

遇见多边形（下）： 边标志算法

华中科技大学软件学院 万琳





提纲

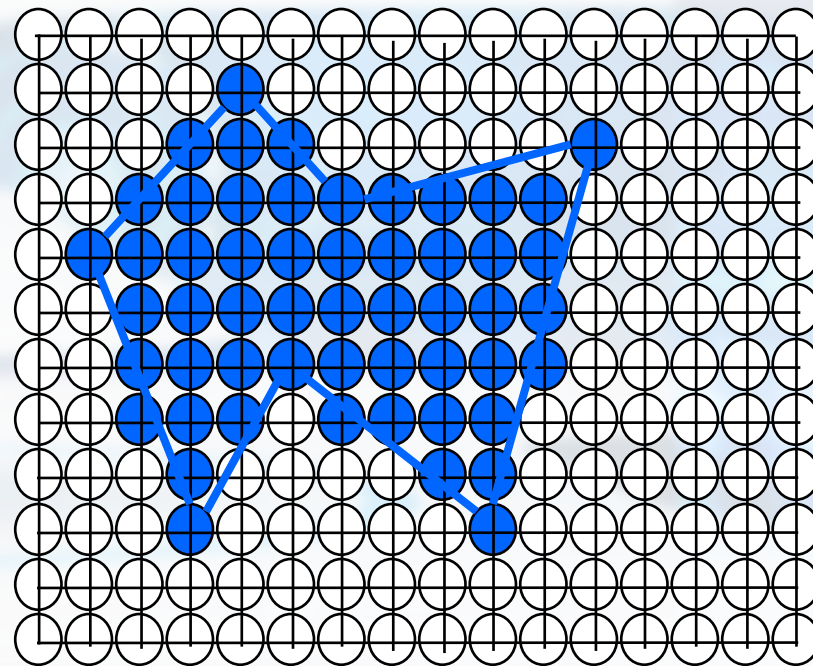
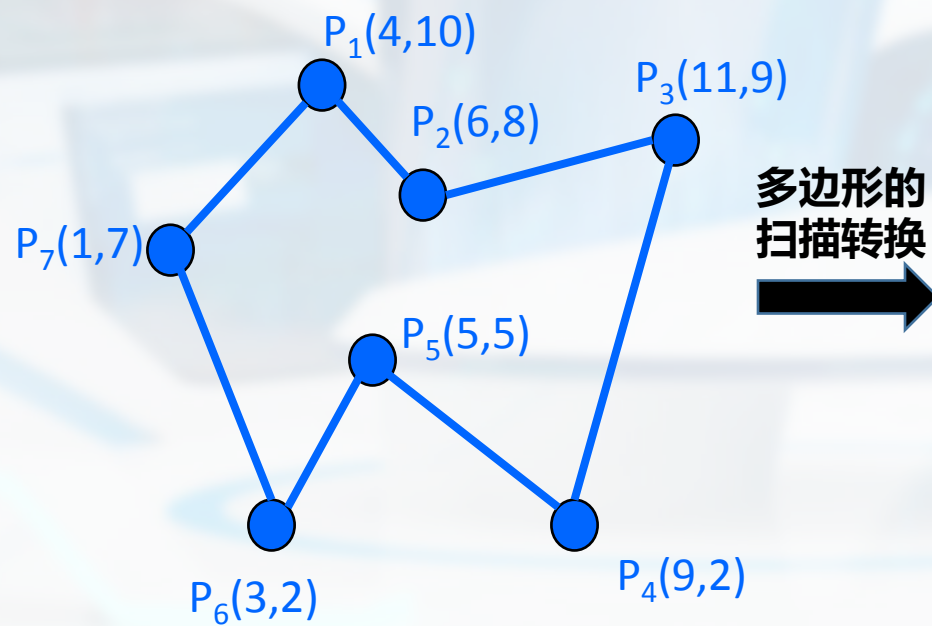
- ① 奇妙的想法
- ② 边标志算法
- ③ 算法分析

1

奇妙的想法

输入：多边形顶点序列 $P_1(x_1, y_1)$ 到 $P_7(x_7, y_7)$

输出：最佳逼近这个多边形的像素点集

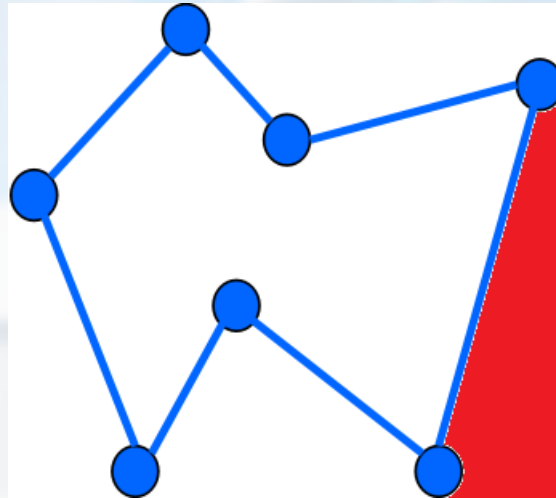
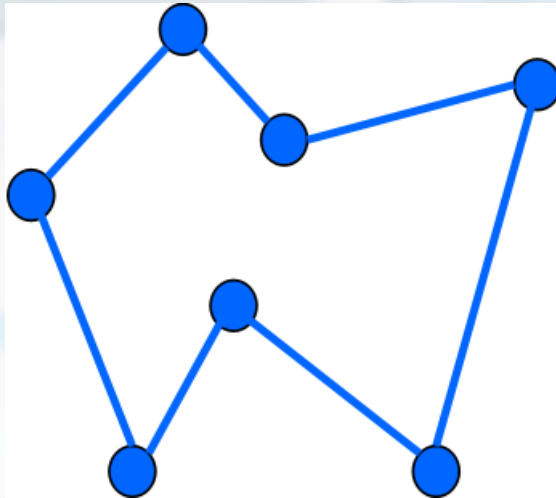


1

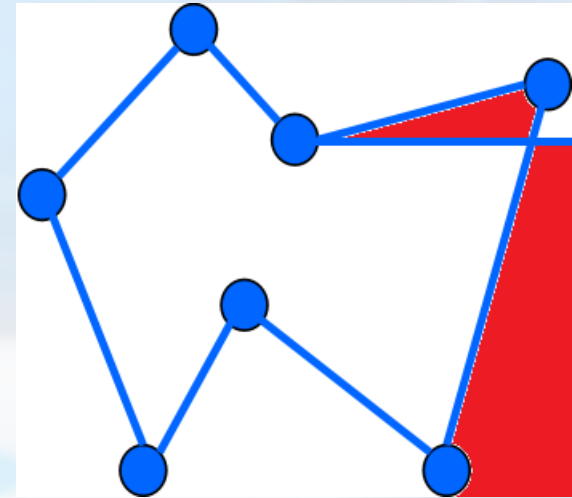
奇妙的想法

输入：多边形顶点序列 $P_1(x_1, y_1)$ 到 $P_7(x_7, y_7)$

输出：最佳逼近这个多边形的像素点集



逐边向右取反

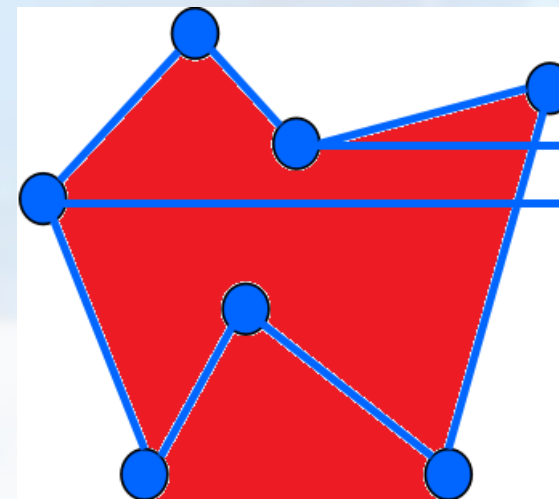
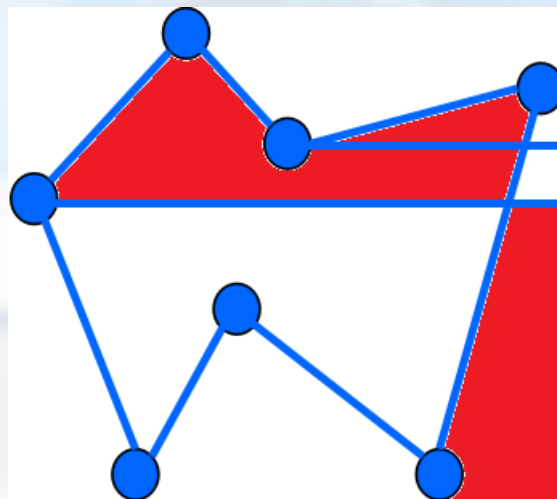
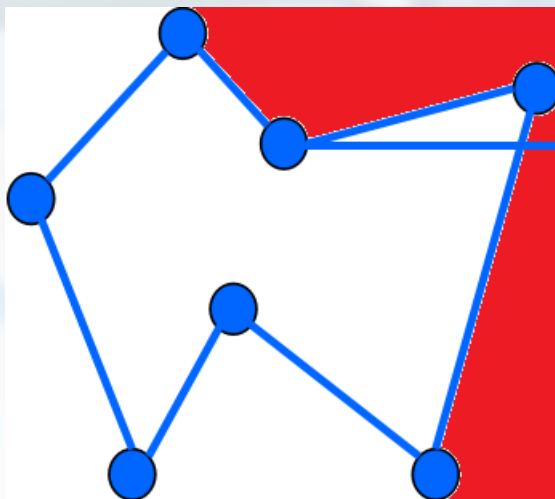


1

奇妙的想法

输入：多边形顶点序列 $P_1(x_1, y_1)$ 到 $P_7(x_7, y_7)$

输出：最佳逼近这个多边形的像素点集



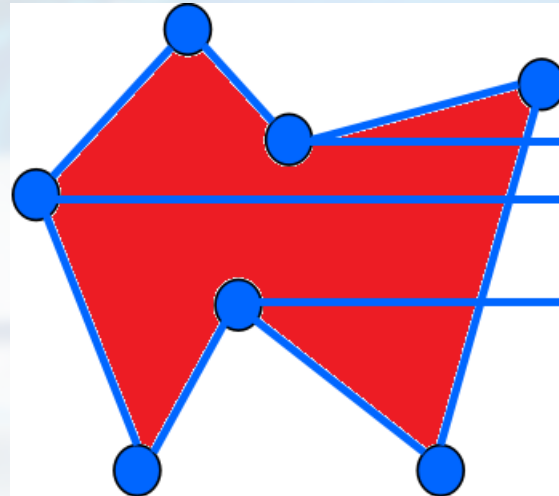
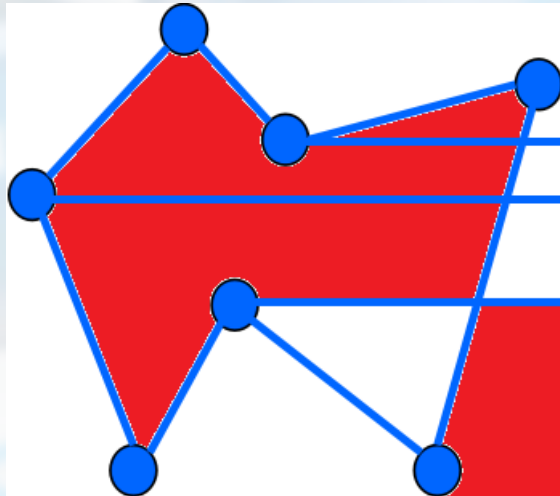
1

奇妙的想法

输入：多边形顶点序列 $P_1(x_1, y_1)$ 到 $P_7(x_7, y_7)$

输出：最佳逼近这个多边形的像素点集

边缘填充
算法

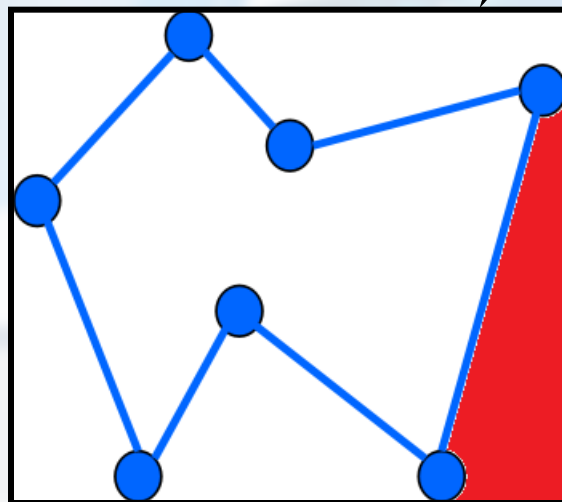


1

奇妙的想法

已经做的简化

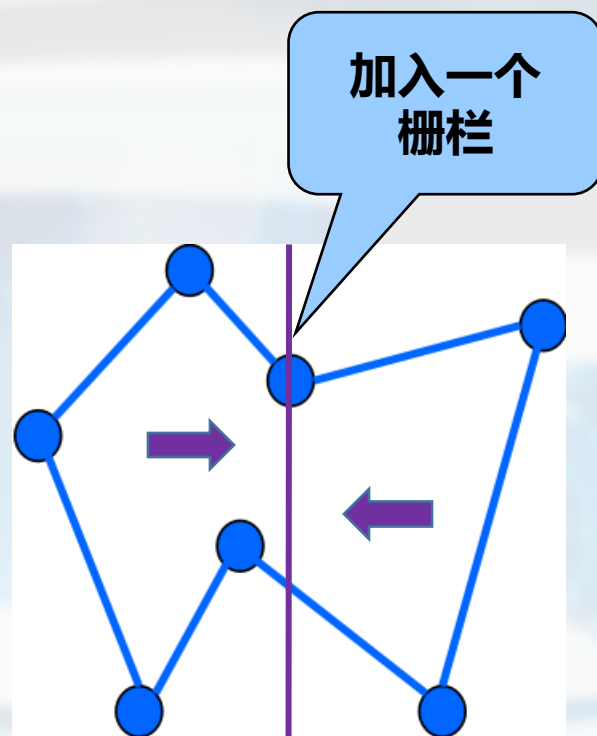
外接矩形



1

奇妙的想法

改进！



2

边标志算法

分为两个步骤：

(1)打标记

(2)填充

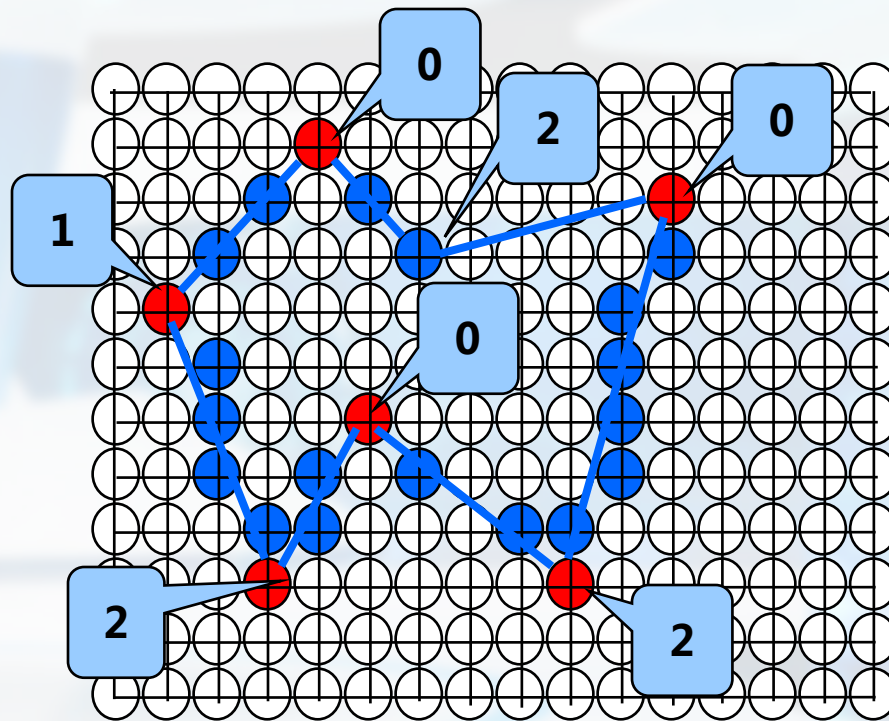
Inside

初值：假

遇到标记点：取反

真：填充；

假：不填充



2

边标志算法

分为两个步骤：

(1)打标记

(2)填充

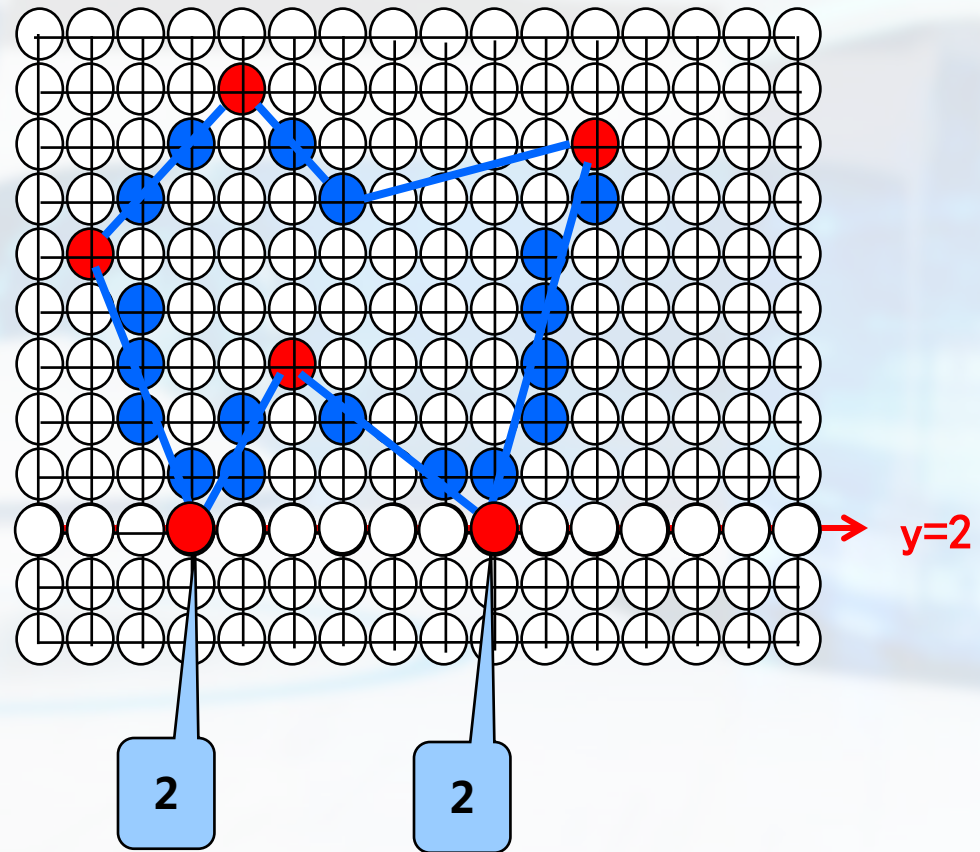
Inside

初值：假

遇到标记点：取反

真：填充；

假：不填充



2

边标志算法

分为两个步骤：

(1)打标记

(2)填充

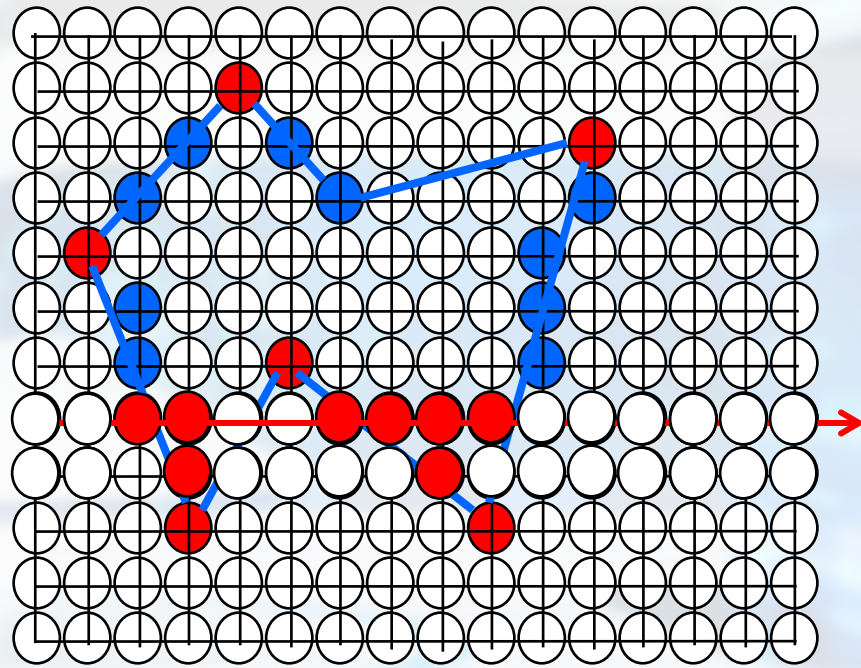
Inside

初值：假

遇到标记点：取反

真：填充；

假：不填充



3

算法分析

- ◆ 边标志算法对每个像素仅访问一次。与边缘填充算法和栅栏填充算法相比，避免了对帧缓存中大量元素的多次赋值，但仍然需逐条扫描线地对帧缓存中的元素进行搜索和比较。
- ◆ 当用软件实现本算法时，速度与Y向连贯性算法相当，但本算法用硬件实现后速度会有很大提高。



谢谢

软件学院 万琳