

# 基于非线性规划理论的凸多面体 最小平移距离算法

周之平 张少博 吴介一 张飒兵

(东南大学 CIMS 中心, 南京 210096)

**摘 要** 凸多面体的最小平移距离问题一直以来都成为计算机图形学的一个研究热点。目前已有的距离算法在稳定性、可实现性、精确度和实现效率这几方面或多或少都存在一定的缺陷。为此,从最小平移距离定义出发,引入广义分离平面概念,提出一种用非线性规划求解距离问题的新算法。算法先定义一对最优广义分离平面以确定凸多面体最小平移距离;然后,将最优广义分离平面对的搜索问题等效变换为非线性规划问题;最后,用非线性优化工具软件对非线性规划问题进行求解,从而确定最小平移距离。实验结果表明:该算法能提供一个准确的距离值和实现向量,其性能优于其他同类算法;迭代次数与多面体的顶点数呈线性关系。此外,该算法只需提供顶点信息即可实现,求解过程中避免了死循环,故实现简单、可靠。因此,此算法是一种快速而有效的距离算法。

**关键词** 凸多面体 最小平移距离 分离平面 实现向量 非线性规划

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2006)10-1487-07

## A Minimum Translational Distance Algorithm of Convex Polyhedra Based on Nonlinear Programming Theory

ZHOU Zhi-ping, ZHANG Shao-bo, WU Jie-yi, ZHANG Sa-bing

(The CIMS Center of Southeast University, Nanjing 210096)

**Abstract** The problem of minimum translational distance (MTD for short) of convex polyhedra is always an active subfield of computer graphics. The current distance algorithms are deficient in such requirements as stability, realizability, accuracy and efficiency more or less. In order to overcome these limitations, the generalized separable plane is introduced based on the definition of MTD and a new algorithm of the MTD problem using nonlinear programming is presented in the paper. This algorithm is carried out as follow. Firstly, the MTD measure is determined by defining the optimal generalized separable plane-pair. Secondly, the problem of searching the optimal plane-pair is equivalent with nonlinear programming problem under some transforms. Finally, a nonlinear optimization software is used to solve the equivalent model, and therefore MTD measure is determined by the solution. The results show that the proposed algorithm performs linearly with the size of model and over the other algorithms in most of the tests. Besides, it can provide both an accurate measure and the witness vector in a few iterations, which are gently linear with the vertex number. In addition, the implementation is simple and reliable, because only the information of vertex is required and the cycle can be avoided. So, it is a fast and efficient distance algorithm.

**Keywords** convex polyhedral, minimum translational distance, separable plane, witness vector, nonlinear programming

基金项目:江苏省自然科学基金重点项目(BK2001204)

收稿日期:2005-12-11; 改回日期:2006-04-29

第一作者简介:周之平(1976~),男。现为东南大学控制理论与控制工程专业博士研究生。主要研究方向为虚拟装配和计算机图形学。E-mail: zhousam@seu.edu.cn

## 1 引言

距离算法的精确性和性能一直是图形学中的研究难点,距离信息及其实现向量等关键信息对于力反馈交互仿真环境、运动规划等图形学应用领域来说尤为重要。

最小平移距离概念最先由 Cameron 和 Cully 在文献[1]提出,它是最小分离距离和最小渗透量的统称。计算最小分离距离的主流算法有 Lin-Canny (LC) 算法<sup>[2]</sup>和 Gilbert-Johnson-Kerrthi (GJK) 算法<sup>[3]</sup>。LC 算法通过对边界迭代搜索最近特征对来实现,GJK 算法在隐式构造 Minkowski 和  $M$  的基础上,利用支撑映射理论,搜索距离原点最近的单纯形,进而计算多面体之间最近距离。但是这两种算法都只能计算分离距离,LC 算法必须构造所有特征 VR 区域,其实现的复杂程度较高;当最近点位于几何中心位置时,GJK 算法会陷入死循环。Bergen 提出了求解凸多面体最小平移距离问题的改进 GJK 算法<sup>[4]</sup>。该算法通过如下策略改善了 GJK 算法性能并使其能准确地求解穿透深度:(1)改进终止判定条件,当原点到单纯形的距离为垂直距离时,如果新搜索到的支撑顶点为单纯形的顶点,则终止迭代过程;(2)当多面体之间发生干涉时,变更搜索向量方向,从原点  $O$  最近的边界面  $f^*$  出发,迭代搜索  $M$  边界上距离原点  $O$  最近的点,从而确定穿透深度;(3)求解穿透深度时,若搜索向量为零向量时,引入附加顶点拓展单纯形对象,再进行迭代搜索。

近年来,也有不少学者从光栅化硬件,高斯球映射等多个角度来求解多面体的穿透深度问题<sup>[5-7]</sup>。文献[5]、[6]采用硬件实现,实现的复杂程度较高。文献[7]采用高斯球映射方法完成特征对搜索,特征映射变换之后可能会出现死循环。此外,也有学者从线性规划<sup>[8-10]</sup>、Minkowski 边界面构造<sup>[11]</sup>、构建 Hop-filed 神经网络模型<sup>[12]</sup>等方面对凸多面体的距离问题进行了深入的研究。文献[8]提出一种伪平移距离概念,并给出了一种线性规划求解方法,但是其不能提供一个准确值。文献[9]提出一种“增长函数”(或称为“膨胀函数”)概念,将最小平移问题转化为线性规划问题,但是其只能提供一个近似量,其数值受多面体最小外接球半径影响很大。文献[10]利用文献[9]的求解原理,计算基于 1 范数的穿透深度,虽然求解过程相对简单,但不能提供一

个近似值。文献[11]从构造  $M$  边界面出发,搜索  $M$  上距离原点最近的面来求解穿透深度,其复杂度很高。文献[12]提出一种神经网络计算模型,收敛速度很慢。

基于上述原因,本文就凸多面体的最小平移距离问题进行深入研究,将其求解转化为最优广义分离平面的搜索问题,利用非线性规划理论来增强算法的精确性和性能。考虑到文献[4]算法的稳定性和准确性,选择文献[4]算法与本文算法进行比较,以突出本文算法的性能。

## 2 凸多面体最小平移距离问题描述

### 2.1 最小平移距离定义

设  $A, B$  为两个凸多面体模型,其顶点序列分别为  $v_{ai}, v_{bj}, i=1, 2, \dots, m, j=1, 2, \dots, n$ 。 $A, B$  可用其顶点的凸组合表示:

$$A: \{x \mid x = \sum \lambda_i v_{ai}, \sum \lambda_i = 1, \lambda_i \geq 0\} \quad (1)$$

$$B: \{y \mid y = \sum \mu_j v_{bj}, \sum \mu_j = 1, \mu_j \geq 0\} \quad (2)$$

$A, B$  的最小平移距离为

$$d^+(A, B) = \begin{cases} \min \{ \|z\|_2, z \in \partial M, A \cap B = \emptyset \} \\ -\min \{ \|z\|_2, z \in \partial M, A \cap B \neq \emptyset \} \end{cases} \quad (3)$$

其中,  $M = \{z \mid z = x - y, \forall x \in A, y \in B\}$ ,  $\partial M$  为  $M$  的边界。

### 2.2 用一对广义分离平面确定凸多面体最小平移距离

根据文献[13]可知,当  $A, B$  可分离时,可以用一对分离平面确定它们的距离。当  $A, B$  发生干涉时,同样可以用一对平行平面确定它们的穿透深度。

**命题 1** 当  $A \cap B \neq \emptyset$  时,若能找到距离最近的一对平面  $H_1, H_2$  (如图 1 所示),其中:

$$H_1(x) = \{x \mid (x - x^*)^T (y^* - x^*) = 0\}$$

$$H_2(y) = \{y \mid (y - y^*)^T (y^* - x^*) = 0\}$$

$x^* \in A, y^* \in B, y^* - x^* \neq 0$ ,使得  $A$  位于  $H_1$  的左半平面,  $B$  位于  $H_2$  的右半平面,则  $x^*, y^*$  构成一对实现点,它们可以唯一确定最小穿透深度及其实现向量。

**证明** 若  $H_1, H_2$  满足上述约束条件,则有

$$\forall x \in A, (x - x^*)^T (y^* - x^*) \geq 0$$

$$\forall y \in B, (y - y^*)^T (y^* - x^*) \leq 0$$

成立。

因此

$$(y - x)^T (y^* - x^*) - (y^* - x^*)^T (y^* - x^*) =$$

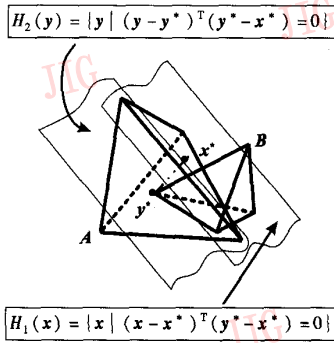


图 1 A, B 干涉时, 分离平面对  $H_1, H_2$

Fig. 1 The separable plane-pair when interfered

$(y - y^*)^T (y^* - x^*) + (x^* - x)^T (y^* - x^*) \leq 0$   
成立。

设  $z = y - x, z^* = y^* - x^*$ , 可知  
 $(z - z^*)^T (-z^*) \geq 0$

成立。因为  $O \in M$ , 所以  $z^* \in \partial M$  必定成立。又因为  $d(H_1, H_2) = \|y^* - x^*\|_2 = \|z^*\| > 0$ , 且  $H_1, H_2$  是距离最近的一对平面, 所以  $\|z^*\| = \min \{\|z\|_2, z \in \partial M\}$ , 即  $H_1, H_2$  可以唯一确定 A, B 的穿透深度。命题成立。

**命题 2** 当  $A \cap B = \emptyset$  时, 若能找到一对平面  $H_1, H_2$  (如图 2 所示), 其中:

$$H_1(x) = \{x \mid (x - x^*)^T (y^* - x^*) = 0\}$$

$$H_2(y) = \{y \mid (y - y^*)^T (y^* - x^*) = 0\}$$

$x^* \in A, y^* \in B$ , 使得 A 位于  $H_1$  的右半平面, B 位于  $H_2$  的左半平面, 则  $x^*, y^*$  构成一对实现点, 它们可以唯一确定最小分离及其实现向量。

**证明** 若  $H_1, H_2$  满足上述约束条件, 则有  $\forall x \in A, (x - x^*)^T (y^* - x^*) \leq 0, \forall y \in B, (y - y^*)^T (y^* - x^*) \geq 0$  成立。

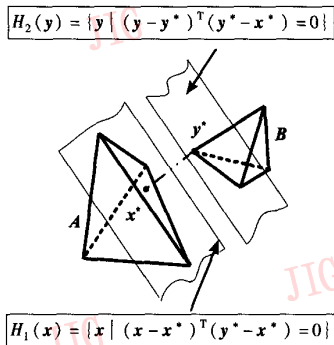


图 2 A, B 分离时, 广义分离平面对  $H_1, H_2$

Fig. 2 The separable plane-pair when separable

因此

$$(y - x)^T (y^* - x^*) - (y^* - x^*)^T (y^* - x^*) \geq 0.$$

设  $z = y - x, z^* = y^* - x^* \neq 0$ , 即  $(z - z^*)^T z^* \geq 0$  成立。

因为  $O \notin M, z^* \in \partial M$ , 进一步可以判定

$$\|z^*\|_2 = \min \{\|z\|_2, z \in \partial M\}$$

又因为

$$d(H_1, H_2) = \|y^* - x^*\|_2 = \|z^*\|_2$$

所以  $H_1, H_2$  可以唯一确定 A, B 的分离距离。

故, 命题成立。

为了不引起概念上的混淆, 称文中所述的  $H_1, H_2$  为一对广义分离平面。

### 3 基于非线性规划理论的凸多面体最小平移距离算法 (NLPBA)

根据 2.2 节可知, 可以通过搜索一对广义分离平面来确定 A, B 的最小平移距离。因为多面体与平面的位置关系可以通过判定其顶点与平面的位置关系确定, 所以, 命题 1 等价于求解非线性规划问题

$$\begin{aligned} & \min_{x \in A, y \in B} \|y - x\|^2 \\ \text{s. t. } & \begin{cases} (v_{a_i} - x)^T (y - x) \geq 0, \forall i = 1, 2, \dots, m \\ (v_{b_j} - y)^T (y - x) \leq 0, \forall j = 1, 2, \dots, n \\ y - x \neq 0 \end{cases} \end{aligned} \quad (4)$$

命题 2 等价于求解非线性规划问题

$$\begin{aligned} & \min \|y - x\|^2 \\ \text{s. t. } & \begin{cases} (v_{a_i} - x)^T (y - x) \leq 0, \forall i = 1, 2, \dots, m \\ (v_{b_j} - y)^T (y - x) \geq 0, \forall j = 1, 2, \dots, n \end{cases} \end{aligned} \quad (5)$$

因为 A, B 可分离时  $H_1, H_2$  是唯一的, 对式 (5) 的目标函数求最小值不影响运算结果。

#### 3.1 NLP 等价模型的提出

通过不等式移项, 可将式 (4) 和 (5) 改写成如下形式:

$$\begin{aligned} & \min_{x \in A, y \in B} \|y - x\|^2 \\ \text{s. t. } & \begin{cases} v_{a_i}^T (y - x) \geq (y - x)^T x, \forall i = 1, 2, \dots, m \\ v_{b_j}^T (y - x) \leq (y - x)^T y, \forall j = 1, 2, \dots, n \\ y - x \neq 0 \end{cases} \end{aligned} \quad (6)$$

和

$$\begin{aligned} & \min_{x \in A, y \in B} \|y - x\|^2 \\ \text{s. t. } & \begin{cases} v_{a_i}^T (y - x) \leq (y - x)^T x, \forall i = 1, 2, \dots, m \\ v_{b_j}^T (y - x) \geq (y - x)^T y, \forall j = 1, 2, \dots, n \end{cases} \end{aligned} \quad (7)$$

对于式(6),因为  $x = \sum_{i=1}^m \lambda_i v_{ai}, \sum_{i=1}^m \lambda_i = 1, \lambda_i \geq 0$ ,

故  $(y-x)^T x = \sum \lambda_i v_{ai}^T (y-x) \geq \min_i \{v_{ai}^T (y-x)\}$  成立。

又因为  $\min_i \{v_{ai}^T (y-x)\} \geq (y-x)^T x$ , 所以,  $(y-x)^T x = \min_i \{v_{ai}^T (y-x)\}$ 。同理可知

$$(y-x)^T y = \max_j \{v_{bj}^T (y-x)\}。$$

对于式(7),同理可知

$$(y-x)^T x = \max_i \{v_{ai}^T (y-x)\}$$

$$(y-x)^T y = \min_j \{v_{bj}^T (y-x)\}$$

所以,分别令

$$s_1 = \min_i \{v_{ai}^T (y-x)\}$$

$$s_2 = \max_i \{v_{ai}^T (y-x)\}$$

$$t_1 = \max_j \{v_{bj}^T (y-x)\}$$

$$t_2 = \min_j \{v_{bj}^T (y-x)\}$$

则式(6)和(7)分别等价于

$$\begin{aligned} & \min_{x \in A, y \in B} \|y-x\|_2^2 \\ \text{s.t. } & \begin{cases} v_{ai}^T (y-x) \geq s, \forall i=1,2,\dots,m \\ v_{bj}^T (y-x) \leq t, \forall j=1,2,\dots,n \\ t-s = \|y-x\|_2^2 \\ y-x \neq 0 \end{cases} \end{aligned} \quad (8)$$

和

$$\begin{aligned} & \min_{x \in A, y \in B} \|y-x\|_2^2 \\ \text{s.t. } & \begin{cases} v_{ai}^T (y-x) \leq s, \forall i=1,2,\dots,m \\ v_{bj}^T (y-x) \geq t, \forall j=1,2,\dots,n \\ t-s = \|y-x\|_2^2 \end{cases} \end{aligned} \quad (9)$$

### 3.2 NLP 模型等价性证明

(1)证明式(8)与式(6)等价。

证明 假设  $t^*, s^*, y^*, x^*$  为式(8)的最优解,有

$$\begin{aligned} x^{*T} (y^* - x^*) &= \sum \lambda_i v_{ai}^T (y^* - x^*) \\ &\geq \sum \lambda_i \min_i \{v_{ai}^T (y^* - x^*)\} \\ &= s^* \end{aligned}$$

成立。同理  $y^{*T} (y^* - x^*) \leq t^*$  成立。故

$$(y^* - x^*)^T (y^* - x^*) - (t^* - s^*) \leq 0$$

要使得式(8)中的等式约束成立,则必定有

$$(y^* - x^*)^T y^* = t^*$$

$$(y^* - x^*)^T x^* = s^*$$

因此,式(8)与式(6)是严格等价的。

(2)证明式(9)与式(7)等价。

设  $t^*, s^*, y^*, x^*$  为式(9)的最优解,同理可以证明

$$(y^* - x^*)^T y^* = t^*$$

$$(y^* - x^*)^T x^* = s^*$$

因此,式(9)与式(7)是严格等价的。

### 3.3 NLP 等价模型的求解

将式(1)和(2)代入式(8)和(9),为了求解方便,引入变量  $\beta = \sum_{i=1}^m \lambda_i v_{ai} - \sum_{j=1}^n \mu_j v_{bj}$ , 则式(8)和

式(9)可以进一步转化为

$$\begin{aligned} & \min \|\beta\|_2^2 \\ \text{s.t. } & \begin{cases} v_{ai}^T \beta \geq s, \forall i=1,2,\dots,m \\ v_{bj}^T \beta \leq t, \forall j=1,2,\dots,n \\ t-s = \|\beta\|_2^2 \\ t-s > 0 \\ \beta - \sum \mu_j v_{bj} + \sum \lambda_i v_{ai} = 0 \\ \sum \lambda_i = 1, 0 \leq \lambda_i \leq 1 \\ \sum \mu_j = 1, 0 \leq \mu_j \leq 1 \end{cases} \end{aligned} \quad (10)$$

和

$$\begin{aligned} & \min \|\beta\|_2^2 \\ \text{s.t. } & \begin{cases} v_{ai}^T \beta \leq s, \forall i=1,2,\dots,m \\ v_{bj}^T \beta \geq t, \forall j=1,2,\dots,n \\ t-s = \|\beta\|_2^2 \\ t-s > 0 \\ \beta - \sum \mu_j v_{bj} + \sum \lambda_i v_{ai} = 0 \\ \sum \lambda_i = 1, 0 \leq \lambda_i \leq 1 \\ \sum \mu_j = 1, 0 \leq \mu_j \leq 1 \end{cases} \end{aligned} \quad (11)$$

这是关于  $m+n+3$  变量,  $m+n+5$  约束的非线性规划问题。一旦模型参数给定,便可以用 LINDO 软件<sup>[14]</sup>来求解。

### 3.4 判定凸多面体位置关系

采用非线性规划求解最小平移距离时,必须事先知道多面体的位置关系,才能建立相应的等价模型。RLP 算法<sup>[15]</sup>是一种判定凸多面体位置关系的快速算法,它只需要提供顶点的信息即可,其性能稳健而可靠,实现起来简单,故利用它判断两多面体是否可分离。其主要思想就是判断下面这个线性规划问题是否有可行解。

$$\min_{\omega, \gamma, y_i, z_j} \frac{1}{m} \sum_{i=1}^m y_i + \frac{1}{n} \sum_{j=1}^n z_j$$

$$\text{s. t. } \begin{cases} y_i + \mathbf{v}_{a_i}^T \boldsymbol{\omega} - \gamma - 1 \geq 0, \forall i = 1, 2, \dots, m \\ z_j - \mathbf{v}_{b_j}^T \boldsymbol{\omega} + \gamma - 1 \geq 0, \forall j = 1, 2, \dots, n \\ y_i, z_j \geq 0, \boldsymbol{\omega} \in \mathbf{R}^3 \end{cases} \quad (12)$$

这是一个关于  $m + n + 2$  变量,  $m + n$  约束的线性规划问题, 可用 LINDO 软件来求解。

### 3.5 NLPBA 算法描述

- (1) 输入多面体  $A, B$  的顶点数据
- (2) 构造形如式(12)的线性模型, 利用 LINDO 软件求解。若有可行解, 则说明  $A, B$  可分离, 转第 3 步; 否则, 说明  $A, B$  发生干涉, 转第 4 步;
- (3) 构造求解分离距离的非线性模型(如式(11)), 转第 5 步;
- (4) 构造求解穿透深度的非线性模型(如式(10)), 转第 5 步;
- (5) 用 LINDO 软件求解计算最小平移距离, 并返回其对应的实现点和实现向量。

## 4 算法性能分析

### 4.1 模型选择

文献[13]采用球状多面体、锥状多面体和柱状多面体这三种对象进行干涉仿真, 但是, 采用锥状多面体和柱状多面体进行动态仿真时, 因为其几何形状具有一定的狭长性, 所以难以设定一条合理的运动路径使得“干涉”和“分离”这两种事件发生的几率均衡。采用椭球状多面体也同样面临这个问题。而采用对空间随机采样方法构造凸包, 凸包的顶点数具有随机性, 不易统计数据。为此, 选择球状多面体对象, 这是因为: (1) 对球表面均匀采样获得的点本身具有凸包性质, 不必再进行凸包构造, 实现起来比较方便; (2) 可以设定适当的运动路径( $A$  绕 XOY 平面作旋转运动,  $B$  作直线运动)使得  $A, B$  的几何位置关系具有一般性, 可以保证“干涉”和“分离”出现的几率均衡; (3) NLPBA 算法和文献[4]的算法都只需要顶点几何信息, 对处于一般几何关系的两球状多面体进行动态仿真可以体现上述算法的一般性能。为表示方便起见, 称文献[4]的算法为 RBGJK (Revised GJK algorithm by Bergen) 算法。

### 4.2 性能比较

对本文算法进行了实现, 程序用 Visual C++ 6.0 编写, 在 PIV 1.7GHZ, 256MB 内存下的 Windows

2000 系统环境中运行, 选择一对球状多面体  $A, B$  测试。

为研究方便起见, 设  $A, B$  的顶点数目均为  $m$  ( $m = n$ ),  $A, B$  分别为由球体  $S_1$  和  $S_2$  表面均匀采样  $m$  个点构成的凸包。其中:  $S_1$  和  $S_2$  的几何形式分别为

$$x^2 + y^2 + z^2 \leq 100$$

$$(x - 20)^2 + (y - 10)^2 + (z - 30)^2 \leq 100$$

$m$  的变化范围为 10 ~ 1000。

对球状多面体  $A, B$  进行动态仿真, 其中  $A$  以  $\omega_A = 0.05 \text{rads/s}$  绕 XOY 平面作旋转运动,  $B$  以速度  $\mathbf{V}_B = -10^{-2}(2\mathbf{i} + \mathbf{j} + 3\mathbf{k})$  作直线运动, 采样间隔为  $\Delta_T = 1\text{s}$ , 计算  $T_k = k\Delta_T$  ( $k = 0 \sim 2000$ ) 时  $A, B$  的距离。对 NLPBA 算法所需的运算时间、存储需求和迭代次数进行统计, 并与 RBGJK 算法进行比较, 实验结果如图 3 ~ 图 6 所示。

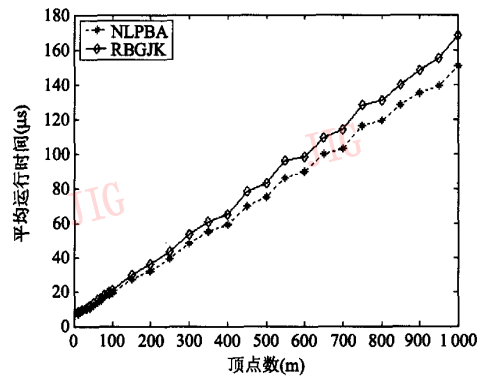


图 3  $A, B$  分离时, 平均运行时间比较

Fig. 3 The comparison of average execution time when two objects are separable

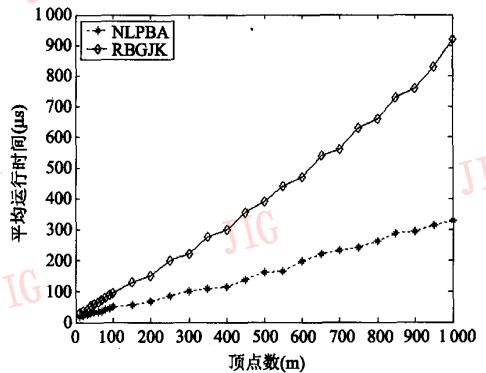


图 4  $A, B$  干涉时, 平均运行时间比较

Fig. 4 The comparison of average execution time when two objects are interfered

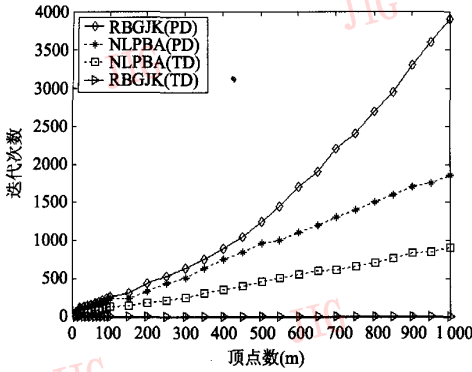


图 5 迭代次数比较(“TD”分离,“PD”干涉)

Fig. 5 The comparison of average iteration times(“TD” or “PD” denotes separable or interfered respectively)

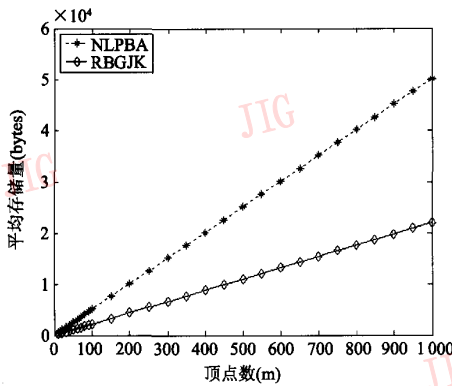


图 6 所需的平均存储量比较

Fig. 6 The comparison of average memory occupied

从图 3 可知,当  $A, B$  分离时, NLPBA、RBGJK 算法求解最小分离距离所需的时间与  $m$  几乎呈线性关系, NLPBA 算法的运行时间比 RBGJK 减少 8% ~ 10% 左右。

根据图 4 可知,当  $A, B$  发生干涉时, RBGJK 算法求解最小穿透深度所需的时间与  $m$  呈平方关系, 相比之下 NLPBA 算法运行时间仍然与顶点数目  $m$  呈线性关系, NLPBA 计算时间比 RBGJK 节省了 40% ~ 60% 左右。

从图 5 可以看出, NLPBA 算法的迭代次数基本上是随  $m$  线性变化的, 而且变化的幅度比较小; 而 RBGJK 在求解分离距离时, 迭代次数基本上是一个常数, 在求解最小穿透深度时, 迭代次数随着  $m$  呈平方关系急剧增加, 这是因为  $M$  的边界面数目与  $m$  呈平方关系。

从图 6 数据可知, 采用 NLPBA 算法大约需 50m.

字节的存储量(非线性模型的数据、指令集), 而 RBGJK 算法大约需要 20m 字节的存储量。

由于 NLPBA 算法采用基于梯度下降的迭代规则, 通过少量的迭代就可找到最优解, 其运算时间和迭代次数与顶点数目  $m$  呈线性关系, 算法效率优于 RBGJK, 特别是求解最小穿透深度值时; 其次, NLPBA 算法可避免陷入死循环, 且仅需要提供顶点坐标信息就可以完成距离计算。因此, NLPBA 是一种快速而有效的距离算法。

### 5 结 论

针对凸多面体对象的最小平移距离问题, 从最小平移距离的定义出发, 将其转化为最优广义分离平面的搜索问题。通过代数变换, 建立非线性等价模型以确定最优广义分离平面, 将最小平移距离问题变换为非线性最优化问题, 利用非线性优化工具对之求解, 并与同类算法进行性能比较。实验结果表明, 该算法能准确计算最小平移距离及其实现向量, 其运行时间与顶点数目呈线性关系, 性能优于 RBGJK 算法。今后, 将对 3 维曲面物体和非凸多面体模型的距离计算进行深入研究。

### 参考文献 (References)

- 1 Cameron S A, Culley R K. Determining the minimum translation distance between two convex polyhedra[A]. In: Proceeding of IEEE International Conference of Robot and Automations [ C ], San Francisco, CA, USA, 1986: 591 ~ 596.
- 2 Lin M C, Canny J F. Fast algorithm for incremental distance calculation[A]. In: Proceeding of IEEE International Conference on Robotics and Automation [ C ], Sacramento, USA, 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 Journal of Robotics and Automation, 1988, 4(2): 193 ~ 203.
- 4 Bergen G. Proximity queries and penetration depth computation on 3D game objects[A]. In: Proceedings of Game Developers Conference [ C ], San Jose, USA, 2001: 821 ~ 837.
- 5 Kim Y J, Otaduy M A, Lin M C. Fast Penetration Depth Computation Using Rasterization Hardware and Hierarchical Refinement[R]. TR02-014, North Carolina, Chapel Hill, USA: Department of Computer Science, University of North Carolina, 2002.
- 6 Kim Y J, Otaduy M A, Lin M C. Fast penetration depth computation for physically-based animation [ A ]. In: Proceeding of ACM

- SIGGRAPH Symposium on Computer Animation [C], New York, USA: ACM Press, 2002; 23 ~ 32.
- 7 Kim Y J, Lin M C, Monocha D. Incremental penetration depth estimation between convex polytopes using dual-space expansion[J]. IEEE Transactions on Visualization and Computer Graphics, 2004, 10(2): 152 ~ 163.
- 8 Zhu Xiang-yang, Ding Han, Xiong You-lun. The pseudo minimum translational distance between convex polyhedral: definitions and properties [J]. Science in China (Series E), 2001, 31(2): 128 ~ 136. [朱向阳, 丁汉, 熊有伦. 凸多面体之间的伪最小平移距离—I. 定义及其性质[J]. 中国科学(E辑), 2001, 31(2): 128 ~ 136.]
- 9 Ong C J. A fast growth distance algorithm for incremental motions [J]. IEEE Transactions on Robotics and Automation, 2000, 16(6): 880 ~ 890.
- 10 Sridharan K. Efficient computation of a measure of depth between convex objects for graphics applications[J]. Computers & Graphics, 2002, 26(5): 785 ~ 793.
- 11 Shih C L, Liu J Y. Computing the minimum directed distances between convex polyhedra [J]. Journal of Information Science and Engineering, 1999, 15(3): 353 ~ 373.
- 12 Yuan J. A neural network measuring the intersection of m-dimensional convex polyhedra [J]. Automatica, 1995, 31(4): 517 ~ 531.
- 13 Heckbert P S. Graphics Gems IV [M], Boston: Academic Press, 1994.
- 14 LINDO API 4.0 [CP/OL]. <http://www.lindo.com/downloads/LAPI-WINDOWS-IA32-4.0.zip>, 2005-05-23.
- 15 Bennett K P, Mangasarian O L. Neural Network Training via Linear Programming [R]. TR948, Wisconsin, Madison, USA: Department of Computer Science, University of Wisconsin-Madison, 1990.