

# lec8 算法2

## 一、几何变换：

### (一) 仿射变换

- 1、齐次坐标形式表示
- 2、可以表示的变换类型：旋转变换，伸缩变换，倾斜变换

### (二) 投影变换

- 1、变换含义：将一个平面投影到另一个平面上

$$\begin{pmatrix} \tilde{r} \\ \tilde{c} \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} r \\ c \\ W \end{pmatrix}$$

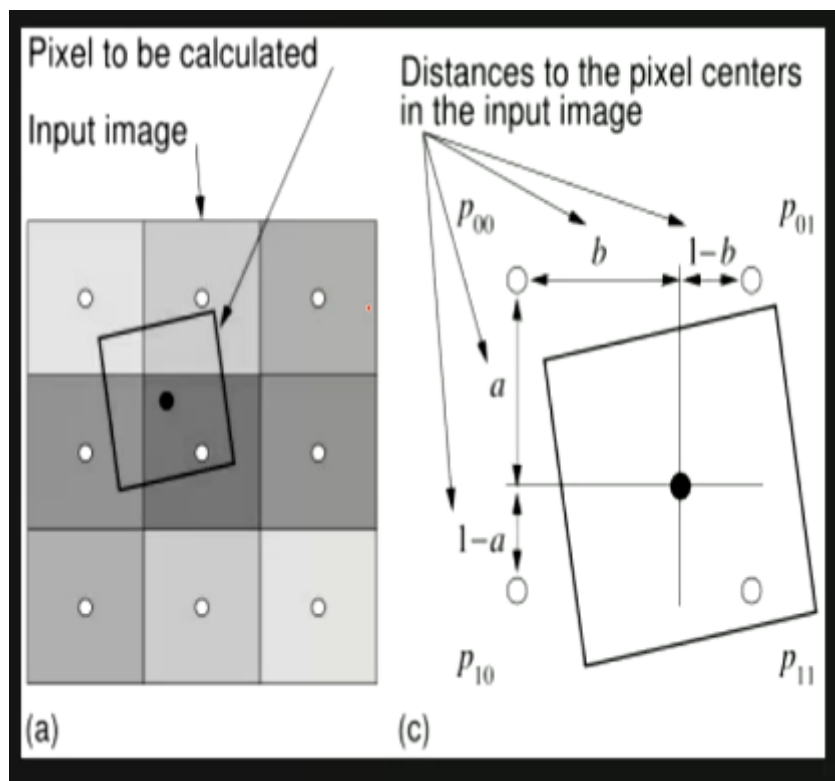
- 2、单应矩阵：自由度为8

### (三) 仿射变换像素值确定方法：插值法

- 1、最邻近插值法：对于某一个像素而言，将其映射到原图像上，然后寻找距离它最近的像素值的值，它在新图像上的像素值就是这个最近的像素值

有锯齿

- 2、**双线性插值**：分别对水平和垂直方向做插值，然后综合



$$\tilde{g} = b(ap_{11} + (1-a)p_{01}) + (1-b)(ap_{10} + (1-a)p_{00})$$

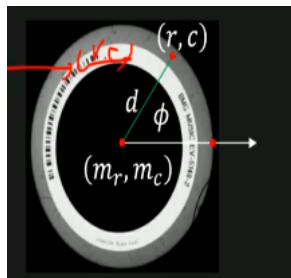
3、双三次插值

#### (四) 平滑

先对原图像做平滑，然后使用插值方法进行仿射变换，可以改善最邻近插值法走样现象

## 二、极坐标变换

### (一) rc与 $d\phi$ 关系



$$r = m_r - d \sin \phi$$

$$c = m_c + d \cos \phi$$

### (二) 映射关系

通过  $d, \phi$  建立二维矩形区域，每个点对应原图上的一个像素点，可以将圆弧展开成矩形

## 三、图像分割

### (一) 二值化

### (二) 亚像素边界

双线性插值法：通过双线性插值法得到一个曲面，然后用所求的边界灰度值平面与曲面相交得到亚像素精度边界

## 四、特征提取：区域特征

### (一) 矩

零阶矩：连通域的面积

$$m_{p,q} = \sum r^p c^q$$

一阶矩（归一化的矩）：连通域的亚像素精度中心点

$$n_{p,q} = \frac{\sum r^p c^q}{A}$$

中心矩（归一化）：

$$\mu_{p,q} = \frac{1}{A} \sum_{(r,c)^T \in R} (r - n_{1,0})^p (c - n_{0,1})^q$$

二阶中心矩：跟椭圆的长轴短轴( $r_1, r_2$ ), 旋转角( $\theta$ )有关

通过  $\frac{r_1}{r_2}$  判断是圆还是椭圆

## (二) 外接矩形

## (三) 轮廓长度

基于链码定义：对角线长度为 $\sqrt{2}$ , 上下左右相邻为1

## (四) 矩形度

$$R = \frac{A}{A_{out}}$$

越接近1越像矩形

## (五) 圆形度：更多考虑boundary

$$C = \frac{P^2}{4\pi A}$$

P是区域的周长, A是区域的面积

## (六) 圆形性

与五合用：先计算圆形性, 再计算region的圆形度

# Lec9 算法3

## 一、灰度图像特征

### (一) 区域灰度均值

### (二) 灰度区域方差

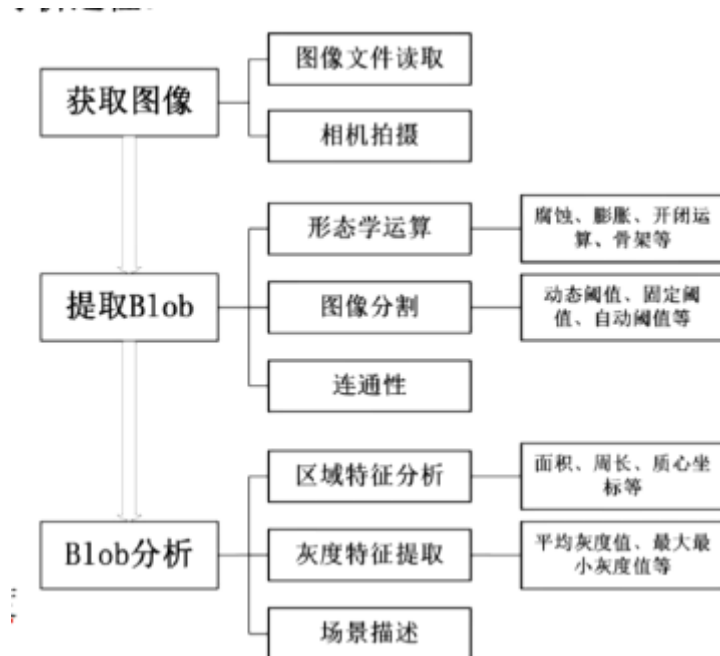
第一个区域的灰度分布作为参考, 让下一副图像的灰度分布接近, 由此减小其他因素的影响

### (三) 矩

$$m_{p,q} = \sum g_{r,c} r^p c^q$$

### (四) 形态学操作

## 二、Blob 分析



### 三、模板匹配

#### (一) 相似性

#### (二) 模板匹配和相对关系确定ROI

#### (三) 算法

##### 灰度匹配

SAD: 绝对值误差--->模板像素和滑动窗口对应位置像素差值

SSD: 平方误差

问题: 亮度变化时(整体灰度变)即使是相同的形状仍然会返回很大的SAD/SSD误差

越接近0质量越高

##### NCC模板匹配:归一化相关系数

1、NCC公式

$$NCC(r, c) = \frac{1}{n} \sum \frac{t(u, v) - m_t}{\sqrt{s_t}} \frac{f(u + r, v + c) - m_f}{\sqrt{s_f}}$$

其中m代表均值, s代表方差

2、NCC取值范围

$$-1 < NCC < 1$$

3、几何衡量相似性: 余弦相似度

将2D区域flatten成一维向量, 计算向量之间的余弦

越接近正负1质量越高

## (四) 灰度匹配加速

计算前项的SAD值，如果已经大于 $nt_s$ ，那么此次模板匹配不需要再进行下去

## (五) NCC加速

- 1、离线计算模板的均值和方差
- 2、提前规划ROI

# Lec10 NCC

## 一、图像金字塔

### (一) 减小复杂度的原理

先在小Scale图像上模板匹配，得到大致的ROI，传到下一层金字塔得到ROI，进一步模板匹配，由此类推

### (二) 构建图像金字塔的原理

原图不断下采样构成金字塔，越上层的金字塔scale感受野越大，scale越小

- 1、隔行抽取采样：会丢失一部分信息
- 2、滤波后采样（不断重复）：高斯核考虑了kernel内邻域像素的信息，随着上采样滤波kernel变为两倍
- 3、层数：最顶层的应该仍然具有一些区分度

### (三) 算法备注

“最顶层阈值参数宽容一些。”

## 二、亚像素精度NCC

### (一) 算法内容

对于已经经过NCC确定的像素点，考察他的邻域一共九个点，用来拟合一个三维曲面，求其极值点，极值点坐标是亚像素精度的模板中心

### (二) 算法流程

1、

$$F(x, y) = AP$$

其中 $F(x, y)$ 是邻域内9个pixel的NCC值

A是由位置信息 $x_i, y_i$ 构成的矩阵

P是待定系数（二次多项式系数）

2、偏导数求极值

### 三、缩放值和旋转角度

#### (一) 旋转：尝试获得角度的Angle of Interest

- 1、离散化模板：构建一系列不同转动角度的模板
- 2、最顶层：用不同旋转角度的模板进行匹配，找到NCC最大的方向
- 3、下一层：得知ROI范围和大致旋转角度后，用待选的模板在待定ROI里继续搜索
- 4、重复上述过程直到最底层

#### (二) 缩放：略

### 四、模板匹配性能增强:基于边缘度量

#### (一) 均方边缘距离

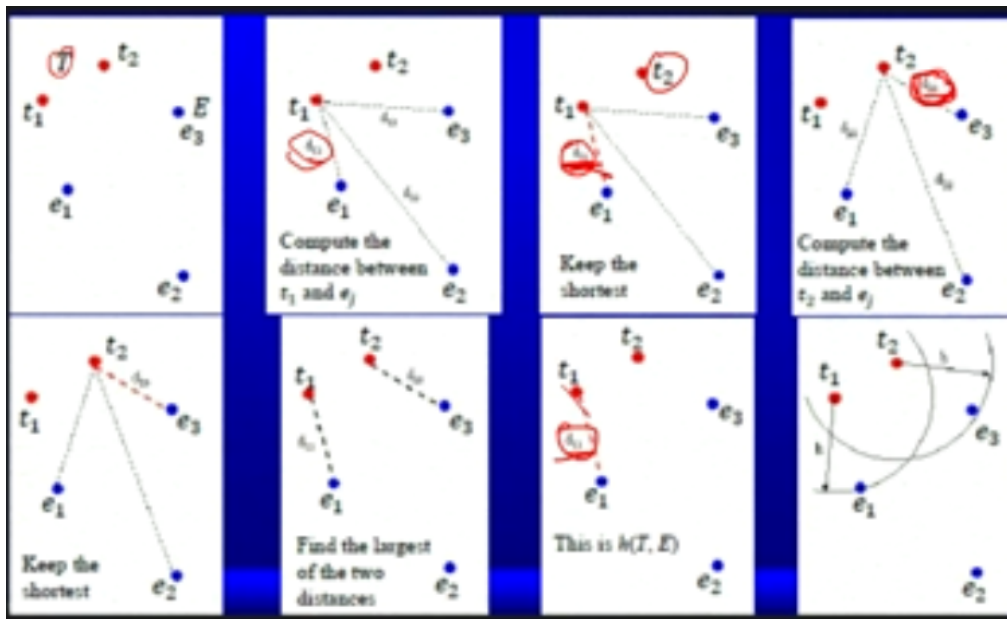
模板边缘点多于图像中边缘点时会导致均方距离过大

#### (二) Hausdorff距离

$$H(E, T) = \max(h(T, E), h(E, T))$$

$h(T, E)$ : T点集到E点集的最小距离里的最大值,

$h(E, T)$ : E点集到T点集的最小距离里的最大值,



## Lec11 增强模板匹配

### 一、广义霍夫变换

#### (一) 边缘点信息

与中心点连线，得到：

- 1、一个由中心点指向边缘点的向量
- 2、向量与水平方向的夹角 $\theta$
- 3、梯度方向

## (二) R-table: 基于模板构建

Index	Edge direction	$r$
0	0 $\phi_0$	$r_0^0(r_0, \theta_0), r_1^0(r_1, \theta_1), \dots$
1	$\Delta\phi$ $\phi_1$	$r_0^1(r_0, \theta_0), r_1^1(r_1, \theta_1), \dots$
2	$2\Delta\phi$ $\phi_2$	$r_0^2(r_0, \theta_0), r_1^2(r_1, \theta_1), \dots$
$\vdots$	$\vdots$	$\vdots r_k^l$
$n$	$n\Delta\phi$ $\phi_n$	$r_0^n(r_0, \theta_0), r_1^n(r_1, \theta_1), \dots$

## (三) 算法流程

对test image上的边缘上的每一个点，计算它的Edge direction (梯度方向)，在上述R-table内寻找可能的几个 $(r, \theta)$ 数对，计算中心 $(x_c, y_c)$ ，然后投票

得票数最高的 $(x_c, y_c)$ 就是所求的中心

## (四) 旋转和缩放

投票表变成4维 $(x_c, y_c, S, \alpha)$