

# 机器视觉复习提纲

---

## lec1:导论

---

### 一、机器视觉基本概念（重点）

#### （一）基本目的（任务，能做什么，优点）

提升产量，提高利润，减少缺陷，Track Trace and Control

#### （二）如何做到

- 1、Measure测量
- 2、Count计数
- 3、Decode解码
- 4、Location定位
- 5、Inspection检测(缺陷检测)

#### （三）机器视觉系统基本组成

##### 基于PC的机器视觉

物体，光源，镜头，相机(CCD,CMOS)，PC(算法+通信)

##### 基于Smart Camera

物体，智能相机

长焦：视野窄，物体大

短焦：视野范围广，物体小

## lec2:如何设计一个机器视觉系统

---

### 一、任务说明书（非重点）

#### （一）Task and Benefit

给任务分类，属于哪一个part

#### （二）Part:清晰的部件说明和充足的数据

- 1、样本形状变化
- 2、样本最大最小尺寸
- 3、需要提取的特征

### (三) Part presentation:零件如何在视野内出现

- 1、索引定位
- 2、移动速度加速度

### (四) Performance Requirement

- 1、准确率
- 2、时间效率

### (五) 信息接口

### (六) 安装空间

### (七) 使用环境

## 二、系统设计

### (一) 相机类型

- 1、面阵相机
- 2、线扫描相机

### (二) FOV

FOV=考虑旋转和便宜的余量+margin余量+sensor长宽比

### (三) 分辨率 (计算题, 大概率会考, 重点)

- 1、相机分辨率: pixel像素
- 2、空间分辨率: mm/pixel
- 3、测量精度

名称	符号	单位
空间分辨率	$R_s$	Mm/Pixel
相机分辨率	$R_c$	Pixel
待检测特征实际尺寸	$S_f$	mm
待检测特征对应最小像素	$N_f$	Pixel

$$R_s \times R_c = FOV$$

$$R_s = \frac{S_f}{N_f}$$

统一:

$$R_c = \frac{FOV}{R_s} = FOV \frac{N_f}{S_f}$$

其中:FOV由实际物体确定,Nf一般取1,Sf由要求精度决定

例题:

$$FOV_{hor} = 7 \times \frac{4}{3} = 9.33mm$$
$$Rc = \frac{FOV}{Rs} = FOV \times \frac{Nf}{Sf}$$
$$Nf = 1, Sf = 0.1mm$$
$$Rc_{ver} = FOV_{ver} * 10 = 70mm$$
$$Rc_{hor} = FOV_{hor} * 10 = 93.3mm$$
$$Rs = \frac{FOV_{len}}{Rc} = 10.3/Rc_{camera} = \dots$$

#### (四) 像素速率

#### (五) 凸透镜成像 (计算题, 大概率不会考)

$$\frac{1}{f'} = \frac{1}{a} - \frac{1}{a'}$$

#### (六) 相机接口

相机镜头接口	C MOUNT	CS MOUNT
/	加adapt ring ->CS MOUNT	/

## Lec3:相机

### 一、基本概念

#### (一) 光电效应

- 1、QE量子效率: 光激发电子的效率, 与感光能力正相关
- 2、满井容量: 更大的pixel成像效果更好
- 3、成像原理: 光子击打Si原子释放电子, 根据电子电量决定pixel的值

#### (二) CMOS和CCD

- 1、CCD: 成像质量高 (感光面积大), 价格高, 速度低
- 2、CMOS: 成像质量低 (感光面积小), 价格低, 速度高

#### (三) 相机参数

- 1、增益: 也就是提高亮度, 噪声会同步增加
- 2、快门方式: 全局快门 (高速物体), 卷帘快门
- 3、触发方式: external外部硬件触发, internal内部软件触发

## 二、相机类型

### (一) 面阵相机

- 1、一些可以节省资源的操作：人为划定ROI, Skipping(跳过一些像素), Binning(残差叠加)
- 2、Bayer Pattern:单个CCD, 分别对三种颜色的像素捕获, 再用插值法补上空白
- 3、3CCD:棱镜分光, 三个CCD分别采取, 价格高, 成像质量好

### (二) 线阵相机

- 1、优点：成像质量比较高, 适用于扫描的场合
- 2、防止拉伸的参数匹配 (计算题, 可能考) :

$$\frac{L_0}{H_C} = \frac{V_0}{V_C}$$

解释：目标物体宽度/线阵相机扫面宽度=目标物体移动速度/线阵相机扫面速度(垂直于扫描方向)

### (三) 相机接口

#### GigE->多相机

- 1、交换机：更方便
- 2、千兆网卡：更高质量

#### IEEE 1394

- 1、同步：保证速度
- 2、异步：保证不丢帧, 保证质量

#### camera link->高速场景

需要额外的采集卡

#### usb3.0->单相机

#### 通用相机接口

## Lec4:镜头

---

### 一、基本概念

#### (一) 小孔成像

#### (二) 近轴近似

$$n_1\alpha_1 = n_2\alpha_2$$

#### (三) 景深DOF

物体的深度超出范围时不能拍摄

## (四) Work distance

镜头与物体之间的距离

可以在不同WD的情况下选用不同的焦距镜头，保证拍摄物体大小基本不变

## (五) 放大倍率Pmag

$$Pmag = \frac{FOV}{Sensor}$$

## 二、镜头对成像质量的影响

### (一) 分辨率

- 1、表示方法：Line Pair的识别能力
- 2、镜头的Resolution和相机的Resolution要匹配

### (二) 对比度

- 1、对比度公式

$$Contrast = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}$$

- 2、工业CCD相机：至少要求10-20%的对比度

人眼对比度识别能力：1-2%

- 3、MTF值：反应镜头分辨率能力和对比度能力

- (1) 公式

$$F_{MTF} = \frac{g_1 - g_2}{255}$$

- (2) 线对对MTF值的影响：频率对MTF的影响

线对越密集MTF一般来说越小

- (3) 分辨率&对比度测试

宽线用来测对比度，细线用来测分辨率，一般来说MTF>0.5 是可以接受的范围

MTF线弯曲：有场弯曲现象

### (三) 景深DoF

- 1、F#

$$F\# = \frac{f}{D}$$

f: 成像距离, D: 光圈大小

和单反里大光圈小景深，小光圈大景深一个道理

F#越大，光圈越小，景深越大，能够拍的清越远的物体

## (四) 畸变

$$D = \frac{AD - PD}{PD}$$

AD: 实际成像位置 PD: 理想成像位置

$D < 0$  负畸变: 桶形畸变

$D > 0$  正畸变: 枕形畸变

## (五) 透视 (畸变) : 近大远小

### 远心镜头 (作业题)

优点:

1、没有透视畸变现象

限制:

1、拍摄区域小于镜头成像直径, FOV被限制

2、放大倍数不变

## 三、镜头曲线

### MTF曲线

### 畸变值曲线

### 相对亮度曲线

## Lec5 光源

---

### 一、LED

#### (一) Solid Angle 立体角

$$d\Phi = \frac{dA \cos\theta}{r^2}$$

$dA$  是空间中的一个小平面

#### (二) 光源能量

1、光通量: 单位时间内发出的光量, 单位为流明 (lm)

2、发光强度: 在给定方向的单位立体角内发射的光通量, 单位为坎德拉(cd)

$1cd = 1lm/sr$  (立体弧度)

3、照度: 面元上光通量和面积的比值, 单位为勒克斯(lx)

$1lx = 1lm/m^2$

4、亮度: 某一方向, 单位投影面积上的光强

## 二、对比度与光源

### (一) 光源类型

光源类型：平面光源，**环形光源**，无影光源（无影灯），积分球光源，**同轴光源**，条形光源，点光源

### (二) 打光方式

- 1、**亮场照明**（照明角度大，接近垂直）：均匀性好，面积相对小
- 2、**暗场照明**（照明角度小于 $45^\circ$ ）：凹凸表现能力强
- 3、垂直照明（垂直）：均匀性好
- 4、**背光照明**：用于测量和形状判断，高精度测量->单色波长短的光
- 5、多角度照明：提取三维信息
- 6、积分球照明：均匀，用于缺陷检测，曲面
- 7、同轴（平行）照明：缺陷检测
- 8、**平面漫射光**
- 9、点光源：与远心镜头搭配使用

### (三) 滤光片&其他

- 1、滤光片：过滤特定波长的光
- 2、偏振片：减小炫光
- 3、减少环境光：用更强的光掩盖掉环境光干扰（例如使用频闪光）；用带通滤波片
- 4、色表：光源散发的颜色
- 5、显色性：光源的物体通过物体反射后，显现物体的颜色

灰度相机显色：

- (1) 用相同或者相近的颜色使得目标颜色更亮（灰度值更高）
- (2) 用相反颜色使得目标颜色更暗
- 6、IR红外光：需要特殊的红外相机
- 7、UV紫外光：根据材料特殊性质

## lec6 算法

---

### 一、二值图像：binary

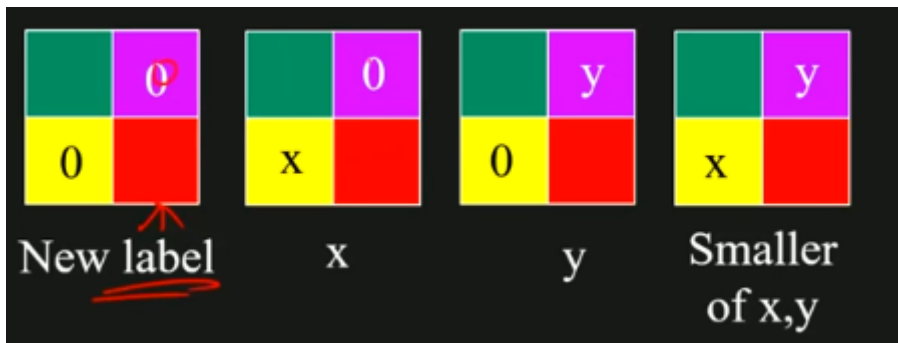
#### (一) 连通域操作Region

- 1、面积：统计所有像素为1的像素数量
- 2、投影：向水平和垂直方向投影（桶排序）->确定物体分布(bounding box)
- 3、行程编码RLE(快速算法)：每一行考虑连续的"1"path -> 有效地求面积和投影
- 4、矩Moments：旋转平移不变性，性质很好
- 5、4邻域与8邻域：前景（物体）用8邻域定义（贪心），背景用4邻域定义（找洞）

## (二) 连通域标记

### 1、经典算法Row\_by\_Row

(1) 第一遍循环：考察像素为1的每个像素的左，上像素邻接关系，全0置新label



(2) 第二遍循环：根据第一步记录的邻接关系连通剩余区域

### 2、行程编码法DFS

(1) 建立行程编码的树：overlap可以选用四邻接和八邻接原则

(2) 每棵树是一个region

## (三) 其他操作

### 1、尺寸滤波器

### 2、欧拉数

$$E = C - H$$

欧拉数=连通域个数-孔洞个数

### 3、Boundary: 用链码找边界

可以获得周长

### 4、Distance: 点与点之间的距离

欧式距离：二范数距离

City-block：一范数距离(绝对值)

Chessboard:  $\max(|r_1 - r_2|, |c_1 - c_2|)$

算法：模板卷积

### 5、图像增强

cv::LUT用于提升速度进行灰度图增强

### 6、辐射度标定

消除非线性畸变，理想的光能与灰度关系：

$$g = aE + b$$



## 7、图像平滑

### 平均滤波

时域平均：只能用于静态场景，N张图片

Padding策略：镜像，补0

时间复杂度： $O(whmn)$ ---结合律加速---> $O(wh(m+n))$

算法加速:

(1) 可分离滤波器->分离法：kernel分成列向量和行向量，然后先用行向量卷积再用列向量卷积

(2) 可迭代滤波器->迭代法： $t_{r,c} = t_{r,c-1} + g_{r,c+m} - g_{r,c-m-1}$

### 高斯滤波

# 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

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

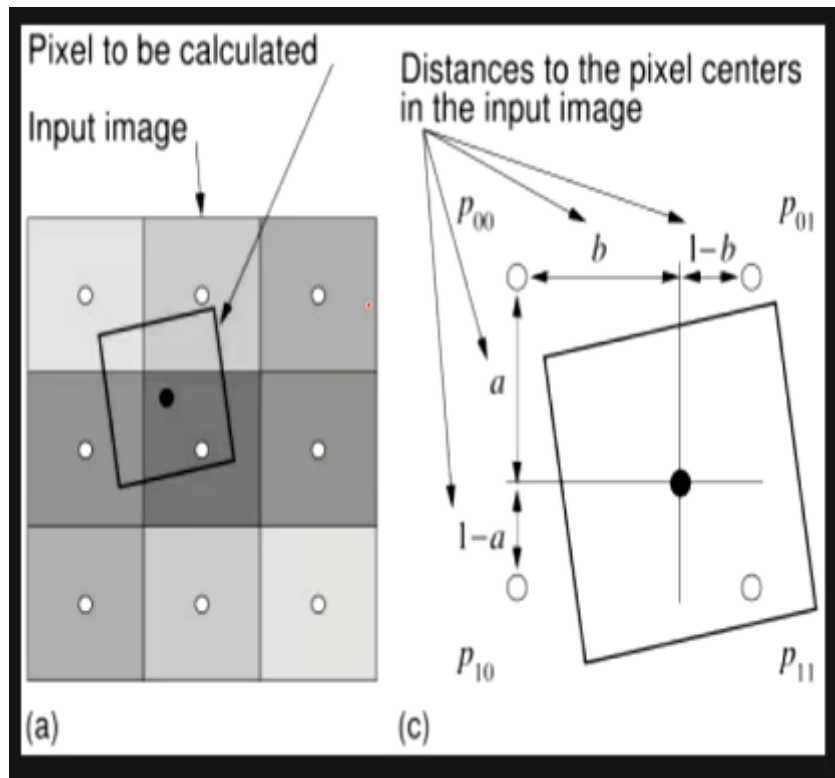
- 2、单应矩阵：自由度为8

### (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



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

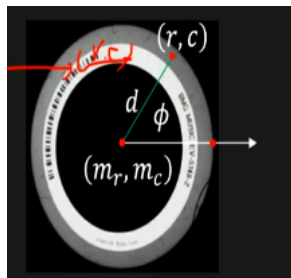
3、双三次插值

#### (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

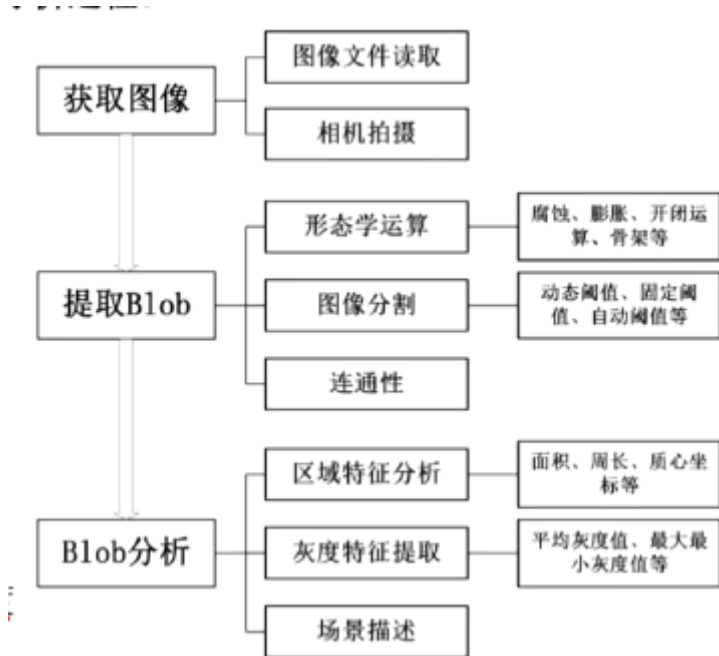
第一个区域的灰度分布作为参考, 让下一副图像的灰度分布接近, 由此减小其他因素的影响

### (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

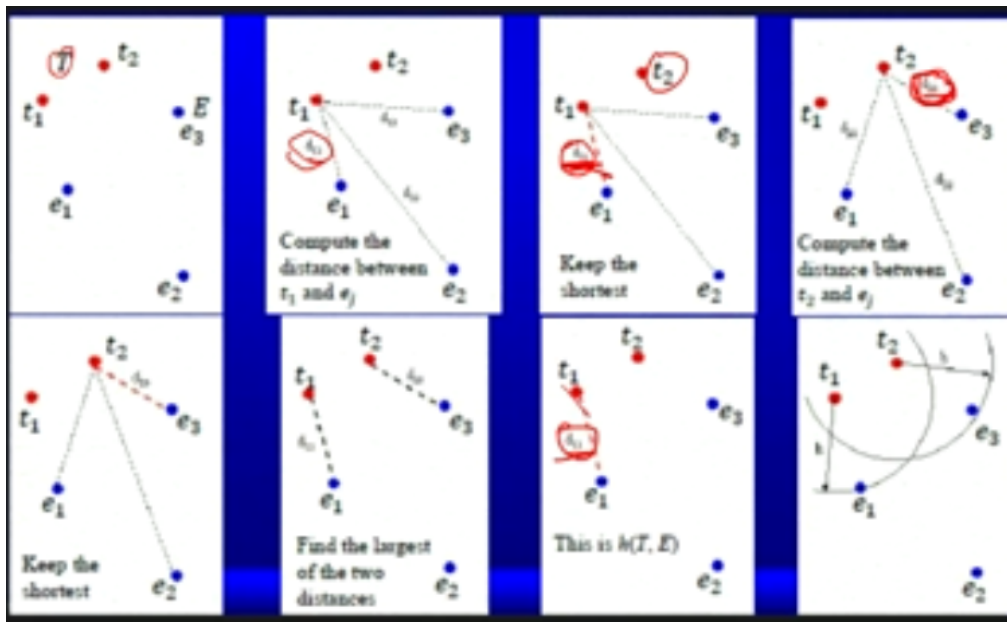
模板边缘点多于图像中边缘点时会导致均方距离过大

#### (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 $(x_c, y_c, S, \alpha)$



# Lec12 测量

## 一、1D Edge

定义：在区域内取一条线段，考察线段上像素的分布



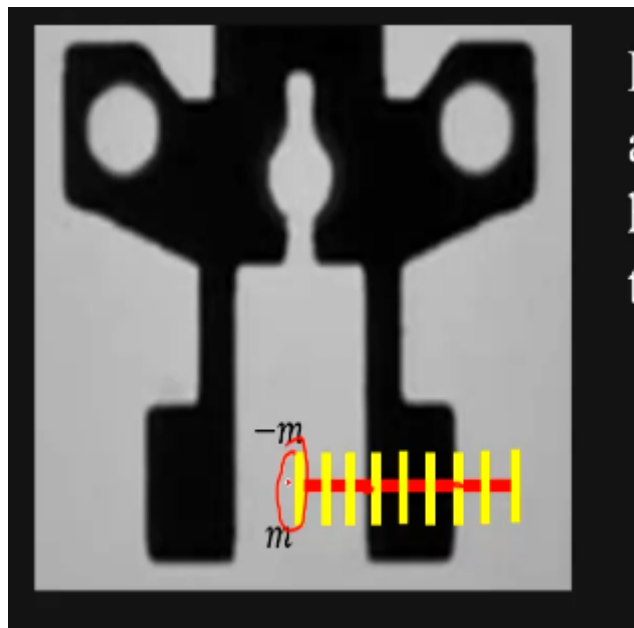
### (一) 极性

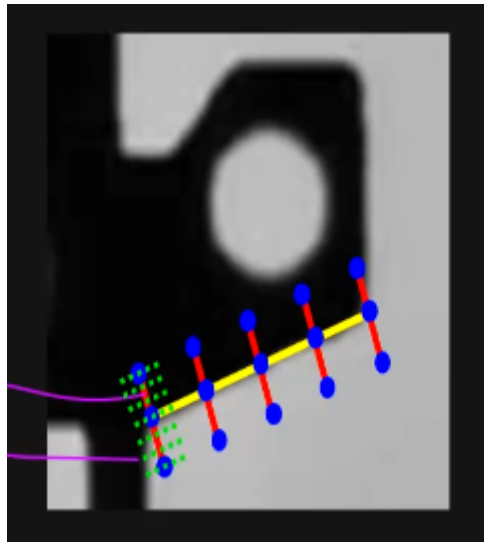
- 1、提取边缘的方向可能会影响边缘的质量
- 2、从一阶导图像能够看出正边缘和负边缘
- 3、ROI可以规定方向，由此区分正边缘和负边缘

### (二) 减少噪声

#### 1、等间隔取样平均

处理无序噪声





## 2、平滑profile

等间隔取样平均后的结果再平均，可以处理有序噪声

## 3、平滑profile:平滑+求导

Canny:高斯核用于平滑，高斯核导数用于计算一阶导，算法能分离不能迭代（递归）

Deriche:能够分离和迭代（递归）

## 4、NMS

沿着profile考察各个梯度点，梯度值大于附近两个时为目标边缘

### （三）亚像素

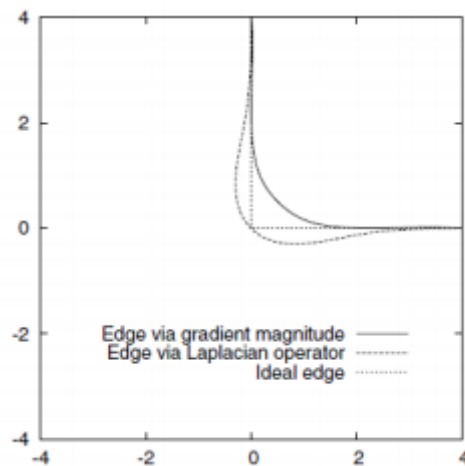
沿着profile考察各个梯度点，用邻近的两个边缘点和他本身拟合一条二次曲线，定点是亚像素级别的边缘

图像目标有旋转和平移怎么办？

模板匹配->仿射变换矩阵->保证ROI区域不变

## 二、2D Edge

### （一）一阶导和二阶导求取边缘的不同

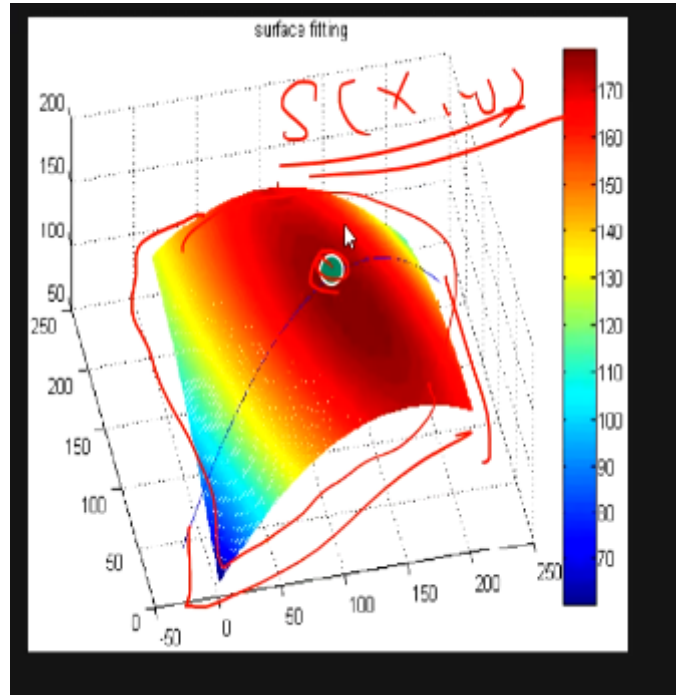


二阶导能够包含角点信息，一般大于实际轮廓----->更多凹凸细节

一阶导一般小于实际轮廓，更多保留平滑边缘----->更加圆滑

## (二) 二维亚像素

对于边缘像素：考察他的  $3 \times 3$  邻域，用9个数据拟合二次曲面（6个参数），在梯度方向构建垂直于图像平面的平面，求平面和二次曲面的交线上的极值点，就是亚像素精度边缘点



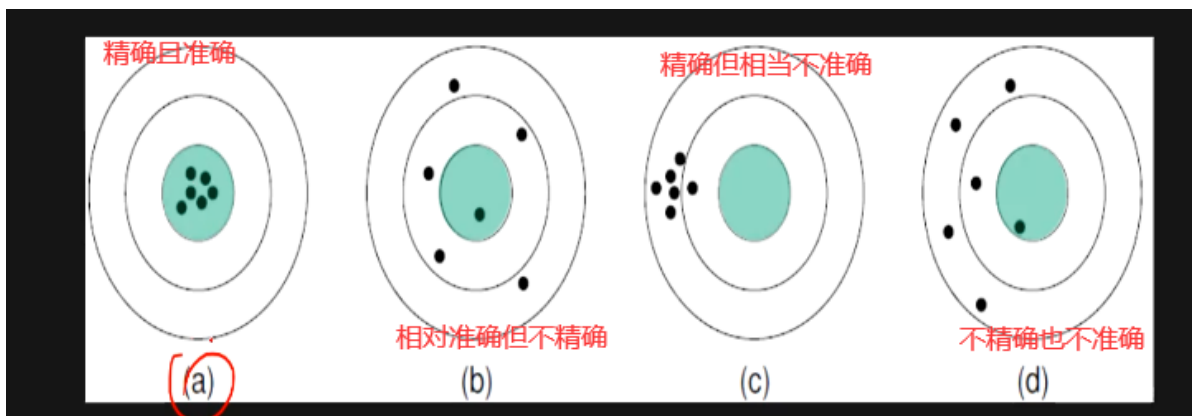
## 三、精度

### (一) 准确度

真实精度：测量值和真实值的偏差，依据绝对误差表示

### (二) 精确度

重复精度：测量值和平均值的接近程度，依据方差表示



## 四、拟合

### (一) 最小二乘直线拟合

## 1、代价函数

$$\epsilon = \sum(\alpha r_i + \beta c_i + \gamma) - \lambda(\alpha + \beta - 1)n$$

矩形拟合形式：有点复杂，摆烂了

## 2、削弱远离直线点的影响:给远离直线的点更小的权重

IRLS算法

- (1) 取权重都是1->拟合一条曲线->根据各个点距离直线的远近确定新的权重
- (2) 使用新的权重拟合直线

## (二) 圆拟合

# 五、轮廓分割

## (一) ?

## (二) 分割直线

1、用途：分段拟合

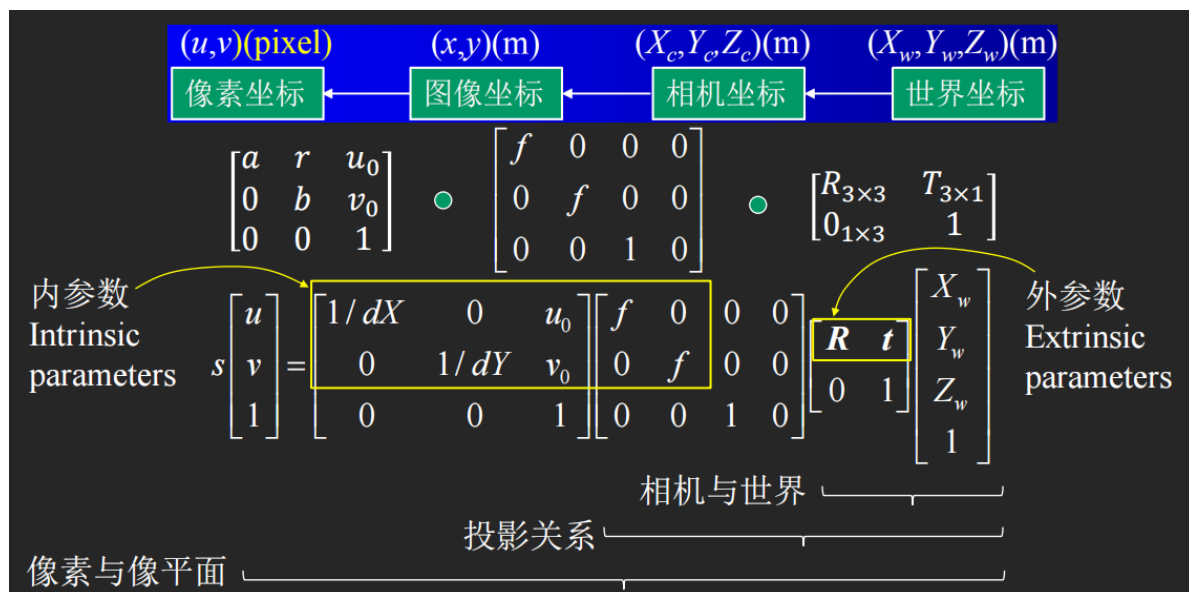
2、算法：定义 $d_{max}$ 为最大容许距离，在区域内不断进行分割，知道边缘点到分割直线的距离小于 $d_{max}$

# Lec13

规则几何形状：1D边缘

不规则形状:2D边缘，再计算亚像素精度boundary

## 一、相机标定



## **二、OCR字符识别**

二值化, 像素投影, 字符分割

## **三、二维码Barcode**

## **四、三维识别**

结构光、飞行时间

## **五、缺陷检测**