

**计算机图形学中二维裁剪算法的研究与总结**

**The Research and Summary of two-dimensional cutting algorithm in Computer Graphics**

院 系：计算机科学与软件工程学院计算机科学技术系

专 业： 计算机科学与技术专业

姓 名： 丛佩金

学 号： 51164500019

指导 教师： 李海晟 副教授

2016年12月

**摘要**

随着计算机技术的发展，计算机图形学也日益成熟。在医学、娱乐、图形艺术、商业、教育培训、科学工程等众多领域，计算机图形学的应用非常普遍。计算机图形学主要研究在计算机中如何构造图形，将用数学模型描述的图形数据采用合适的算法转换为屏幕上图形的显示。计算机图形学学科研究的对象为二维图形学和三维图形学及其显示和变化情况，点、线、面为二维图形学范畴，几何体和场等数学构造方法则为三维图形学范畴。现在，计算机图形学的一些基本算法已经成了固化在硬件中的规范软件包，这个学科也日趋成熟和完善，但是依然有很多算法还需要不断的改进才能应用到实际中，而裁剪算法就是其中之一。本文主要对二维裁剪算法进行研究与总结，为以后的学习打下坚实的基础。

**【关键词】：**计算机图形学 二维裁剪算法

**Abstract**

With the development of computer technology, computer graphics is increasingly mature. In medicine, entertainment, graphic arts, business, education training, science and engineering, etc, computer graphics applications are common. Computer graphics mainly studies how to construct the graphics in the computer, and it will use mathematical model to describe the graphics data using appropriate algorithm into the screen graphic display. Computer graphics discipline research object for 2d graphics and 3d graphics and its display and change. Dot, line, face for 2d graphics category, such as geometry and field mathematical structural method is for 3d graphics category. Now, some basic computer graphics algorithm has already formed the curing of standard package in the hardware, the subject has become more and more mature and perfect. But there are a lot of algorithms need to be improved to be applied to practice, cutting algorithm is one of them. This paper focused on the study and summary of the 2d cutting algorithm to lay a solid foundation for further study.

**【Keywords】：**computer graphics 2d cutting algorithm

一、引言

随着计算机技术的发展，计算机图形学也日益成熟，与很多领域都有着密不可分的联系，在我们日常生活中，也成了随处可见的必须部分。本篇论文主要对计算机图形学中二维裁剪算法进行研究与总结，比较不同算法的优缺点，为以后的进一步学习打下坚实的基础。裁剪算法，是计算机图形学中很多重要问题的基础，即从数据集合中识别指定区域内或指定区域外图形部分的过程。裁减用途很广泛，最典型的就是确定场景中位于指定区域内的景物部分，按裁剪对象来分，裁减算法大致分为以下几种：点裁剪、直线段裁剪、区域多边形裁剪、曲线裁剪和文字裁剪。接下来，本篇论文会对这几种裁剪算法分别进行介绍。裁剪也有很多方面的应用，主要包括：使用实体造型创建对象、在三维视图中标示出可见面、对图形的一部分进行删除、复制或移动操作、防止图形边界混淆、从特定场景中抽取指定部分等。由此可见，研究二维裁剪算法是非常重要的。

论文接下来按照如下部分进行介绍：第二大部分主要为介绍各种二维裁剪算法；第三大部分主要对二维裁剪算法进行比较与总结。

二、二维裁剪算法介绍

在使用计算机处理图形信息时，计算机内部存储的图形往往比较大，而屏幕显示的只是图的一部分。因此需要确定图形中哪些部分落在显示区之内，哪些落在显示区之外，以便只显示落在显示区内的那部分图形。这个选择过程称为裁剪。最简单的裁剪方法是把各种图形扫描转换为点之后，再判断各点是否在窗内。但那样太费时，一般不可取。这是因为有些图形组成部分全部在窗口外，可以完全排除，不必进行扫描转换。所以一般采用先裁剪再扫描转换的方法。这部分主要对几种二维裁剪算法进行介绍。

1. 二维点的裁剪
2. 主要算法

如果点P（x,y）满足下列不等式，则保存该点用于显示（裁剪区域一般是一个正则矩形，其边界位于）：

如果上述四个不等式有任何一个不满足，则裁剪掉该点（将不会存储

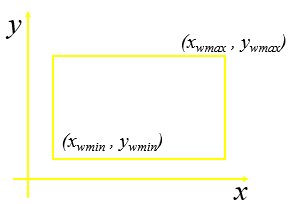
或显示）。

1. 相关应用

此算法主要用于包含云、海面泡沫、烟或者爆炸等用小圆或小球这样的粒子进行建模的场景。

1. 二维线段的裁剪
2. 裁剪目的

已知线段的两个端点P1和P2，观察矩形裁剪窗口W，判断线段P1P2是否落在裁剪窗口之内并找出其位于内部的部分。

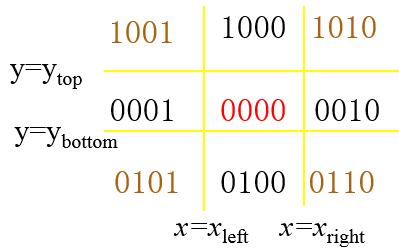


1. 主要算法

2．1 Cohen-Sutherland算法

最早开发的快速线段裁剪算法，得到了广泛应用，通过初始测试来减少求交计算，从而提高效率，其特点是对显然不可见线段的快速判别，可以直接推广到三维。

裁剪窗口的四边的延伸线将平面分割成九个区域，给这些区域编码，根据线段的两个端点所在的区域编码，快速判别线段与窗口的关系。编码规则：若区域位于窗口的四边：的可见侧编码为0，不可见侧编码为1。使得每个区域有如下形式的唯一编码：



特点：1）窗口的编码为：“0000”；

2）四个角区域编码为2个“0”，2个“1”；

3）四个含窗口某一边的区域编码为1个“1”，3个“0”；

具体算法：已知线段P1P2两个端点P1，P2和窗口W，对P1，P2分别进行编码，为P1\_code,P2\_code;

1. 线段两端点的编码为其所在区域的编码；
2. 若两端点的编码中有一位同为1，则该线段在窗口外，裁剪完成，否则转下步；
3. 若两端点的编码全为0，则显示该线段，裁剪完成，否则转下步；
4. 依left,right,bottom,top顺序，求线段的交点，重复第二步；

算法描述：P1=（X1，Y1），P2=（X2，Y2）

Step1(编码):P1编码，P1\_code,P2编码，P2\_code

Step2:

2.1与边进行比较

2.1.1 P1\_code.与P2\_code.比较，同为“1”则拒绝；

2.1.2 P1\_code.与P2\_code.比较，其中一个为“1”，则

（1）与边求交点，重新编码；

（2）拒绝在不可见侧，去掉一部分；

（3）将交点的编码赋给删除的端点的编码；

2.2与边比较（步骤同上）；

2.3与边比较（步骤同上）；

2.4与边比较（步骤同上）；

2．2 中点分割算法

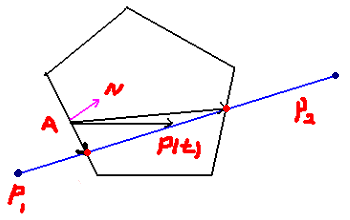
Cohen-Sutherland算法需要求线段与窗口边界的交点，而中点分割算法将上述直接求交点的过程用折半查找（即求中点）的方法来代替。中点分割方法也采用对线段端点进行编码的方式来判断完全可见和显然完全不可见的线段。对于既不是完全可见也不是显然完全不可见的线段，中点分割算法分别求离两个端点最近的可见点，这两个最近的可见点（如果存在的话）之间的线段就是原线段的可见部分。

中点分割算法是Cohen-Sutherland算法的硬件版本，其只需要做加法和移位，不要做乘除法，因此用硬件实现既简单又有效。

2．3 Cyrus-Beck算法

Cyrus-Beck算法（简称CB算法），也称作参数化裁剪算法，其将线段表示成参数方程的形式，求线段与窗口边界的交点的参数，确定了线段的可见部分之后再用交点的参数计算交点的坐标。Cyrus-Beck算法不仅适用于矩形裁剪窗口。而且适用于凸多边形裁剪窗口，更加具有一般性。

考虑一个凸多边形区域R和一条线段P1P2，要求计算线段落在区域R中的部分，如下图所示，假定A是区域R边界上的一点，N是区域边界在点A处的内法向量，线段P1P2的参数方程表示为P（t）。



P（t）=（P2-P1）t+P1（0<=t<=1）

对于线段上任意一点P（t），有三种情形：

1. N（P（t）-A）<0,这时P（t）必在多边形外侧；
2. N（P（t）-A）=0,这时P（t）必在多边形边界或其延长线上；
3. N（P（t）-A）>0,这时P（t）必在多边形内侧；

由凸多边形的性质可知，P（t）在凸多边形内的充要条件是：对于凸多边形边界上任意一点A和该处的内法向量N都有

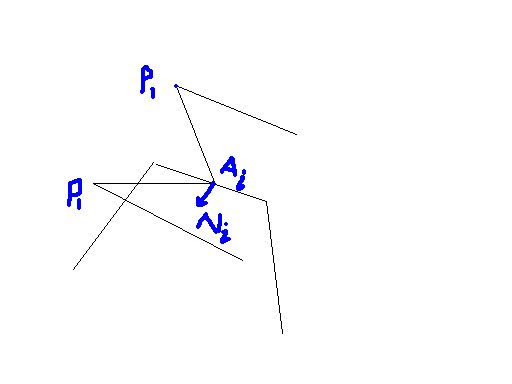
N（P（t）-A）>0。

现在假设多边形有n条边，在每条边上取一点和该点处的内法向量（i=1，2，…n）,可见线段的参数满足如下不等式：

把直线方程P（t）代入得：

若对于某个i，有，这时,P2P1与对应

平行，无交点，如下图所示，这时有两种情况：线段在区域的外侧或内侧。



如果是在区域外侧，可直接判断线段在多边形外；如果在区域内

侧，则可忽略继续处理其他边。

若不等式的解为：

记,

,则判别方法为：

若,则两者是可见线段的端点参数，否则有,则整条

线段在区域外部。

上式解得几何意义非常明显。以P1P2方向与内法向量的内积符

号分为两组。一组为下限组，一组为上限组，下限组以

为特征，表示在该处沿P1P2方向前进将接近或

进入多边形内侧，上限组以为特征，表示在该

处沿P1P2方向前进将越来越远的离开多边形区域。

2．4 Liang-Barsky算法

Liang-Barsky算法（简称LB算法）可以看作是Cyrus-Beck算法的特殊情况，此时裁剪窗口不是一般的凸多边形，而是矩形窗口。Liang-Barsky算法中所用的量如下所示，

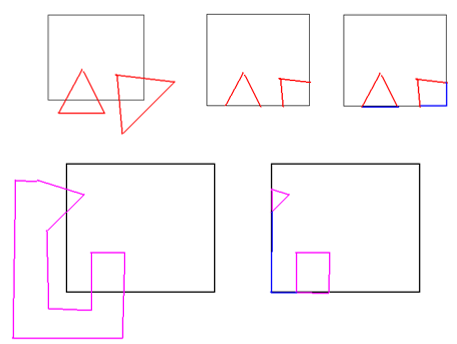
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 边 | 外法向量 | 边上一点 |  |  |  |
| 上  Y=Yt | (0,1) | (X,Yt) | (X0-X,Y0-Yt) | (Y0-Yt) | (Y1-Y0) |
| 下  Y=Yb | (0,-1) | (X,Yb) | (X0-X,Y0-Yb) | -(Y0-Yb) | -(Y1-Y0\_ |
| 左  X=Xr | (1,0) | (Xr,Y) | (X0-Xr,Y0-Y) | (X0-Xr) | (X1-X0) |
| 右  X=Xl | (-1,0) | (Xl,Y) | (X0-Xl,Y0-Y) | -(X0-Xl) | -(X1-X0) |

由于Cyrus-Beck算法适用于凸多边形裁剪窗口，更加具有一般性，因此当裁剪窗口为矩形时，其效率不高。而Liang-Barsky算法是针对矩形窗口的Cyrus-Beck算法，效率比一般的Cyrus-Beck算法更高。

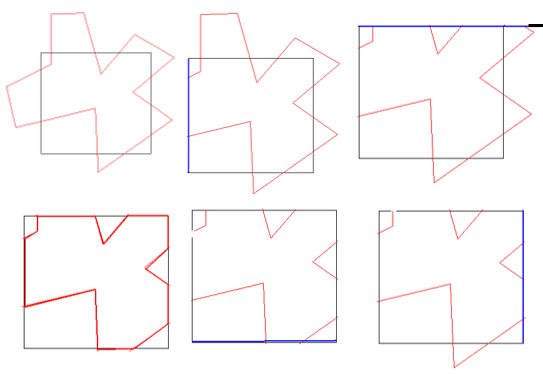
2．5 NLN（Nicholl-Lee-Nicholl）算法

点P0有三种情况：

1. P0在窗口内；
2. 如果P1也在窗口内，那么P0P1完全可见；
3. 否则，从P0向窗口的四个角点引射线，将窗口外部划分为T、B、R、L四个区域，P1位与哪一个区域中，就求P0P1与对应边界的交点，P0到交点之间部分分为线段的可见部分。
4. P0在左边区，从P0向窗口的四个角点引射线，得到L、LR、LT、LB四个区域，这四个区域确定了P0P1与窗口边界相交的情况；
5. 如果P1在区域L中，我们就求P0P1与左边界的交点，交点到P1之间部分分为线段的可见部分；
6. 如果P1在区域LB中，我们分别求P0P1与左边界以及P0P1与下边界的交点，两个交点之间部分分为线段的可见部分；
7. 如果P1不在L、LR、LT、LB这四个区域中，那么P0P1完全不可见；
8. P0在左上角区；
9. 从P0向窗口的四个角点引射线，得到两种可能的区域划分；
10. 如果P1在L、T、LB、TB、TR、LR等区域中，可以求P0P1与对应边界的交点，得到线段的可见部分；
11. 如果P1不在上述区域中，那么P0P1完全不可见；
12. 多边形填充区的裁剪



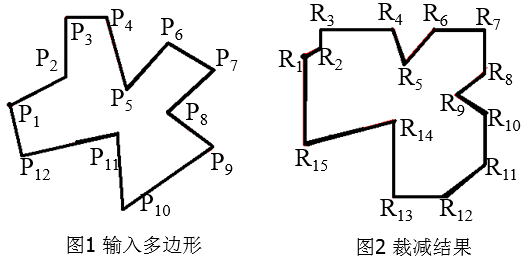
* + - 1. 主要算法
  1. Sutherland-Hodgeman算法



算法的输入是以顶点序列表示的多边形，用P1P2…Pn表示把P1连到P2，P2连

到P3，…最后把Pn连到P1所成的多边形，算法的输出也是一个顶点序列，构成

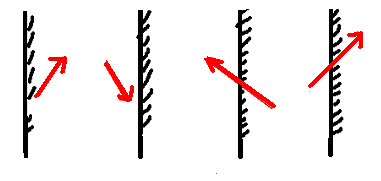
一个或多个多边形，如下图所示。



算法的每一步，考虑以窗口的一条边以及延长线构成的裁剪线。该线把平面分成

两部分：一部分包含窗口，称为可见一侧；另一部分称为不可见一侧。依序考虑

多边形各条边的两个端点S、P，它们与裁剪线的位置关系只有四种，如下图所示，



每条线段端点S、P与裁剪线比较之后，可输出0至2个顶点。对于情况（I）

两端点S、P都在可见一侧，则输出P，（II）若S、P都在不可见一侧，则输出

0个顶点。(III)若S在可见一侧，P在不可见一侧，则输出线段SP与裁剪线的

交点。（IV）若S在不可见一侧，P在可见一侧，则输出线段SP与裁剪线的交

点和线段终点P。

* 1. Weile-Atherton算法

被裁剪的多边形称为主多边形，裁剪多边形称为裁剪区域，

主多边形被裁剪后形成的新边界由主多边形和裁剪区域的部分边界共同构成。在窗口边界上从出交点沿逆时针方向到达另一个与多边形的交点，如果该点是已处理的点，则走向下一步。如果是新交点，则继续按逆时针方向处理多边形直到遇到已处理的顶点。

主要算法：

1)求交点，建立P\_intersect\_list;建立Q\_intersect\_list

2)交点排序P\_intersect\_list， Q\_intersect\_list分别排序，

排序标准按照各自的参数，并且记住同一点的位置。

3）按照顺序输出。

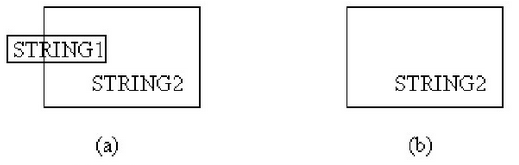
（四）曲线的裁剪

这里以“圆的裁剪“为例，方法如下：

1. 先裁剪再扫描转换
2. 先作包围盒测试，排除完全可见或显然完全不可见的情况；
3. 如果圆的包围盒与窗口相交，将圆分为四个象限，对每个象限再作包围盒测试；
4. 如果包围盒仍然与窗口相交，那么我们用解方程组得方法计算圆与窗口边界的交点，再对得到的圆弧进行扫描转换；
5. 边扫描转换边裁剪
6. 对于空心圆，可以在扫描转换时逐点进行判断，只画那些位与窗口内部的像素；
7. 对于实心圆，可以在扫描转换时逐区段进行判断，只画区段位于窗口内部的部分；
8. 文字的裁剪

这部分对文字的裁剪，主要考虑到字符串精度、字符精度以及点精度等条件；

1. 基于字符串



1. 基于字符



1. 基于构成字符的最小元素
2. 点阵字符：点；
3. 矢量字符：线、多边形；

三、总结与展望

通过以上对二维裁剪算法的分析，Cohen-Sutherland裁剪算法是一个最早开发

的线段裁剪算法，采用编码方法实现了对完全可见和完全不可见线段的快速判断，这使得该算法应用广泛，但此算法并不能判断出所有的完全不可见线段，在进行了求交运算后才能判断直线是否完全可见，而对完全不可见直线进行求交运算就降低了该算法的运行效率，为了提高该算法的运行效率，有很多工作对此算法进行了改进。中点分割算法和Cohen-Sutherland算法一样，首先对直线段的端点进行编码，核心思想是通过二分逼近来确定直线段与窗口的交点，是Cohen-Sutherland算法的硬件版本，其只需要做加法和移位，不要做乘除法，因此用硬件实现既简单又有效。Cyrus-Beck算法能够使用任意多边形对一条直线段进行裁剪，算法思想简单但是运算量较大，也不适用来计算边数多的多边形，同时此算法也只适用于凸多边形，但是却比Cohen-Sutherland算法更有效，目前也有相关工作对此算法进行了改进，使其能够扩展到对凹多边形的处理。梁-Barsky算法是针对标准矩形窗口提出的更快的直线段裁剪算法，把二维裁剪的问题化成二次一维裁剪问题，而把裁剪问题转化为解一组不等式的问题，进而改善了Cohen-Sutherland算法中全部摒弃的判断只适合于那些仅在窗口同一侧线段的不足之处，使得图形在处理上更加简单，更加方便，弥补了其他一些算法在图形构建上的缺陷。（Nicholl-Lee-Nicholl）算法通过在裁剪窗口周围创立多个区域，在求交之前尽量更多的进行区域测试，从而达到避免多次求交的目的，NLN算法需要线段的斜率分别与任意一个端点到四个角点连线的斜率进行比较，但是多次的斜率运算会降低裁剪的效率，也有很多工作对此算法进行了改进来提高裁剪效率。