

The Geometric Iteration Method for Computing the Minimum Distance between Two Spatial Circles

Xiaowu Li¹, Zhinan Wu², Lin Wang¹, Mingsheng Zhang¹

¹College of Information Engineering, Guizhou Minzu University, Guiyang Guizhou

²School of Mathematics and Computer Science, Yichun University, Yichun Jiangxi

Email: lixiaowu002@126.com, wanglin@gzmu.edu.cn, zhangmscomputer@163.com

Received: Nov. 1st, 2015; accepted: Nov. 18th, 2015; published: Nov. 24th, 2015

Copyright © 2015 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Computing the minimum distance between two spatial circles is the base of collision detection and intersection in the fields of computer graphics, computer-aided design and computer-aided geometric design. This paper has completely analyzed and discussed the minimum distance problem between two spatial circles for their spatial position relationships. If the two central axes of two spatial circles are not paralleled, we have presented the algorithm for computing the minimum distance between two spatial circles based on the geometric iterative method. Besides, if two central axes of two spatial circles have an intersection, we also have presented two pairs of corresponding points of the minimum distance for two spatial circles based on the geometric iterative method; if two central axes of two spatial circles are paralleled or coincident, we have directly provided the analytical expressions for computing the minimum distance between two spatial circles. Numerical examples illustrate that the algorithms are efficient and robust.

Keywords

Two Spatial Circles, The Minimum Distance, Geometric Iteration Method, Central Axis

几何迭代方法计算空间两圆之间的最近距离

李小武¹, 吴志男², 王 林¹, 张明生¹

¹贵州民族大学信息工程学院, 贵州 贵阳

²宜春学院, 数学与计算机科学学院, 江西 宜春

Email: lixiaowu002@126.com, wanglin@gzmu.edu.cn, zhangmscomputer@163.com

收稿日期: 2015年11月1日; 录用日期: 2015年11月18日; 发布日期: 2015年11月24日

摘 要

空间两圆之间的最近距离计算是计算机图形学、计算机辅助设计、计算机辅助几何设计等领域进行碰撞检测和相交计算问题的基础。本文对任意位置关系的空间两圆的最近距离进行了完整分析与讨论。空间两圆的中心轴不平行时, 提出了基于几何迭代方法的空间两圆最近距离的求解算法, 当空间两圆中心轴有交点时, 本文给出了两圆最近距离的两对对应点; 当空间两圆的中心轴平行或重合时, 本文给出两圆最近距离的解析表达式。最后通过若干例子显示本文方法的稳定性和有效性。

关键词

空间两圆, 最近距离, 几何迭代方法, 中心轴

1. 引言

空间物体之间的最近距离计算是计算机图形学、计算机辅助设计、计算机辅助制造、计算机辅助几何设计、机器人、数控加工等领域中的重要话题。空间两物体之间的最近距离不仅是机辅助设计/计算机辅助制造中的碰撞检测必要环节之一, 同时也是计算机图形学中的碰撞检测中的重要环节[1] [2]。对于多面体之间最近距离的研究有[2]-[5], 关于空间两曲面最近距离的研究有[2] [6]小雕等[7]给出了基于点圆最近距离计算的圆环面/球面求交算法。该算法直接判断处理无交、相切于一点、交于一个圆以及交于两个圆等简单情况; 对于其他情况, 通过求解一元四次关键方程得到相应的关键值, 对大圆参数区间进行划分, 然后通过简单的符号判断来确定相交的参数子区间, 并直接给出交曲线段的参数表达式。文献[8]把圆环面之间的最近距离问题转换成三维空间中两圆的最小距离问题, 并用群论的方法证明了这个问题没有闭式解, 需要求解一个一元八次方程, 但文献没有给出具体求解过程。刘晓明等[9]指出文献[8]的各种位置关系判断并不是完全的。文献[10]用迭代方法求解空间两圆的最近距离问题, 首先在空间任意取一点, 直接计算该点至第一圆的最近距离的坐标点, 然后再由该坐标点直接计算第二圆的最近距离的坐标点, 在反复迭代直至求出空间两圆的最近距离。该算法虽然能求出两圆全局的最近距离, 但该算法没有给出空间两圆最近距离的两对对应点。文献[11]也设计了一个迭代算法求出两个空间椭圆的最近距离问题。文献[10] [11]相同之处在于都采用了迭代思想来逼近两圆或两椭圆的最近距离, 都没有分析讨论最近距离的两对对应点以及中心轴平行和重合的特殊位置情况。文献[12]以几何方法为基础, 通过判断圆环中心圆之间的位置关系来判定相交区域, 并运用数值分析方法精确计算出每条交线的初始点, 然后用跟踪法求出交线。文献[9]证明了空间两圆 Hausdorff 距离总可以表示为它们的一对共线法向点的距离, 给出了计算空间两圆的共线法向点的详细过程, 并对这些共线法向点进行筛选, 求出了空间两圆的最近距离和 Hausdorff 距离。文献[9]还给出了两圆环包含、分离和相交时的充要条件, 解决了两圆环面之间的最近距离计算问题, 从理论上对两圆环的位置关系给出了严格的判别方法, 但具体的实现却依赖于用数值方法求解八次方程。最近关于空间曲线最近距离的研究有[13]-[19]。

本文利用几何迭代方法完整地研究了空间两圆任意位置关系的最近距离的算法。当空间两圆中心轴不平行时,对任意初始迭代点,求出该点至第一圆的最近距离的正交投射点,由正交投射点经过逆坐标变换和坐标变换正交投射至第二圆得到第二圆的正交投射点,然后再把该点作为初始迭代点重复上述步骤反复迭代,直至求出空间两圆的最近距离。当空间两圆的中心轴有公共点时,还可以求出空间两圆最近距离的另外一对对应的点;当空间两圆的中心轴平行,文章给出了简化版的两圆最近距离的迭代算法,当第二圆的正交投射至第一圆所在平面时,与第一圆有交点时,我们还给出了另外一对两圆最近距离的对应点。此外,文章还为两圆的中心轴平行或重合情况给出了求解最近距离的表达式。上述实例显示本文算法是有效和鲁棒的。

2. 算法分析与实现及实例

2.1. 两中心轴一般位置(不平行)的几何迭代算法

$$\text{第一个圆 } C_0: \begin{cases} x = R \cos(\theta) \\ y = R \sin(\theta) \\ z = 0 \end{cases} \text{ 和第二原始位置圆 } C_1: \text{ 为 } \begin{cases} x = R_1 \cos(u) \\ y = R_1 \sin(u) \\ z = 0 \end{cases}, \text{ 圆 } C_1 \text{ 经过坐标变换得到圆 } C_2,$$

其中圆 C_2 的中心轴的向量和圆心点分别为 $\langle V_x, V_y, V_z \rangle$ 和 (p_x, p_y, p_z) 。

目标是求圆 C_0 和 C_2 的最近距离,空间两圆最近距离的基本思路为:

- I) 正交投射空间任意点至第一圆 C_0 得到对应的正交投射点;
- II) 正交投射第一圆上的正交投射点至第二圆 C_2 得到对应的正交投射点;
- III) 反复进行上述两步直至求出第一圆 C_0 与第二圆 C_2 之间最近距离。

下面按照三步步骤思路具体实现空间两圆最近距离的相关算法。

由于圆 C_1 的中心轴的向量和圆心点分别是 $\langle V_x, V_y, V_z \rangle = \langle 0, 0, V_z \rangle (V_z \neq 0)$ 和 $(p_x, p_y, p_z) = (0, 0, 0)$ 。故圆 C_1 转换为圆 C_2 的坐标变换过程为

$$\begin{cases} x' = p_x + \frac{\cos(\beta)}{\sin(\gamma)} x + \cos(\alpha) \frac{\cos(\gamma)}{\sin(\gamma)} y + \cos(\alpha) z, \\ y' = p_y - \frac{\cos(\alpha)}{\sin(\gamma)} x + \cos(\beta) \frac{\cos(\gamma)}{\sin(\gamma)} y + \cos(\beta) z, \\ z' = p_z - \sin(\gamma) y + \cos(\gamma) z, \end{cases} \quad (1)$$

$$\text{其中 } \cos(\alpha) = \frac{V_x}{\sqrt{V_x^2 + V_y^2 + V_z^2}}, \cos(\beta) = \frac{V_y}{\sqrt{V_x^2 + V_y^2 + V_z^2}}, \cos(\gamma) = \frac{V_z}{\sqrt{V_x^2 + V_y^2 + V_z^2}}, \sin(\gamma) = \frac{\sqrt{V_x^2 + V_y^2}}{\sqrt{V_x^2 + V_y^2 + V_z^2}}。$$

由坐标变换(1),可知对应的逆坐标变换

$$\begin{cases} x = \frac{\cos(\beta) \sin(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)} (x' - p_x) - \frac{\cos(\alpha) \sin(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)} (y' - p_y), \\ y = \frac{\cos(\alpha) \cos(\gamma) \sin(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)} (x' - p_x) + \frac{\cos(\beta) \cos(\gamma) \sin(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)} (y' - p_y) - \sin(\gamma) (z' - p_z), \\ z = \frac{\cos(\alpha) \sin^2(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)} (x' - p_x) + \frac{\cos(\beta) \sin^2(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)} (y' - p_y) + \cos(\gamma) (z' - p_z) \end{cases} \quad (2)$$

由坐标变换(1)和逆坐标变换(2),给出计算空间两圆 C_0 和 C_2 之间最近距离的算法。

算法 1:

Step 1: 正交投射第二圆 C_2 上任意点 (v_x, v_y, v_z) 或第二圆 C_2 圆心 (p_x, p_y, p_z) 至第一圆 C_0 , 得到对应的正交投射点 $\left(\frac{v_x \cdot R}{\sqrt{v_x^2 + v_y^2}}, \frac{v_y \cdot R}{\sqrt{v_x^2 + v_y^2}}, 0 \right)$

Step 2: 利用逆坐标变换公式(2), 对第一圆 C_0 上的正交投射点 $(x', y', z') = \left(\frac{v_x \cdot R}{\sqrt{v_x^2 + v_y^2}}, \frac{v_y \cdot R}{\sqrt{v_x^2 + v_y^2}}, 0 \right)$ 进行逆坐标变换(2)得到点 (x_0, y_0, z_0) , 其中

$$\begin{cases} x_0 = \frac{\cos(\beta)\sin(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)}(x' - p_x) - \frac{\cos(\alpha)\sin(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)}(y' - p_y), \\ y_0 = \frac{\cos(\alpha)\cos(\gamma)\sin(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)}(x' - p_x) + \frac{\cos(\beta)\cos(\gamma)\sin(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)}(y' - p_y) - \sin(\gamma)(z' - p_z), \\ z_0 = \frac{\cos(\alpha)\sin^2(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)}(x' - p_x) + \frac{\cos(\beta)\sin^2(\gamma)}{\cos^2(\alpha) + \cos^2(\beta)}(y' - p_y) + \cos(\gamma)(z' - p_z) \end{cases}$$

Step 3: 计算点 (x_0, y_0, z_0) 正交投射至第二原始位置的圆 C_1 的正交投射点 $\left(\frac{x_0 R_1}{\sqrt{x_0^2 + y_0^2}}, \frac{y_0 R_1}{\sqrt{x_0^2 + y_0^2}}, 0 \right)$

Step 4: 正交投射点 $\left(\frac{x_0 R_1}{\sqrt{x_0^2 + y_0^2}}, \frac{y_0 R_1}{\sqrt{x_0^2 + y_0^2}}, 0 \right)$ 经过坐标变换(1)得到并记作点 (xR_1, yR_1, zR_1)

Step 5: 令 $(xR_1, yR_1, zR_1) = (v_x, v_y, v_z)$, 反复重复上述四步, 直至第一圆 C_0 上的点 $\left(\frac{v_x \cdot R}{\sqrt{v_x^2 + v_y^2}}, \frac{v_y \cdot R}{\sqrt{v_x^2 + v_y^2}}, 0 \right)$ 与第二圆 C_2 上的点 (xR_1, yR_1, zR_1) 之间距离最短, 即圆 C_0 和 C_2 之间最短距离为

$$d = \sqrt{\left(\frac{v_x R}{\sqrt{v_x^2 + v_y^2}} - xR_1 \right)^2 + \left(\frac{v_y R}{\sqrt{v_x^2 + v_y^2}} - yR_1 \right)^2 + zR_1^2}.$$

当两圆中心轴有公共点时, 这时两圆的最近距离有两对对应点, 投射第二圆 C_2 的中心轴 $l_2: \begin{cases} x = p_x + V_x \cdot t \\ y = p_y + V_y \cdot t \\ z = p_z + V_z \cdot t \end{cases}$

至第一圆 C_0 得到直线 $l_1: \begin{cases} x = p_x \cdot t \\ y = p_y \cdot t \\ z = 0 \end{cases}$. 点 $(v'_x, v'_y, 0) = \left(\frac{v_x \cdot R}{\sqrt{v_x^2 + v_y^2}}, \frac{v_y \cdot R}{\sqrt{v_x^2 + v_y^2}}, 0 \right)$ 关于直线 l_1 的对称点为:

$$\begin{aligned} & \left(\frac{p_x^2 \cdot v'_x + 2p_x \cdot p_y \cdot v'_y - p_y^2 \cdot v'_x}{p_x^2 + p_y^2}, -\frac{p_x^2 \cdot v'_y - 2p_x \cdot p_y \cdot v'_x - p_y^2 \cdot v'_y}{p_x^2 + p_y^2}, 0 \right) \\ \text{令} & \left(\frac{p_x^2 \cdot v'_x + 2p_x \cdot p_y \cdot v'_y - p_y^2 \cdot v'_x}{p_x^2 + p_y^2}, -\frac{p_x^2 \cdot v'_y - 2p_x \cdot p_y \cdot v'_x - p_y^2 \cdot v'_y}{p_x^2 + p_y^2}, 0 \right) = (v_x, v_y, v_z) \text{ 作为算法 1 的初始迭代点,} \end{aligned}$$

重复上述五步直至求出两圆的另外一对最近距离的对应点。空间两圆中心轴一般位置和垂直的示意图见图 1 和图 2。

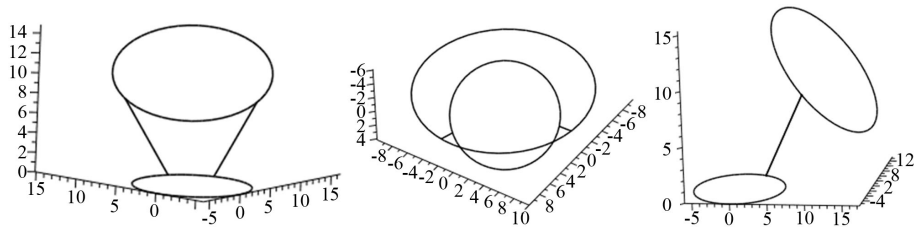


Figure 1. Schematic figure for general position of central axes of two spatial circles
图 1. 空间两圆中心轴一般位置的示意图

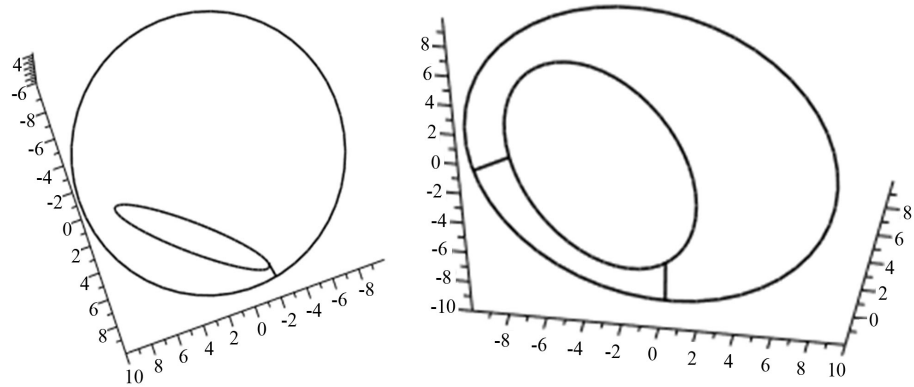


Figure 2. Schematic figure for perpendicular position of central axes of two spatial circles
图 2. 空间两圆中心轴垂直的示意图

注释 1: 当空间两圆两中心轴互相垂直且有公共点, 圆心点在 z-轴上的特殊位置时, 给出了具体的

解析分析表达式。这时第一圆和第二原始位置的圆分别为 $C_0: \begin{cases} x = R \cos(\theta) \\ y = R \sin(\theta) \\ z = 0 \end{cases}$ 和 $C_1: \begin{cases} x = R_1 \cos(\theta) \\ y = R_1 \sin(\theta) \\ z = 0 \end{cases}$, 经过坐

标变换(1)后变成第二圆 $C_2: \begin{cases} x = 0 \\ y^2 + (z - p_z)^2 = R_1^2 \end{cases}$, 其中, 中心轴的向量和圆心点分别为 $\langle V_x, V_y, V_z \rangle = \langle V_x, V_y, 0 \rangle$ (不失一般性假定 $V_y \neq 0$) 和 $(p_x, p_y, p_z) = (0, 0, p_z)$ 。不难知道这时第一圆 C_0 的上点 $(0, R, 0)$ 是第二圆 C_2 上

$\left(0, \frac{R \cdot R_1}{\sqrt{R^2 + p_z^2}}, \left(p_z - \frac{p_z R_1}{\sqrt{R^2 + p_z^2}}\right)\right)$ 的正交投射点, 同理第一圆 C_0 的上点 $(0, -R, 0)$ 是第二圆 C_2 上

$\left(0, -\frac{R \cdot R_1}{\sqrt{R^2 + p_z^2}}, \left(p_z - \frac{p_z R_1}{\sqrt{R^2 + p_z^2}}\right)\right)$ 的正交投射点, 所以两圆的最近距离为 $|\sqrt{R^2 + p_z^2} - R_1|$ 。另外, 第二圆 C_2

的点 $(0, 0, p_z - R_1)$ 至第一圆 $C_0: \begin{cases} x = R \cos(\theta) \\ y = R \sin(\theta) \\ z = 0 \end{cases}$ 的正交投射点集就是第一圆 C_0 本身, 则最近距离为

$\sqrt{R^2 + (p_z - R_1)^2}$ 。由于 $|\sqrt{R^2 + p_z^2} - R_1| < \sqrt{R^2 + (p_z - R_1)^2}$, 所以这时两圆的最近距离为 $|\sqrt{R^2 + p_z^2} - R_1|$ 。

更特别的, 当第二圆 C_2 的圆心点与第一圆 C_0 圆心点重合时, 这时两圆的最近距离变为 $|R - R_1|$ 。

注释 2: 文献[10]给出了 Kurush-Kuhn-Tucker 优化条件, 并给出了在优化条件下非线性 Kurush-Kuhn-Tucker 系统的封闭式解。对于两圆盘在同一中心轴上和在同一水平面上, 没有给出这两种特殊位置的解。

文献[11]首先给出了点至空间椭圆最近距离的一元四次方程准确解,然后利用迭代方法在第一椭圆上任选一点求出至第二空间椭圆最近距离及对应点,再由对应点求出至第一椭圆最近距离及对应点,反复迭代直至求出两椭圆最近距离及对应点。当空间两椭圆最近距离两对对应点时没有给出对应分析和讨论。另外对空间两椭圆中心轴是同一轴时最近距离有无数对应点没有给出分析;当两椭圆是在同一平面时,也没有给出对应分析和讨论。我们不仅给出了空间两圆在任意位置的迭代方法最近距离及对应点,还分析和讨论了空间两圆最近距离有两对对应点位置分析结论。此外,我们还对两圆在同一平面位置给出了完整的分析,也就是说,我们给出了空间两圆在空间任意位置的所有情况讨论。

2.2. 两中心轴平行的几何迭代算法

两中心轴平行,此时两圆的表达式可分别表示为第一圆 C_0 :
$$\begin{cases} x = R \cos(\theta) \\ y = R \sin(\theta) \\ z = 0 \end{cases}$$
 和第二圆 C_2 :
$$\begin{cases} x = p_x + R_1 \cos(u) \\ y = p_y + R_1 \sin(u) \\ z = p_z \end{cases}$$

其中,中心轴为 $\langle V_x, V_y, V_z \rangle = \langle 0, 0, V_z \rangle$, 圆心点为 (p_x, p_y, p_z) 。若用**算法 1** 计算两圆的最近距离,发现坐标变换(1)和逆坐标变换(2)已失效,为此提出相应的两圆最近距离的**算法 2**:

算法 2:

Step 1: 正交投射空间任意点 (v_x, v_y, v_z) 至第一圆 C_0 得到对应的正交投射点

$$(x_0, y_0, z_0) = \left(\frac{v_x \cdot R}{\sqrt{v_x^2 + v_y^2}}, \frac{v_y \cdot R}{\sqrt{v_x^2 + v_y^2}}, 0 \right)$$

Step 2: 计算点 (x_0, y_0, z_0) 正交投射至第二圆 C_2 的正交投射点 (x_1, y_1, z_1) , 其中

$$\begin{cases} x_1 = x_0 + (p_x - x_0) \cdot t_0 \\ y_1 = y_0 + (p_y - y_0) \cdot t_0, \text{ 此时 } t_0 = 1 - \frac{R_1}{\sqrt{(p_x - x_0)^2 + (p_y - y_0)^2}} \\ z_1 = p_z \end{cases}$$

Step 3: 令 $(x_1, y_1, z_1) = (v_x, v_y, v_z)$, 反复重复上述两步,直至第一圆 C_0 上的点 $\left(\frac{v_x \cdot R}{\sqrt{v_x^2 + v_y^2}}, \frac{v_y \cdot R}{\sqrt{v_x^2 + v_y^2}}, 0 \right)$

与第二圆 C_2 上的点 (x_1, y_1, z_1) 之间距离最短,即圆 C_0 和 C_2 之间最短距离为

$$d = \sqrt{\left(\frac{v_x R}{\sqrt{v_x^2 + v_y^2}} - x_1 \right)^2 + \left(\frac{v_y R}{\sqrt{v_x^2 + v_y^2}} - y_1 \right)^2 + z_1^2}$$

当 $|R_1 - R| < \sqrt{p_x^2 + p_y^2} < R_1 + R$ 时,两圆的最近距离有两对对应点,投射第二圆 C_2 的中心点

(p_x, p_y, p_z) 至第一圆 C_0 平面得到点 $(p_x, p_y, 0)$, 连接两点 $(p_x, p_y, 0)$ 和点 $(0, 0, 0)$ 得到直线 l_1 :
$$\begin{cases} x = p_x \cdot t \\ y = p_y \cdot t \\ z = 0 \end{cases}$$

点 $(x_0, y_0, 0) = \left(\frac{v_x \cdot R}{\sqrt{v_x^2 + v_y^2}}, \frac{v_y \cdot R}{\sqrt{v_x^2 + v_y^2}}, 0 \right)$ 关于直线 l_1 的对称点为

$$\left(\frac{p_x^2 \cdot x_0 + 2p_x \cdot p_y \cdot y_0 - p_y^2 \cdot x_0}{p_x^2 + p_y^2}, -\frac{p_x^2 \cdot y_0 - 2p_x \cdot p_y \cdot x_0 - p_y^2 \cdot y_0}{p_x^2 + p_y^2}, 0 \right)$$

令 $\left(\frac{p_x^2 \cdot x_0 + 2p_x \cdot p_y \cdot y_0 - p_y^2 \cdot x_0}{p_x^2 + p_y^2}, -\frac{p_x^2 \cdot y_0 - 2p_x \cdot p_y \cdot x_0 - p_y^2 \cdot y_0}{p_x^2 + p_y^2}, 0 \right) = (v_x, v_y, v_z)$ 作为**算法 2**的初始迭代点,

重复上述三步直至求出两圆的另外一对最近距离的对应点。

2.3. 中心轴平行的空间两圆最近距离解析表达式

第一圆 $C_0: \begin{cases} x = R \cos(\theta) \\ y = R \sin(\theta) \\ z = 0 \end{cases}$ 和第二圆 $C_2: \begin{cases} x = p_x + R_1 \cos(u) \\ y = p_y + R_1 \sin(u) \\ z = p_z \end{cases}$, 其中第二圆 C_2 中心轴为

$\langle V_x, V_y, V_z \rangle = \langle 0, 0, V_z \rangle$, 圆心点为 (p_x, p_y, p_z) 。空间两圆中心轴平行的示意图见图 3。

分下列几种情况具体分析

1) 若 $p_x^2 + p_y^2 > (R + R_1)^2$, 则空间两圆最近距离的对应点分别为 $\left(\frac{p_x R}{\sqrt{p_x^2 + p_y^2}}, \frac{p_y R}{\sqrt{p_x^2 + p_y^2}}, 0 \right)$ 和

$\left(p_x - \frac{p_x R_1}{\sqrt{p_x^2 + p_y^2}}, p_y - \frac{p_y R_1}{\sqrt{p_x^2 + p_y^2}}, p_z \right)$, 其最近距离为 $\sqrt{\left(\sqrt{p_x^2 + p_y^2} - (R + R_1) \right)^2 + p_z^2}$ 。

2) 若 $(R - R_1)^2 \leq p_x^2 + p_y^2 \leq (R + R_1)^2$, 则对对应点为 $(x_1, y_1, 0)$, (x_1, y_1, p_z) 和 $(x_2, y_2, 0)$, (x_2, y_2, p_z) , 其距离为 $|p_z|$, 其中

$$x_1 = \frac{p_x (R^2 - R_1^2 + p_x^2 + p_y^2) + \sqrt{p_y^2 \left((R + R_1)^2 - (p_x^2 + p_y^2) \right) \left((p_x^2 + p_y^2) - (R - R_1)^2 \right)}}{2(p_x^2 + p_y^2)}$$

$$y_1 = \frac{p_y (R^2 - R_1^2 + p_x^2 + p_y^2) - \sqrt{p_x^2 \left((R + R_1)^2 - (p_x^2 + p_y^2) \right) \left((p_x^2 + p_y^2) - (R - R_1)^2 \right)}}{2(p_x^2 + p_y^2)}$$

$$x_2 = \frac{p_x (R^2 - R_1^2 + p_x^2 + p_y^2) - \sqrt{p_y^2 \left((R + R_1)^2 - (p_x^2 + p_y^2) \right) \left((p_x^2 + p_y^2) - (R - R_1)^2 \right)}}{2(p_x^2 + p_y^2)}$$

$$y_2 = \frac{p_y (R^2 - R_1^2 + p_x^2 + p_y^2) + \sqrt{p_x^2 \left((R + R_1)^2 - (p_x^2 + p_y^2) \right) \left((p_x^2 + p_y^2) - (R - R_1)^2 \right)}}{2(p_x^2 + p_y^2)}$$

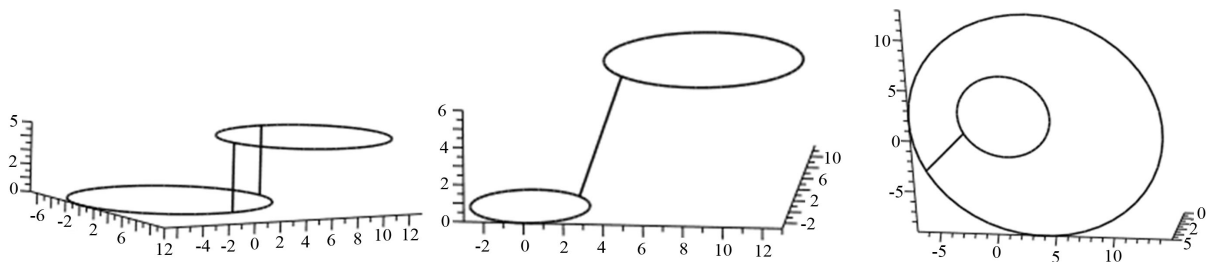


Figure 3. Actual examples for parallel position of central axes of two spatial circles

图 3. 空间两圆中心轴平行的实例

3) 若 $0 < p_x^2 + p_y^2 < (R - R_1)^2$,

当 $R < R_1$, 两圆最近距离的对应点为

$$\left(\frac{-p_x R}{\sqrt{p_x^2 + p_y^2}}, \frac{-p_y R}{\sqrt{p_x^2 + p_y^2}}, 0 \right) \text{ 和 } \left(p_x - \frac{p_x R_1}{\sqrt{p_x^2 + p_y^2}}, p_y - \frac{p_y R_1}{\sqrt{p_x^2 + p_y^2}}, p_z \right)$$

$$\text{其距离为 } \sqrt{\left(\sqrt{p_x^2 + p_y^2} + R \right)^2 \left(1 - \frac{R_1}{\sqrt{p_x^2 + p_y^2}} \right)^2} + p_z^2;$$

当 $R > R_1$, 两圆最近距离的对应点为

$$\left(\frac{p_x R}{\sqrt{p_x^2 + p_y^2}}, \frac{p_y R}{\sqrt{p_x^2 + p_y^2}} \right) \text{ 和 } \left(p_x + \frac{p_x R_1}{\sqrt{p_x^2 + p_y^2}}, p_y + \frac{p_y R_1}{\sqrt{p_x^2 + p_y^2}} \right)$$

$$\text{其距离为 } \sqrt{\left(\sqrt{p_x^2 + p_y^2} + R_1 \right)^2 \left(1 - \frac{R}{\sqrt{p_x^2 + p_y^2}} \right)^2} + p_z^2$$

2.4. 中心轴重合的空间两圆最近距离解析表达式

$$\text{这时第一圆 } C_0: \begin{cases} x = R \cos(\theta) \\ y = R \sin(\theta) \\ z = 0 \end{cases}, \text{ 第二圆 } C_2: \begin{cases} x = R_1 \cos(\theta) \\ y = R_1 \sin(\theta) \\ z = p_z \end{cases}. \text{ 两圆最近距离对应点集为 } \begin{cases} x = R \cos(\theta) \\ y = R \sin(\theta) \\ z = 0 \end{cases} \text{ 和 } \begin{cases} x = R_1 \cos(\theta) \\ y = R_1 \sin(\theta) \\ z = p_z \end{cases}$$

$$\begin{cases} x = R_1 \cos(\theta) \\ y = R_1 \sin(\theta) \\ z = p_z \end{cases}, \text{ 两个式子的参数 } \theta \text{ 必须同时取同一值, 所以两圆的最近距离为 } \sqrt{(R_1 - R)^2 + p_z^2}.$$

例 1: 空间两圆 C_0 和 C_2 的半径分别为 $R = 6, R_1 = 8$, 第二圆 C_2 的中心轴的向量和圆心分别为 $\langle 4, 7, 6 \rangle$ 和 $(10, 8, 9)$, 由算法 1 可知, 两圆最近距离对应点分别为 $(2.2566117602763080, 5.5594697016336607, 0)$ 和 $(5.5251451389773982, 13.600640848138522, 5.4491555845201255)$. 最近距离为 10.24875800. 此例是针对算法 1 中的空间两圆中心轴没有公共点时两圆最近距离只有唯一一对对应点的解释。

例 2: 空间两圆 C_0 和 C_2 的半径分别为 $R = 10, R_1 = 6$, 第二圆 C_2 的中心轴的向量和圆心分别为 $\langle 4, 5, 0 \rangle$ 和 $(-4, -5, 4)$, 由算法 1 可知, 两圆最近距离第一对对应点分别为 $(-9.7379538193036241, -2.2742593108768315, 0)$ 和 $(-7.9338517047101518, -1.8529186362318786, 0.7410600411832663)$, 另外一对对应点分别为 $(-0.081189952715625401, -9.9996704041472304, 0)$ 和 $(-0.0661482952898482, -8.1470813637681214, 0.7410600411832664)$, 最近距离为 1.995365226. 此例是针对算法 1 中的空间两圆中心轴有公共点时两圆最近距离有两对对应点的说明。

3. 总结与展望

本文完全地给出了空间两圆任意位置的最近距离的迭代算法和解析表达式, 当空间两圆中心轴不平行, 利用几何迭代法提出了空间两圆最近距离的算法; 当空间两圆中心轴平行时, 提出了简化版的空间两圆最近距离的迭代算法; 当中心轴平行或重合时, 我们提出了空间两圆最近距离的解析表达式。文章提供的迭代算法对初始迭代点不敏感且是有效的。下一步将在此基础上尝试给出空间圆环面/球面、圆环面/圆环面、椭圆环面/圆环面、椭圆环面/椭圆环面等与空间两圆有关联的若干

情况分析。

基金项目

国家自然科学基金(61263034); 国家统计局基金资助项目(2014LY011); 贵州省科学技术基金([2014]2092), ([2014]2093)); 贵州省“模式识别与智能系统”重点实验室建设项目(黔科合计[2009]4002); 贵州省“信息处理与模式识别”研究生教育创新基地。

参考文献 (References)

- [1] Ho, S., Sarma, S. and Adachi, Y. (2001) Real-Time Interference Analysis between a Tool and an Environment. *Computer-Aided Design*, **33**, 935-947. [http://dx.doi.org/10.1016/S0010-4485\(00\)00117-2](http://dx.doi.org/10.1016/S0010-4485(00)00117-2)
- [2] Johnson, D. and Cohen, E. (1998) A framework for Efficient Minimum Distance Computations. *Proceedings of the IEEE Conference on Robotics and Automation*, **36**, 78-84. <http://dx.doi.org/10.1109/robot.1998.681403>
- [3] Cameron, S. and Enhancing, G.J.K. (1997) Computing Minimum and Penetration Distances between Convex Polyhedra. *Proceedings of the IEEE Conference on Robotics and Automation*, 3112-3117. <http://dx.doi.org/10.1109/ROBOT.1997.606761>
- [4] Gilbert, E.G., Johnson, D.W. and Keerthi, S.S. (1988) A Fast Procedure for Computing the Distance between Complex Objects in Three-Dimensional Space. *IEEE Journal of Robotics and Automation*, **4**, 193-203. <http://dx.doi.org/10.1109/56.2083>
- [5] Lin, M.C. (1993) Efficient Collision Detection for Animation and Robotics. Ph. D Thesis, University of California, Berkeley.
- [6] Snyder, J. (1995) An Interactive Tool for Placing Curved Surfaces without Interpenetration. *Computer Graphics. SIGGRAPH*, Los Angeles, 6-11 August 1995, 209-218.
- [7] 陈小雕, 雍俊海, 郑国勤, 等. 圆环面/球面求交算法[J]. 计算机辅助设计与图形学学报, 2005, 17(6): 1202-1206.
- [8] Neff, C.A. (1990) Finding the Distance between Two Circles in Three-Dimensional Space. *IBM Journal of Research and Development*, **34**, 770-775. <http://dx.doi.org/10.1147/rd.345.0770>
- [9] 刘晓明, 刘长远, 胡强, 雍俊海. 计算两圆环面之间的最近距离[J]. 计算机辅助设计与图形学学报, 2011, 23(2): 240-246.
- [10] Almohamad, H.A. and Selim, S.Z. (2003) An Algorithm for Computing the Distance between Two Circular Disks. *Applied Mathematical Modelling*, **27**, 115-124. [http://dx.doi.org/10.1016/S0307-904X\(02\)00080-X](http://dx.doi.org/10.1016/S0307-904X(02)00080-X)
- [11] Kim, I.S. (2006) An Algorithm for Finding the Distance between Two Ellipses. *Communications Korean Mathematical Society*, **21**, 559-567. <http://dx.doi.org/10.4134/CKMS.2006.21.3.559>
- [12] 王日昀, 宁涛, 席平, 等. 圆环与圆环求交算法中初始点的计算[J]. 工程图学学报, 2004, 25(1): 47-51.
- [13] Chen, X.D., Chen, L.Q., Wang, Y.G., Xu, G., Yong, J.H. and Paul, J.C. (2009) Computing the Minimum Distance between Two Bézier Curves. *Journal of Computational and Applied Mathematics*, **229**, 294-301. <http://dx.doi.org/10.1016/j.cam.2008.10.050>
- [14] Chen, X.D., Ma, W.Y., Xu, G. and Paul, J.C. (2010) Computing the Hausdorff Distance between Two B-Spline Curves. *Computer-Aided Design*, **42**, 1197-1206. <http://dx.doi.org/10.1016/j.cad.2010.06.009>
- [15] Chang, J.W., Choi, Y.K., Kim, M.S. and Wang, W.P. (2011) Computation of the Minimum Distance between Two Bézier Curves/Surfaces. *Computers & Graphics*, **35**, 677-684. <http://dx.doi.org/10.1016/j.cag.2011.03.025>
- [16] Bai, Y.B., Yong, J.H., Liu, C.Y., Liu, X.M. and Meng, Y. (2011) Polyline Approach for Approximating Hausdorff Distance between Planar Free-Form Curves. *Computer-Aided Design*, **43**, 687-698. <http://dx.doi.org/10.1016/j.cad.2011.02.008>
- [17] Ma, Y.P., Tu, C.H. and Wang, W.P. (2012) Distance Computation for Canal Surfaces Using Cone-Sphere Bounding Volumes. *Computer Aided Geometric Design*, **29**, 255-264. <http://dx.doi.org/10.1016/j.cagd.2011.10.007>
- [18] Sundar, B.R., Chunduru, A., Tiwari, R., Gupta, A. and Muthuganapathy, R. (2014) Footpoint Distance as a Measure of Distance Computation between Curves and Surfaces. *Computers & Graphics*, **38**, 300-309. <http://dx.doi.org/10.1016/j.cag.2013.10.032>
- [19] Blasco, A. and Pérez-Díaz, S. (2015) Characterizing the Finiteness of the Hausdorff Distance between Two Algebraic Curves. *Journal of Computational and Applied Mathematics*, **280**, 327-346. <http://dx.doi.org/10.1016/j.cam.2014.12.005>