

机器视觉技术

孙明竹

sunmz@nankai.edu.cn



机器人与信息自动化研究所

Institute of Robotics & Automatic Information System



南开大学

Nankai University

第二部分 图像处理基础

Image Processing

课程目标

第一层次：编程

理解机器视觉/计算机视觉的概念
掌握视觉程序设计方法

第二层次：基本算法

掌握视觉算法细节
通过算法改进完成视觉应用

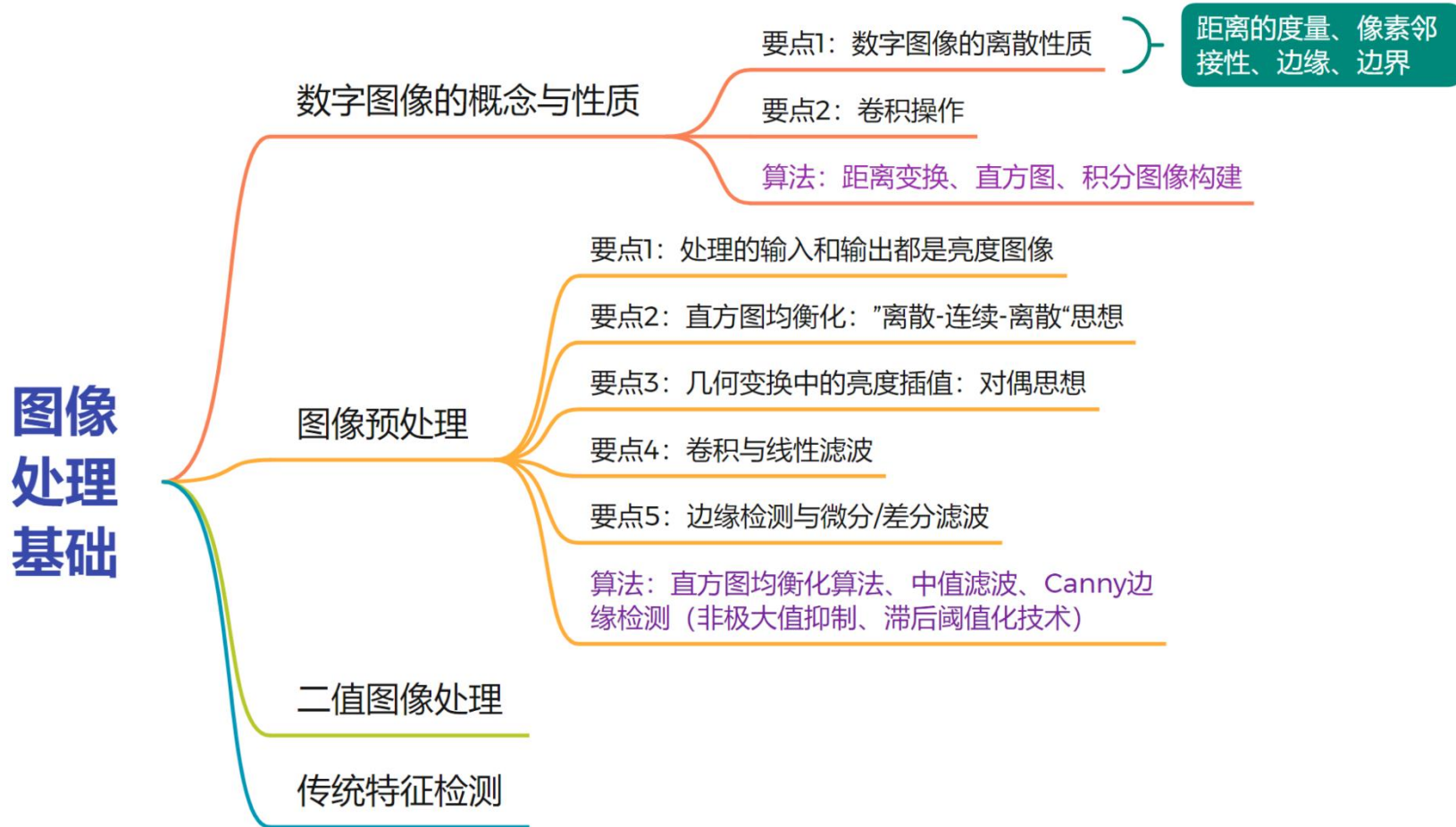
第三层次：视觉核心

通过数学物理方法、结合机器学习算法解决视觉问题

成绩分配

	理论课(3学分)		实验课(1学分)	合计
理论部分	期末考试60分中的80%	平时作业40分中的50%		68*3
编程部分	期末考试60分中的20%	平时作业40分中的50%	编程实践100分	32*3+100

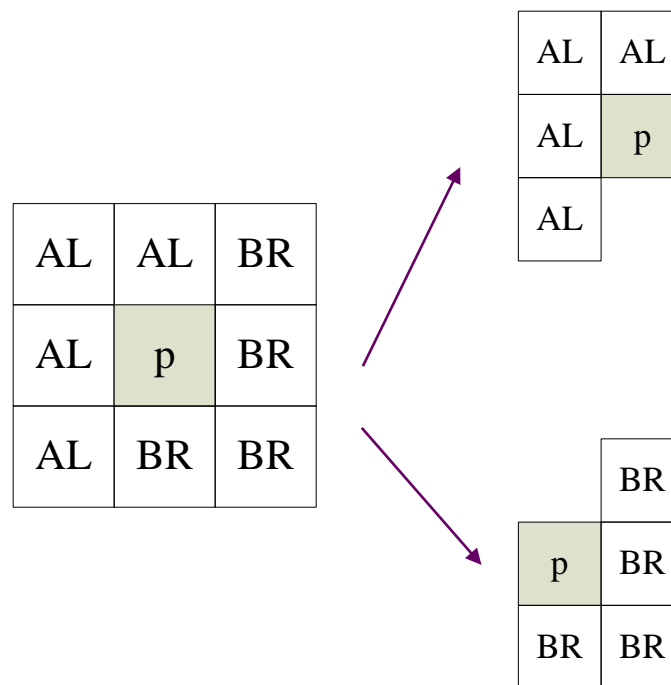
内容总结1



1) 距离变换

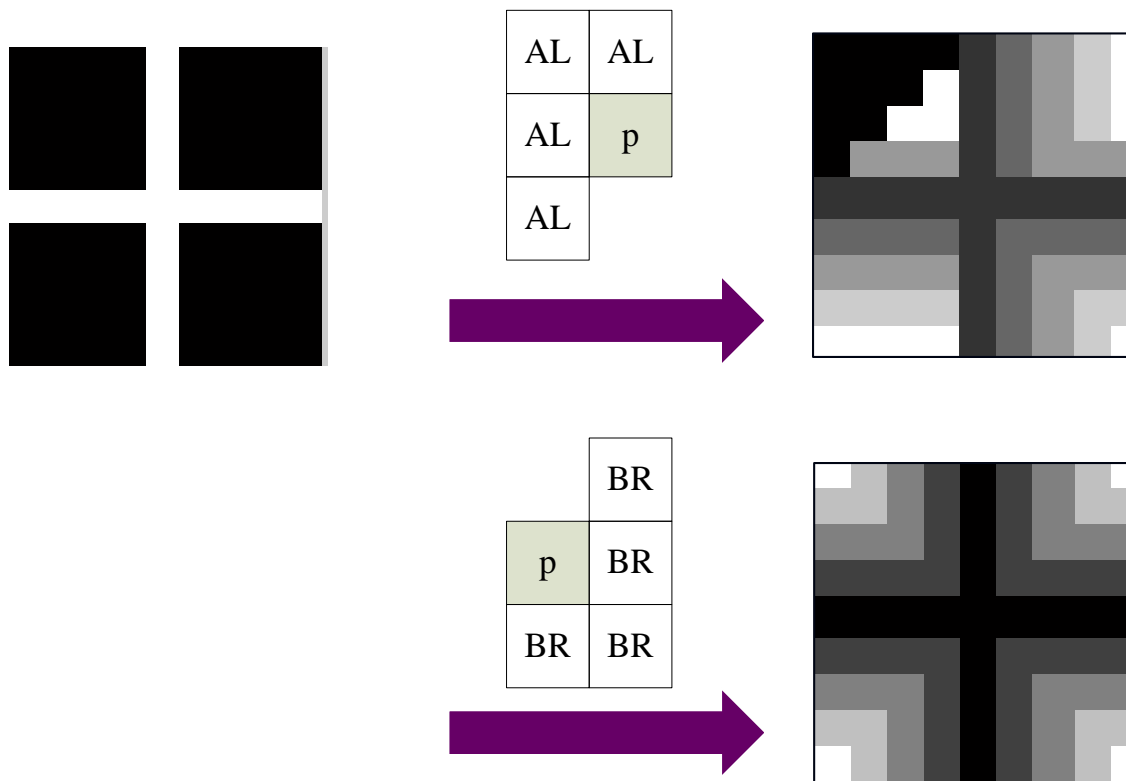
➤ 距离变换(distance transform)/斜切(chamfering)

- 设计3*3的图像掩膜，使用掩膜遍历图像
- 把掩膜中心 p 的8邻接像素分成两个掩膜
- 根据掩膜覆盖的图像位置的距离变换结果，计算中心像素 p 的距离变换结果



1) 距离变换

- 距离变换(distance transform)/斜切(chamfering)
 - 距离变换给出图像中的每个像素与某个图像子集的距离



1) 距离变换

算法：距离变换/斜切（按照某种距离度量D，对大小为M×N的图像的子集S计算距离变换）

1. 创建大小为M×N的二维数组F，子集S元素位置置为0，其他位置置为无穷大
2. 遍历图像，从上到下，从左至右。利用上方和左侧邻接像素(AL集合)，计算：
$$F(p) = \min_{q \in AL} [F(p), D(p, q) + F(q)]$$
3. 遍历图像，从下到上，从右至左。利用下方和右侧邻接像素(BR集合)，计算：
$$F(p) = \min_{q \in BR} [F(p), D(p, q) + F(q)]$$
4. 数组F的结果即为子集S的斜切

2) 卷积操作

➤ 卷积(Convolution)的定义

二维函数 $f(x, y)$ 和 $h(x, y)$ 的卷积 $g(x, y)$ 记为 $f * h$ ，由积分定义：

$$\begin{aligned} g(x, y) &= f(x, y) * h(x, y) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x-a, y-b)h(a, b)dad b \end{aligned}$$

➤ 卷积的性质

- 交换率 $f * h = h * f$
- 结合律 $(f * h_1) * h_2 = f * (h_1 * h_2)$
- 微分交换 $\frac{d}{dx}(f * h) = \frac{df}{dx} * h = f * \frac{dh}{dx}$

2) 卷积操作

➤ 二维离散卷积 $g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x-a, y-b)h(a, b)dad b$



$$g(m, n) = \sum_{k=0}^{M_2-1} \sum_{l=0}^{N_2-1} f(m-k, n-l)h(k, l)$$

$$= \sum_{k=0}^{M_1-1} \sum_{l=0}^{N_1-1} f(k, l)h(m-k, n-l)$$

$f(x, y)$ 尺寸: $M_1 \times N_1$

$h(x, y)$ 尺寸: $M_2 \times N_2$

$m = 0, 1, L, M_1 + M_2 - 2;$

$n = 0, 1, L, N_1 + N_2 - 2;$

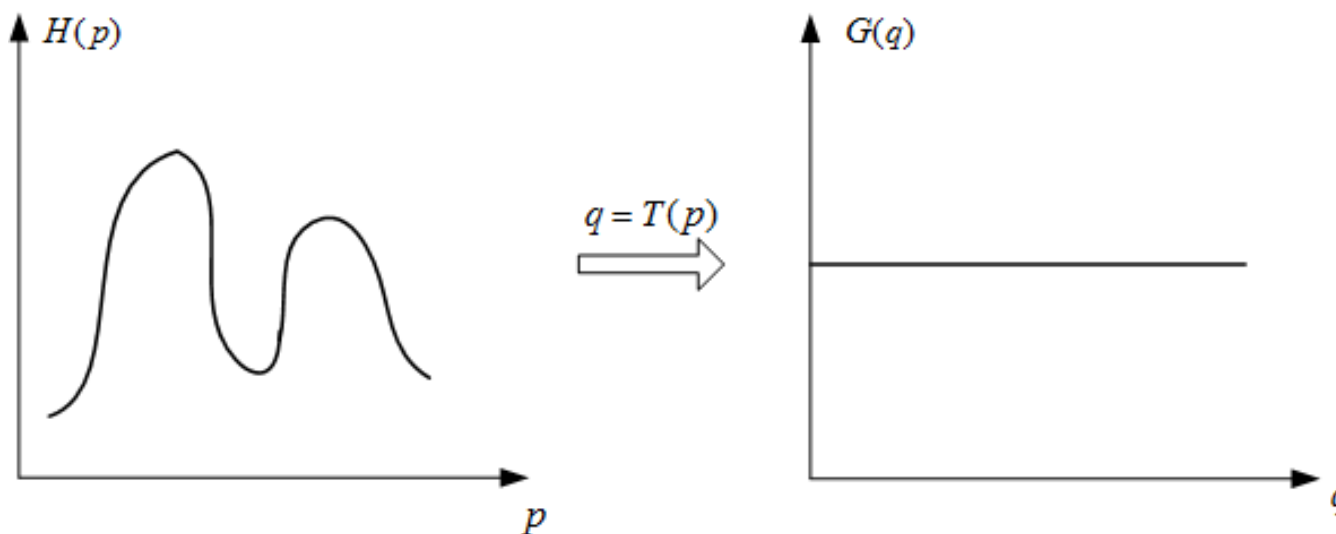
➤ h : 卷积核 / 卷积掩模 / 窗口  编程使用

➤ 手工计算离散卷积的步骤  手工计算使用

➤ 反转、相乘、相加

3) 直方图均衡化

- 直方图均衡化的目标是创建一幅在整个亮度范围内具有相同亮度分布的图像
- 输入直方图 $H(p)$ ，输入亮度范围为 $[p_0, p_k]$ ，直方图均衡化的目标：找到一个单调的像素亮度变换 $q=T(p)$ ，使输出直方图 $G(q)$ 在整个输出亮度范围 $[q_0, q_k]$ 内是均匀的



3) 直方图均衡化

- 直方图均衡化增强了靠近直方图极大值附近的亮度的对比度，减小了极小值附近的对比度
- 直方图可看作离散的概率密度函数，变换T是单调的

$$\sum_{i=0}^k H(p_i) = \sum_{i=0}^k G(q_i)$$

- 若图像有M行N列，均衡化后的直方图G(q)对应着均衡的概率密度函数，其函数值为一个常数

$$G(q_i) = \frac{MN}{q_k - q_0}$$

3) 直方图均衡化

- 公式代入：当考虑“理想的”连续概率密度时，可以得到精确的均衡化直方图

$$\int_{p_0}^p H(s)ds = \int_{q_0}^q \frac{MN}{q_k - q_0} ds = \frac{MN(q - q_0)}{q_k - q_0}$$

- 得到像素亮度变换T

$$q = T(p) = \frac{(q_k - q_0)}{MN} \int_{p_0}^p H(s)ds + q_0$$

- 积分称为累计直方图，在数字图像中用求和近似，结果直方图并不是理想均衡的

$$q = T(p) = \frac{(q_k - q_0)}{MN} \sum_{s=p_0}^p H(s) + q_0$$

3) 直方图均衡化

算法：直方图均衡化

1. 对于 $k(256)$ 个亮度级、大小为 $M \times N$ 的图像，创建长为 k 的数组 H ，初始化为0

2. 形成图像直方图 H

3. 形成累计直方图 H_c $H_c[0] = H[0]$

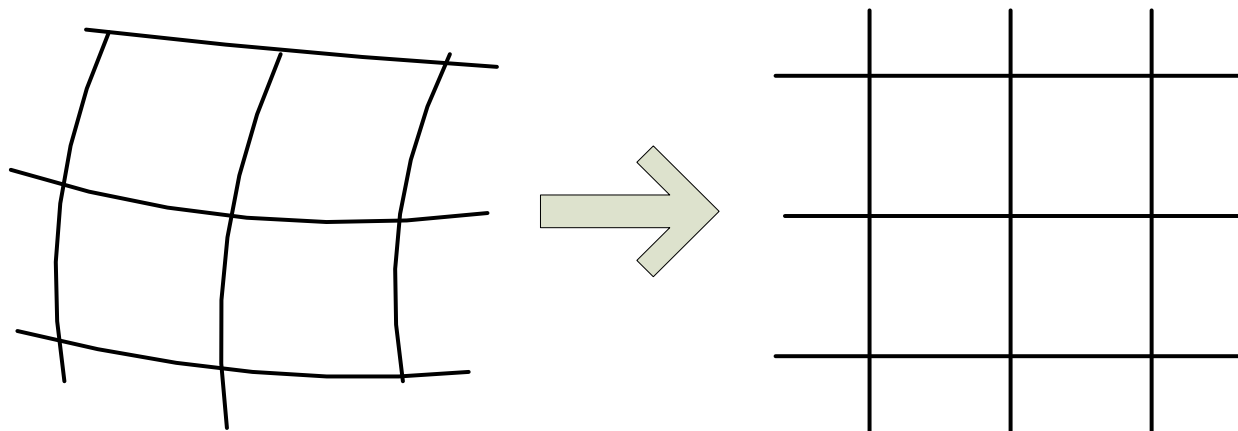
$$H_c[p] = H_c[p-1] + H[p] \quad p=1,2,L,k-1$$

4. 设置 $T[p] = \text{round}(\frac{k-1}{MN} H_c[p]), p = 0,1,L,k-1$

5. 重新扫描图像，根据查找表获得变换结果

4) 几何变换

- 几何变换用于消除图像采集时的几何失真
- 几何变换的基本步骤
 - 1) 像素坐标变换：将输入图像的像素位置映射到输出图像，输出点的坐标是连续数值
 - 2) 亮度插值：确定输出图像中整数坐标点的亮度值



像素坐标变换

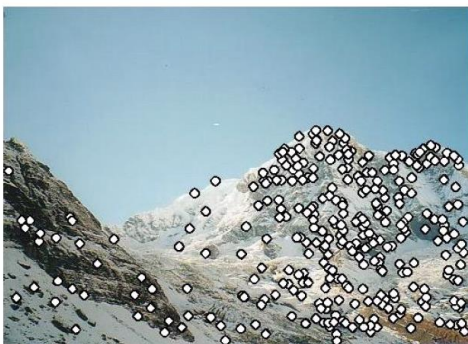
➤ SIFT特征检测



(a) Image 1



(b) Image 2



(c) SIFT matches 1



(d) SIFT matches 2

仿射变换






$$\begin{cases} x' = a_0 + a_1x + a_2y \\ y' = b_0 + b_1x + b_2y \end{cases}$$

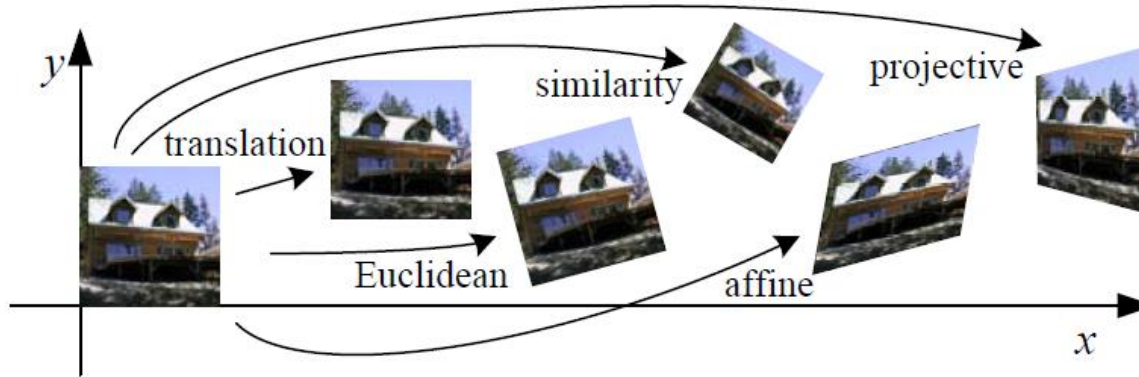
- 确定变换系数需要三个对应点对
- 仿射变换包含了典型的几何变换，如平移、旋转、缩放

补充：像素坐标变换

➤ 齐次坐标变换

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = [*] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

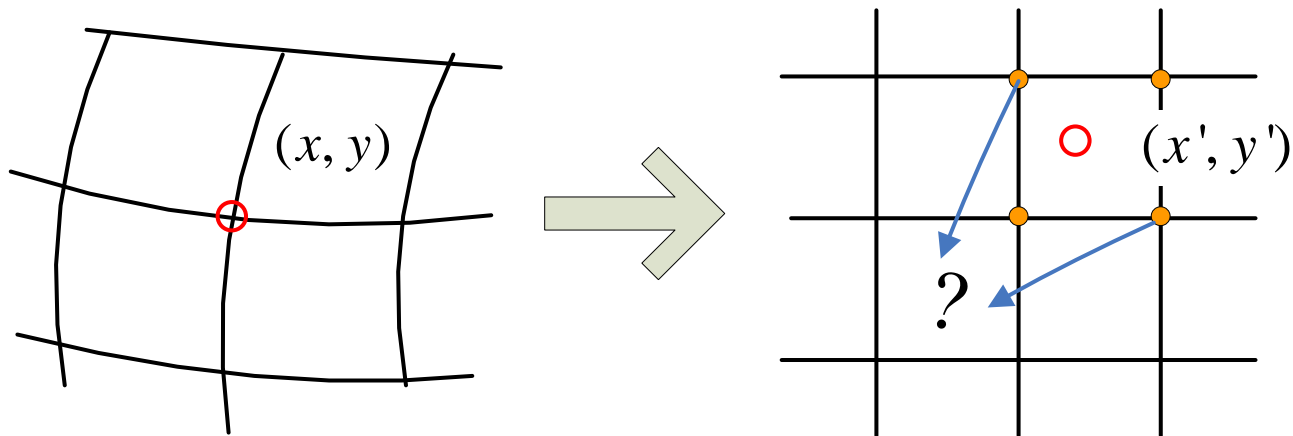
Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} I & & t \end{bmatrix}_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} R & & t \end{bmatrix}_{2 \times 3}$	3	lengths	
similarity	$\begin{bmatrix} sR & & t \end{bmatrix}_{2 \times 3}$	4	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{2 \times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3 \times 3}$	8	straight lines	



亮度插值

➤ 进行亮度插值的原因

- 像素亮度变换将原始图像中的 (x, y) 变换到了新位置 (x', y') 一般说来, 新坐标点的位置都是非整数
- 对于数字图像, 需要得到离散栅格上的数值, 输出图像栅格上每个像素的亮度值可通过邻域非整数采样点的亮度插值得



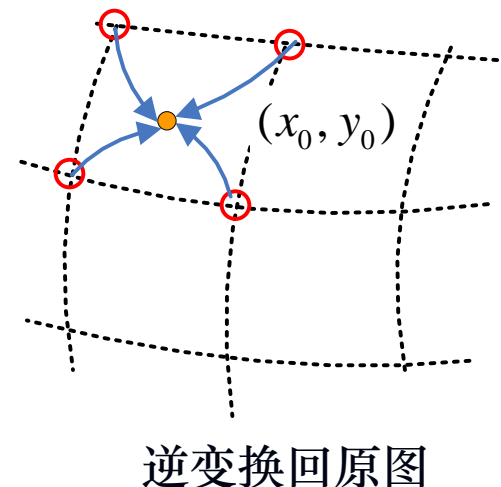
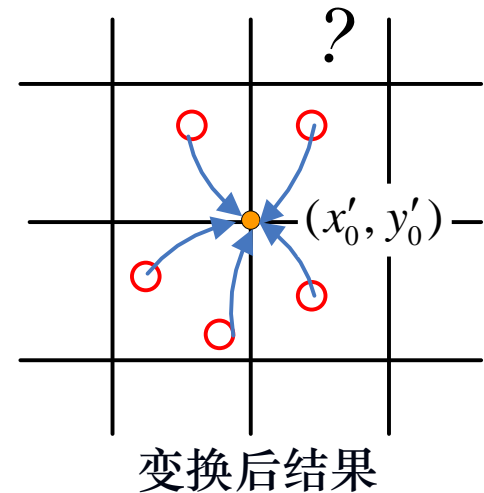
亮度插值

➤ 用对偶方法完成亮度插值

- 确定对应于输出图像离散栅格点在输入图像中原来点的亮度
- 假设计算在输出图像**离散栅格**上像素 (x'_0, y'_0) 的亮度值
- 输入图像中对应点 (x_0, y_0) 的坐标可通过像素坐标逆变换得到

$$(x_0, y_0) = \mathbf{T}^{-1}(x'_0, y'_0)$$

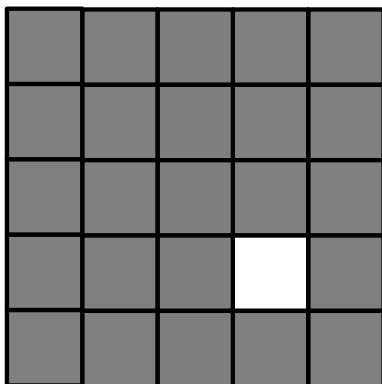
- 对输入图像进行亮度插值，获得 $(x_0, y_0) / (x'_0, y'_0)$ 的亮度值



练习1: 距离变换

对于如下图像子集（白色表示1，灰色表示0）

- 请根据距离变换的定义，给出下图按照城区距离 D_4 进行距离变换的结果
- 请根据距离变换的算法，给出下图按照棋盘距离 D_8 进行距离变换，**第一次遍历**图像的结果



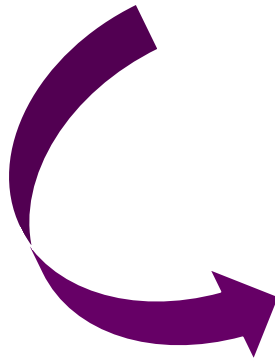
练习2: 直方图均衡化

对于如下有10个亮度级、大小为 6×6 的图像，试给出对该图像进行直方图均衡化的过程和结果

$$I = \begin{bmatrix} 0 & 3 & 5 & 6 & 1 & 0 \\ 3 & 3 & 3 & 1 & 1 & 1 \\ 9 & 3 & 0 & 0 & 1 & 9 \\ 9 & 2 & 0 & 0 & 4 & 9 \\ 2 & 2 & 2 & 4 & 4 & 4 \\ 0 & 2 & 7 & 8 & 4 & 0 \end{bmatrix}$$

$$T[p] = \text{round}\left(\frac{k-1}{MN} H_c[p]\right), p = 0, 1, \dots, k-1$$

3) 直方图均衡化



练习3 卷积与线性滤波

对于如下 4×4 的数字图像 I ，使用“复制 (replicate)”方式填充边界，且输出图像与输入图像尺寸相同，分别给出平滑滤波器 h_1 ，一阶差分滤波器 h_2 ，二阶差分滤波器 h_3 的滤波结果，并通过结果比较三个滤波器的性能

$$I = \begin{bmatrix} 1 & 1 & 8 & 8 \\ 1 & 1 & 8 & 8 \\ 1 & 1 & 8 & 8 \\ 1 & 1 & 8 & 8 \end{bmatrix} \quad h_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, h_3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

内容总结2



图像处理基础

数字图像的概念与性质

图像预处理

二值图像处理

要点1: 最简单的图像分割: 图像二值化

要点2: 内边界、外边界、扩展边界 (数字图像的性质)

要点3: 膨胀腐蚀、开闭运算的理解与计算

算法: 阈值算法、内边界跟踪算法

传统特征检测

要点1: 稀疏特征 (Harris角点、SIFT特征点)

要点2: 稠密特征 (HOG特征)

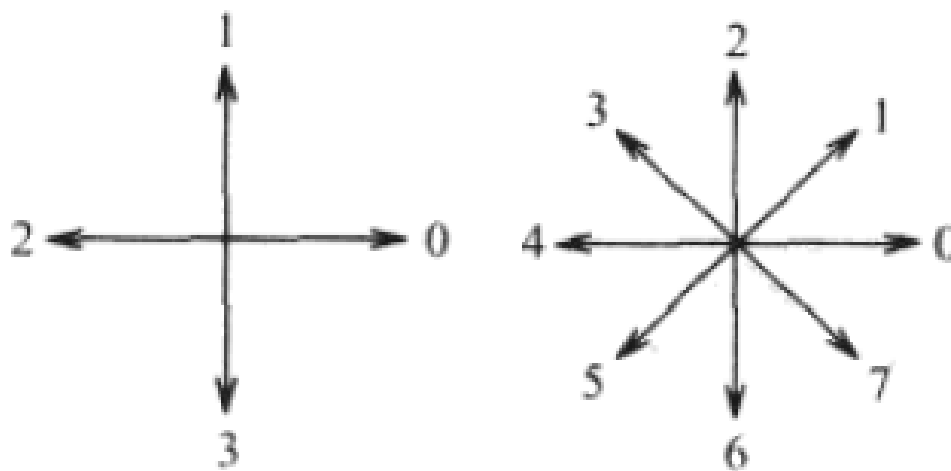
要点3: 点 - 线的对偶性质的Hough变换

算法: Harris角点检测、SIFT特征检测、HOG特征检测、基于Hough变换的直线/圆检测

5) 二值图像边界跟踪

算法：内边界跟踪

1. 从左上方开始搜索图像，直到找到一个新区域的一个像素 P_0 ， P_0 是区域边界的起始像素
2. 定义变量 dir ，存储从前一个边界元素到当前边界元素的前一个移动方向，置 $dir=3$ (4-邻接跟踪) / 7 (8-邻接跟踪)



算法：内边界跟踪（续）

3. 按照逆时针方向搜索当前像素的 3×3 邻域，从以下的方向开始搜索邻域：

4-邻接跟踪： $(dir+3) \bmod 4$

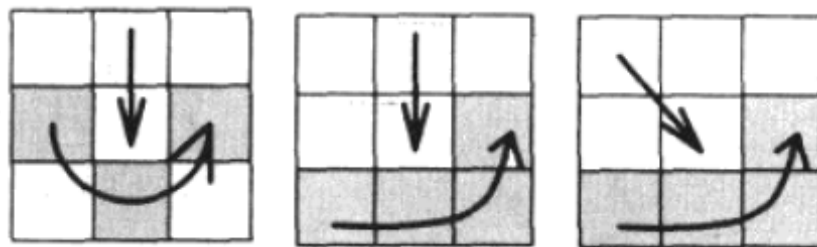
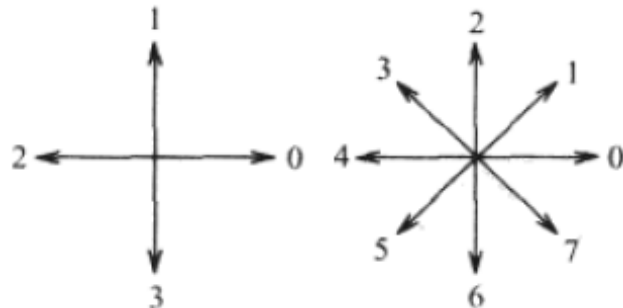
8-邻接跟踪： $(dir+7) \bmod 8$ (dir 为偶数)

$(dir+6) \bmod 8$ (dir 为奇数)

找到的第一个与当前像素值相同的像素是一个新的边界元素 P_n ，更新 dir 数值

4. 如果当前的边界元素 P_n 等于第二个边界元素 P_1 ，且前一个边界元素 P_{n-1} 等于 P_0 ，则停止；否则，重复第3步

5. 检测到的内边界由像素 P_0, \dots, P_{n-2} 构成



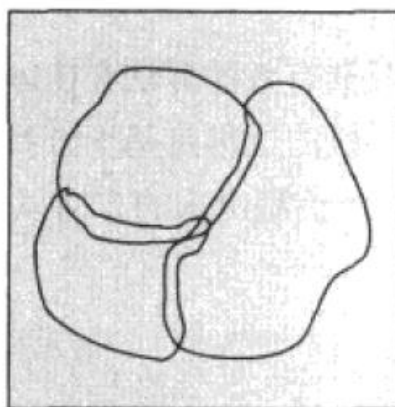
6) 扩展边界

➤ 扩展边界

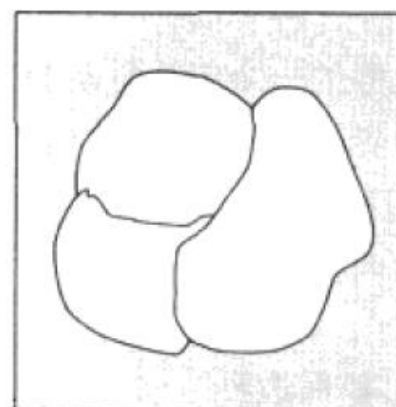
- 定义了相邻区域之间存在的单一公共边界，可以使用标准像素坐标来描述
- 保留了外边界有用的性质
- 边界形状与内边界完全相同，只是边界像素位置分别向下和向右平移了半个像素



内边界



外边界



扩展边界

6) 扩展边界

3	2	1
4	P	0
5	6	7

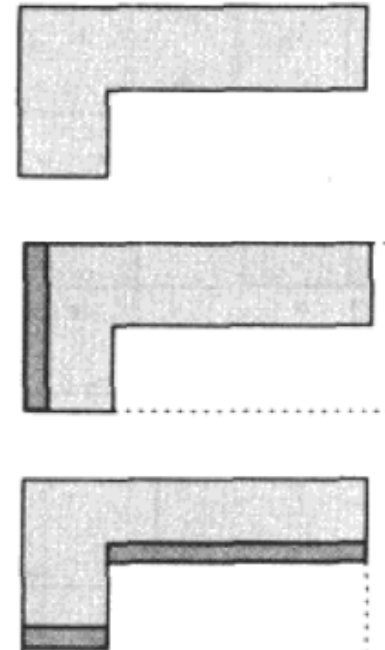
➤ 扩展边界用8-邻接定义

- $P_4(P)$ 表示像素P直接左像素, $P_0(P)$ 表示像素P直接右像素
- $P_2(P)$ 表示像素P直接上像素, $P_6(P)$ 表示像素P直接下像素

➤ 区域R的内边界像素

如果 $P_4(P) \in Q$, 则 P 是 R 的左像素, 标识为 $LEFT(R)$
 如果 $P_0(P) \in Q$, 则 P 是 R 的右像素, 标识为 $RIGHT(R)$
 如果 $P_2(P) \in Q$, 则 P 是 R 的上像素, 标识为 $UPPER(R)$
 如果 $P_6(P) \in Q$, 则 P 是 R 的下像素, 标识为 $LOWER(R)$

(Q 代表区域 R 外的像素)



6) 扩展边界

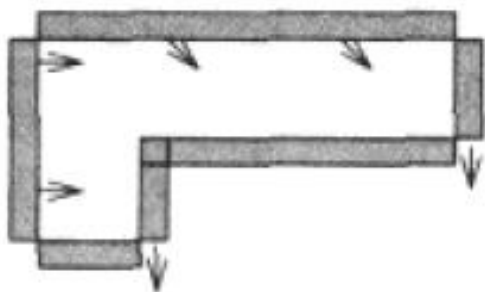
- 扩展边界EB定义为满足下述条件的像素 P, P_0, P_6, P_7 的集合

$$\begin{aligned} EB = & \{P : P \in LEFT(R)\} \cup \\ & \{P : P \in UPPER(R)\} \cup \\ & \{P_6(P) : P \in LOWER(R)\} \cup \\ & \{P_6(P) : P \in LEFT(R)\} \cup \\ & \{P_0(P) : P \in RIGHT(R)\} \cup \\ & \{P_7(P) : P \in RIGHT(R)\} \end{aligned}$$

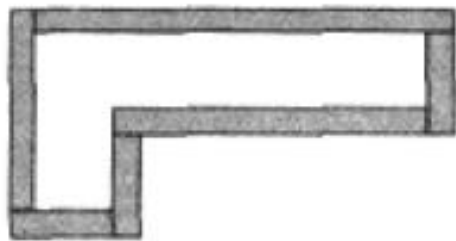
3	2	1
4	P	0
5	6	7

6) 扩展边界

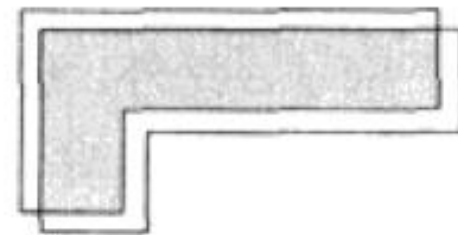
- 从外边界建立扩展边界
 - 向下和右移位所有的UPPER外边界点一个像素
 - 向右移位所有的LEFT外边界点一个像素
 - 向下移位所有的RIGHT外边界点一个像素
 - LOWER外边界保持不变



外边界



扩展边界

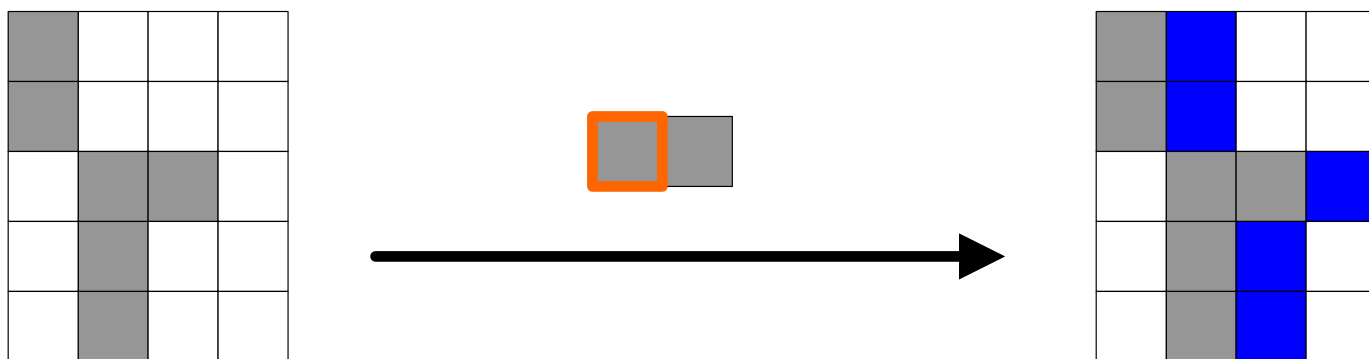


扩展边界与自然
物体边界具有
相同的形状与尺寸

7) 膨胀与腐蚀

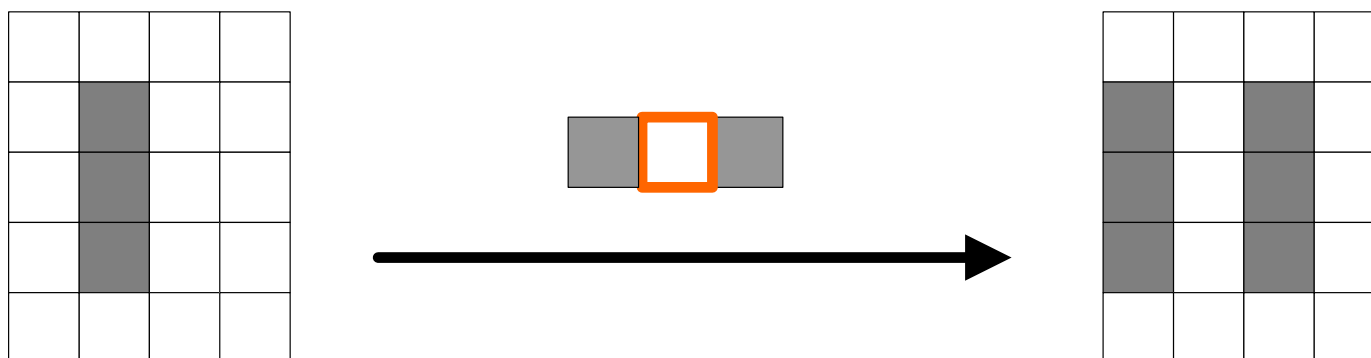
➤ 膨胀的实现过程

- 将结构元素B的原点移至集合A的某一点
- 将结构元素B中的点与该点相加，得到对集合中该点的膨胀结果
- 对集合A中所有元素重复该过程



7) 膨胀与腐蚀

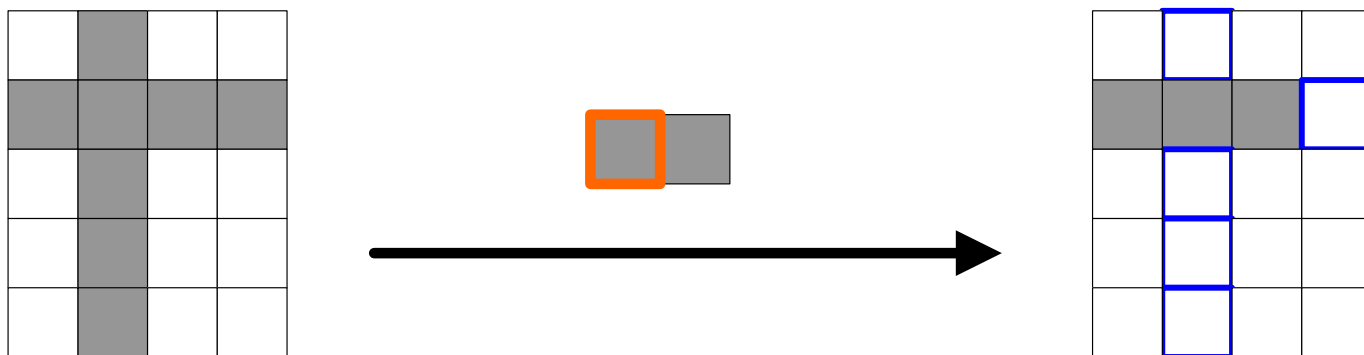
- **膨胀的实现过程** $A \oplus B = \{p \in \varepsilon^2, p = a + b, a \in A \text{ 且 } b \in B\}$
 - 将结构元素B的原点移至集合A的某一点
 - 将结构元素B中的点与该点相加，得到对集合中该点的膨胀结果
 - 对集合A中所有元素重复该过程



7) 膨胀与腐蚀

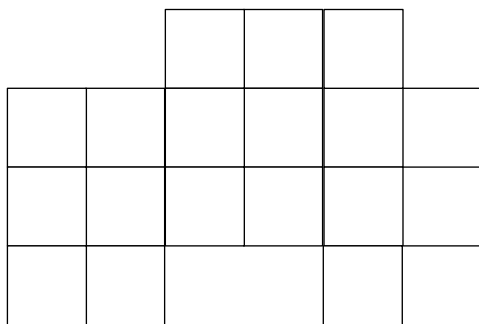
➤ 腐蚀的实现过程

- 将结构元素B的原点移至集合A的某一点
- 若结构元素B属于集合A，则保留该点作为腐蚀结果
- 对集合A中所有元素重复该过程

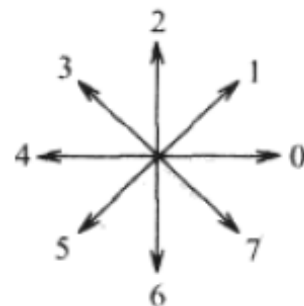


练习4 内边界跟踪

请给出如下二值图像使用8-邻域进行内边界跟踪结果，及对应的移动方向dir值序列

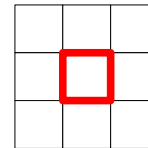
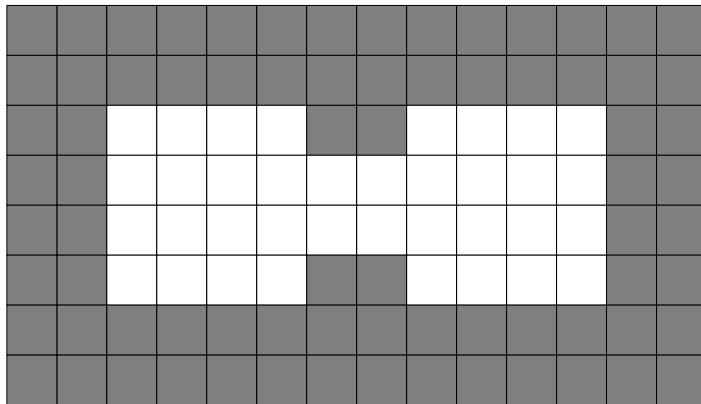


8-邻接跟踪: $(dir+7) \bmod 8$ (dir为偶数)
 $(dir+6) \bmod 8$ (dir为奇数)



练习5 开闭运算

对于如下二值图像（白色表示1，灰色表示0），若结构元素为ele，分别计算该图像的二值开运算和二值闭运算的结果，并比较开运算和闭运算的作用



总结内容2

➤ 衡量图像特征的标准

- 亮度不变性、旋转不变性、尺度不变性
- 噪声敏感性

➤ 边缘-角点-特征点-稠密特征

- 边缘检测：检测图像上灰度的剧烈变化 ➡ 图像梯度
- Harris角点检测：在图像邻域各个方向上检测灰度的剧烈变化 ➡ 使用图像x和y方向的差分 I_x/I_y
- SIFT特征：在DoG尺度空间检测极值点+图像梯度方向直方图
- HOG特征：梯度方向直方图+机器学习

The End



机器人与信息自动化研究所

Institute of Robotics & Automatic Information System



南開大學
Nankai University