model based on vertex coordinate; section 4 illustrates the algorithm presented in this paper and at last we give the conclusion and the further study that we will do in the future.

II. ALGORITHM FOUNDATION

Definition 1. The convex combination of points $\chi_i, \cdots \chi_k$ is given by $\chi = \sum_{i=1}^{k} \lambda_i \chi_i$, where $\chi_i \in V, \lambda_i \geq 0, i = 1, \cdots; k$ and $\sum_{i=1}^{k} \lambda_i = 1$.

Definition 2. Let $A \subset V$, a subset of A is called convex set if $(x, y) = \{\lambda x + (1-\lambda) y \mid 0 < \lambda < 1\} \subset A$ for every $x, y \in A$.

Definition 3. Let $A \subset V$, the convex hull of A is given by the intersection of all the convex subset of A, denoted by $co(A)$.

Theorem 1. Let $A \subset V$, $co(A)$ equals to all the finite convex combination of the elements of A.

Proof. Suppose B is all the finite convex combination of the elements of A. If $\chi_1, \cdots, \chi_k \in A$, which is belonged

* This work is partially supported by the natural science fund of Hubei province.(Grant No2005ABA243)

to $co(A)$, then $\sum_{i=1}^{k}\lambda_i\chi_i \in co(A), \forall\lambda_i = 1.$ So $B \subset co(A)$. Let $x, y \in B$, here, it means the existence of $x_i, y_i \in A, \lambda_i \geq 0, \mu_i \geq 0, \sum_{i=1}^{k}\lambda_i = 1, \sum_{j=1}^{m}\mu_j = 1$, which make $x = \sum_{i=1}^{k}\lambda_i\chi_i, y = \sum_{j=1}^{m}\mu_j y_j$. The convex combination of x, y is given by $\lambda x + (1-\lambda)y = \sum_{i=1}^{k}\lambda\lambda_i x_i + \sum_{j=1}^{m}(1-\lambda)\mu_j y_j, \lambda \in (0,1)$, but $\lambda\lambda_i \geq 0, (1-\lambda)\mu_j \geq 0$ and $\sum_{i=1}^{k}\lambda\lambda_i + \sum_{j=1}^{m}(1-\lambda)\mu_j y_j = \lambda + (1-\lambda) = 1$,so $\lambda x + (1-\lambda)y \in B$ and B are convex sets. We have known $B \supset A$, so $B \supset co(A)$. From above proof, we can get $B = co(A)$. According to this, we can conclude that $co(x_1, \cdots x_m) = \{\sum_{i=1}^{m}\lambda_i x_i \mid \lambda_i \geq 0, \sum_{i=1}^{m}\lambda_i = 1\}$.

Definition 4. Convex polyhedron generated by $x_1, \cdots, x_m$ is defined by the $co(x_1 \cdots x_m\}$, where A is finite points set, written as $A = \{x_{1,} \cdots x_m\}$, and $x_1, \cdots x_m$ is called generated element.

Definition 5. Let $C = co(a_{1,} \cdots a_m)$, if an arbitrary point $a_1, \cdots a_m$ cannot be represented by the other points, then $a_1, \cdots a_m$ is called convex polyhedron vertex.

Theorem 2. If A is a $m * n$ matrix, $r(A) = m, b \in R^m$, S is a nonempty convex polyhedron, where $S = \{x \mid Ax = b, x \geq 0, b \geq 0\}$ .Let $x_1, \cdots x_k$ be the vertex of S , then $x \in S$ if and only if the existence of $x = \sum_{i=1}^{k}\lambda_i x_i$, where $\lambda_1 \cdots \lambda_k \geq 0, \sum_{i=1}^{k}\lambda_i = 1$.

Proof. We prove the necessity of this proposition firstly. For every $x \in S$, we can write it as $x = \lambda_1 x_1 + \lambda_2 x_2 + \cdots \lambda_j x_j$, where $\lambda_1 + \lambda_2 + \cdots \lambda_j = 1$, $\lambda_1, \lambda_2, \ldots, \lambda_j \geq 0$ and j is arbitrary integer. If $j \leq k$, then $x_1, \cdots, x_k$ is linear correlation, which is inconsistent with the proposition, so only $j > k$ is considered. In the situation of $k < j$, $j-1$ vectors $x_1 - x_j, x_2 - x_j, \cdots, x_{j-1} - x_j$ must be linear correlation. There are $j-1$ real numbers which are not all zero,making $v_1(x_1 - x_j) + v_2(x_2 - x_j) + \cdots + v_{j-1}(x_{j-1} - x_j) = 0$, denoted by $v_j = -\sum_{i=1}^{j-1}v_i$,here $v_1 x_1 + v_2 x_2 + \cdots + v_j x_j = 0$ and $\sum_{i=1}^{j}v_i = 0$ .Obviously, j numbers v_i / λ_i are not all zero, if we use η to denote the maximum, then $\eta > 0$ and the following formulas come into existence:

$$x = x - 0 = \sum_{i=1}^{j}(\lambda_i - \frac{v_i}{\eta})x_i ,$$

$\sum_{i=1}^{j=2}(\lambda_i - \frac{v_i}{\eta}) = \sum_{i=1}^{j}\lambda_i - \frac{1}{\eta}\sum_{i=1}^{j}v_i = 1$, $\lambda_i - \frac{v_i}{\eta} \geq 0$. Also, there is an h which is subject to $0 \leq h \leq j$, making $\lambda_h - \frac{v_h}{\eta} = 0$. In this way, we eliminate the polynomial of x_h in the expression of x . The calculating process is continued until the vertex number in the right of equation isn't greater than k , and then the necessity of the proposition is proved. Lastly, according to the definition of convex polyhedron, we can easily prove the sufficiency condition of the proposition.

III. DISTANCE MODEL

Suppose there are n object models in three-dimensional space, whose positions and poses always change along with time. Generally speaking, the propose of collision detection is judging whether the collision has happened or not. In view of computation geometry, we can apprehend collision detection like this: suppose R^3 is a three-dimensional space, which is represented by three-dimensional geometry coordinate system F_u ; if the set occupied by model i in F_u is denoted by F_i , obviously F_i is the subset of F_u, then four-dimensional geometry coordinate system C_n is constituted as F_u changes along with time; if subset of C_n given by the movement track of model i is denoted by C_i , then collision detection is actually judging whether or not $C_1 \cap C_2 \cap \cdots \cap C_n \neq \Phi$. Next, vertex is used to represent convex polyhedron, and in view of optimization, following algorithm determines whether intersection has happened between two convex polyhedra by computing their minimum distance or not.

Suppose convex polyhedron A is denoted by $A=\left\{x \mid x=\sum_{i=1}^{n_A}\lambda_i x_i, \sum_{i=1}^{n_A}\lambda_i=1, \lambda_i \geq 0\right\}$

And convex polyhedron B is represented by $B=\left\{y \mid y=\sum_{i=1}^{n_B}\sigma_i y_i, \sum_{i=1}^{n_B}\sigma_i=1, \sigma_i \geq 0\right\}$,

where n_A, n_B are the vertex numbers of objects A, B and $x_i, y_i \in R^3$ are the vertex coordination at a certain time respectively. According to definition of collision detection above, we know that objects A, B are separate if $A \cap B = \Phi$ and there is no collision; on the contrary, objects A, B are collided if $A \cap B \neq \Phi$ and the distance between them is 0.

Reference [8] showed that non-linear programming is solvable that aims at seeking the minimum value of n-variable function in special region. As a rule, variables of n functions are denoted by X , which is a vector of n-dimensional, and n functions are written as $f(x)$. Assume that S is a subset of domain of $f(x)$, and then the following formula (fs) $\begin{cases}\min & f(x) \\ s.t. & x \in s\end{cases}$ means that computing the minimum value of $f(x)$ in S , containing the minimum point x^* . Here x^* is the optimized solution and the corresponding function value f^* is the optimized value. Actually, optimization mentioned above is "how to do in the best way". Such model can end detecting collision and computing the minimum distance. Let space Euclid distance be the minimized function $f(x)$, that is $f(x)=\| y-x \|$, where $\forall x \in A, \forall y \in B$. On account of objects A, B , we can obtain the following mathematical model, which is used to detect collision by tracking the distance between two objects.

$$\begin{aligned}&\text{Min} \quad d_{A,B}=\| y-x \| \\ &\text{s.t.} \quad x=\sum_{i=1}^{n_A}\lambda_i x_i \\ &\quad y=\sum_{j=1}^{n_B}\sigma_j y_j \\ &\quad \sum_{i=1}^{n_A}\lambda_i=1 \\ &\quad \sum_{j=1}^{n_B}\sigma_j=1 \\ &\quad \lambda_i \geq 0 \quad i=1,\cdots,n_A \\ &\quad \sigma_j \geq 0 \quad j=1,\cdots,n_B\end{aligned} \tag{1}$$

Among them, λ_i, σ_j are variables and $x_i, y_j \in R^3$, which are constant, are the vertex coordination at a certain time. The meaning of the model is that: searching points λ_i, σ_j , which can minimize $d_{A,B}$, in the equations and inequalities satisfied restriction; if $d_{A,B}=0$, then there is collision between objects A, B and the point determined by λ_i, σ_j is the collided point; if $d_{A,B}>0$, then there is no collision between objects A, B and $d_{A,B}$ is the minimum distance. Note that, if the collision happens between vertex and vertex, vertex and edge or vertex and face, the collided point coordinate is only one; but if the collision happens between edge and edge, edge and face or face and face, the collided point coordinate is not just only one, because there may be more than one pair of collided point.

The algorithm mentioned above actually gets the solution by minimizing the Euclid distance step by step. After the obtained the vertex information, we know that the complexity of this algorithm is mainly determined by the complexity of optimization. Then we use reduced gradient to analyse complexity. As we know, changing radix operation at most needs $n-m$ times, detecting convergence at most needs $n-m$ times and the average inquiry time which adopts inexact inquiry method of one-dimensional is 2~3 [8]. Among the algorithm presented by this paper, $n=n_A+n_B$ and $m=2,$ so the algorithm complexity is $\mathrm{O}(n_A+n_B)^2$.

IV. Simulation Results

A. Collision between two convex polyhedra

Given the vertex coordinate of two tetrahedron, that is, A : (0,1,0) (1,0,0) (0,0,1) , (0,0,0) , and B : (1,0,0) (2,-2,0) (0,-2,2) (0,-2,0) , we can obtain an arbitrary point coordinate in object A and B respectively.

$$\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}=\lambda_1\begin{pmatrix}0\\1\\0\end{pmatrix}+\lambda_2\begin{pmatrix}1\\0\\0\end{pmatrix}+\lambda_3\begin{pmatrix}0\\0\\1\end{pmatrix}+\lambda_4\begin{pmatrix}0\\0\\0\end{pmatrix} \tag{2}$$

$$\begin{pmatrix}y_1\\y_2\\y_3\end{pmatrix}=\sigma_1\begin{pmatrix}1\\0\\0\end{pmatrix}+\sigma_2\begin{pmatrix}2\\-2\\0\end{pmatrix}+\sigma_3\begin{pmatrix}0\\-2\\2\end{pmatrix}+\sigma_4\begin{pmatrix}0\\-2\\0\end{pmatrix}. \tag{3}$$

Using the mathematical model (1) in Section 3, we have:

$$\begin{aligned}&\text{Min}\, d_{AB}=\sqrt{(\sigma_1+2\sigma_2-\lambda_2)^2+(2\sigma_2+2\sigma_3+2\sigma_4+\lambda_1)^2+(2\sigma_3-\lambda_3)^2} \\ &\text{s.t.} \quad \lambda_1+\lambda_2+\lambda_3+\lambda_4=1 \\ &\qquad \sigma_1+\sigma_2+\sigma_3+\sigma_4=1\end{aligned} \tag{4}$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$$
$$\sigma_1, \sigma_2, \sigma_3, \sigma_4 \geq 0$$

The nonlinear programming model is solved by using MATLAB6.5 [10]. Then, we have $\lambda_1 = 0.0000$, $\lambda_2 = 1.0000$, $\lambda_3 = 0$, $\lambda_4 = -0.0000$, $\sigma_1 = 1.0000$, $\sigma_2 = -0.0000$, $\sigma_3 = 0$, $\sigma_4 = 0.0000$, $d_{A,B} = 1.6548e-034$, it approximately equals to 0, so there is collision between the objects

If we put above values into (2) or (3), then we can get the collided point (1, 0, 0), consistent with the supposed situation.

B. Separation of two convex polyhedra

Suppose the vertex coordinate of tetrahedron A is (0,1,0),(1,0,0),(0,0,1),(0,0,0) and the vertex coordinate of tetrahedron B is (1,-1,0) , (2,-2,0) ,(0,-2,2) (0,-2,0) , an arbitrary point coordinate in object A or B is given by (5) and (6) respectively.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \lambda_1 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \lambda_3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \lambda_4 \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (5)$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \sigma_1 \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + \sigma_2 \begin{pmatrix} 2 \\ -2 \\ 0 \end{pmatrix} + \sigma_3 \begin{pmatrix} 0 \\ -2 \\ 2 \end{pmatrix} + \sigma_4 \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix} \quad (6)$$

Compute in the same way, we have:

$\lambda_1 = 0.0000$, $\lambda_2 = 1.0000$, $\lambda_3 = 0.0000$, $\lambda_4 = 0.0000$; $\sigma_1 = 1.0000$, $\sigma_2 = 0.0000$, $\sigma_3 = -0.0000$, $\sigma_4 = 0.0000$, $d_{A,B} = 1$, so two objects separate, and the minimum distance is 1. The result shows there is no collision between the objects, and is consistent with the supposed situation.

CONCLUSION

It is feasible that the method uses convex hull of vertex to represent convex polyhedron and detects collision by the distance of two objects. It converts the problem of computing the distance between two objects into nonlinear programming and greatly improves the efficiency of execution. What's more, this algorithm can calculate the collided point, and is ready for providing necessary data for collision response. The mainly further studies are: firstly, doing research on the intersection of objects that are more than two ones; secondly, putting time variable T into the distance model, and making this detection method adapt to the situation of object movement.

REFERENCES

[1] Jiaoying Shi.Basic virtual reality and applied algorithm[M].Beijing: science press,2002,416～240.

[2] M.C. Lin and J.F. Canny. A fast algorithm for incremental distance calculation[C]. IEEE Conf. on Robotics and Automation, 1991,1008～1014.

[3] Gilbert E.G., Johnson D.W., Keerthi S.S.. A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space [J]. IEEE Trans. on Robotics and Automation, 1988,4(2): 193～203.

[4] Stephen Cameron. A Comparison of Two Fast Algorithms for Computing the Distance between Convex Polyhedra[C], Proc.Of IEEE transactions on Robotics and Automation, 1996; 13(6): 915-920.

[5] Cameron SA. Collision Detection by Four-Dimensional Intersection Testing [J]. IEEE Trans. Robotics and Automation, 1990,6(3): 291～302.

[6] Stephen Cameron. A Study of the Clash Detection Problem in Robotics [J]. Int. Conf. Robotics and Automation, March 1985,488～493.

[7] Kou Shushun.Convex analysis and convex quadratic programming [M].Tianjin: Tianjin university press,1993,4～5.

[8] Liu Baoguang. Nonlinear programming [M].Beijing: Beijing university of science and technology press,1988,142～143.

[9] Zhiqiang Liu, Jiazhen Hong and Yang Hui. A summary of collision detection [J]. Transaction of software, 1999:10(5): 545-551.

[10]Science product development center of Feishi. Computation and design of assistant optimization with MATLAB 6.5 [M].Beijing:electronic industry press,2003,20～40.