视锥体裁剪几何算法研究

于海燕[1]， 张 帅[1]， 余沛文[1]， 何援军[2]

(1. 东华大学机械工程学院，上海 201620；2. 上海交通大学计算机系，上海 200240)

摘 要：从几何角度出发，以投影理论为指导，设计了一种降维投影的视锥体裁剪几何算法。基本思想是基于视锥体构建计算坐标系，在计算坐标系下，向两个投影平面做正投影。空间中被裁剪线段与视锥体的位置关系被简化为投影平面内线段与等腰梯形的关系。这种几何化的降维方法有利于解决空间几何奇异问题。构建了空间视锥体裁剪中线段与视锥体的各种位置关系的测试样本，特别是 78 种处于几何奇异状态的位置关系，用于综合评估算法的速度和稳定性。用 C 语言在 VC++平台上分别实现了投影降维的视锥体裁剪几何算法、经典的 Liang-Barsky 算法和与 6 个面分别求交的一般算法。在定性分析基础上，利用测试样本对 3 种算法做了计算速度与稳定性方面的测试对比。

关 键 词：视锥体裁剪；几何算法；投影理论；几何奇异

中图分类号：TP 391.72 **DOI**：10.11996/JG.j.2095-302X.2017010001

文献标识码：A **文 章 编 号**：2095-302X(2017)01-0001-04

View Frustum Culling Geometric Algorithm

YU Haiyan[1], ZHANG Shuai[1], YU Peiwen[1], HE Yuanjun[2]

(1. College of Mechanical Engineering, Donghua University, Shanghai 201620, China;
2. Department of Computer Science & Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: From the point of view of geometry, a view frustum culling geometric algorithm is designed based on projective theory in descriptive geometry. The basic idea is that a proper computational coordinate system is built, where the space position relation of the view frustum and a line segment is transformed into the plane position relation of a trapezoid and the line segment by simple orthographic projection. This geometric dimension reduction method is beneficial to solve space geometric singular issue. A test sample is designed which includes all kinds of typical position relations of the view frustum and the line segment to comprehensively evaluate algorithm's speed and stability. And especially, there are 78 kinds of geometric singular relations. At last, our view frustum culling geometric algorithm, classical Liang-Barsky algorithm and a basic algorithm to solvethe problem of intersection with 6 planes have been implemented on Visual C++ with C program. On the basement of qualitative analysis, these 3 algorithms have been tested on speed and stability.

Keywords: view frustum culling; geometric algorithm; projection theory; geometric singular

随着三维几何模型愈发复杂、逼真，虚拟环境的场景规模越来越大，如何有效减少绘制对象，降低三维模型的复杂度，是三维显示系统中快速稳定绘制复杂场景的关键所在。近年来，可见性的判断是一种有效方法，在大规模的复杂场景中，虽然需要绘制对象的数量大幅增加，但对于观察

收稿日期：2016-06-22；定稿日期：2016-09-05

第一作者：于海燕(1974–)，女，黑龙江牡丹江人，副教授，博士。主要研究方向为 CAD/CG。E-mail：yuhy@dhu.edu.cn

通信作者：张 帅(1993–)，男，河南驻马店人，硕士研究生。主要研究方向为 CAD/CG。E-mail：1543965553@qq.com

者来说未必增加了很多，一般而言可见实体的数目要远远小于实体数量总和。例如，观察者在房间内，可看到房间内的部分物体还可通过门窗看到外面部分的物体；观察者站在房间外，则只能看到房间的外表面，房间内部环境是不可见的。实际绘制量与场景规模之间存在着很大差异，这是可见性研究的基础和依据[1]。可见性剔除算法(visibility culling)[2-3]可分为 3 类：遮挡剔除(occlusion culling)、背面剔除(beck-face culling)和视域剔除(view frustum culling)[4]。视域剔除又被称之为视锥体裁剪，是计算机图形学中一个重要的基础算法，视锥体裁剪算法在计算机图形学中主要应用于虚拟现实环境和 CAD/CAM 等工程领域。

视锥体(frustum)是指场景中摄像机可见的一个锥体范围。由上、下、左、右、近、远共 6 个面组成。在视锥体内部的景物是可见的，反之则不可见。为提高性能，一般只对与视锥体有交集的对象进行绘制。视锥体裁剪算法要实现的功能就是视锥体和线段求交，将视锥体内的部分显示出来，外部的视为不可见。通过视锥体裁剪算法能判定出大量不可见的部分，大大减少了送入绘制管道的数量，从而极大地提高了图形的显示性能[5]。

1 相关研究

常见的视锥体裁剪算法有包围盒裁剪算法、参数形式的裁剪算法、基于编码的裁剪算法、混合裁剪算法、一般裁剪算法和其他算法[6]。其中 Liang-Barsky 算法[7]作为一种比较成熟的算法，常被很多新算法作为比较对象。Liang-Barsky 裁剪算法是将被剪裁线段用含参数的表达式表示，并通过代数运算推导得到参数的值，将三维裁剪问题转化为 3 次一维裁剪问题，再将一维裁剪问题转化为解一组关于参数的不等式，即可得到线段在视锥体内可见部分的参数值范围。此算法逻辑简单、通用性强、计算量少、程序简洁、且计算量相对稳定，是一种高效的参数形式的裁剪算法。此外，视锥体的表面是由 6 个面围成的，求线段在视锥体内的可见部分即求线段与 6 个面的交点，再找到 2 个有效交点即为所求，这就是一般裁剪算法的原理；这种算法思路清晰，核心是求视锥体 6 个平面的方程，然后再求线段与每个面的交点；整个过程在世界坐标系下进行，不需任何坐标变换、分区、编码等其他操作，且求解速度较快，求解全程采用代数计算，是一种典型的代数算法。

三维线段裁剪是3D显示系统都要做的底层技术，对算法的效率要求较高[8]。由于算法的速度是衡量一种算法优劣的核心因素，因此对三维线段裁剪算法速度的研究一直也是国内外研究的热点。但这些算法不足有两点：①这些算法均以代数方法为主，几何的方法只是作为辅助，从算法的描述上来讲，公式繁多、交流不便、几何意义不明显；无论是上文提到的参数化的 Liang-Barsky 算法，还是代数运算的一般算法都存在这些问题，使得程序的可读性不强。②这些算法大多只关注算法的速度而未涉及算法的稳定性，特别是几何奇异位置，当系统稳定性非常差时可导致整个系统的瘫痪。

针对以上两个问题，依据画法几何投影理论，从几何的角度[9]来解决上述问题，提出了基于几何以及投影降维的视锥体裁剪算法(视锥体裁剪几何算法)，将复杂的三维问题转化为简单的二维问题。同多数以代数为主的算法相比，这种具有几何意义的算法降低了复杂度，通过降维使得几何奇异关系变得相对清晰、简单。其次，可改造性比较强。算法每一步都有其相对应的几何意义，所以当调试程序出错时，能够相对容易地发现算法的错误与不足，进行修改或完善。上述几何算法的优点是很多代数为主的算法所不具备的。通过对几何奇异情况分类方法的讨论，设计了一套相对客观、详细的算法的测试样本，选择高效且具代表性的 Liang-Barsky 算法和 6 面求交一般算法，以及本文的几何算法，在 VC++平台上分别实现并进行全面、对等测试。

2 计算坐标系的建立

通常情况下，世界坐标系下的视锥体以及直线段在空间所处的位置是随机的，为了使计算方便，需要进行坐标系转换，建立一个全新的计算坐标系，然后选择合适的投影面做直线段与视锥体的正投影。计算坐标系的选取应尽量使投影后的平面几何关系简单，便于计算。对于一个视锥体 A_1、A_2、A_3、A_4、P_b 和 P_t 是已知的(图 1)，由这些点能确定唯一的视锥体。在三维空间，从任意一点出发的 3 个不共面的向量能构成一个三维的坐标系，3 个向量可定义为：

(1) 以后端面的中心 P_b 作为计算坐标系原点，向量 $\overrightarrow{P_bP_t}$ 作为 z 轴正向，设 $\boldsymbol{n}_1=\dfrac{\overrightarrow{P_bP_t}}{\left|\overrightarrow{P_bP_t}\right|}=(a_1,b_1,c_1)$ 作为 z 轴正向单位向量。

(2) 以向量 $\overrightarrow{P_bA_1}$ 作为 x 轴正向，设 $\boldsymbol{n}_2=\dfrac{\overrightarrow{P_bA_1}}{\left|\overrightarrow{P_bA_1}\right|}=(a_2,b_2,c_2)$ 作为 x 轴正向单位向量。

(3) 最后由 $\boldsymbol{n}_3=\boldsymbol{n}_1\times\boldsymbol{n}_2=(a_3, b_3, c_3)$，得到 y 轴正向单位向量。

则由 P_b 点以及互相垂直的 3 个单位向量 $\boldsymbol{n}_1$，$\boldsymbol{n}_2$，$\boldsymbol{n}_3$，构成了一个新的坐标系，如图 1 所示。

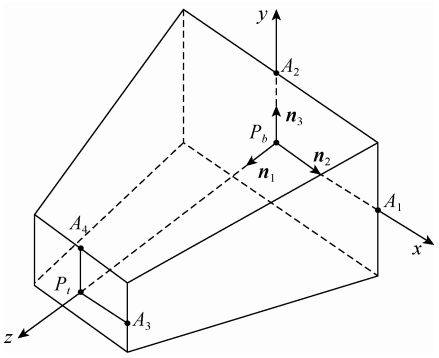

图 1　计算坐标系下的视锥体

由世界坐标系下 $P_b(P_b.x,\ P_b.y,\ P_b.z)$变为计算坐标系原点(0, 0, 0)，可以得到一个平移矩阵的元素 d_1，d_2，d_3 以及其逆矩阵元素 D_1，D_2，D_3。结合 $\boldsymbol{n}_1$，$\boldsymbol{n}_2$，$\boldsymbol{n}_3$ 可以得到世界坐标系到计算坐标系的变换矩阵 $\boldsymbol{T}_{xyz_x^*y^*z^*}$及其逆矩阵 $\boldsymbol{T}^{-1}_{xyz_x^*y^*z^*}$，由于 $\boldsymbol{n}_1$ 作为 z 轴，所以有

$$\boldsymbol{T}_{xyz_x^*y^*z^*}=\begin{pmatrix} a_2 & a_3 & a_1 & 0\\ b_2 & b_3 & b_1 & 0\\ c_2 & c_3 & c_1 & 0\\ d_2 & d_3 & d_1 & 1\end{pmatrix}\text{和}$$

$$\boldsymbol{T}_{x^*y^*z^*_xyz}=\boldsymbol{T}^{-1}_{xyz_x^*y^*z^*}=\begin{pmatrix} a_2 & b_2 & c_2 & 0\\ a_3 & b_3 & c_3 & 0\\ a_1 & b_1 & c_1 & 0\\ D_2 & D_3 & D_1 & 1\end{pmatrix}$$

3　裁剪原理

基于几何及投影降维的视锥体裁剪算法原理：首先建立计算坐标系，并选取两个投影面(V 面和 W 面)，利用画法几何理论建立 V/W 投影体系，在这个计算坐标系下，视锥体在 V 面上的投影(T_V)与在 W 面上的投影(T_W)均为等腰梯形(图 2)。在投影 V 面和 W 面内分别得到线段与等腰梯形的平面几何关系；然后建立平面投影与空间实体的对应关系，将空间直线的投影在 V 面与 W 面上分 2 次裁剪，其交集即为三维裁剪结果。最后将裁剪结果逆变换回原始坐标系。具体过程如下：

(1) 确定 V 面和 W 面。选取视锥体的两个对称平面为对应于三投影面体系的 V 面和 W 面，两平面的交线为 z 轴。如图 1 所示，选取由 P_b、P_t、A_1 3 点构成的平面作为正投影平面(V 面)，由 P_b、P_t、A_2 3 点构成的平面作为侧投影平面(W 面)，只需这两个投影平面就能完全确定直线段和视锥体的空间位置关系。视锥体在 V 面和 W 面上的正投影均为等腰梯形，记作 T_V 与 T_W，直线段在两平面的投影均为线段。如图 2 所示。

(2) 确定梯形和线段参数。要将空间中视锥体裁剪 3D 线段被转化为平面内的等腰梯形裁剪 2D 线段，因此需先确定等腰梯形和被裁剪 2D 线段的参数，其是指计算坐标系下的参数。由于知道视锥体 8 个顶点和 8 条边，所以就能得到这两个面中两个梯形每个点的坐标和每条边方程，就能构建梯形裁剪平台。同样对于被裁剪 3D 线段，两端点已知，经过变换矩阵 $\boldsymbol{T}_{xyz_x^*y^*z^*}$做正向坐标变换得到计算坐标系下的 3D 线段端点，再向 V 面和 W 面投影，就得到了被裁剪的 2D 线段。到此，裁剪前的预处理工作结束。

(3) 裁剪。三维裁剪问题被转化为两个二维平面内的裁剪问题，即 2D 直线段与梯形求交。如果空间线段 P_1P_2 所在的直线在 V 面上的投影与梯形 T_V 的边界存在交点 I_V，那么 I_V 所对应的空间点在 W 面上的投影必须在梯形 T_V 内或边上才有效。只有 I_V 有效，才能在 T_V 上求出 P_1P_2 相应的裁剪点(I_V 本身或靠近 I_V 的那个交点)；否则，在 T_V 上的交点无效，同样在 T_W 上执行类似的操作。分别在两个平面内进行线段与梯形的求交，取两个平面内相交部分的交集。如图 2 所示，空间直线段 P_1P_2 在 V 面上的投影为 $P_{1V}P_{2V}$，在 W 面上的投影为 $P_{1W}P_{2W}$。P_1P_2 被 T_V 裁剪后得到的可见部分为 $Q_{1V}Q_{2V}$，被 T_W 裁剪后得到的可见部分为 $Q_{1W}Q_{2V}$。取二者的交集，最后得到 P_1P_2 被视锥体裁剪后的可见部分为 $Q_{1V}Q_{2W}$[10]。

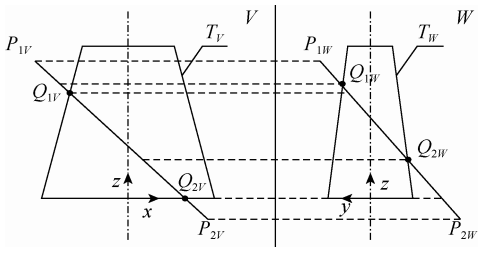

图 2　几何算法中视锥体二次裁剪原理图

(4) 逆变换裁剪结果。由于上述裁剪过程是在计算坐标系下完成，而视锥体与线段初始状态是在世界坐标系下确定的，因此得到的裁剪结果并非实际结果；需要将裁剪结果经过逆变换矩阵 $\boldsymbol{T}^{-1}{}_{xyz_x^*y^*z^*}$进行坐标变换，变换回原始世界坐标系，即为最终所求。

4 样本的设计与测试

为了评估算法性能，需要设计一套比较完整的测试样本。在世界坐标系下，固定视锥体的位置，之后讨论线段与视锥体的位置关系分类。首先，根据视锥体与线段交点的个数来分类，交点个数分为无交点，1 个交点，2 个交点，无数多个交点。其次，直线段穿越视锥体的方式可分为直线段左右穿越和上下穿越。如果指直线段和 4 个侧面相交或者直线段在正平面中斜率绝对值约小于 1 称之为左右穿越；如果直线段主要和前后端面相交或者直线段在正平面的斜率绝对值约大于 1 称之为上下穿越。最后，根据相对运动的观点，让线段在其所在的直线上渐渐移动，相对于视锥体可以得到不同的相对位置关系，分别为相离、碰撞、相交和内含。将空间处于分离与接触临界状态的位置关系定义为碰撞，这种临界的奇异情况是需要着重考虑的。按照上述办法分类，可以得到 78 组被裁剪线段样本，其包含直线段与视锥体顶点、边界线和边界面处于奇异状态的各种情况，使得裁剪算法的检测不会处于一种盲目、随机的状态。

为了实际客观评估视锥体裁剪算法，选择 Liang-Barsky 算法、一般算法以及几何化投影降维算法，利用 C 语言在 VC++6.0 平台上对上述 78 组样本进行分析测试。在算法的直观性及可读性方面，本文几何算法相对于另外两个算法具有明显优势，因为算法与几何意义相对应，当调试算法程序出错时，可快速发现错误并进行修改。在稳定性方面，对各种空间关系，特别是几何奇异状态，几何算法与 Liang-Barsky 算法均能正确处理，二者稳定性相当，而一般算法会出错。考虑到计算精度也可能导致不稳定，几何算法和 Liang-Barsky 算法的出错率也远比一般算法要低。因此几何算法的稳定性与 Liang-Barsky 算法相当，但要比一般算法更加稳定。速度方面，最后在经过预处理之后的标准坐标系下，对上述 78 组样本重复 10 万次裁剪计算，其计算时间的参考比例为：

Liang-Barsky：线面求交：投影降维=4243：4228：4212

说明 3 种算法的计算效率在同一数量级上，几何算法的速度略快于其他 2 种算法。

5 结 束 语

本文从几何的角度研究了视锥体裁剪算法。该算法降低了复杂度，结合坐标变换与画法几何投影理论，将复杂的空间视锥体裁剪问题转化为二次简单的平面等腰梯形裁剪，直观易懂，可改造性强。在统一的平台上，实现 3 种典型视锥体算法，针对算法的稳定性展开讨论，设计了一套测试样本，利用测试样本较客观、完整地测试出各算法的综合性能，给出了 3 种算法的稳定性及速度对比。在程序调试及分析试过程中发现，几何算法表现出可改造性强、鲁棒性强和逻辑简单、直观易懂等特点，使得这种几何化的算法在工程仿真等复杂系统以及稳定性要求较高的场合中有广阔的应用前景。

参 考 文 献

[1] 普建涛, 查红彬. 大规模复杂场景的可见性问题研究[J]. 计算机研究与发展, 2005, 42(2): 236-246.

[2] YAGEL R, RAY W. Visibility computation for efficient walkthrough of complex environments [J]. Presence, 1995, 5(1): 45-60.

[3] ZHANG H, MANOCHA D, HUDSON T, et al. Visibilityculling using hierarchical occlusion maps [J]. Computer Graphics, 1997, 31(3): 77-88.

[4] 封春升, 王莉莉, 杨跃东. 基于视域剔除和图像缓存技术的复杂场景快速绘制方法[J]. 系统仿真学报, 2006, 18(1): 94-98.

[5] 胡国瓒, 陈先桥. 三维图形系统中的物体裁剪问题[J]. 计算机工程与应用, 2007, 43(15): 92-94.

[6] KODITUWAKKU R, WIJEWEERA K, CHAMIKARA M. Anefficientline clipping algorithm for 3D space [J]. International Journal of Advanced Research in Computer Science and Software Engineering, 2012, 2(5): 96-101.

[7] LIANG Y D, BARSKY B A. A new concept and method for line clipping [J]. ACM Transactions on Graphics, 1984, 3(1): 1-22.

[8] 何援军. 几何计算[M]. 北京: 高等教育出版社, 2013: 237-239.

[9] 于海燕, 蔡鸿明, 何援军. 图学计算基础[J]. 图学学报, 2013, 34(6): 1-5.

[10] 余沛文. 视锥体裁剪几何算法与测试方法研究[D]. 上海: 东华大学, 2016.