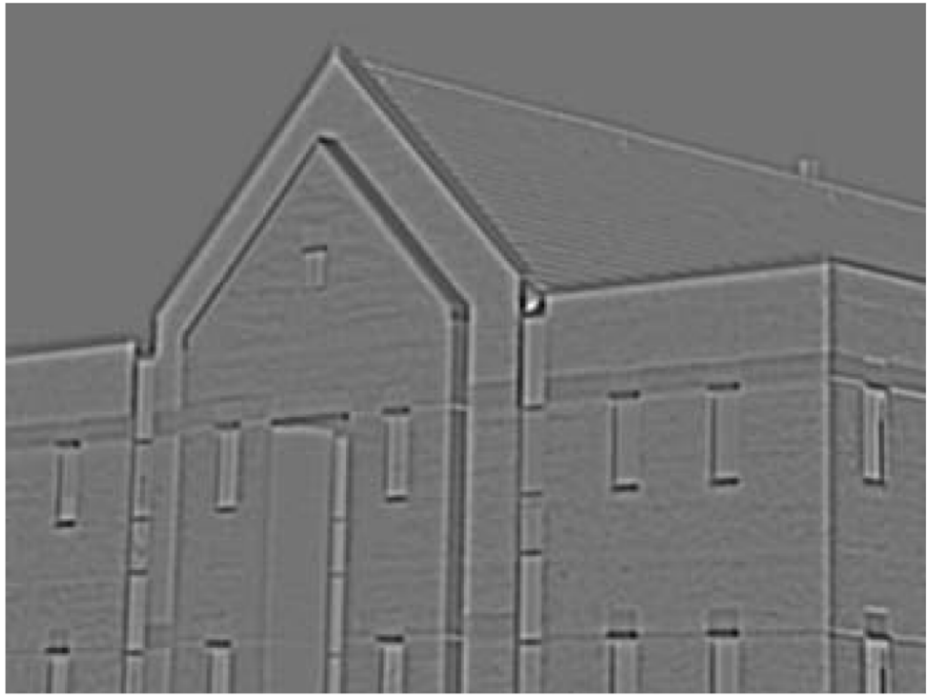
举例

- Marr-Hildreth边缘检测器

原图



前两步骤的结果

举例

- Marr-Hildreth边缘检测器

产生闭环，“意大利空心粉”效应

零交叉（阈值为0）

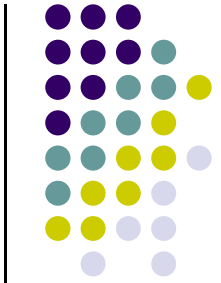
举例

- Marr-Hildreth边缘检测器

零交叉（阈值为0）

零交叉（阈值为最大值的4%）

扩展



1. 考虑不同的尺度
   - 尝试不同的σ，保留共同的零交叉
2. 使用高斯差分（DoG）来近似LoG

$$\mathrm{DoG}(x,y) = \frac{1}{2\pi\sigma_1^2}e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2}e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

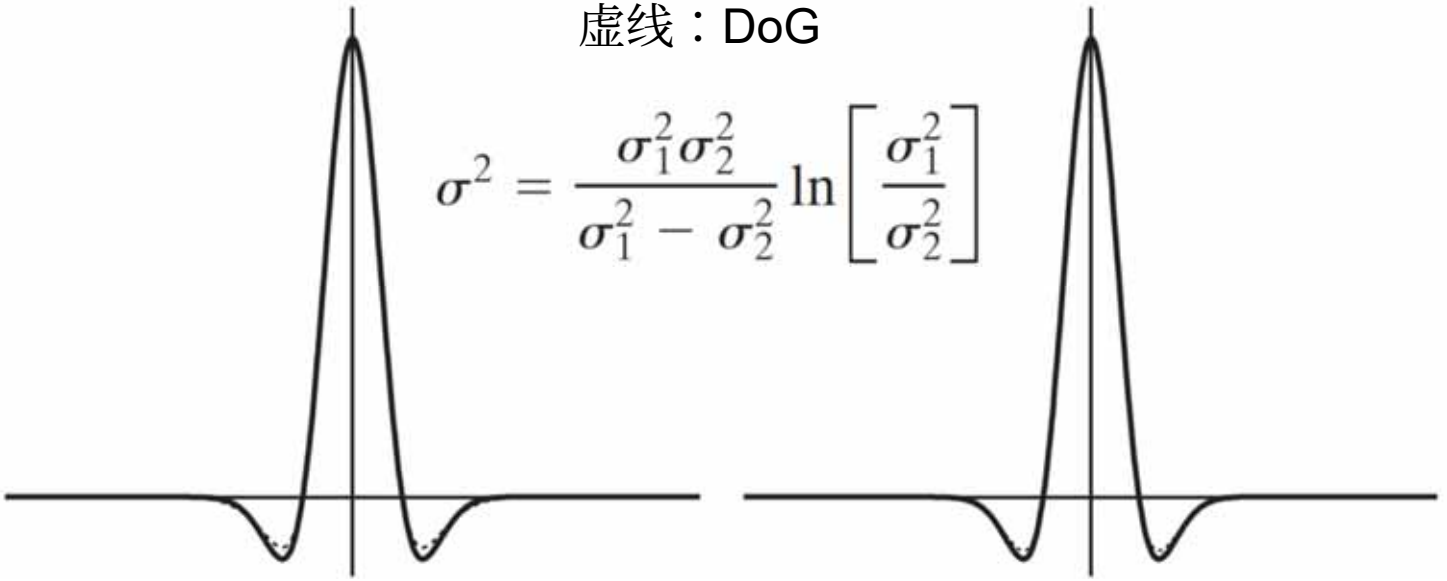
   - 人视觉系统的某些通道对方向和频率是有选择的（标准差比值为1.75:1）
   - 采用1.6:1，可以建模上述现象，并很好地近似LoG

扩展

1. 考虑不同的尺度
   - 尝试不同的σ，保留共同的零交叉
2. 使用高斯差分（DoG）来近似LoG

实线：LoG
虚线：DoG

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 - \sigma_2^2}\ln\left[\frac{\sigma_1^2}{\sigma_2^2}\right]$$



1.75:1　　1.6:1 拟合LoG更好

坎尼（Canny）边缘检测器

1. 低错误率
   - 所有边缘都被找到，并且没有伪响应

2. 边缘点应被很好地定位
   - 已定位的边缘必须尽可能接近真实边缘

3. 单一的边缘点响应
   - 对每个真实边缘点，检测器仅返回1个点

坎尼（Canny）边缘检测器

- 数学分析
  - 考虑加性高斯白噪声污染的1维台阶边缘
  - 高斯一阶导数是近似最优的检测器

- 拓展到二维情况
  - 挑战：边缘可能是任意方向
  - 使用二维高斯函数平滑图像
  - 基于梯度寻找边缘的方向

坎尼（Canny）边缘检测器

1. 高斯函数平滑输入图像f

$$f_s(x, y) = G(x, y) \star f(x, y)$$

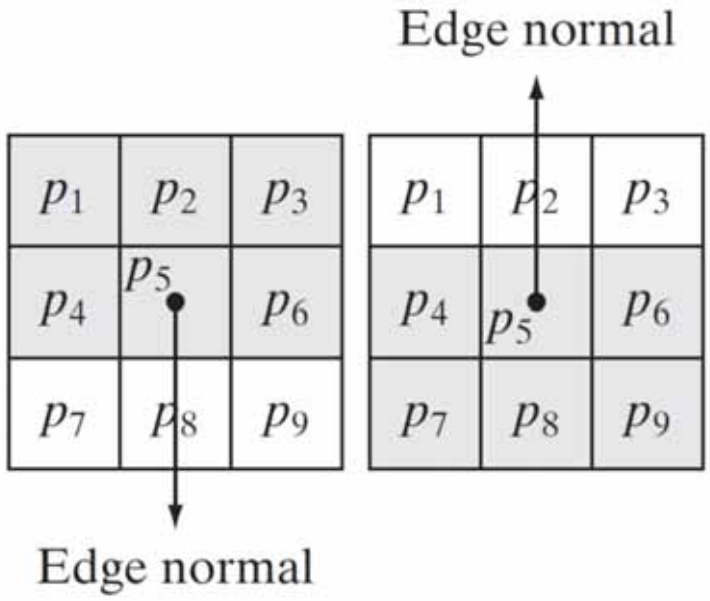
   - 其中 $$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

2. 计算图像f_s的梯度
   - 梯度大小 $$M(x, y) = \sqrt{g_x^2 + g_y^2}$$
   - 梯度方向 $$\alpha(x, y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$
   - 其中$g_x = \partial f_s/\partial x,\ g_y = \partial f_s/\partial y$

坎尼（Canny）边缘检测器

3. 非最大抑制
   - 目的：把梯度生成的粗边缘变细
   - 指定梯度（边缘法线）的多个离散方向
     - 4种边缘：水平、垂直、$+45°$、 $-45°$

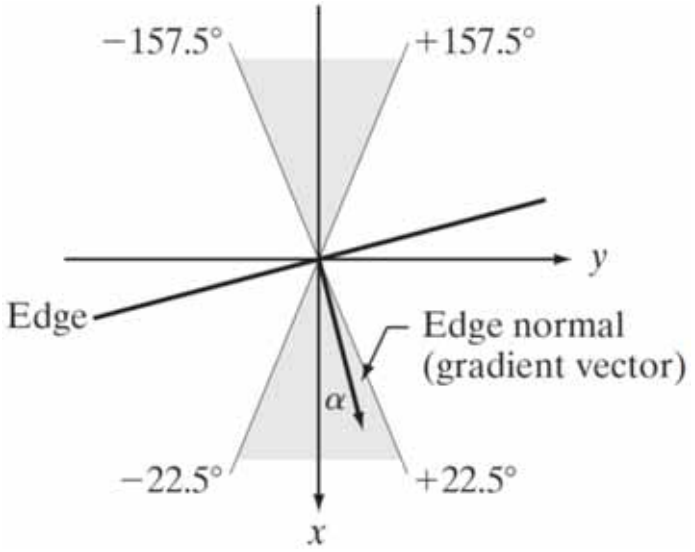
水平边缘的
两种梯度方向



坎尼（Canny）边缘检测器

3. 非最大抑制
   - 目的：把梯度生成的粗边缘变细
   - 指定梯度（边缘法线）的多个离散方向
     - 4种边缘：水平、垂直、$+45°$、 $-45°$
   - 根据梯度（边缘法线）的方向确定边缘的方向

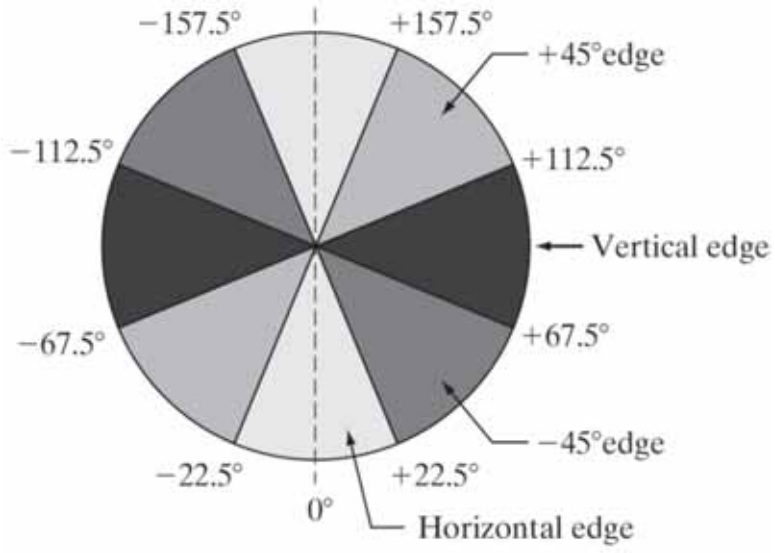
水平边缘
的梯度范围



坎尼（Canny）边缘检测器

3. 非最大抑制
   - 目的：把梯度生成的粗边缘变细
   - 指定梯度（边缘法线）的多个离散方向
     - 4种边缘：水平、垂直、$+45°$、 $-45°$
   - 根据梯度（边缘法线）的方向确定边缘的方向

4种边缘
的梯度范围



坎尼（Canny）边缘检测器

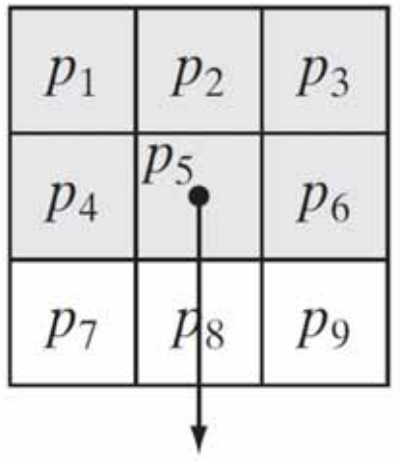
3. 非最大抑制
   - 考虑(x, y)为中心的3×3区域
   - 考虑4个方向：水平、垂直、$+45°$、 $-45°$
   - 确定和梯度$a(x, y)$最接近的方向d_k
   - 如果$M(x, y)$的值比(x, y)在d_k方向的任一邻居数值小，对其抑制：

$$g_N(x, y) = 0$$

   - 否则，保留：

$$g_N(x, y) = M(x, y)$$

p_2和p_8



坎尼（Canny）边缘检测器

4. 滞后阈值
   - 目的：减少伪边缘点
   - 两个阈值：低阈值T_L、高阈值T_H
   - 两个阈值的比值为：2:1或3:1
   - 利用T_H阈值化
     - 强边缘点 $g_{NH}(x, y) = g_N(x, y) \geq T_H$
   - 利用T_L阈值化

$$g_{NL}(x, y) = g_N(x, y) \geq T_L$$

     - g_{NL}包含g_{NH}的所有非零元素

坎尼（Canny）边缘检测器

4. 滞后阈值
   - 去掉g_{NL}中和g_{NH}重复的点

$$g_{NL}(x, y) = g_{NL}(x, y) - g_{NH}(x, y)$$

     - 弱边缘点
5. 连通性分析
   a. 遍历g_{NH}中的每一个点p
      ① 保留g_{NL}中和p连通（例如8连通）的点
   b. 去掉g_{NL}剩余的点
   c. 合并g_{NH}和g_{NL}

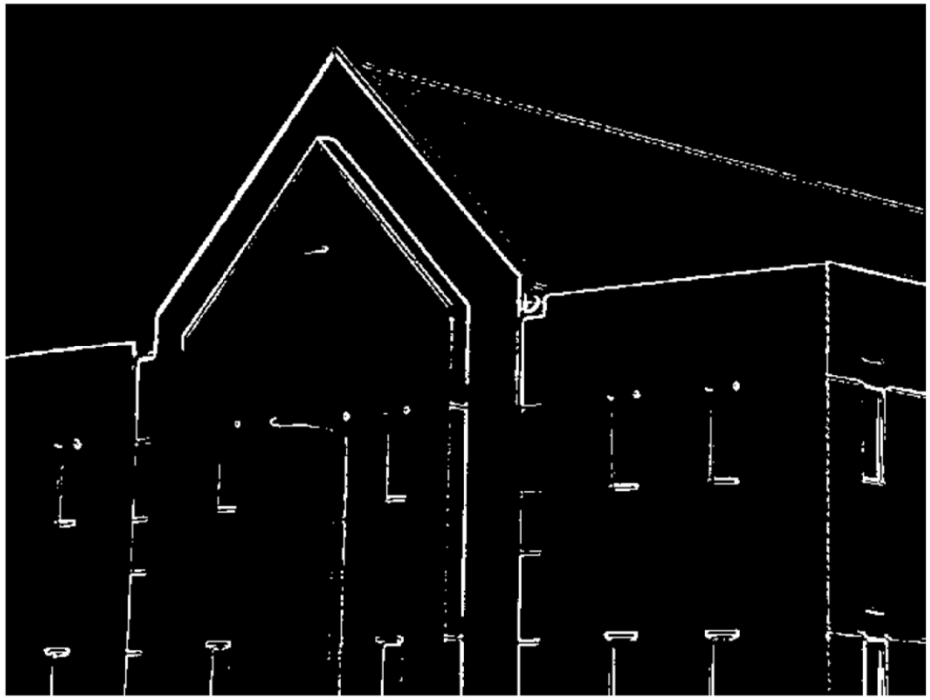
坎尼（Canny）边缘检测器

1. 高斯函数平滑输入图像
2. 计算图像的梯度
   - 梯度大小、梯度角度
3. 非最大抑制
   - 得到细边缘
4. 滞后阈值
   - 检测边缘
5. 连通性分析
   - 连接边缘

举例

- 基本边缘检测
  - 高斯平滑图像→梯度阈值化

原图



梯度阈值化

举例

- 高级边缘检测

Marr-Hildreth边缘检测器

坎尼边缘检测器

举例

- 高级边缘检测

Marr-Hildreth边缘检测器

坎尼边缘检测器

举例

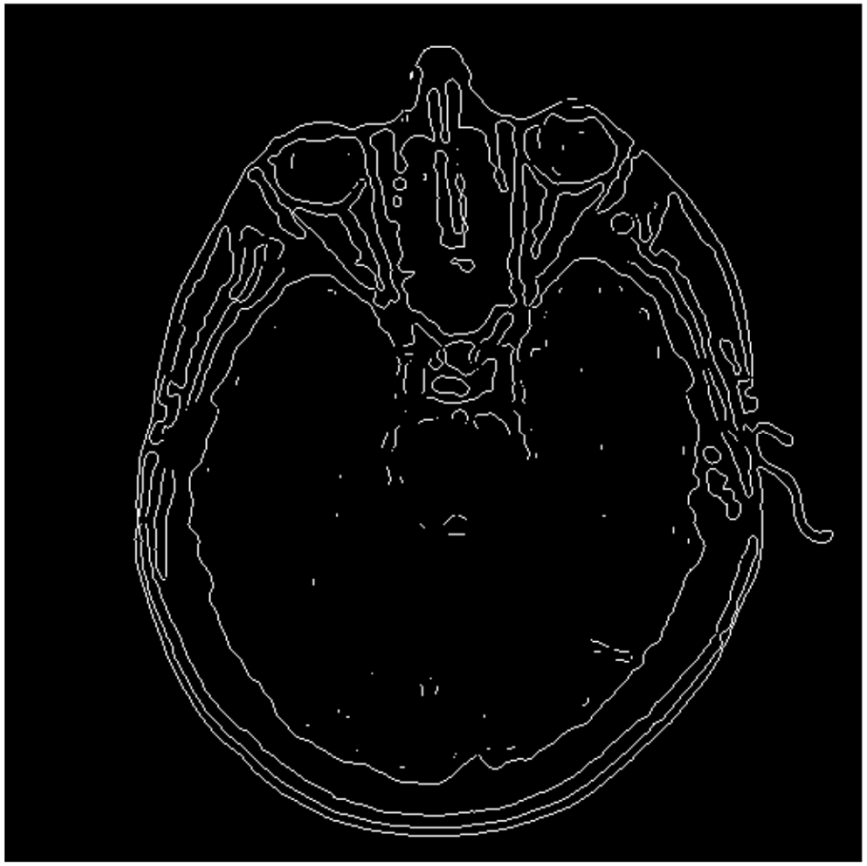
- 基本边缘检测

原图

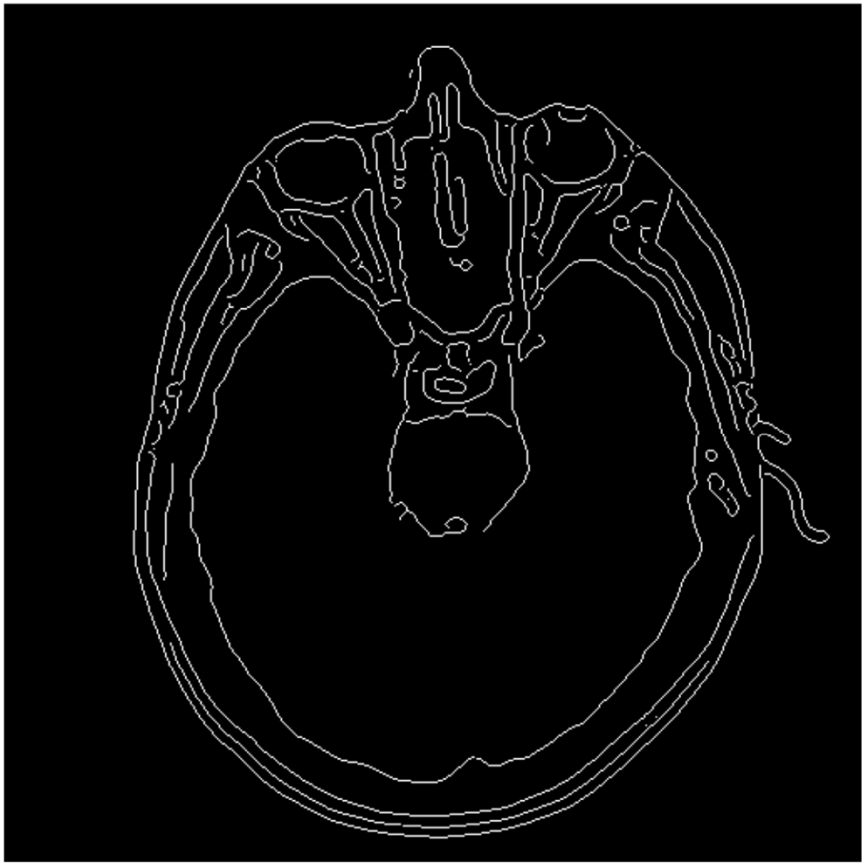
梯度阈值化

举例

- 高级边缘检测

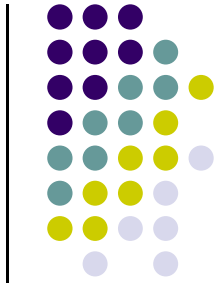


Marr-Hildreth边缘检测器



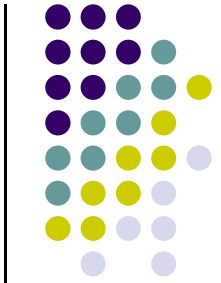
坎尼边缘检测器

提纲



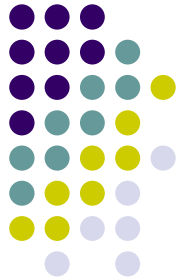
- 基础知识
- 点、线、边缘检测
  - 背景知识
  - 孤立点的检测
  - 线检测
  - 边缘模型
  - 基本边缘检测
  - 高级边缘检测
- 边缘连接和边界检测

背景



- 边缘检测的结果不完美
  - 噪声
  - 不均匀照明导致的边缘间断
  - 虚假的灰度值不连续
- 边缘连接
  - 将边缘像素组合成有意义的边缘或区域边界
  1. 局部处理
  2. 区域处理
  3. 全局处理（使用霍夫变换）

提纲



- 边缘连接和边界检测
  - 局部处理
  - 区域处理
  - 全局处理

局部处理

1. 分析每个候选点(x, y)邻域内像素的特点
2. 将依据某准则相似的点连接起来

a. 基于梯度大小判断相似

$$|M(s,t) - M(x,y)| \leq E$$

- (s, t)在(x, y)的邻域内

b. 基于梯度方向判断相似

$$|\alpha(s,t) - \alpha(x,y)| \leq A$$

如果大小和方向都相似，则连接(s, t)和(x, y)

局部处理

- 简化算法（计算简单）
  1. 计算输入图像$f(x,y)$的梯度大小和方向
     - 梯度大小$M(x,y)$，梯度方向$\alpha(x,y)$
  2. 依据下式生产二值图像

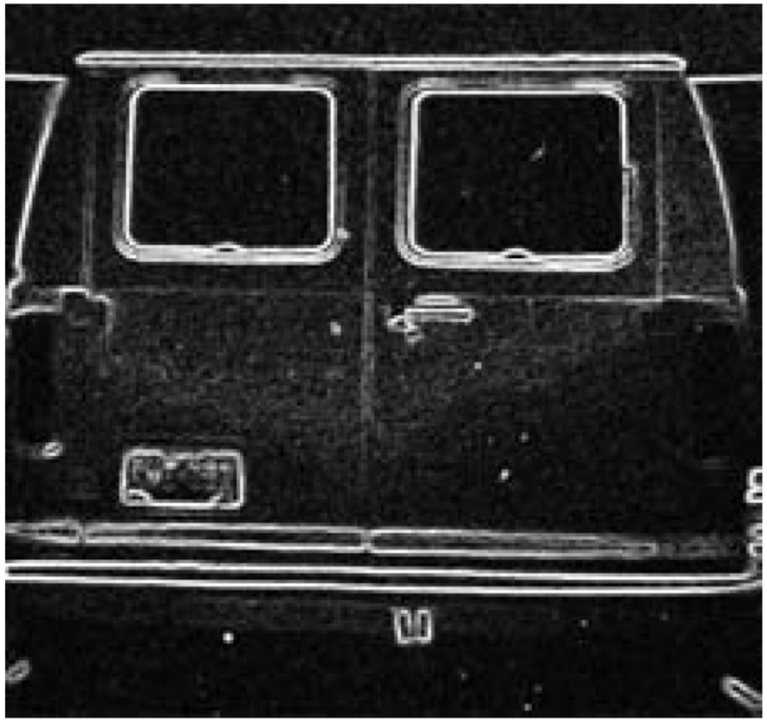
$$g(x,y)=\begin{cases}1 & \text{if } M(x,y)>T_M \text{ AND } \alpha(x,y)=A\pm T_A\\ 0 & \text{otherwise}\end{cases}$$

     - T_M为阈值、A为特定角度、T_A为允许的带宽
  3. 逐行扫描，填充长度不超过K的空隙
  4. 以角度θ旋转$g(x,y)$，重复第3步，再反旋转

举例

- 寻找车牌

汽车尾部图像

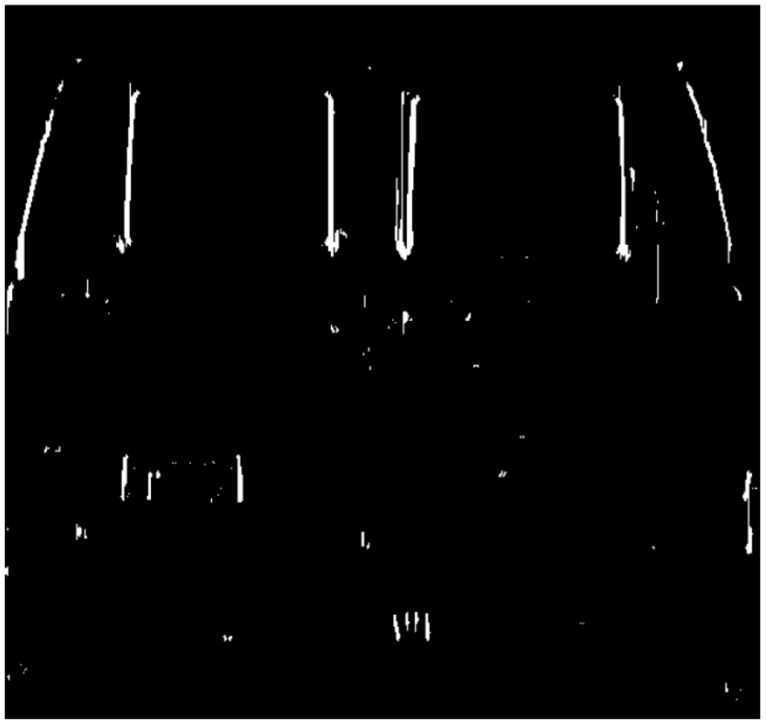


梯度大小图像

举例

- 寻找车牌

水平连接的边缘像素



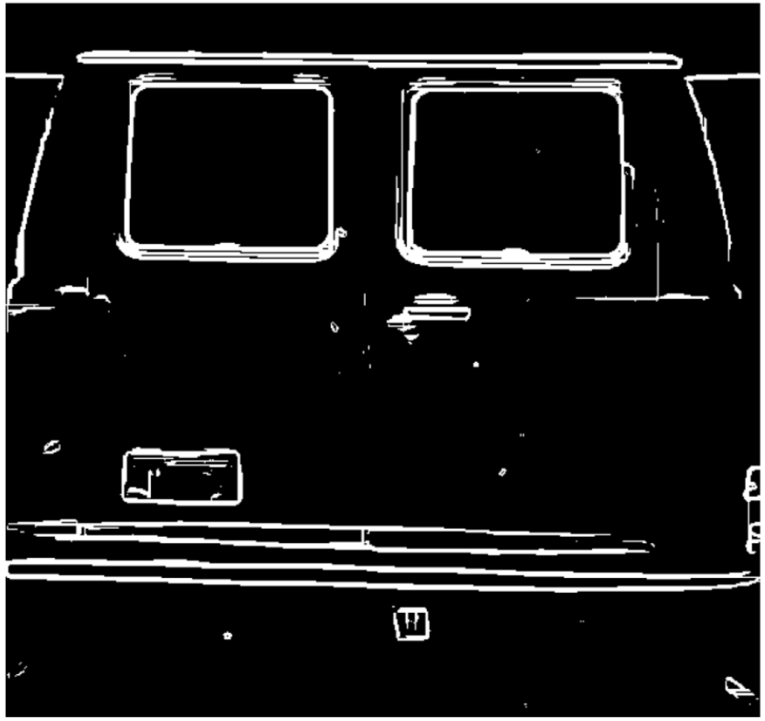
垂直连接的边缘像素

举例

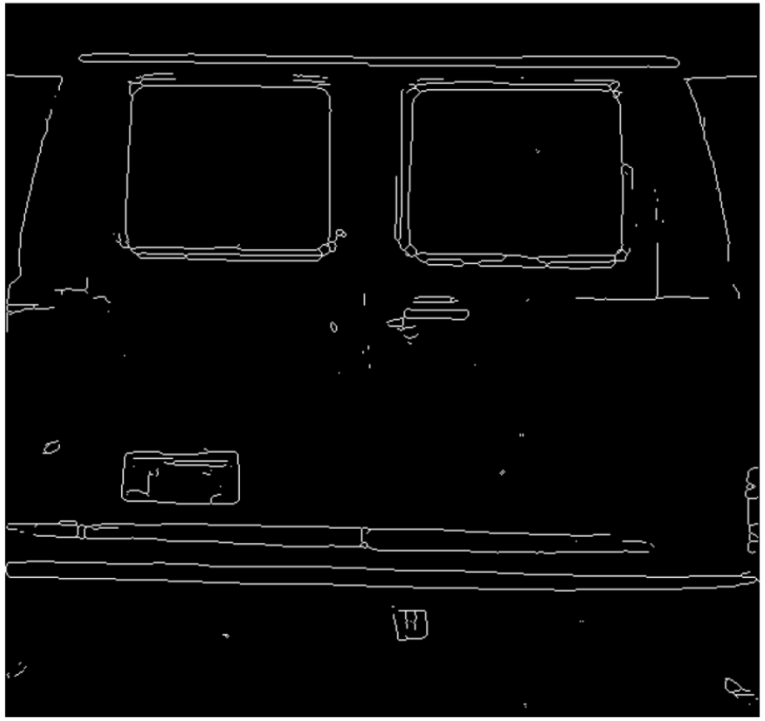
- 寻找车牌

采用比较大的夹角范围

美国车牌的长宽比是2:1



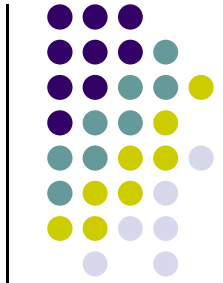
合并后的图像



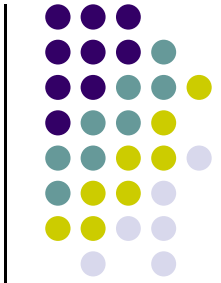
细化后的图像

提纲

- 边缘连接和边界检测
  - 局部处理
  - 区域处理
  - 全局处理



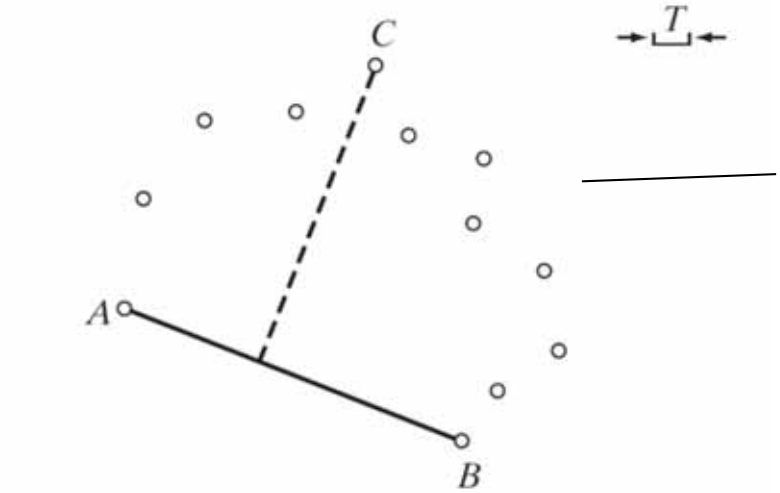
区域处理



- 前提
  - 感兴趣区域的位置已知
  - 预先知道属于边界的像素点
- 目标：基于区域连接像素，近似区域边界
- 方法
  - 函数近似
    - 为已知点拟合一条2维曲线
  - 多边形近似
    - 实现容易、捕捉基本形状特征、表示简单

举例

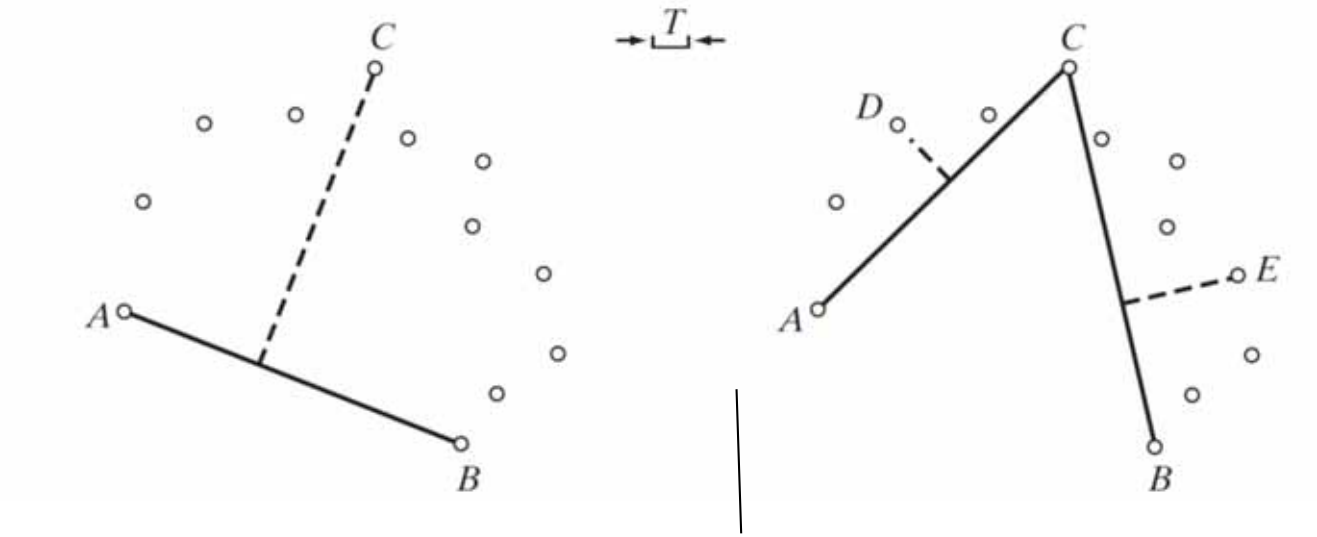
- 已知A、B是曲线的端点



1、用直线连接A、B
2、计算所有点离直线AB的距离
3、找到最远的点C
4、如果距离大于阈值T，把C当做一个顶点

举例

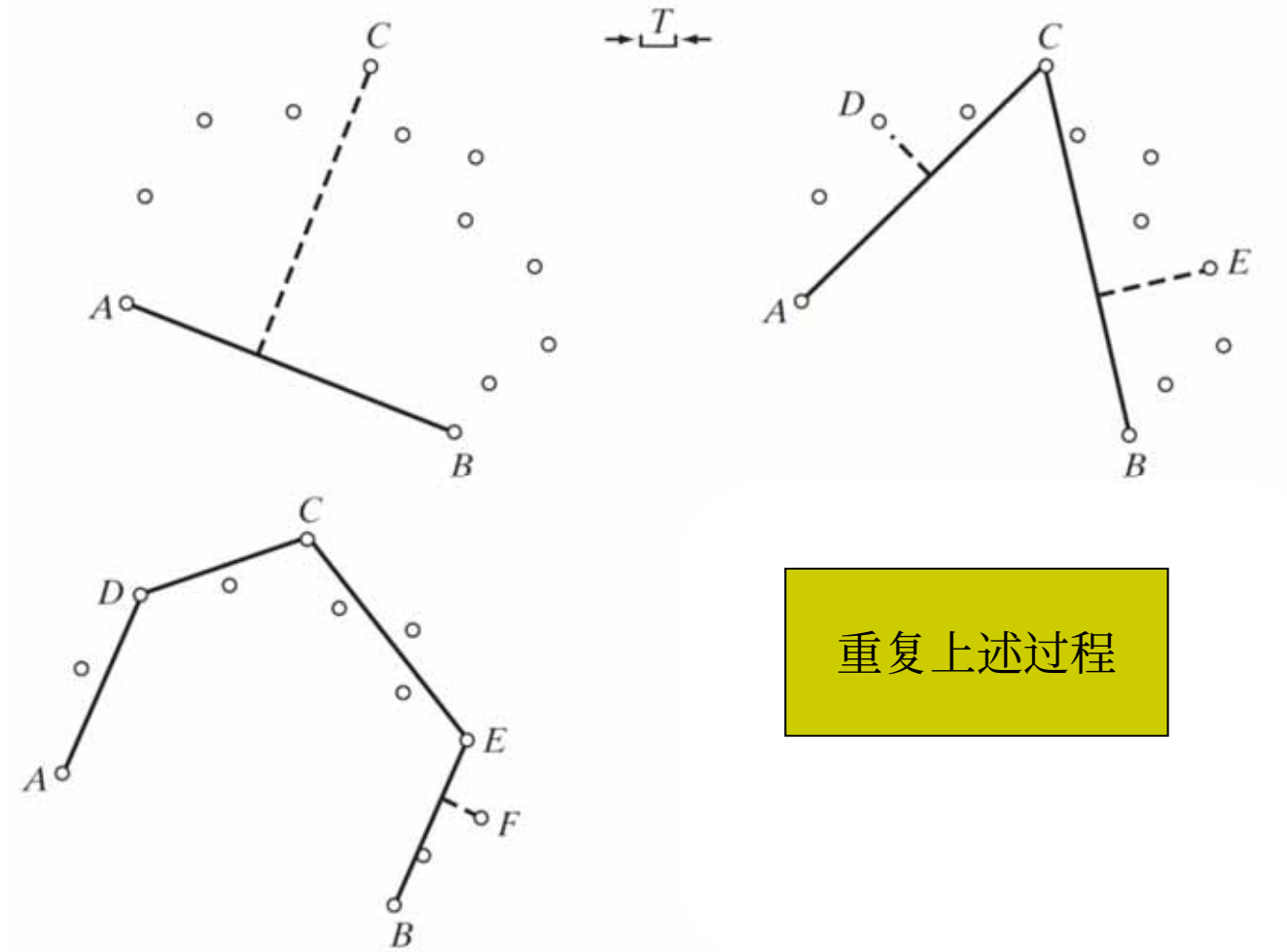
- 已知A、B是曲线的端点



1、用直线连接A、C
2、计算AC之间的点离直线AC的距离
3、找到最远的点D
4、如果距离大于阈值T，把D当做一个顶点

举例

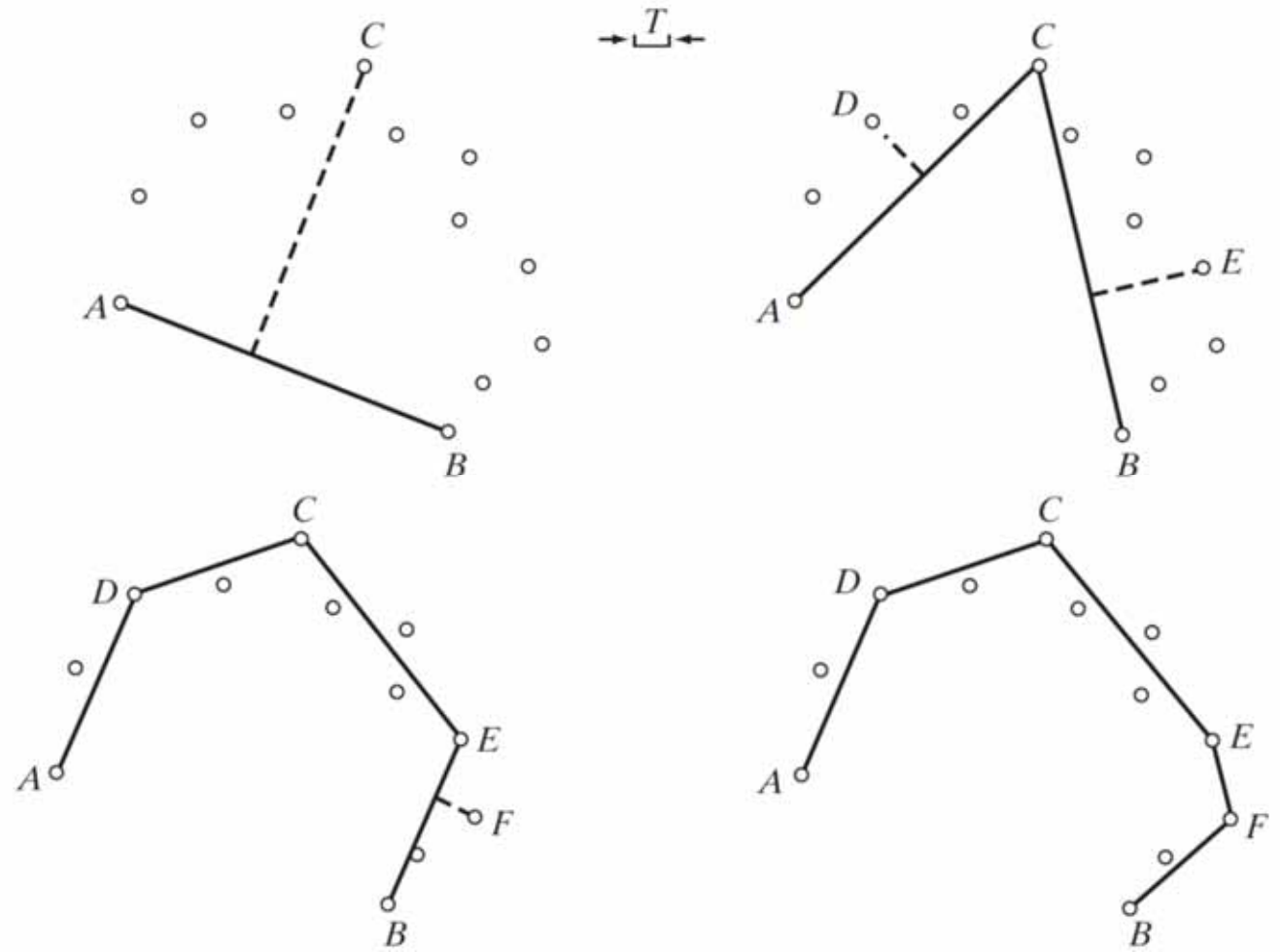
- 已知A、B是曲线的端点



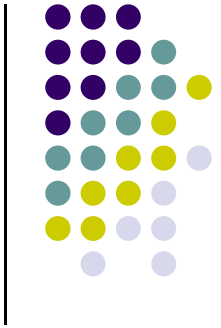
重复上述过程

举例

- 已知A、B是曲线的端点

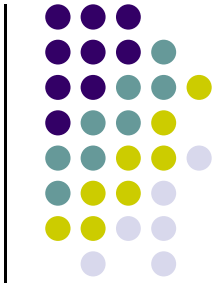


算法设计



- 前提
  - 两个起始点
  - 所有的点必须排序
    - 顺时针、逆时针
- 判断曲线类型
  - 边界线段（开放曲线）
    - 存在两个间距较大的连续点（可作为起始点）
  - 边界（闭合曲线）
    - 连续点之间的距离比较均匀
      - 两端的点为起始点

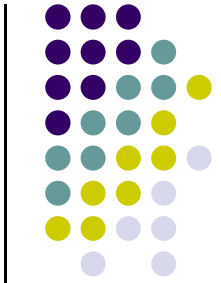
区域处理算法



1. 令P是一个已排序、不重复的二值图像中的序列。指定两个起始点A和B。它们是多边形的两个起始顶点。
2. 指定一个阈值T，以及两个空堆栈“开”(OPEN)和“闭”(CLOSED) 。
3. 如果P中的点对应于一条闭合曲线，则把B放到“开”和“闭”中，并把A放到“开” 中。

   如果对应于一条开放曲线，则把A放到“开”中，而把B放到“闭”。
4. 计算从“闭”中最后一个顶点到“开”中最后一个顶点的线的参数。

区域处理算法

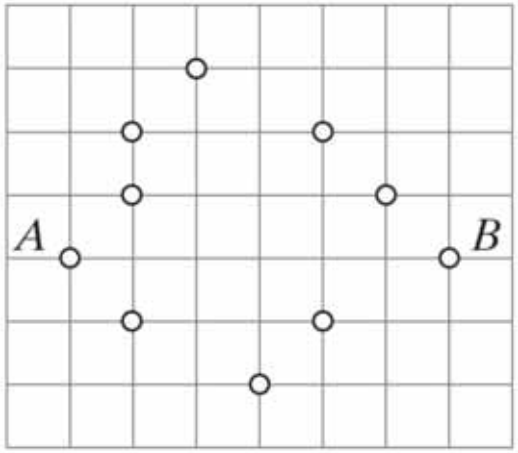


5. 寻找属于序列P、且在步骤4中直线的两个顶点之间的点；计算这些点与直线的距离，选择具有最大距离D_{max}的点V_{max}。
6. 如果$D_{max} > T$，则把V_{max}作为一个新顶点放在“开”堆栈的末尾。转到步骤4。
7. 否则，从“开”中移除最后一个顶点，并把它作为“闭”的最后一个顶点插入。
8. 如果“开”非空，转到步骤4。
9. 否则，退出。“闭”中的顶点就是拟合P中的点的多边形的顶点。

举例

- 闭合曲线
- 顺时针排序
- A、B为起点

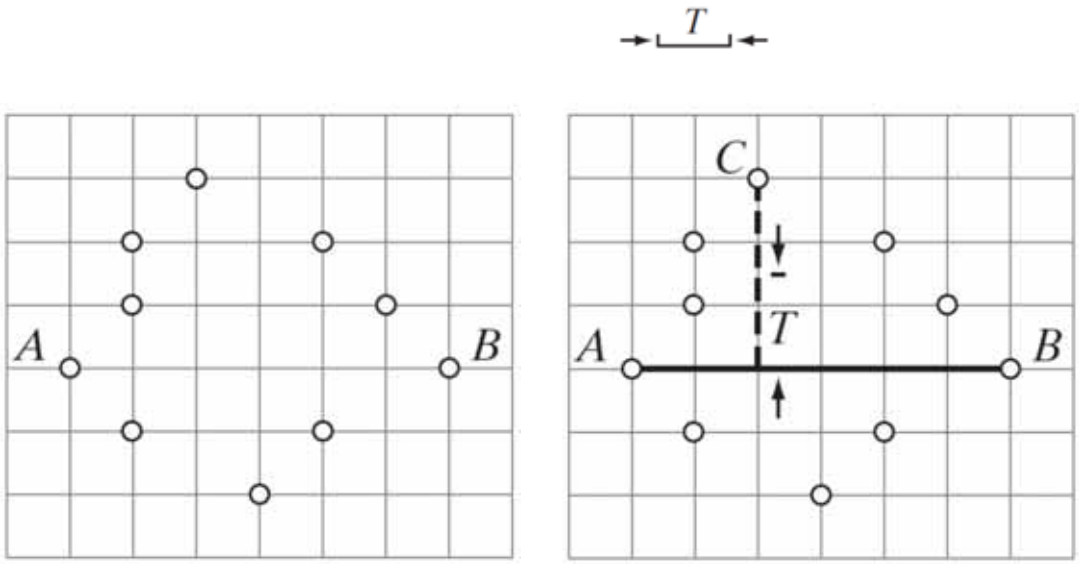
CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B



举例

- 闭合曲线
- 顺时针排序
- A、B为起点

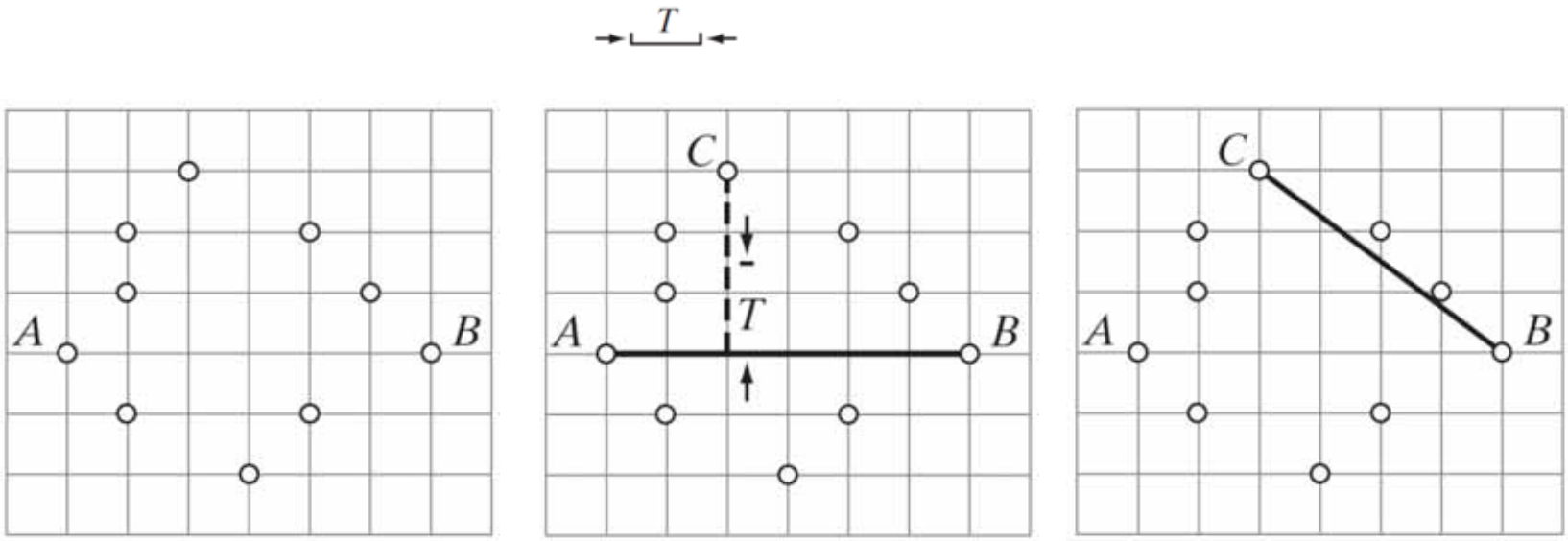
CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C



举例

- 闭合曲线
- 顺时针排序
- A、B为起点

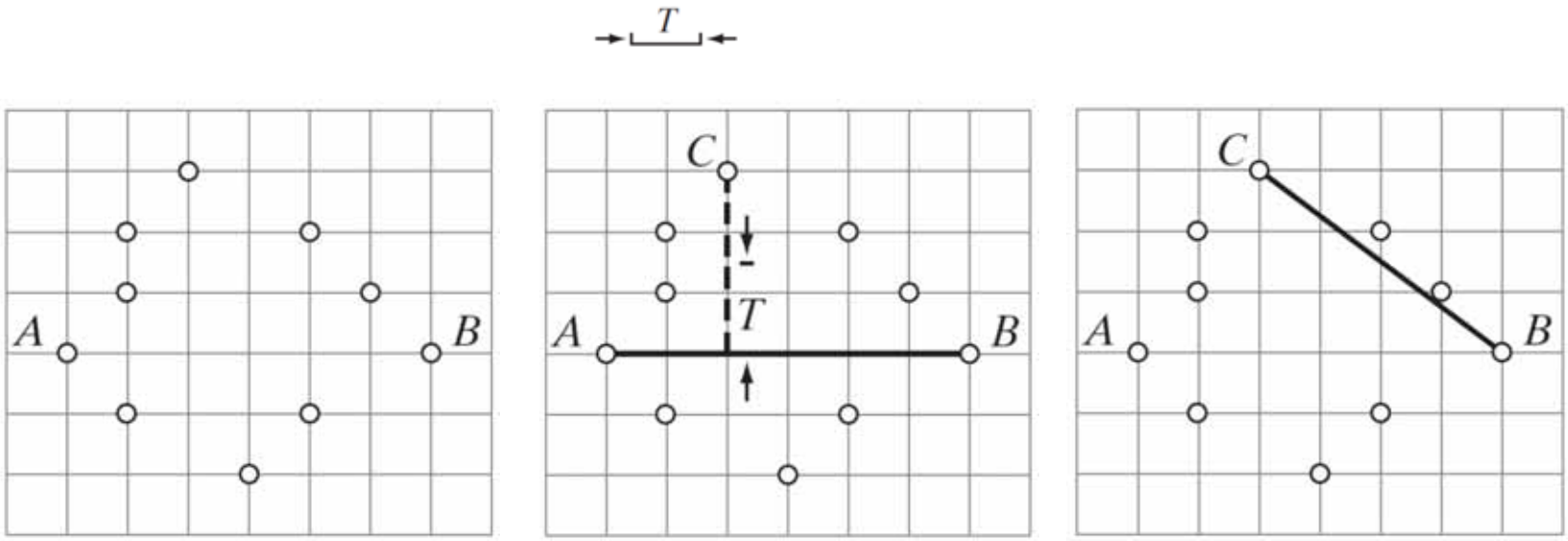
CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—



举例

- 闭合曲线
- 顺时针排序
- A、B为起点

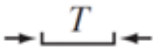
CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—

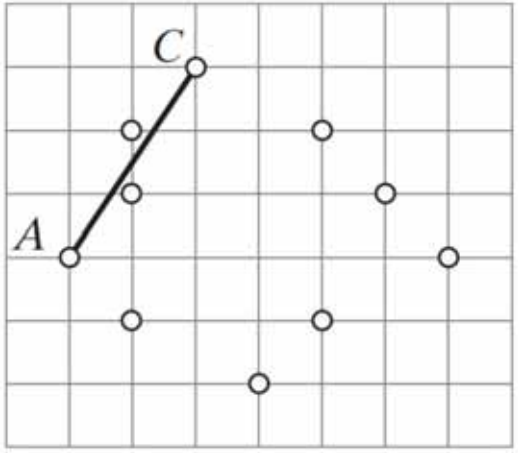


举例

- 闭合曲线
- 顺时针排序
- A、B为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—

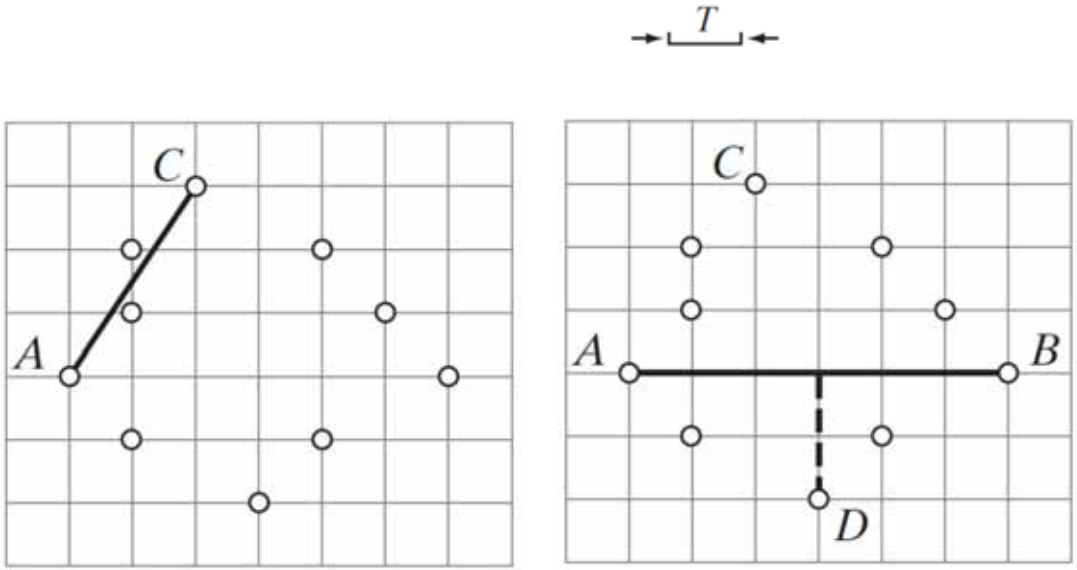




举例

- 闭合曲线
- 顺时针排序
- A、B为起点

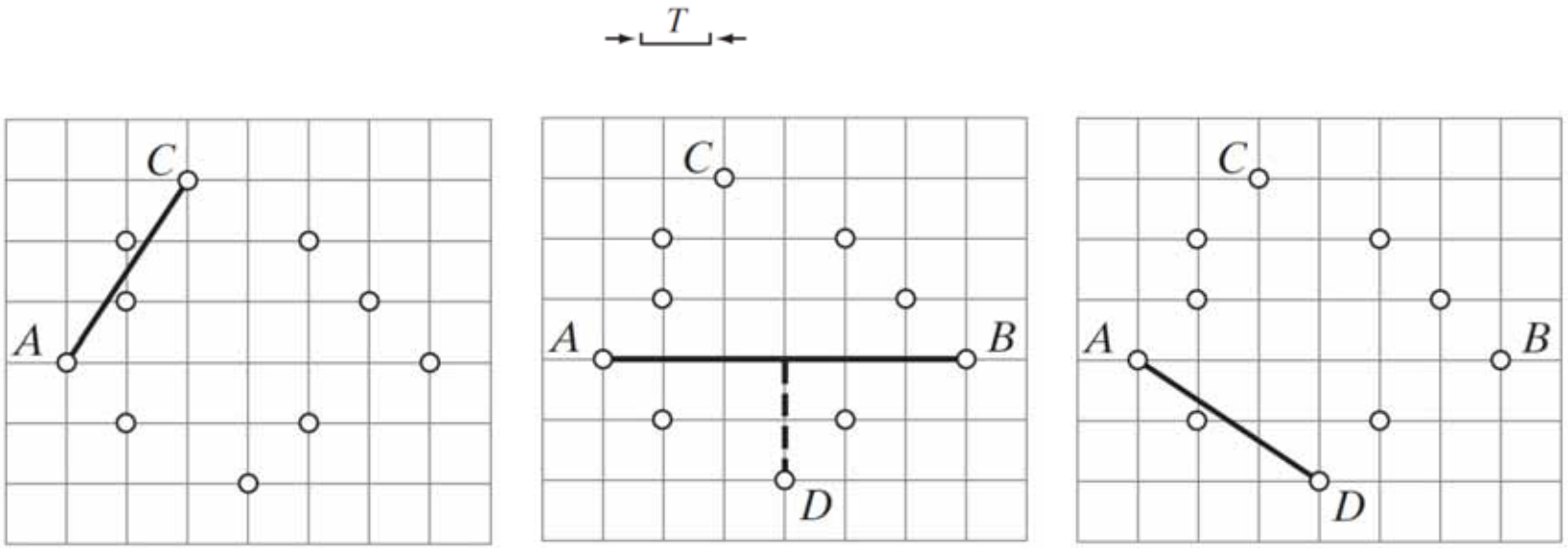
CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—
B, C, A	B	(AB)	D



举例

- 闭合曲线
- 顺时针排序
- A、B为起点

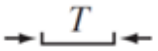
CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—
B, C, A	B	(AB)	D
B, C, A	B, D	(AD)	—

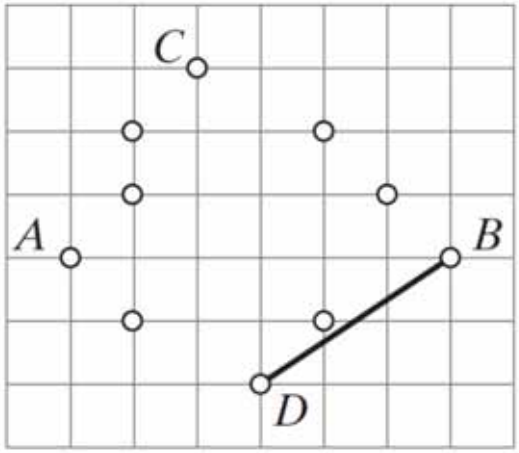


举例

- 闭合曲线
- 顺时针排序
- A、B为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—
B, C, A	B	(AB)	D
B, C, A	B, D	(AD)	—
B, C, A, D	B	(DB)	—





举例

- 闭合曲线
- 顺时针排序
- A、B为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—
B, C, A	B	(AB)	D
B, C, A	B, D	(AD)	—
B, C, A, D	B	(DB)	—
B, C, A, D, B	Empty	—	—