

多边形裁剪

刘世光

天津大学计算机学院

天津大学计算机科学与技术学院

主要内容

- 多边形裁剪

Sutherland-Hodgman多边形裁剪算法;**Weiler-Atherton**算法;

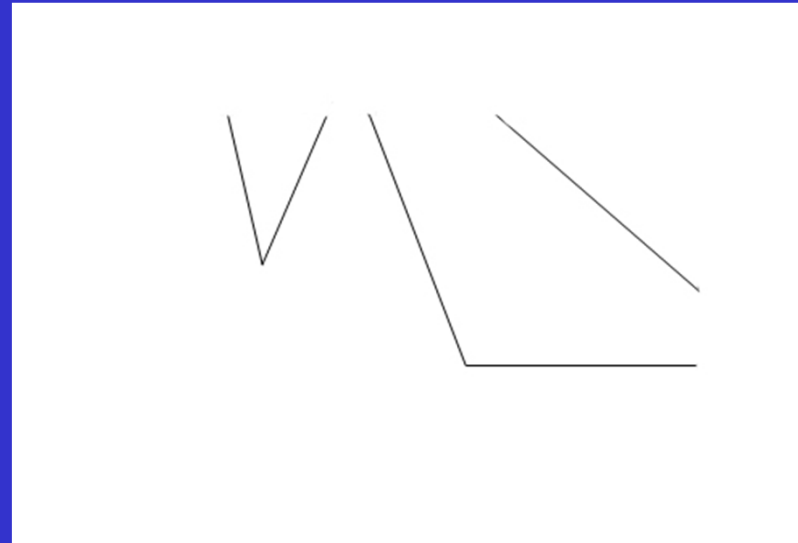
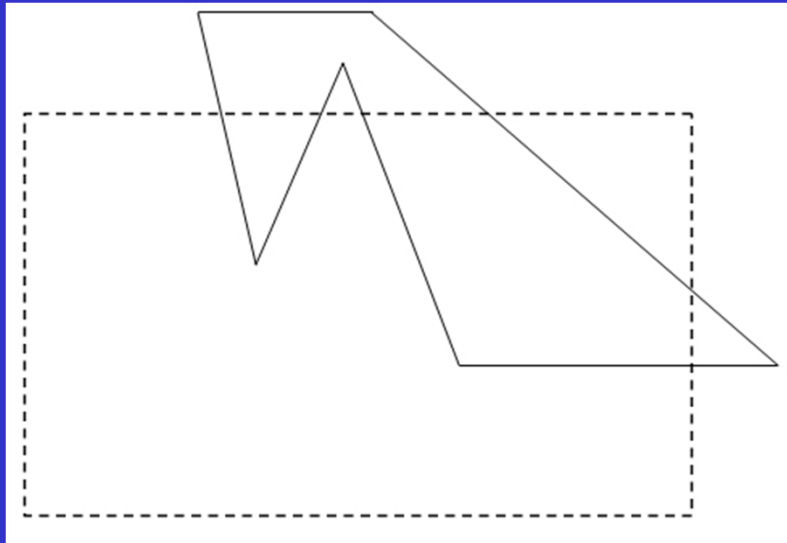
- 曲线裁剪、字符裁剪

多边形裁剪与线段裁剪

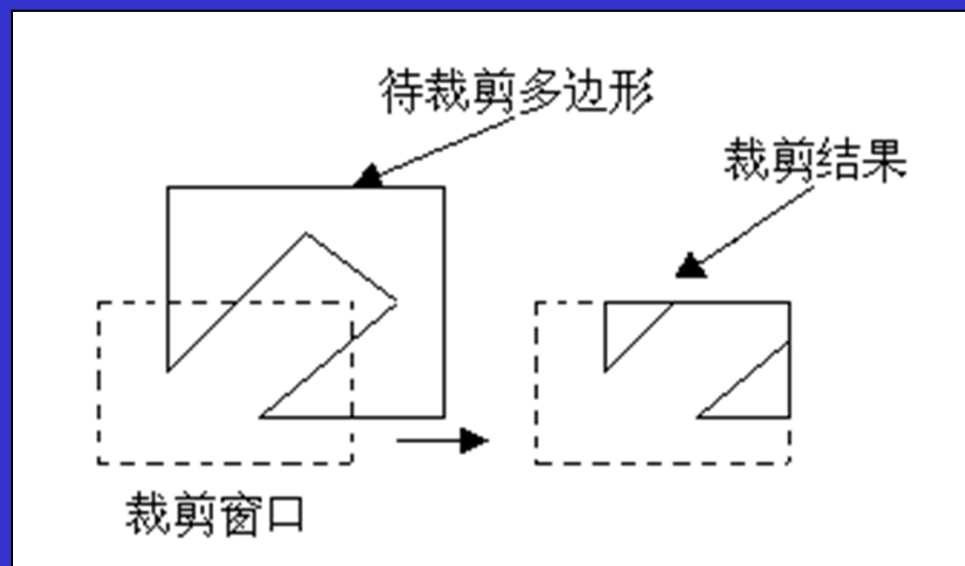
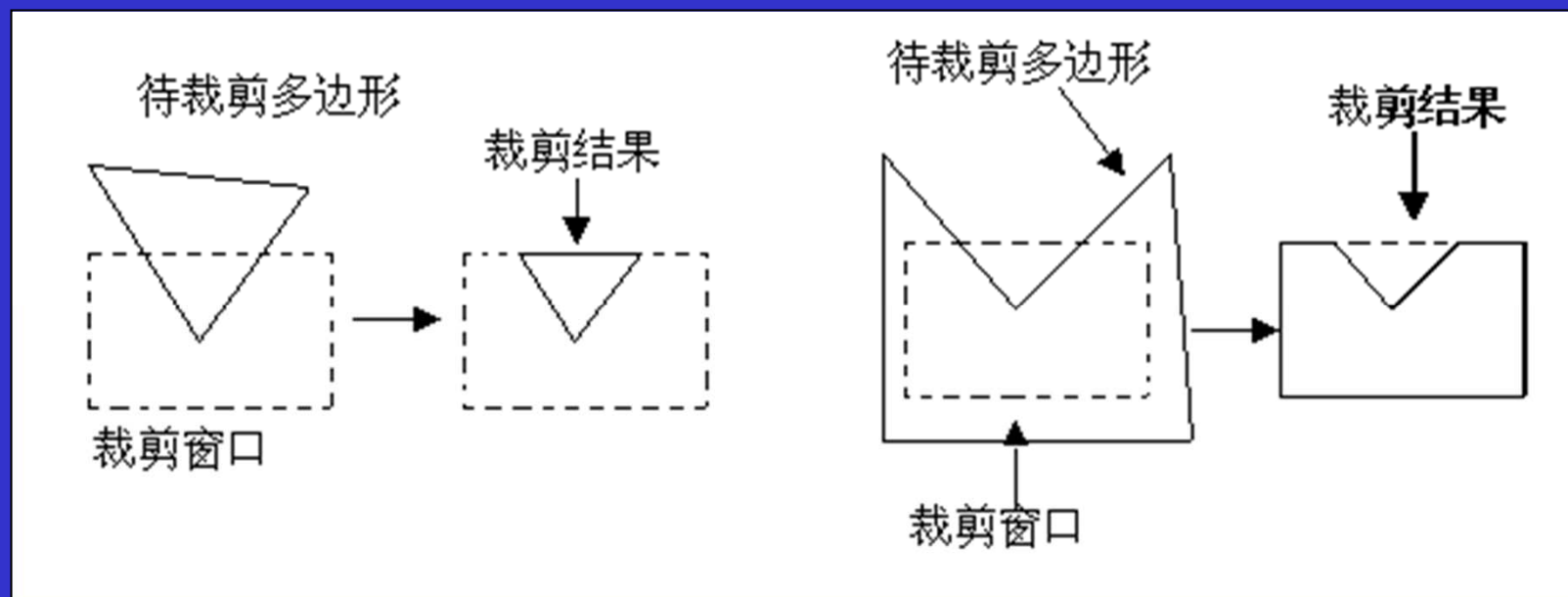
- 使用线段裁剪进行多边形裁剪，则裁剪后的边界将显示为一系列不连接的线段，没有关于如何形成裁剪后的封闭多边形的完整信息。

多边形裁剪

- 新的问题：
 - 边界不再封闭
 - 产生多个部分



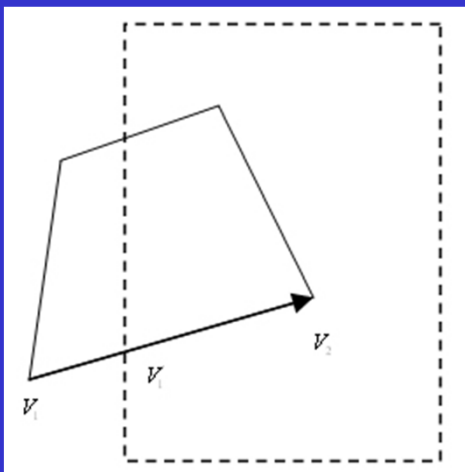
多边形裁剪



Sutherland-Hodgman算法

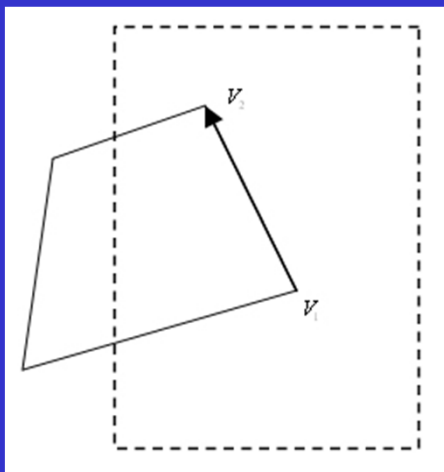
- 由Sutherland和Hodgman提出的裁剪凸多边形填充区的高效算法。
- 其总体策略是顺序地将每一线段的一对顶点送给一组裁剪器（左、右、下、上）。一个裁剪器完成一对顶点的处理后，该边裁剪后留下的坐标值送给下一个裁剪器。

- 用裁剪边界对多边形的边裁剪时的四种情况：



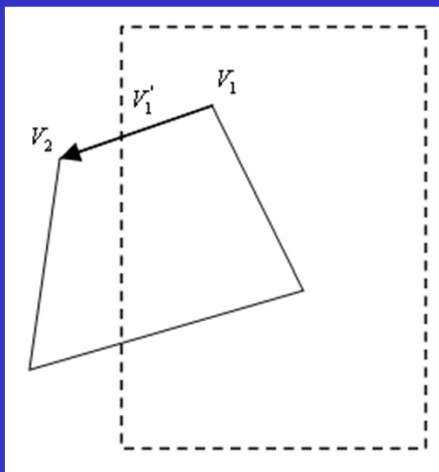
外——→内

输出 V_1', V_2



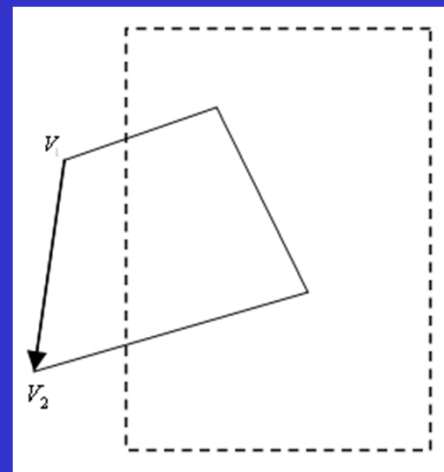
内——→内

输出 V_2



内——→外

输出 V_1'



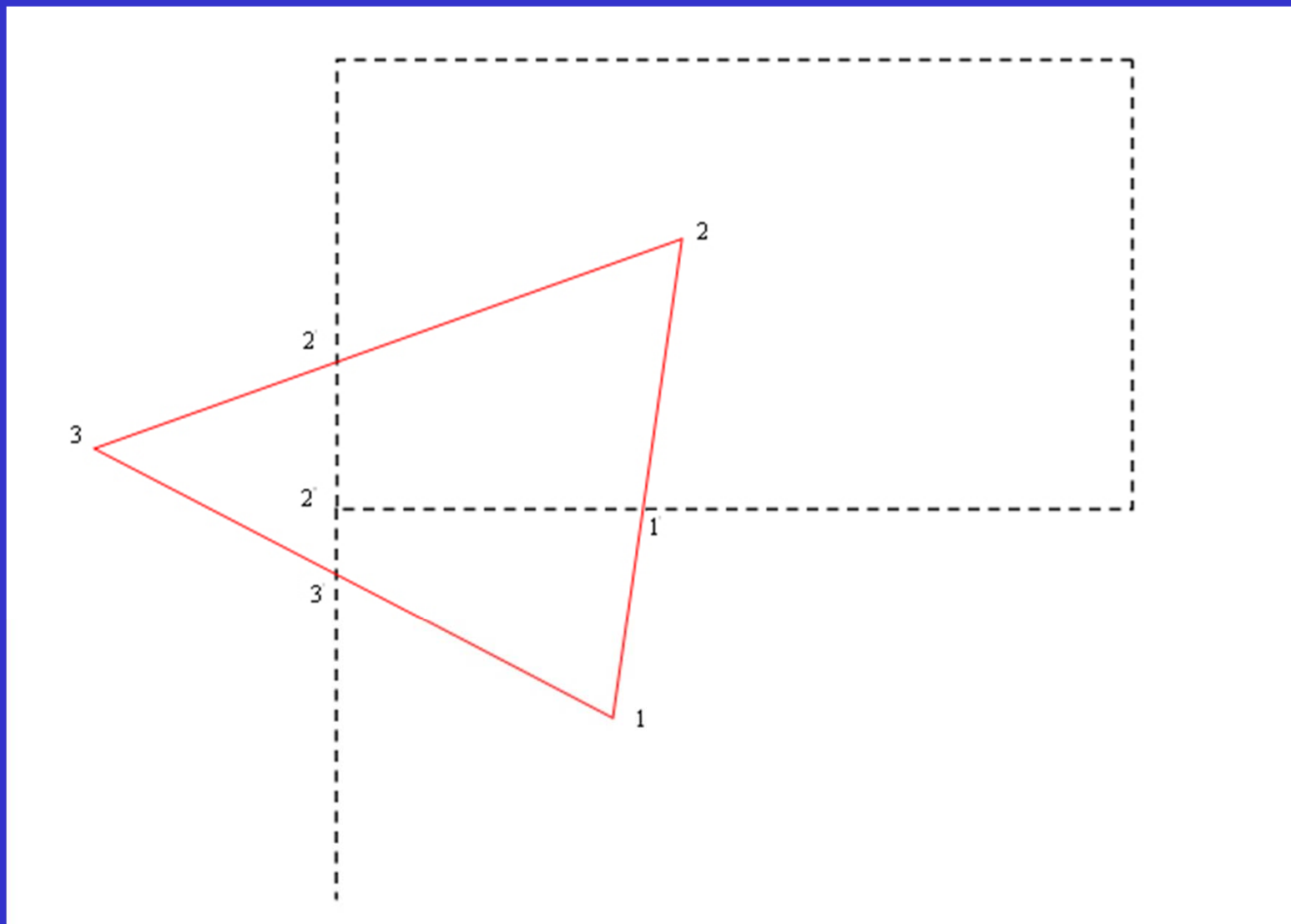
外——→外

输出: 无

实现步骤:



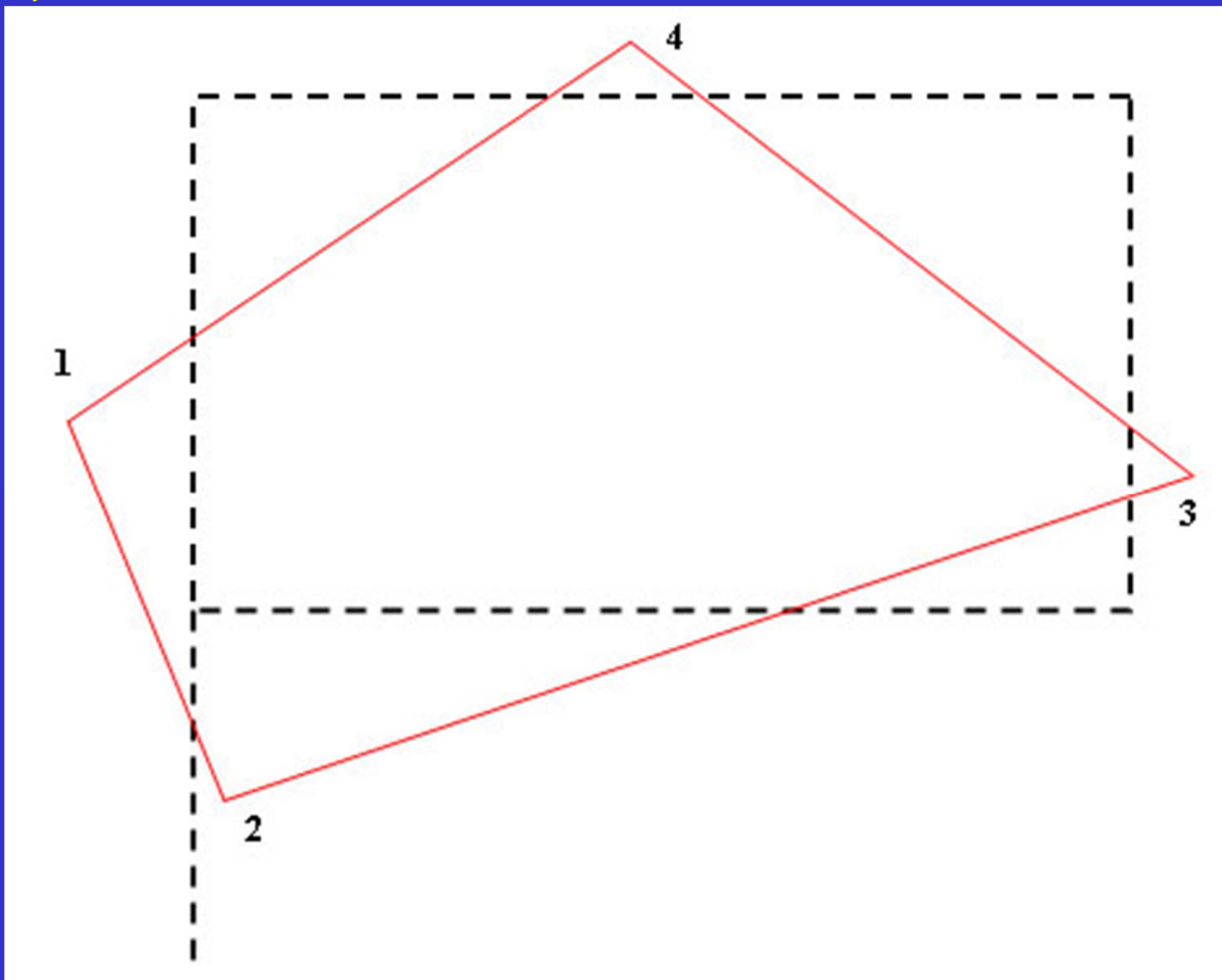
举例:



Sutherland-Hodgman算法

- 小结：
 - 逐边裁剪：两次分解
 - 第一次分解：将多边形关于矩形窗口的裁剪分解为它关于窗口四条边所在直线的裁剪；
 - 第二次分解：将多边形关于一条直线的裁剪分解为多边形各边关于该直线的裁剪。

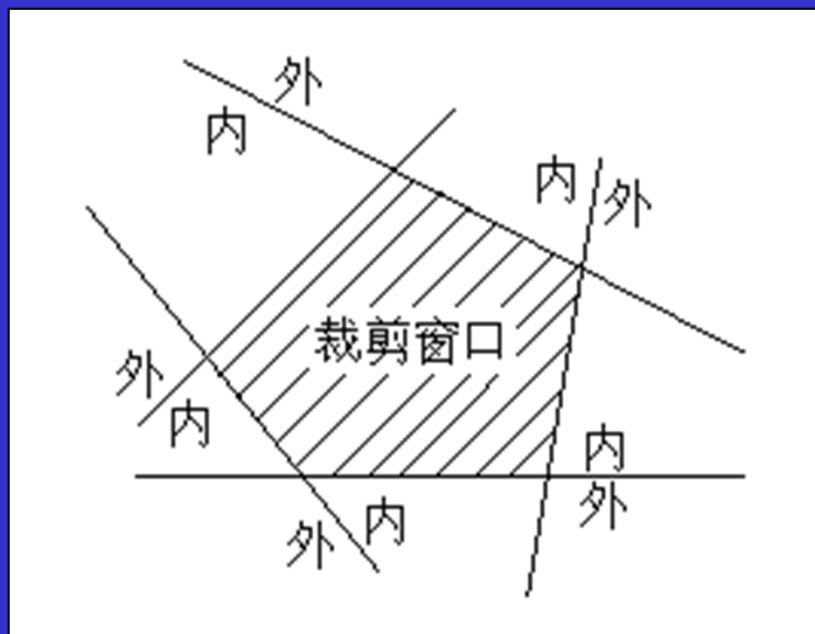
其他例子:



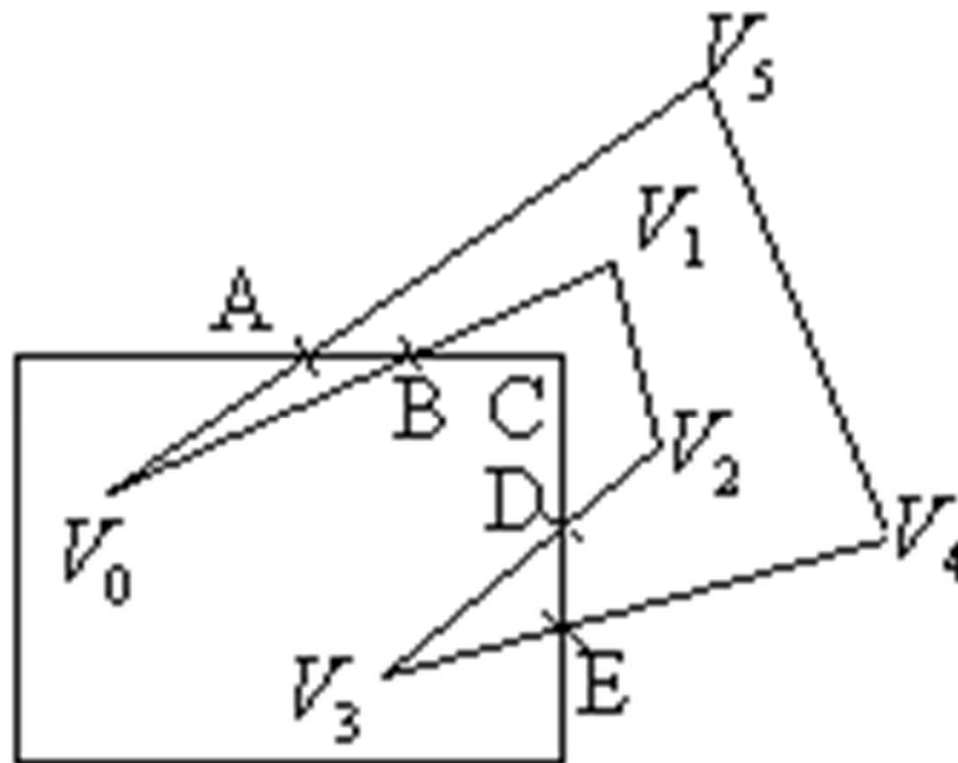
Sutherland-Hodgman算法

– 推广

- 关于任意凸多边形窗口的裁剪



练习：



Sutherland-Hodgman算法

- 算法的不足之处:

裁剪凹多边形时，可能显示一条多余的直线，原因是该算法只有一个输出顶点队列。

- 改进方法:

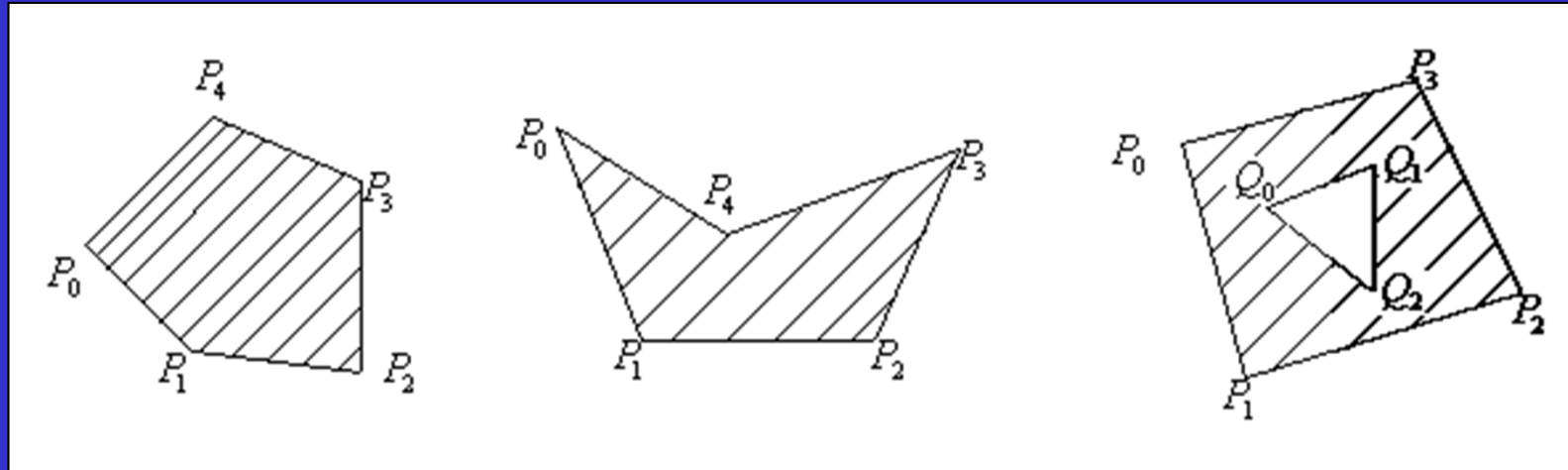
- 1、将凹多边形分解为两个或者更多的凸多边形。
- 2、将输出顶点队列分为两个或多个；
- 3、使用更一般的凹多边形裁剪算法。

Weiler-Atherton算法

- 这个算法是一个通用的多边形裁剪算法，可用于裁剪凸多边形或凹多边形。

Weiler-Atherton算法

- 裁剪窗口为任意多边形



- 主多边形：被裁剪多边形，记为A
- 裁剪多边形：裁剪窗口，记为B

Weiler-Atherton算法

— 多边形顶点的排列顺序

原则：使多边形区域位于有向边的左侧

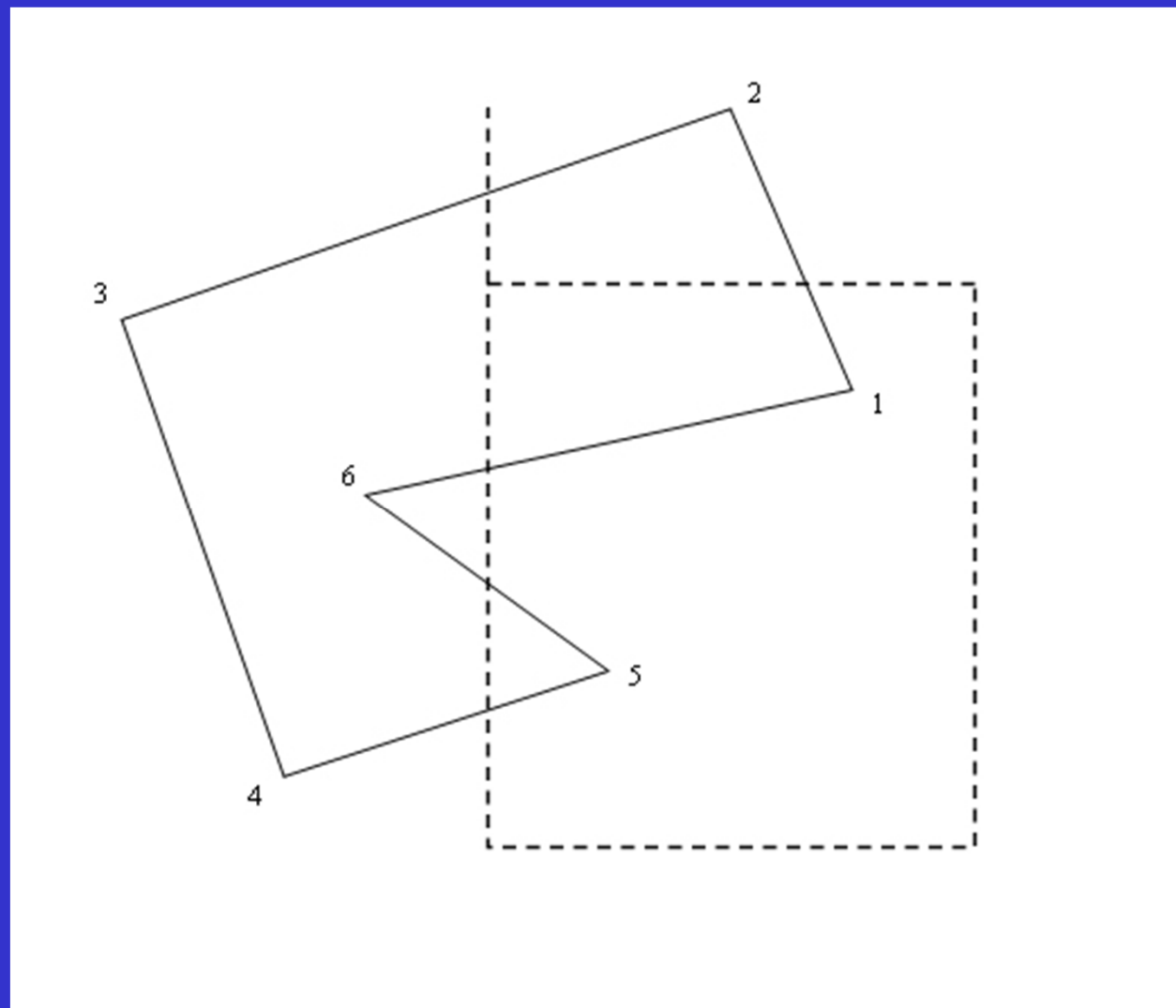
— 两类交点

- 进点：主多边形边界由此进入裁剪多边形内
- 出点：主多边形边界由此离开裁剪多边形区域

Weiler-Atherton算法

- 算法步骤（逆时针的多边形填充区域）
 - 1、按逆时针处理多边形填充区，直到一对内-外顶点与边界相遇（出点）；
 - 2、在窗口边界上从出交点沿逆时针到达另一个多边形的交点。如果该点是处理边的点，则走向下一步。如果是新交点，则继续按逆时针处理多边形直到遇见已处理的顶点。
 - 3、形成裁剪后该区域的顶点队列。
 - 4、回到出交点并继续按逆时针处理多边形的边。

举例：



Weiler-Atherton算法

– 推广：

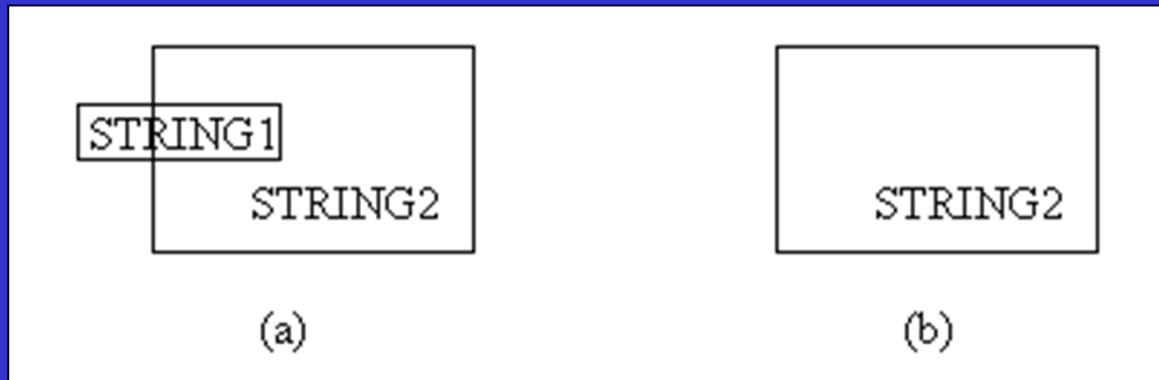
可用任意多边形窗口来处理任意多边形填充区。

非线性裁剪窗口边界的多边形裁剪

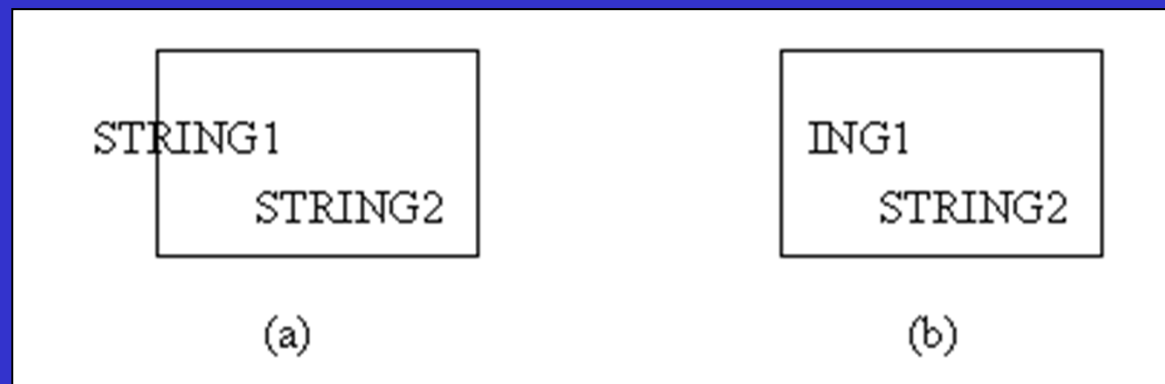
- 首先用直线段逼近边界，然后使用一般的多边形裁减窗口的裁剪算法对其处理。

字符裁剪

- 基于字符串



- 基于字符



字符裁剪

- 基于构成字符的最小元素
 - 点阵字符：点
 - 矢量字符：线、多边形（曲线裁剪）

