

计算机视觉——目标分割

2025年春季

桑农 王岳环



第6章 目标分割

- 什么是图像分割：按照一定的规则，把图像分成一些联通区域的组合
- 图像分割的基本策略，基于灰度（或其它特征）两个基本特性：
 - 不连续性——区域之间
 - 边界分割法（点、线、边）
 - 边缘连接分割法（hough变换）
 - 相似性——区域内部
 - 阈值分割法
 - 面向区域的分割
 - 数学形态学图像处理



第6章 目标分割

- 6.1 轮廓搜索
- 6.2 主动轮廓模型



引子

- 传统意义上的边缘主要定义亮度、颜色等低层图像特征突变，是一种低层视觉行为。
- 轮廓与边界定义了目标的外表形状，确定了区域之间的分界线，它们是人类与计算机进行目标识别的重要特征。
- 轮廓对于识别目标的重要性（目标定位、目标种类、目标方向等）





6.1 轮廓搜索

- **轮廓搜索**是一种基于轮廓的目标分割方法，将检测边缘点和连接边界结合进行，边检测边连接
- 从种子点开始，将图像中的边缘候选点根据物体轮廓线先验知识连接成轮廓
- 通过在图中搜索最小代价通路找到闭合边界
- 考虑了图像中全局信息，在图像噪声影响较大时仍可取得较鲁棒的分割结果



6.1 轮廓搜索

- **6.1.1 图搜索**
 - 把轮廓搜索转化为图结构中搜索代价最小的路径
- **6.1.2 动态规划**
 - 借助于启发性知识来减少搜索计算量



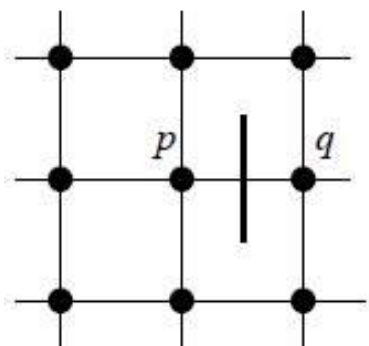
6.1.1 图搜索

- 一个图可表示为 $G = [N, A]$ ，其中 N 是一个有限非空的**结点集**， A 是一个无序结点对的集。集 A 中的每个结点对 (n_i, n_j) 称为一段**弧** ($n_i \in N, n_j \in N$)
- 有向图&无向图
- 子结点&父结点
- 确定一个结点的子结点的过程称为对该结点的展开

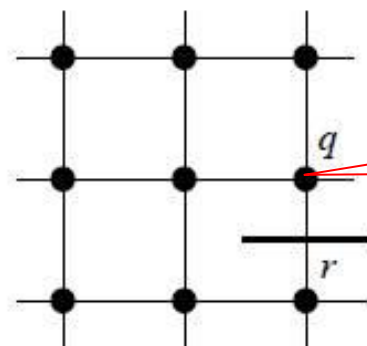


6.1.1 图搜索

- 对任一段弧 (n_i, n_j) 都可定义一个代价，记为 $c(n_i, n_j)$
- 通路 $n_1, n_2, n_i, \dots, n_k$ 的总代价为 $C = \sum_{i=2}^K c(n_{i-1}, n_i)$



(a)



像素

(b)

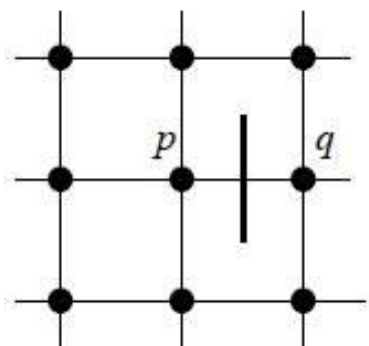
$$c(p, q) = H - [f(p) - f(q)]$$

- 其中H为图像中最大灰度值， $f(p)$ 代表p点的灰度值

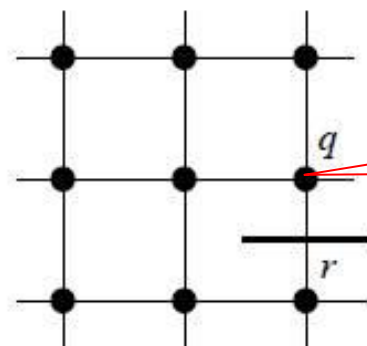


6.1.1 图搜索

- 对任一段弧 (n_i, n_j) 都可定义一个代价，记为 $c(n_i, n_j)$
- 通路 $n_1, n_2, n_i, \dots, n_k$ 的总代价为 $C = \sum_{i=2}^K c(n_{i-1}, n_i)$



(a)



(b)

$$c(p, q) = H - [f(p) - f(q)]$$

- 与灰度差成反比
- 与梯度大小成反比

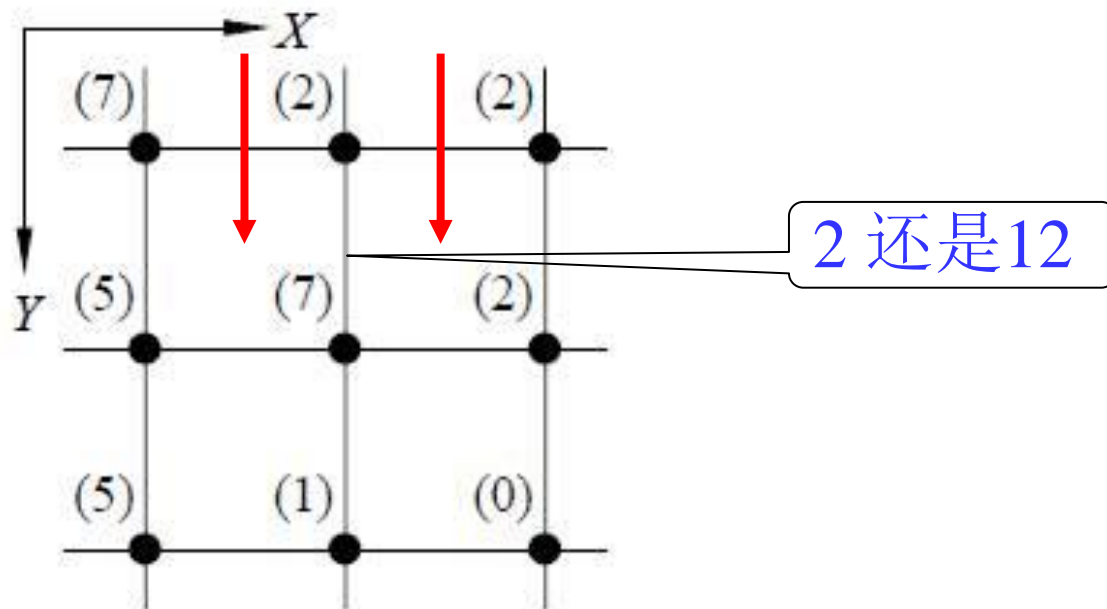


6.1.1 图搜索

$H \Rightarrow$

$$c(p, q) = H - [f(p) - f(q)]$$

- 思考：轮廓搜索中图结构是有向图还是无向图？

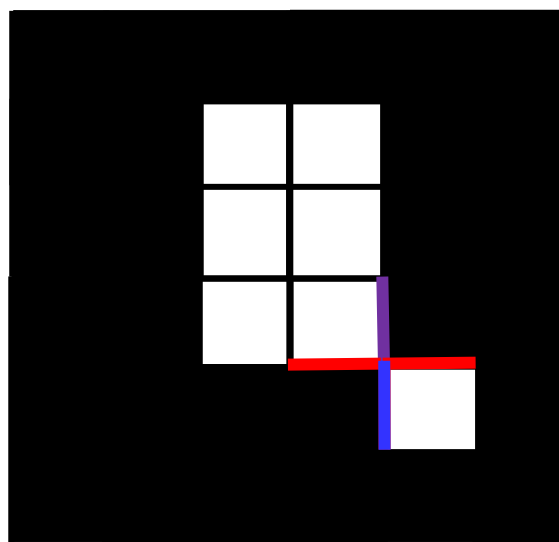
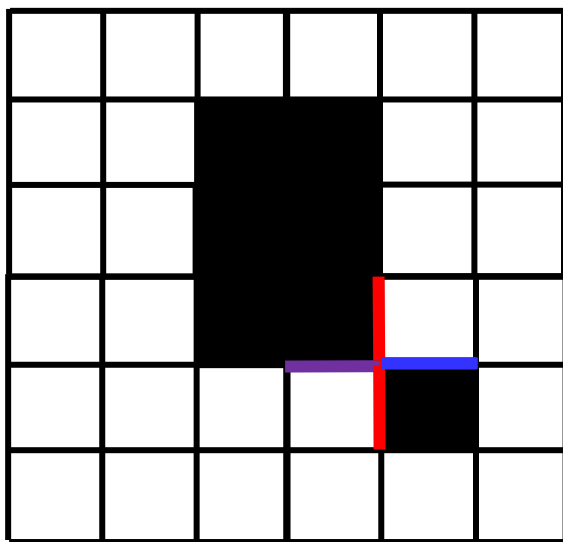




6.1.1 图搜索

$$c(p, q) = H - [f(p) - f(q)]$$

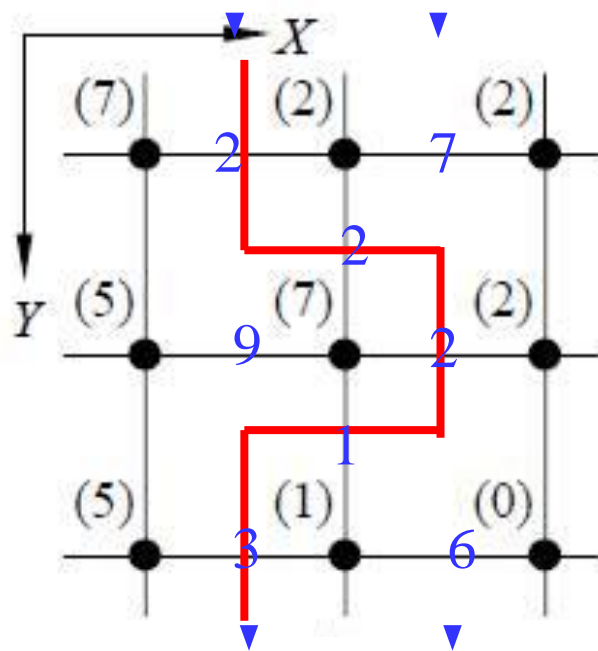
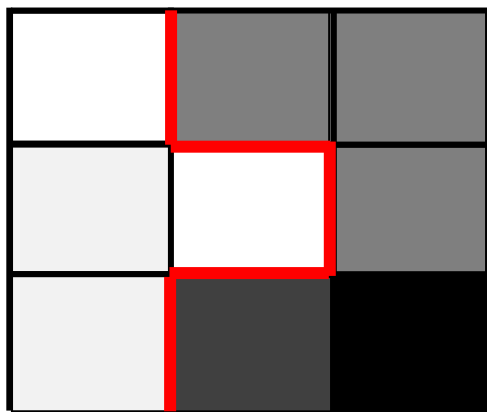
- 思考：轮廓搜索中图结构是有向图还是无向图？
- 定义：顺着边缘方向，p为右边点，q为左边点





6.1.1 图搜索

- 代价函数 $c(p, q) = H - [f(p) - f(q)]$
- 利用图搜索技术从上向下可检测出如图所示的对应大梯度边缘元素的边界段





6.1.1 图搜索

$$c(p, q) = H - [f(p) - f(q)]$$

定义：顺着边缘方向，
p为右边点，q为左边点

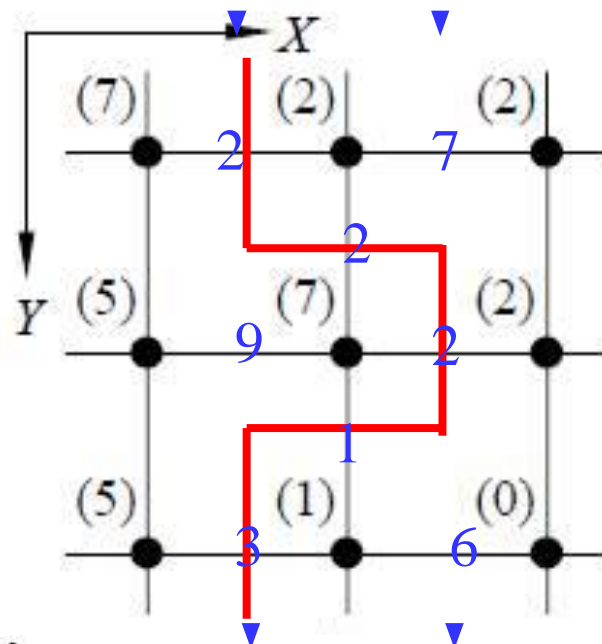
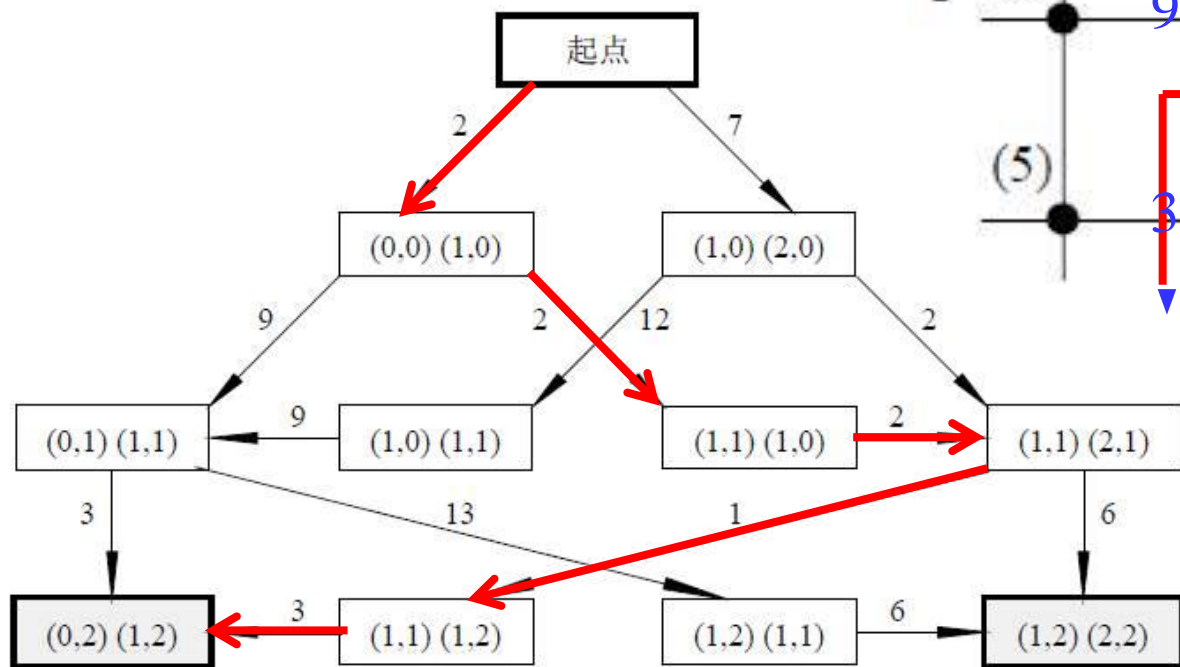
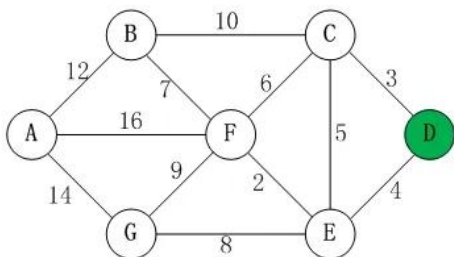


图 6.1.3 用于检测边界的搜索图



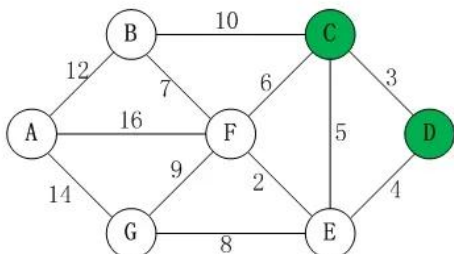
迪杰斯特拉(Dijkstra)算法



第1步:
选取顶点D

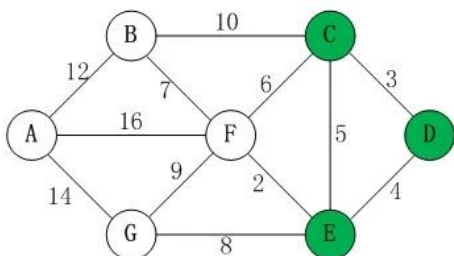
$S = \{D(0)\}$
 $U = \{A(\infty), B(\infty), C(3), E(4), F(\infty), G(\infty)\}$

注:
(01) S 是已计算出最短路径的定点的集合
(02) U 是未计算出最短路径的定点的集合
(03) $C(3)$ 表示顶点C到起点D的最短距离是3



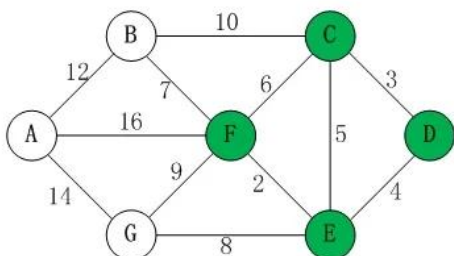
第2步:
选取顶点C

$S = \{D(0), C(3)\}$
 $U = \{A(\infty), B(13), E(4), F(9), G(\infty)\}$



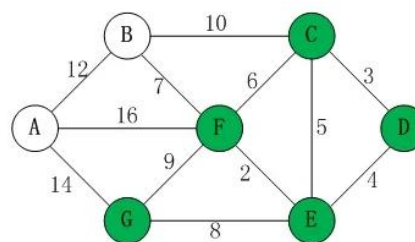
第3步:
选取顶点E

$S = \{D(0), C(3), E(4)\}$
 $U = \{A(\infty), B(13), F(6), G(12)\}$



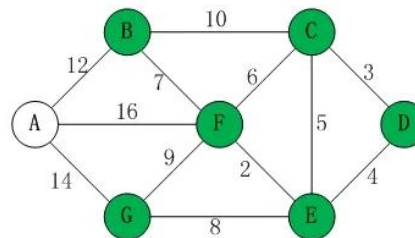
第4步:
选取顶点F

$S = \{D(0), C(3), E(4), F(6)\}$
 $U = \{A(22), B(13), G(12)\}$



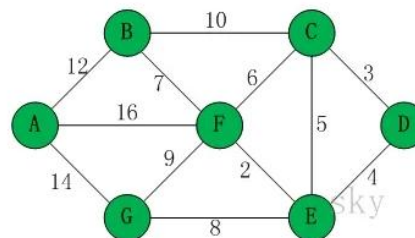
第5步:
选取顶点G

$S = \{D(0), C(3), E(4), F(6), G(12)\}$
 $U = \{A(22), B(13)\}$



第6步:
选取顶点B

$S = \{D(0), C(3), E(4), F(6), G(12), B(13)\}$
 $U = \{A(22)\}$



第7步:
选取顶点A

$S = \{D(0), C(3), E(4), F(6), G(12), B(13), A(22)\}$

<https://blog.csdn.net/q826364410>



迪杰斯特拉(Dijkstra)算法

1. 通过Dijkstra计算图G中的最短路径时，需要指定一个起点D(即从顶点D开始计算)。
2. 此外，引进两个数组S和U。S的作用是记录已求出最短路径的顶点(以及相应的最短路径长度)，而U则是记录还未求出最短路径的顶点(以及该顶点到起点D的距离)。
3. 初始时，数组S中只有起点D；数组U中是除起点D之外的顶点，并且数组U中记录各顶点到起点D的距离。如果顶点与起点D不相邻，距离为无穷大。
4. 然后，从数组U中找出路径最短的顶点K，并将其加入到数组S中；同时，从数组U中移除顶点K。接着，更新数组U中的各顶点到起点D的距离。
5. 重复第4步操作，直到遍历完所有顶点。



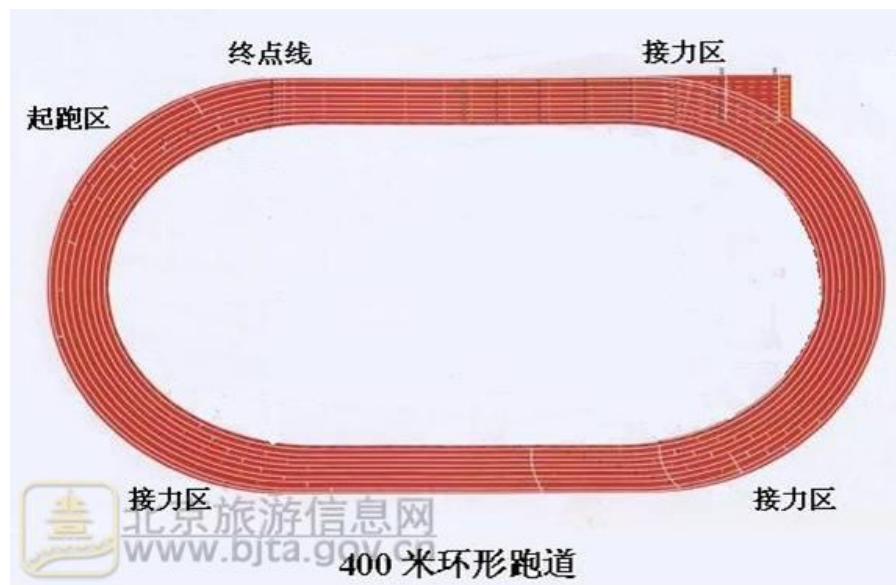
6.1.2 动态规划

- 基于图搜索的轮廓提取，为求得最小代价所需的计算量是很大的。因为对结点的扩展次序是任意的。
- 寻找次最优（贪婪算法）



6.1.2 动态规划

- 动态规划:
- $r(n)$: $s \rightarrow n \rightarrow e$ 的最小代价通路的估计代价
- $g(n)$: $s \rightarrow n$ $h(n)$: $n \rightarrow e$
- $r(n) = g(n) + h(n)$





6.1.2 动态规划

- 动态规划:
- $r(n) = g(n) + h(n)$
- $g(n)$: 用图搜索方法
- $h(n)$: 根据某些启发性知识得到



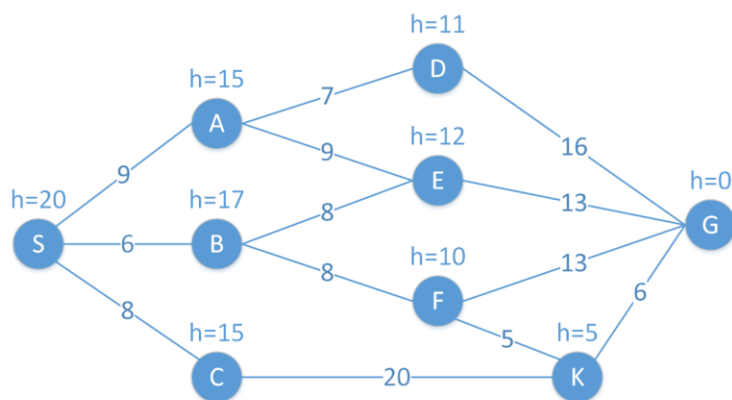
6.1.2 动态规划 (A*算法)

- 借助有关具体问题的启发性知识减少搜索
- 图搜索的算法由以下几个步骤构成
 - (1) 将起始结点标记为OPEN并置 $g(s) = 0$
 - (2) 如果没有结点OPEN, 失败退出, 否则继续
 - (3) 将根据式 $r(n) = g(n) + h(n)$ 算得的估计代价 $r(n)$ 为最小的OPEN结点标记为CLOSE
 - (4) 如果 n 是目标结点, 找到通路 (可由 n 借助指针上溯至 s) 退出, 否则继续



6.1.2 动态规划

- (5) 展开结点 n ，得到它的所有子结点（如果没有子结点，返回步骤(2)）
- (6) 如果某个子结点 n_i 还没有标记，置 $g(n_i) = g(n) + c(n, n_i)$ ，标记它为OPEN并将指向它的指针返回到结点 n
- (7) 如果子结点 n_i 已标记为OPEN或CLOSE，根据 $g'(n_i) = \min[g(n_i), g(n) + c(n, n_i)]$ 更新它的值。将其 g' 值减小的CLOSE子结点标记为OPEN，并将原来指向所有其 g' 值减小的子结点的指针重新指向 n 。返回步骤(2)

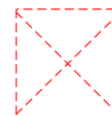


S	0
S	20
S	20

前序	g
当前	h
	f

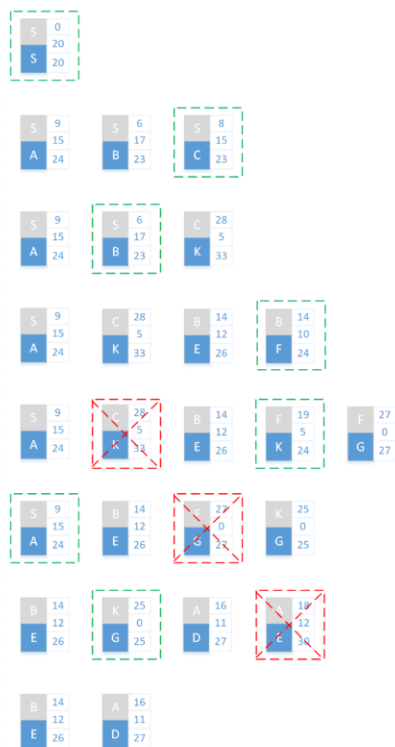


最优值

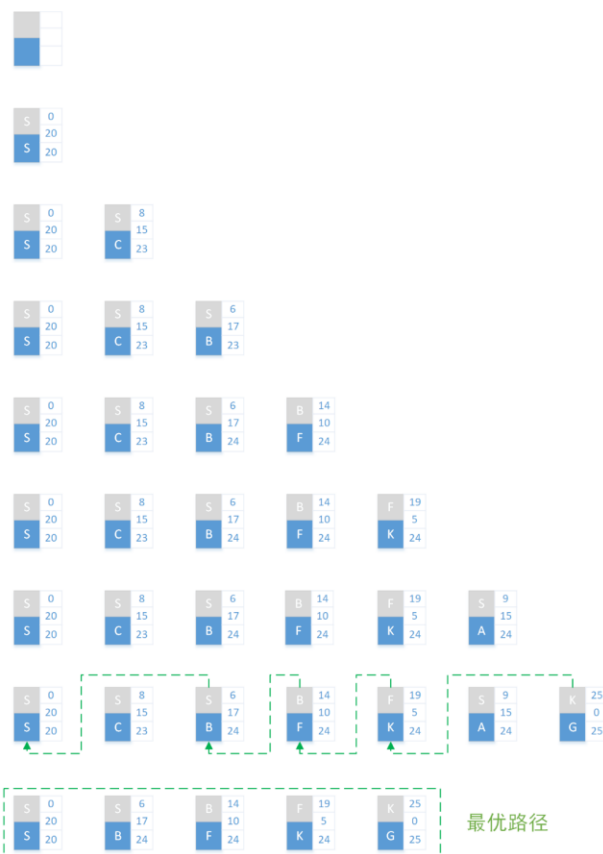


被去除

OPEN



CLOSED



最优路径



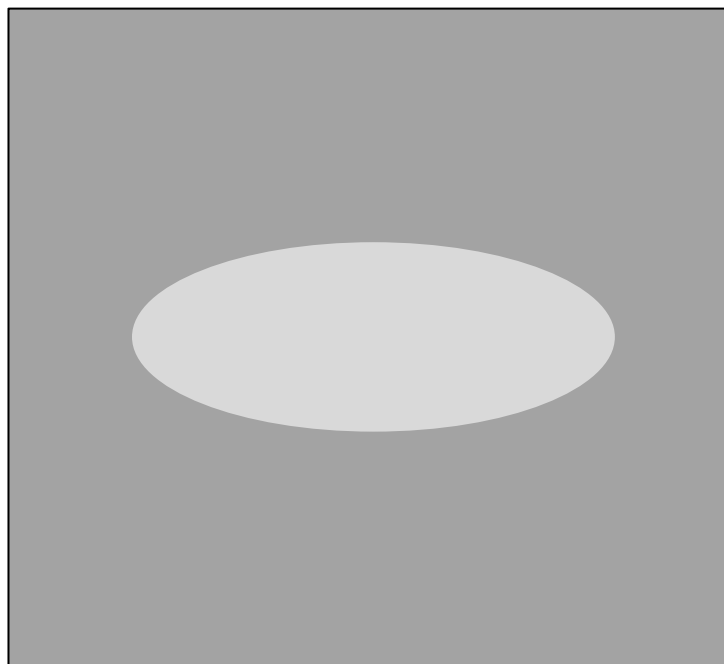
6.1 轮廓搜索

- 与边缘检测技术比较：
 - 边缘检测等分割属于并行边界技术
 - 图搜索、动态规划属于串行边界技术
 - 轮廓搜索实际上将边缘点的检测融入代价函数的计算，从而把边缘检测和边界连接结合起来



封闭轮廓搜索

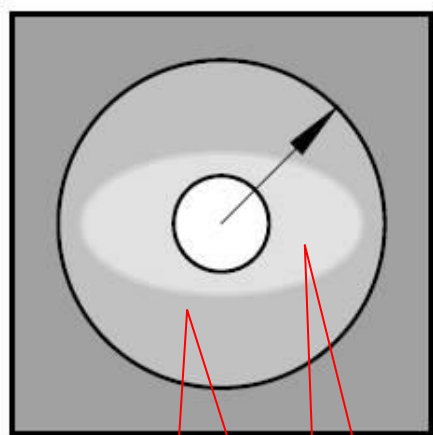
- 轮廓搜索可用于起始点和终点确定的情况
- 实际图像中轮廓搜索是一个封闭区域





封闭轮廓搜索

- 对图像进行极坐标变换
- 同时解决确定起始点和判断搜索是否结束这两个问题



(a)

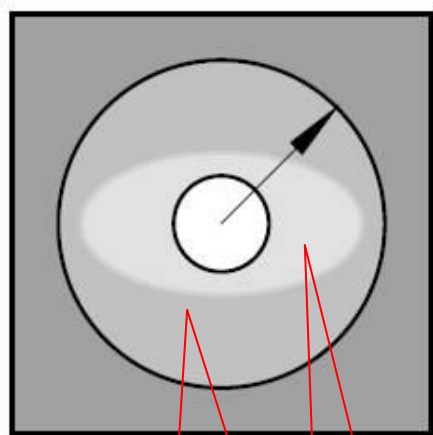
ROI(环形)

目标区域



封闭轮廓搜索

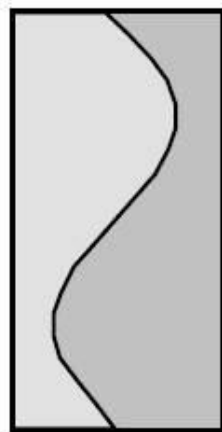
- 对图像进行极坐标变换
- 同时解决确定起始点和判断搜索是否结束这两个问题



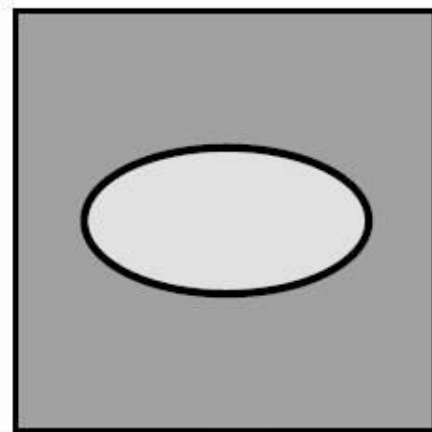
(a)



(b)



(c)



(d)

ROI(环形)

目标区域



6.2 主动轮廓模型

- 由底层数据，自下而上驱动的目标分割方法中，底层的错误将被带给高层无法修正
- **Snakes: active contour models**
 - Kass, 1987, ICCV
 - 对传统的视觉计算理论的挑战
 - 设计这样一个能量函数: 其局部极值组成了高层视觉处理可能的解, 通过加入合适的能量项做能量最小化不落入局部极值而得到需要的解



6.2 主动轮廓模型

- 6.2.1 主动轮廓
- 6.2.2 能量函数
 - 主动轮廓是一种基于边缘信息的目标分割方法
 - 思想：初步的封闭曲线逼近
 - 主动轮廓模型也称蛇模型

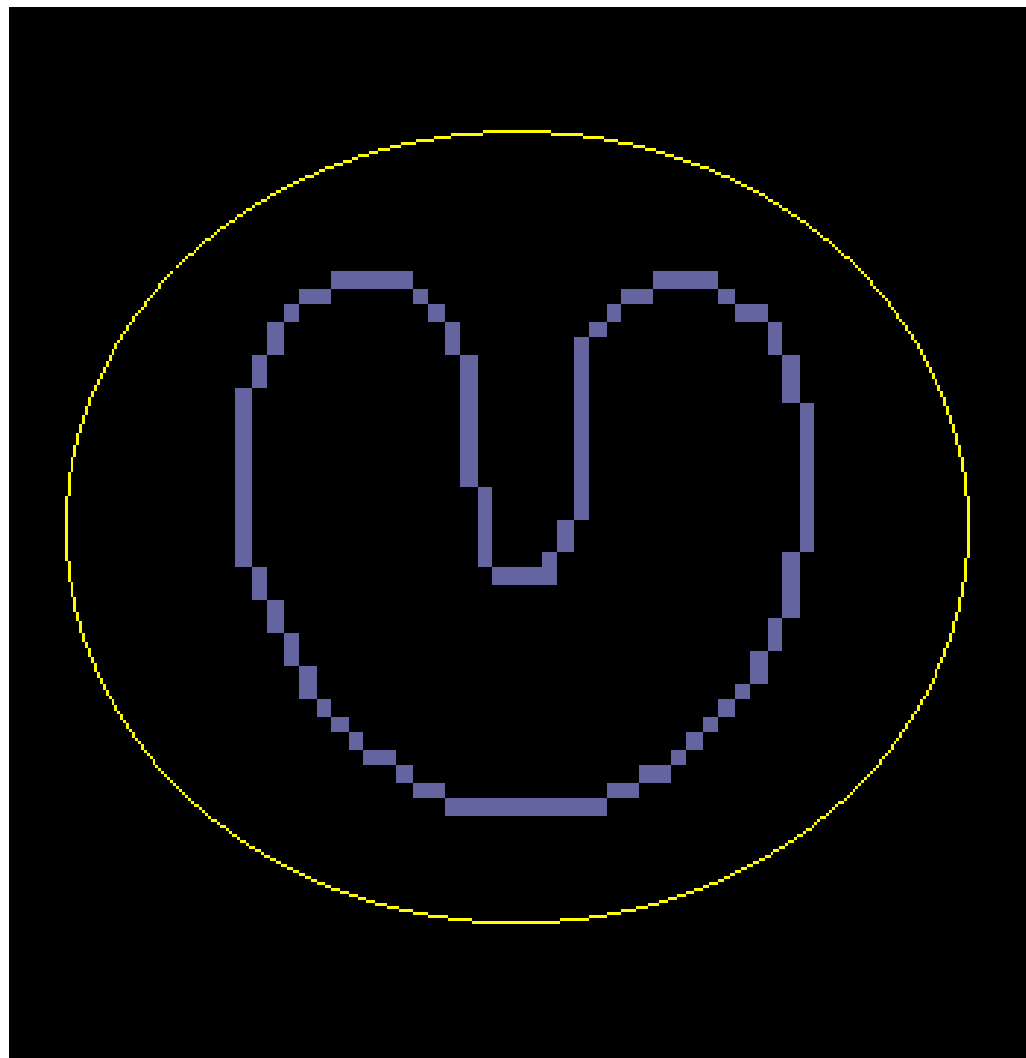


6.2 主动轮廓模型

- 基本原理是表征拟合误差的“能量”为最小化的曲线。
- 设对于拟合目标有一个待选曲线集,定义能量函数与待选集中每一条曲线相关联,能量函数的设计原则就是:有利属性要能导致能量缩小。
- 有利属性包括:曲线连续、平滑、曲线与高梯度区域接近以及其他一些具体的先验知识。
- 活动轮廓在取值范围内移动时,就能在能量函数的指导下收敛到局部边界,且能保持曲线的连续和平滑。



6.2 主动轮廓模型





6.2 主动轮廓模型

图像上一组排序的点的集合

$$V = \{v_1, \dots, v_L\}$$

$$v_i = (x_i, y_i), \quad i = \{1, \dots, L\}$$

$$E_i(v_i') = \alpha E_{\text{int}}(v_i') + \beta E_{\text{ext}}(v_i')$$

内部能量
函数

外部能量
函数

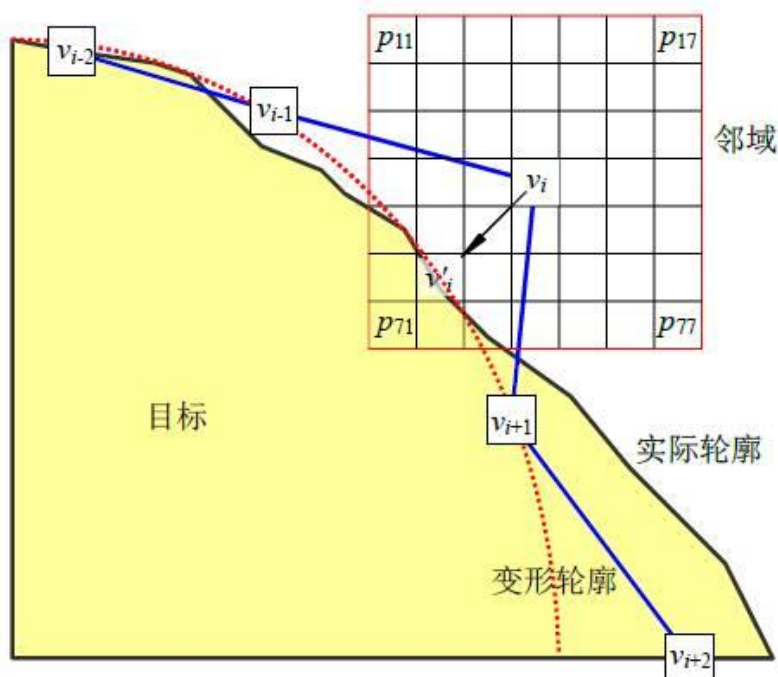


图 6.2.1 主动轮廓上点的移动



6.2 主动轮廓模型

1. 内部能量函数

$$E_{int} = 1/2 (\alpha | v'(s) |^2 + \beta | v''(s) |^2)$$

(1) 弹性势能

(2) 弯曲势能

$v'(s)$ 和 $v''(s)$: 是 v 对 s 的一阶和二阶导数, 系数 α 、 β 分别是控制蛇模型的弹性和刚性, 这些参数操纵着模型的物理行为和局部连续性



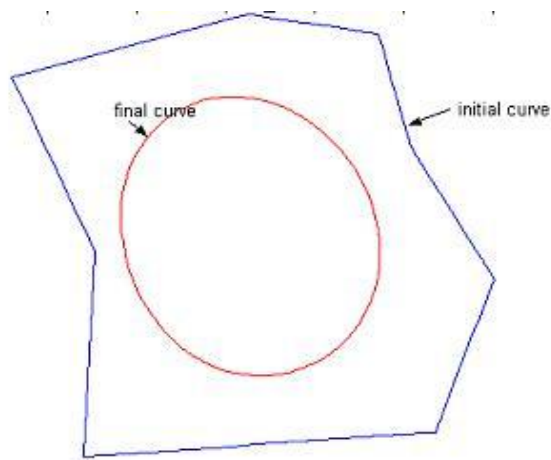
6.2 主动轮廓模型

1. 内部能量函数

弹性力：使轮廓长度减小

一阶导数 $v'(s)$ ，反映曲线的弹性

若轮廓上不存在其它力，则弹性力使曲线收缩为一个点





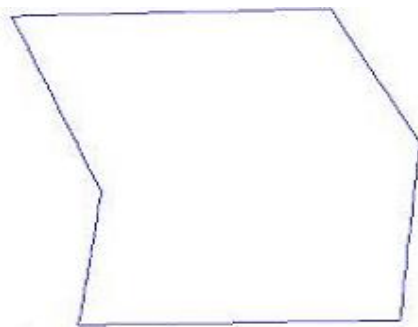
6.2 主动轮廓模型

1. 内部能量函数

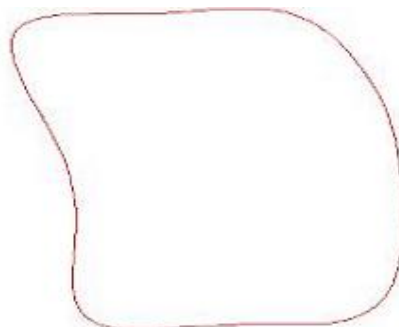
弯曲力：使轮廓平滑

二阶导数 $v''(s)$ ，反映曲线的刚性

弯曲力使曲线逐渐趋于平滑。若曲线上不存在其它力，则弯曲力使曲线最终成为一个圆



初始曲线



在弯曲力下变形后得到的曲线



6.2 主动轮廓模型

1. 内部能量函数

连续能量

$$E_{con}(v'_i) = \frac{1}{I(V)} ||v'_i - \gamma(v_{i-1} + v_{i+1})||^2$$

膨胀能量

$$E_{bal}(v'_i) = n_i \cdot [v_i - v'_i]$$



6.2 主动轮廓模型

2. 外部能量函数

将变形模板向感兴趣的特征位置吸引。一般根据特定的问题而定义

$$\beta E_{\text{ext}}(v_i) = mE_{\text{mag}}(v_i) + gE_{\text{grad}}(v_i)$$

(1) 图像灰度能量

$$E_{\text{mag}}(v_i') = I(v_i')$$

(2) 图像梯度能量

$$E_{\text{grad}}(v_i') = -n_i \cdot \nabla I(v_i')$$



6.2 主动轮廓模型

2. 外部能量函数

$$\beta E_{\text{ext}}(v_i) = mE_{\text{mag}}(v_i) + gE_{\text{grad}}(v_i)$$

(1) 图像灰度能量

$$E_{\text{mag}}(v_i') = I(v_i')$$

将轮廓吸向高或低的灰度区域;

m 为正, 轮廓向低灰度区域移动



6.2 主动轮廓模型

2. 外部能量函数

$$\beta E_{\text{ext}}(v_i) = mE_{\text{mag}}(v_i) + gE_{\text{grad}}(v_i)$$

(2) 图像梯度能量

$$E_{\text{grad}}(v_i') = -n_i \cdot \nabla I(v_i')$$

图像中梯度

将变形轮廓向图像边缘吸引

目标边界处法向量应和梯度方向接近

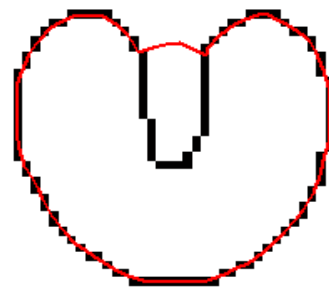


6.2 主动轮廓模型

- Snake模型的特点
 - Snake模型的优点
 - 图像数据、初始估计、目标轮廓及基于知识的约束统一于一个特征提取过程中;
 - 经适当地初始化后, 它能够自主地收敛于能量极小值状态;



6.2 主动轮廓模型



- Snake模型的特点
 - Snake模型的缺点
 - 对初始位置敏感, 需要依赖其他机制将Snake放置在感兴趣的图像特征附近;
 - 它有可能收敛到局部极值点, 甚至发散.
 - 对于边界上的凹点无法有效跟踪 (图像边缘捕获域问题。尺度空间中由粗到精地极小化能量可以极大地扩展捕获区域和降低计算复杂性)



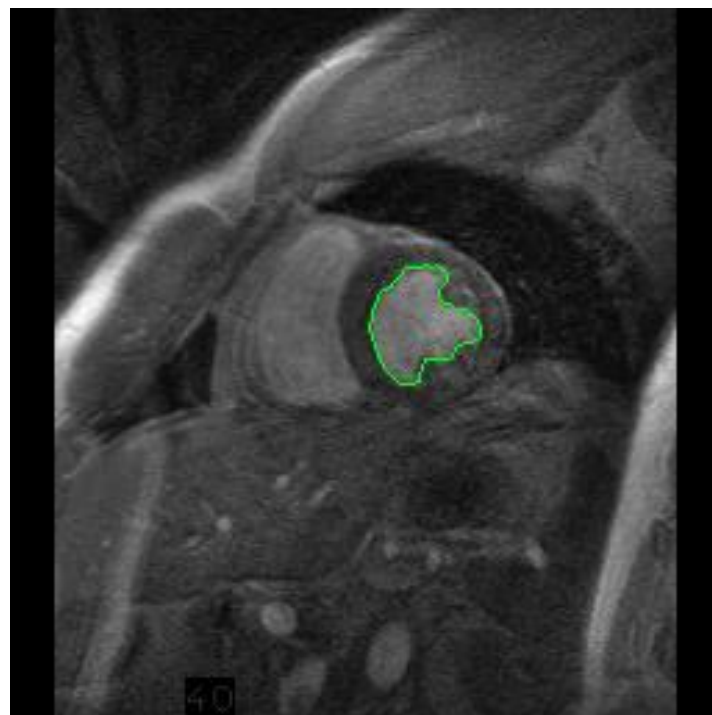
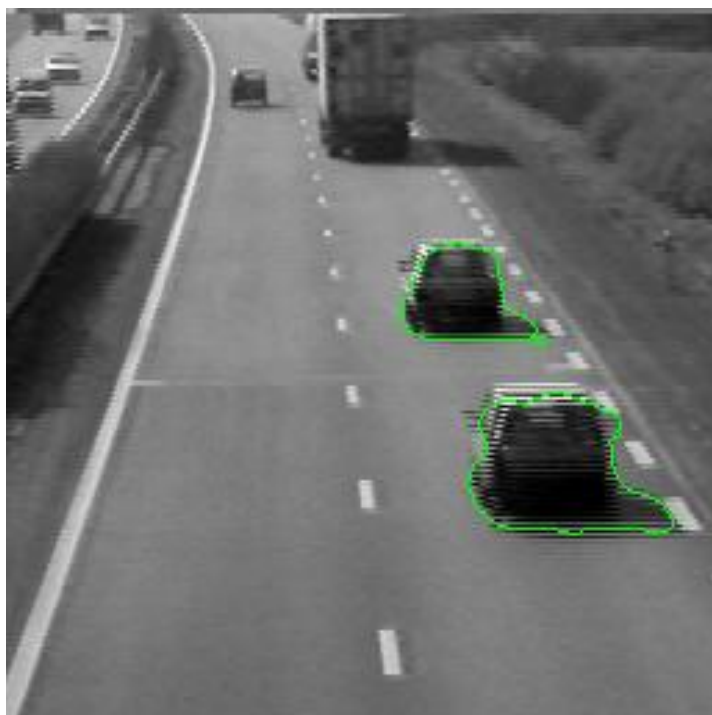
6.2 主动轮廓模型应用

- 轮廓跟踪
 - 目标检测识别
 - 使目标提取从一开始就处于受控的状态,并且直接面向所希望提取的目标。同时还给出了对目标轮廓形状的约束,从而可以快速准确的对目标进行定位.
 - 三维重建
 - 三维表面用序列图像中的轮廓曲线来表达和重建.
 - 立体匹配
 - 如果两幅图像中的轮廓线(基元) 互相对应, 那么沿着3D 轮廓的其他几何形状应具有相似性.



6.2 主动轮廓模型应用

- 运动跟踪





6.2 主动轮廓模型应用

- 运动跟踪





总结

- 主动轮廓模型（snake）回顾
- 内部能量 $E_{int} = 1/2 (\alpha | \underline{v'(s)} |^2 + \beta | \underline{v''(s)} |^2)$
 - snake模型的弹性和刚性
 - 弹性势能
 - 一阶导数 $v'(s)$ ，反映曲线的弹性
 - 作用：使轮廓收缩
 - 弯曲势能
 - 二阶导数 $v''(s)$ ，反映曲线的刚性
 - 作用：使轮廓形状平滑
- 外部能量



总结

- 主动轮廓模型（snake）回顾
- 内部能量 $E_{int} = 1/2 (\alpha | \underline{v'(s)} |^2 + \beta | \underline{v''(s)} |^2)$
- 外部能量
 - 外部能量 E_{ext} 决定着向某种固定的特征移动蛇模型,吸引蛇模型到显著的图像特征。因为这些特征只能根据特定的问题而定义,所以一般的外部能量函数不易确定。因此, E_{ext} 没有统一的数学表达式,必须从问题本身的特性出发,根据实际情况处理



作业

6.1 将左图中指出的所有边缘元素都标在右图所示的图像上，并计算最小代价通路的代价。

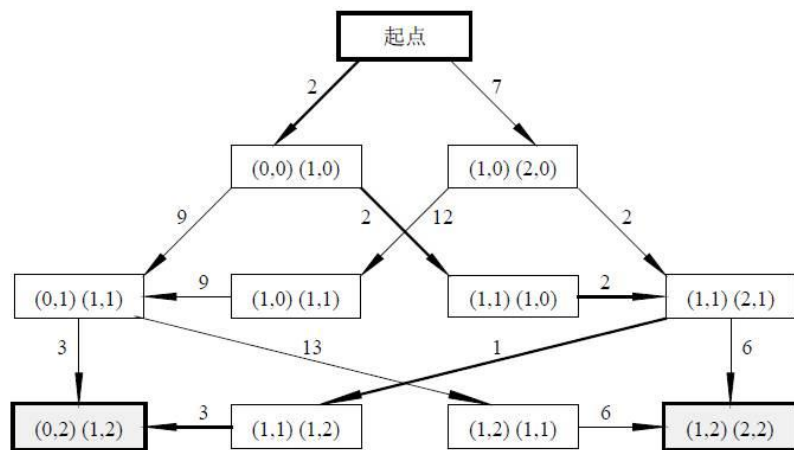
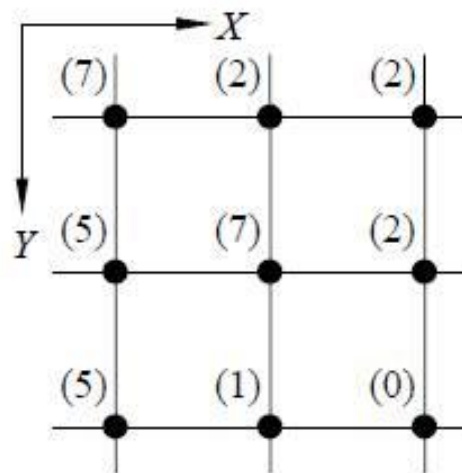


图 6.1.3 用于检测边界的搜索图



6.5 试讨论主动轮廓内部能量函数中加权系数 c 和 b 的作用，以及取值不同时（太大或者太小）对最终轮廓形状的影响

$$\alpha E_{\text{int}}(v_i) = cE_{\text{con}}(v_i) + bE_{\text{bal}}(v_i)$$



The end !