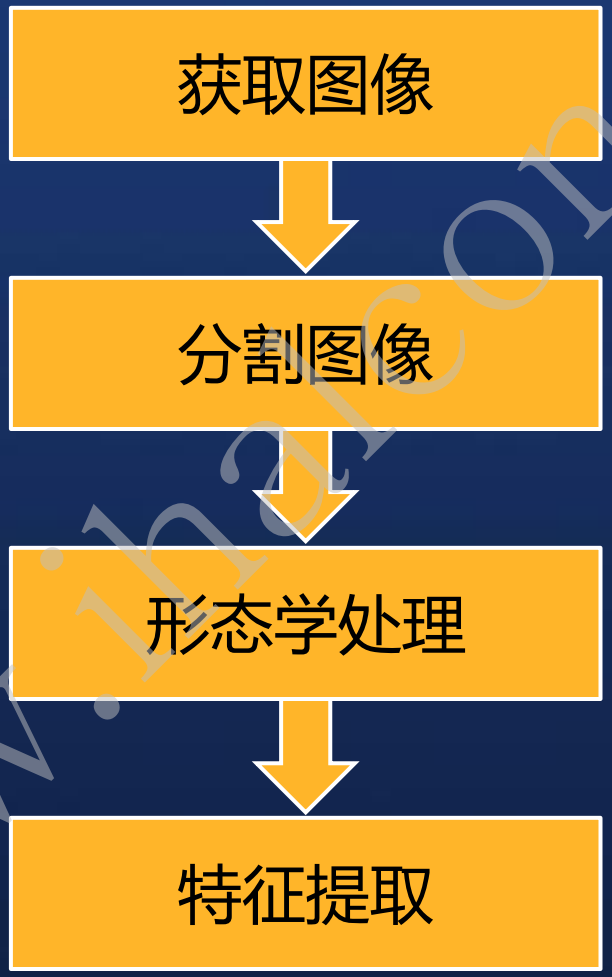
BLOB分析

主讲人： 杨永勇



分割方法——分割图像

阈值

- Global, fast, automatic, binary, character, dual, hysteresis

比较

- Difference checking, dynamic threshold

区域生长

- Local, multi channel, expand

拓扑

- Watersheds, pouring, maximums, plateaus

threshold算子——分割图像

- threshold 算子
  - 最简单的
  - 最快的
  - 使用频率最高的方法
- 如果目标体与背景之间存在灰度差，则threshold首先被使用
- 对于一些困难的案例，可以用使用阴影校正
- threshold 算子的定义:

$$R' = \{(x, y) \in R \mid g_{\min} \leq g(x, y) \leq g_{\max}\}$$

- 如果环境稳定，阈值可在离线状态下一次确定
- 如果照明或者物体表面是变化的，对比度可以标准化或者每幅图像分别确定一个阈值

Threshold的确定——分割图像

- 对于照明条件变化的情况下，确定阈值的另外一种方法就是图像的直方图
- 假设物体和背景之间有着明显的灰度差
- 直方图将会有明显的两个波峰：一个是物体，另一个是背景
- 在物体和背景之间的最小值
- 另外假设物体有稳定的灰度值
- ⇨ 这些阈值对于物体而言是直方图中的最小值
- 问题: 通常这些最小值不是特明显
- ⇨ 平滑直方图：比如1D高斯滤波

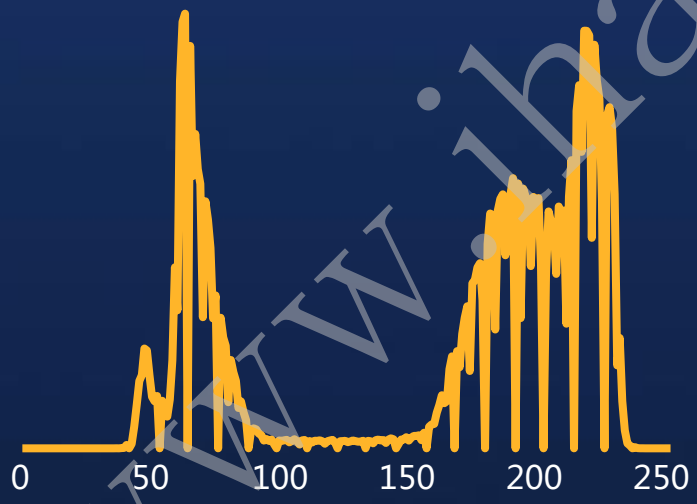
分割: 直方图——分割图像

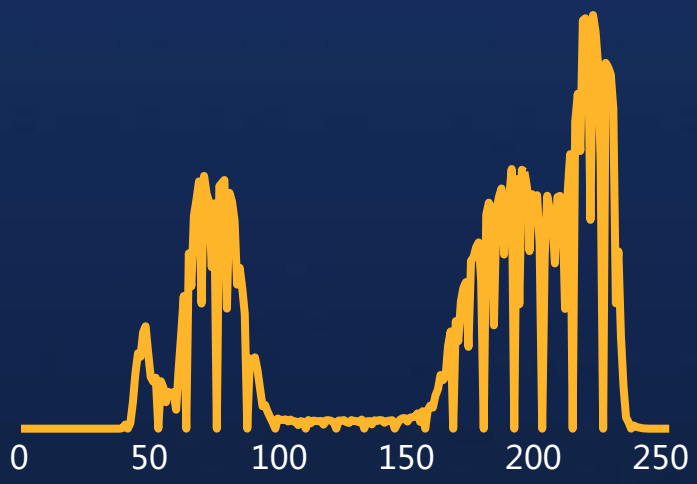
自动全局阈值分割方法

- 计算直方图
- 寻找出现频率最多的灰度值（最大值）
- 在threshold中使用与最大值有一定距离的值作为阈值

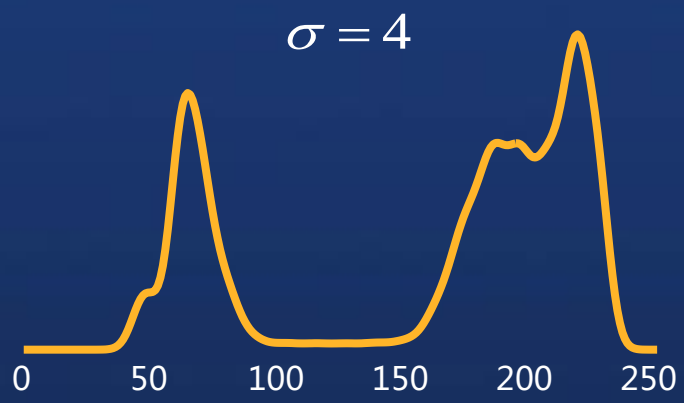
```
gray_histo (Image, Image, AbsoluteHisto, RelativeHisto)
PeakGray := sort_index(AbsoluteHisto)[255]
threshold (Image, Region, 0, PeakGray-25)
```

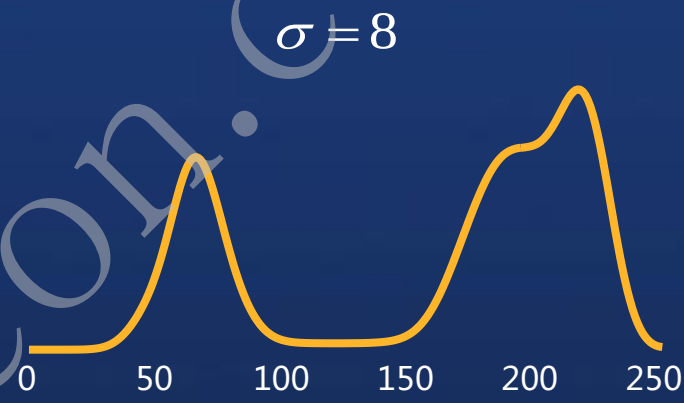
Threshold的确定——分割图像





Threshold的确定——分割图像



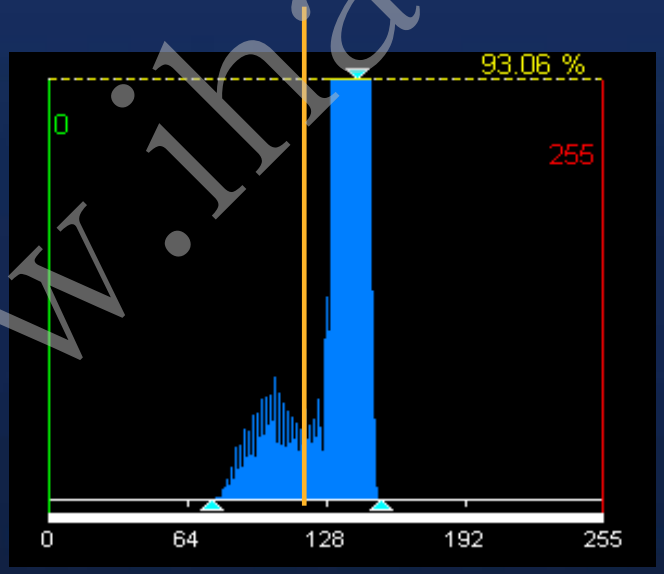


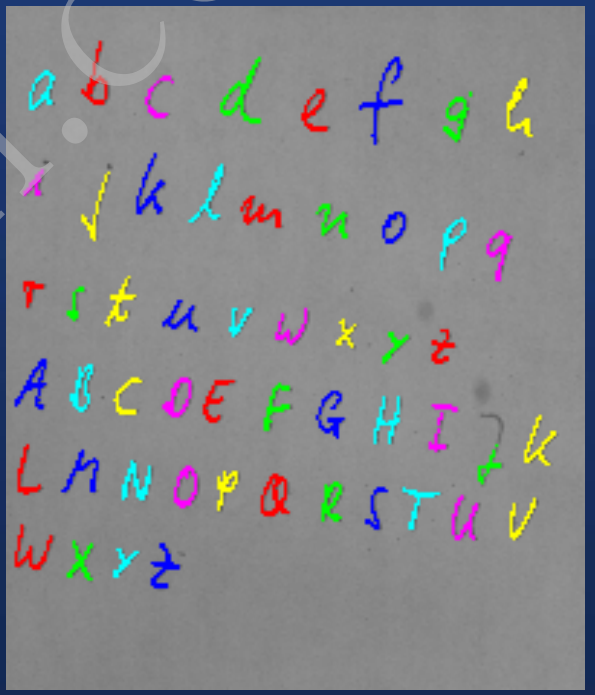
分割: Binary Threshold ——分割图像

自动地全局阈值方法

- 多次迭代平滑灰度直方图
- 查找两个波峰
- 使用threshold找到两波峰之间的最小值

```
bin_threshold (Image, Region)
```





分割: Dynamic Threshold ——分割图像

- 对于一些应用来说，确定一个全局阈值是不可能的，比如，因为
  - 没有通用的参考图像来确定阴影校正
  - 图像的背景是非均匀的
- 物体在局部范围内通常比背景亮些或者黑些
- 在这种情况下，寻找一个固定阈值来区分物体和背景是不太容易的
- 问题: 局部邻域的确定
- 方法：局部邻域可以由平滑滤波器来确定 (比如, **mean_image** 或 **binomial_filter**)

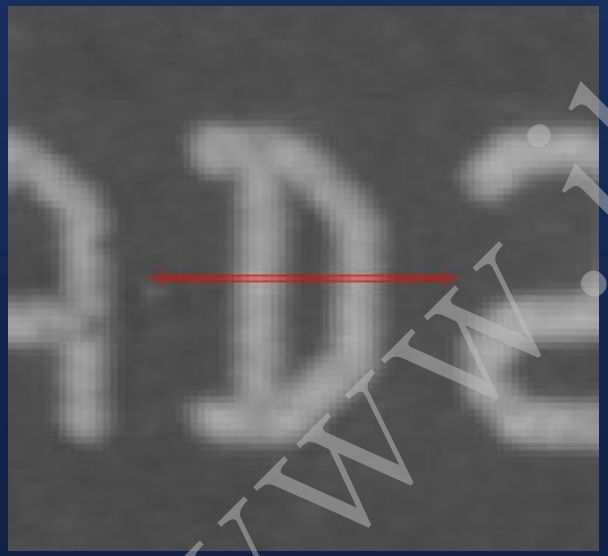
分割: Dynamic Threshold——分割图像

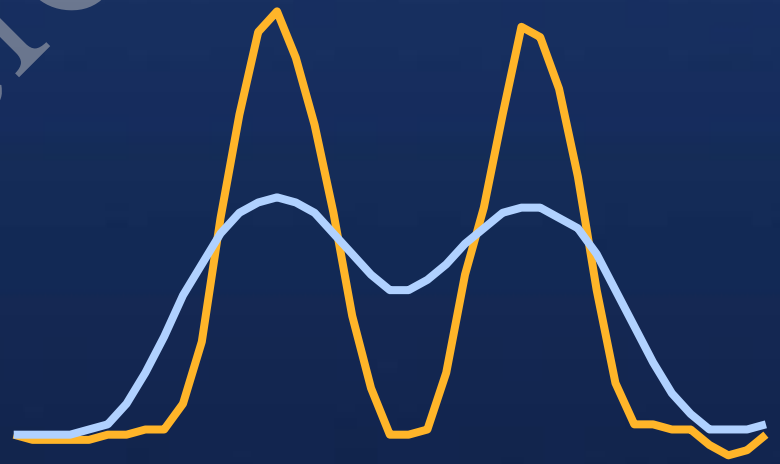
dynamic threshold的定义： s表示输入的平滑图像

$$R' = \{(x, y) \in R \mid g(x, y) - s(x, y) \geq t\}$$

or

$$R' = \{(x, y) \in R \mid g(x, y) - s(x, y) \leq -t\}$$



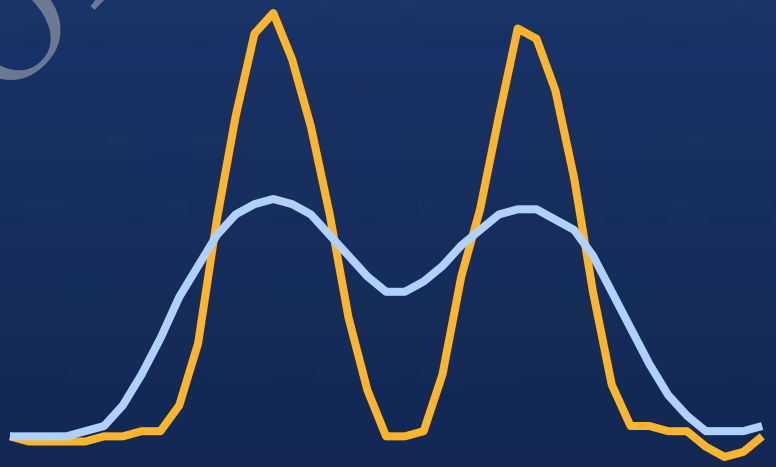


灰度值轮廓图

平滑后的灰度轮廓图

分割: Dynamic Threshold——分割图像

- 滤波的mask尺寸确定了做能分割出来物体的最大尺寸
- 经验之谈
  - `mean_image`的滤波尺寸 > 被提取物体的直径
- 类似条件应用到Gauss filter
- 如果滤波掩码尺寸太大，那么相邻非常近的物体将会连在一块
- 动态阈值也会返回沿着边缘的处理结果



灰度值轮廓图
平滑后的灰度轮廓图

分割: Dynamic Threshold——分割图像

- 局部阈值方法
- 通过平均图像灰度来确定局部邻域
- 使用的滤波掩码尺寸大于字符的笔划宽度,推荐2×D+1
- 选出的所有像素要比局部邻域要黑，也就是比均值滤波后的图像要黑

```
mean_image (Image, ImageMean, 21, 21)
dyn_threshold (Image, ImageMean,
               Region, 15, 'dark')
```

高级动态阈值算子——分割图像

- 动态阈值是一种使用广泛的分割方法，在很多应用领域中是比较重要的
- HALCON提供了一个高级算子: **var_threshold**
- 这个算子有着以下特征：
  - 较好地分开前景和背景
  - 对不合适的参数设置不敏感
  - 因为只使用单个算子就能实现，所以编程容易
- 缺点是相比**dyn_threshold**而言需要更长的执行时间

动态阈值：不同的图像——分割图像

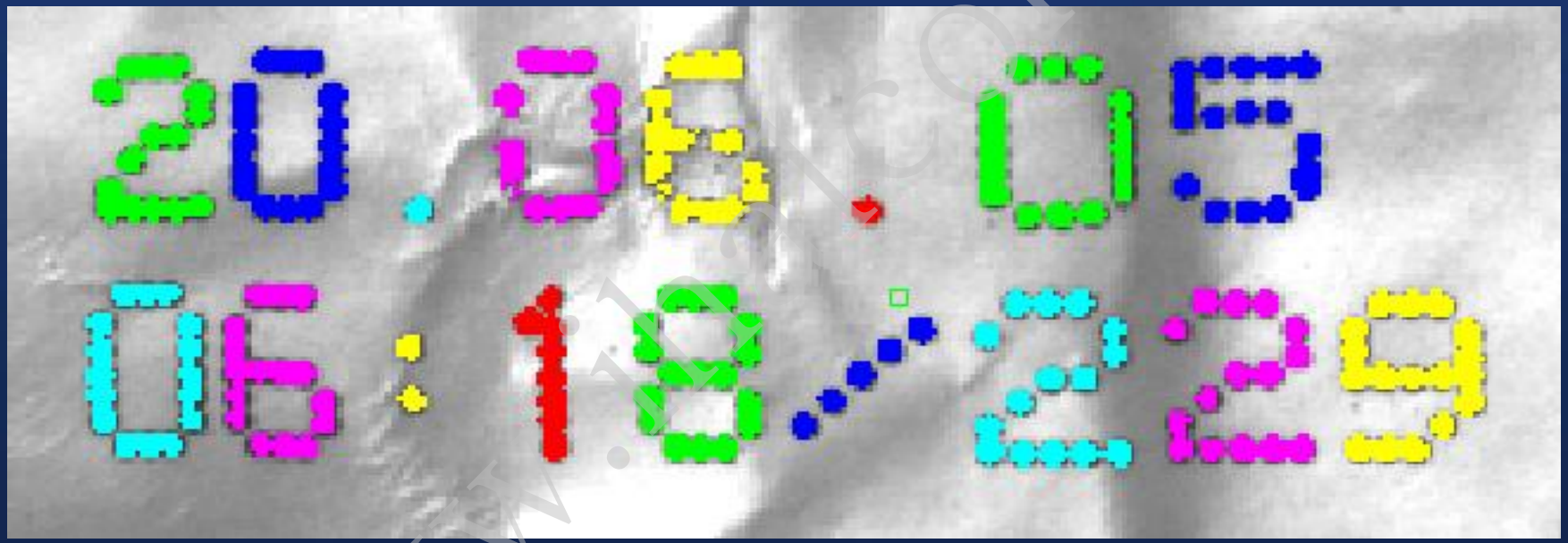
原始图像：
带有不均匀的背景

dyn_threshold：
在字符周围会有“电晕”

var_threshold：
每个字符有着更均匀的背景

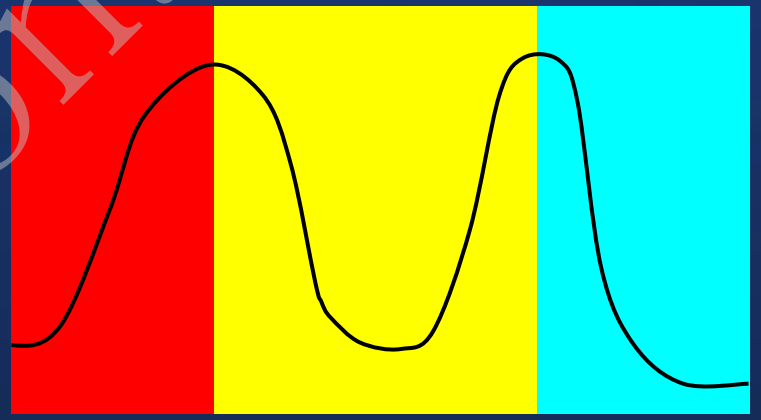
var_threshold: 示例——分割图像

光学字符识别



分水岭分割——分割图像

- watersheds 是基于灰度的拓扑学来分割一幅图像
- 一幅图像可以诠释为一座山脉：较高的灰度值作为山峰，同时较低的灰度值作为山谷
- 在山脉中提取出分水岭
- 在两个盆地之间相应会有一个山脊
- 参数Basins 包含了这些盆地, 同时 Watersheds包含了一些分水岭



- Watersheds 会每幅图像返回单个区域, 同时Basins 包含一系列区域，就是每个不相交的盆地
- 在多数情况下，在分割之前去平滑图像是非常必要的

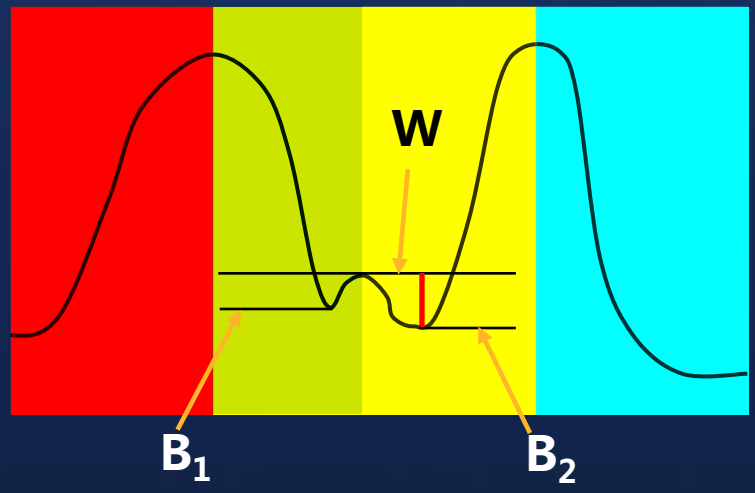
高级Watershed 分水岭——分割图像

- 如果对比度差异比较少，`watersheds_threshold` 可以使得相邻的盆地合并
- 即使在有噪声的条件下，也可以分割区域
- 噪声降低滤波器的应用可以被忽略，这将会提高形状提取的精度
- 这算子比如`watersheds`, 支持`uint2` 和`float` 图像

方法：

- 分割盆地可以使用常规的分水岭算法
- 如果相邻的一对盆地的对比度比给定的阈值要小，那么它们可以成功地合并:

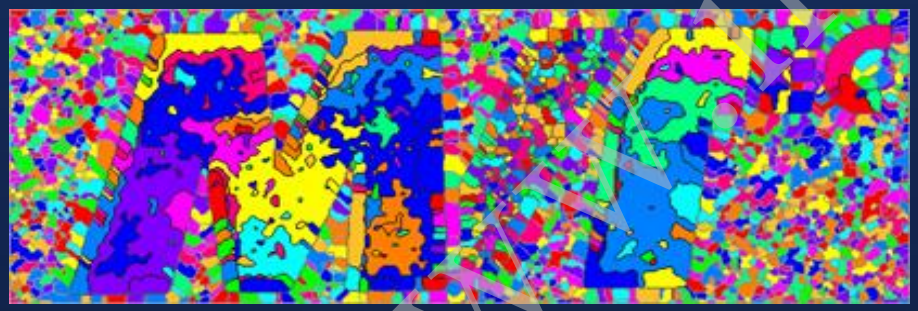
$$\max(W - B_1, W - B_2) < Threshold$$



Watershed: 示例——分割图像

比较: `watersheds` ↔ `watersheds_threshold`

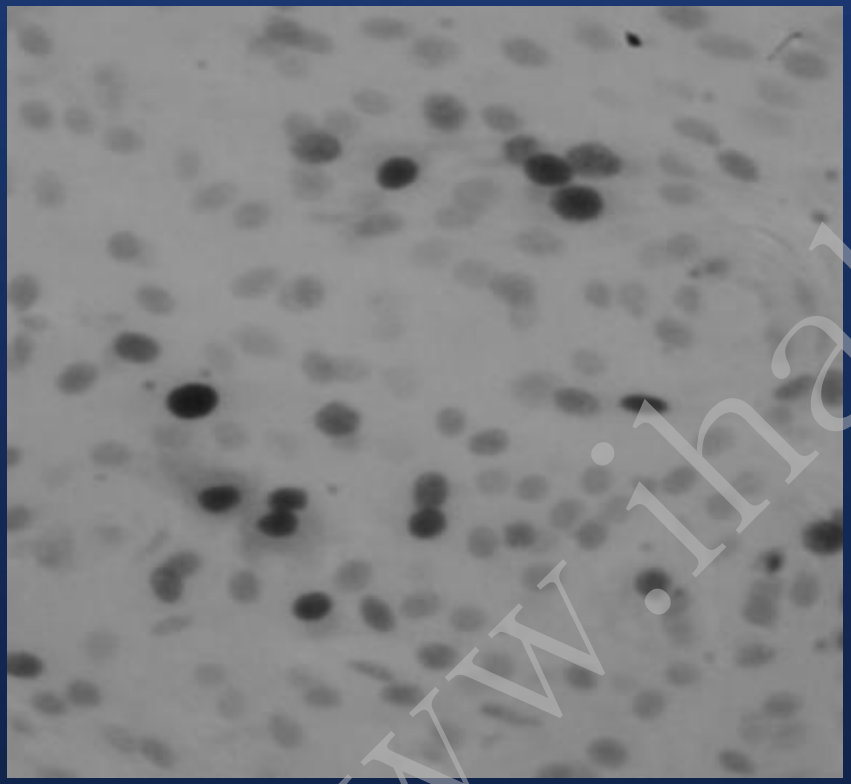
`watersheds`

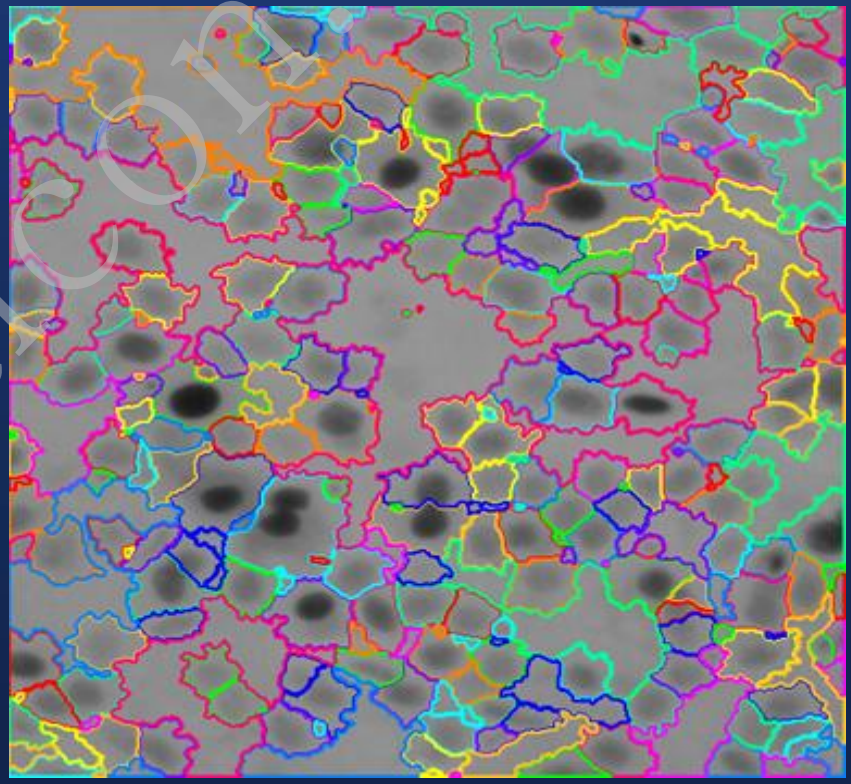


`watersheds_threshold`

Watershed示例——分割图像

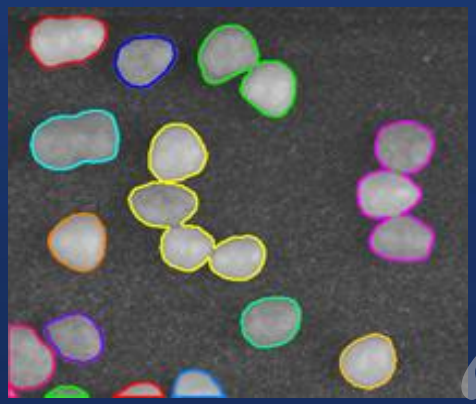
医学: 细胞分割

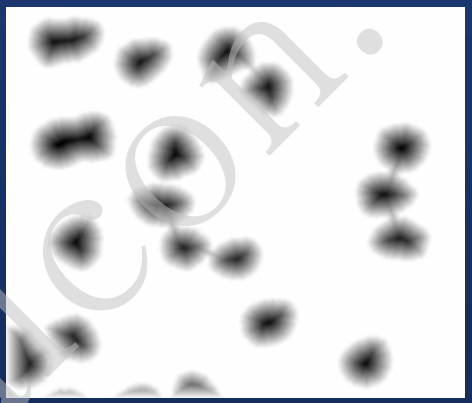


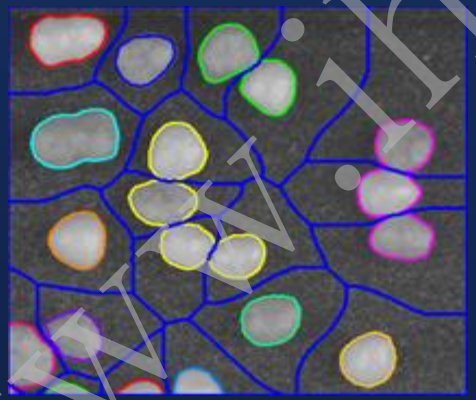


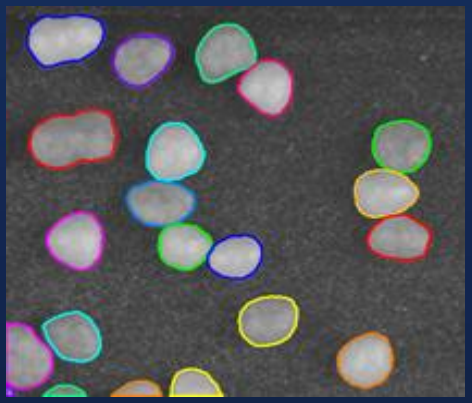
Watershed: 示例——分割图像

制药:使用distance transformation来计算粒子个数







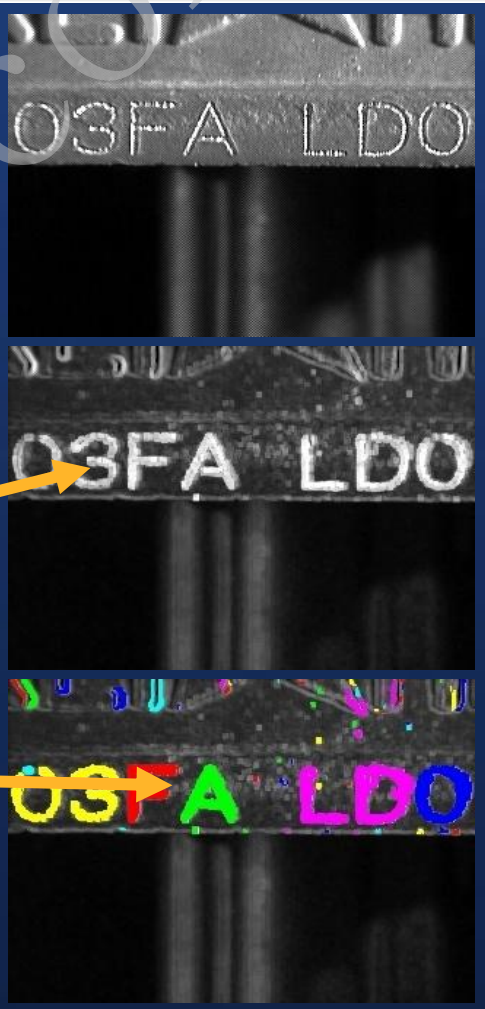


分割: 金属表面——分割图像

纹理滤波

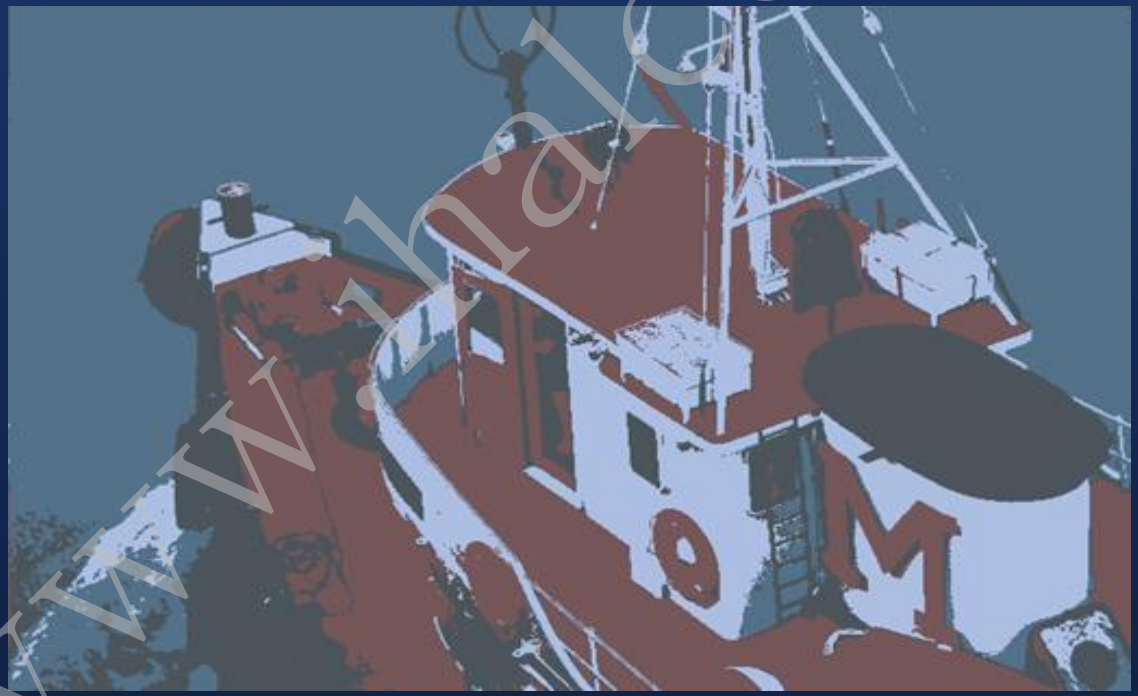
- 增强了由于字符本身反射所引起的局部高亮对比度
- 在高通滤波的结果上选择区域
- 去除小区域

```
gray_range_rect (Image, ImageResult, 7, 7)
threshold (ImageResult, Region, 128, 255)
connection (Region, ConnectedRegions)
select_shape (ConnectedRegions, SelectedRegions,
              'area', 'and', 1000, 99999)
```



多通道分类——分割图像

- HALCON 提供了多个分类器 (SVM, MLP, GMM), 可以用来多通道像素分类
- 最简单的情况下，可以应用到彩色图像上
- 因为有好的精度，所以能获得一个很好的分割效果



算子: 形态学——形态学处理

- 特征是独一无二的
- 任意的结构元素
- 任意尺寸的结构元素
- 非常有效的处理
- 有62个相关算子
- 经典算子
  - `Erosion, dilation, opening, closing`
- 高级算子
  - `top-hat, bottom-hat, hit-or-miss, boundary`
- 特殊算子
  - `Fitting, pruning, thickening, thinning, skeleton`

Union ——形态学处理

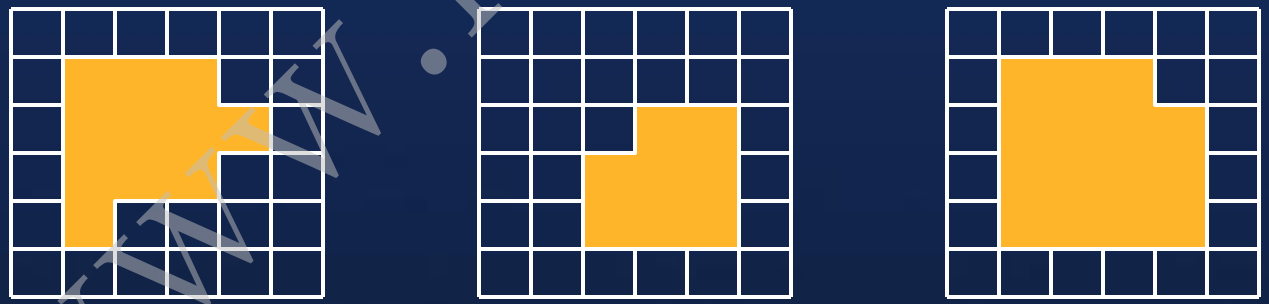
定义 $R \cup S = \{x \mid x \in R \vee x \in S\}$

算子

- **union1**: 把所有region合并成一个变量
- **union2**: 把第二个参数里的所有区域统一到第一个参数的每一个区域中去

用法

- 结合原有的形状来生成区域
- 合并分割结果



R S $R \cup S$

Intersection ——形态学处理

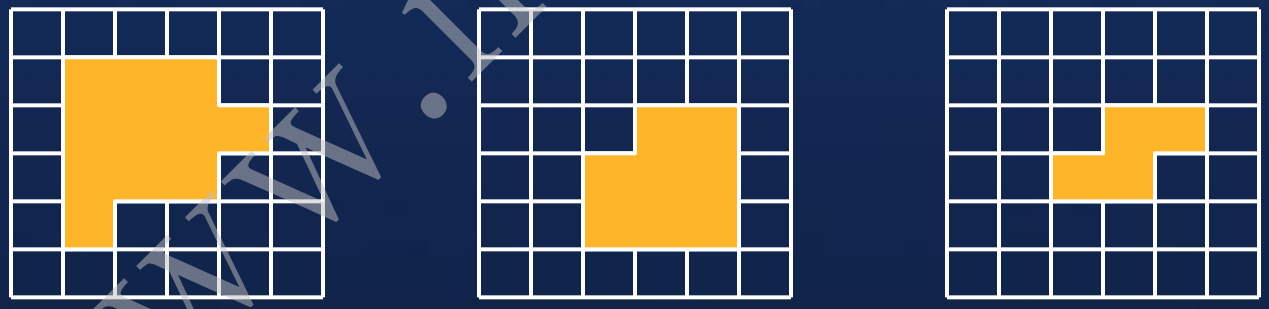
定义 $R \cap S = \{x \mid x \in R \wedge x \in S\}$

算子

- **intersection**: 第一个参数中的每一个区域与第二个参数中的所有区域进行相交

用法

- 结合原有形状来生成图像
- 返回两区域共同点



Difference ——形态学处理

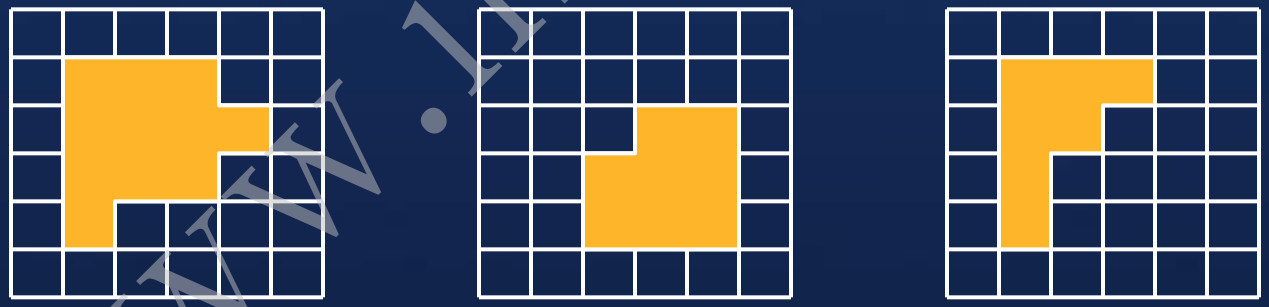
定义 $R \setminus S = \{x \mid x \in R \wedge x \notin S\}$

算子

- **difference**: 去掉第二个参数与第一个参数共有的区域

用法

- 结合原有形状来生成图像
- 返回的点是在一个区域中出现但不在另一区域中



Complement ——形态学处理

定义 $\overline{R} = \{x \mid x \notin R\}$

算子

- `complement`: 计算每个输入区域的补集

用法

- 得到的结果不是分割出来的区域

R $\overline{R}$

Translation ——形态学处理

定义

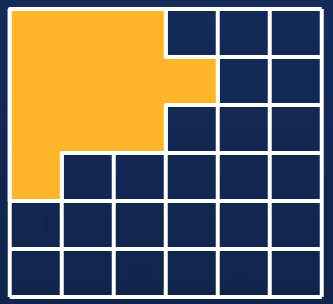
$$R_t = \{x \mid x - t \in R\} = \{y \mid y = x + t \; x \in R\}$$

算子

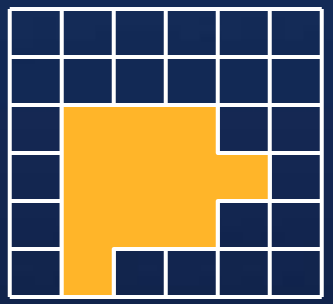
- **move_region**: 整数精度来变换区域

用法

- 适合模板区域的位置
- 用来提取边缘边界 (配合**difference**算子)



R



$R_{(2,1)}$

Transposition ——形态学处理

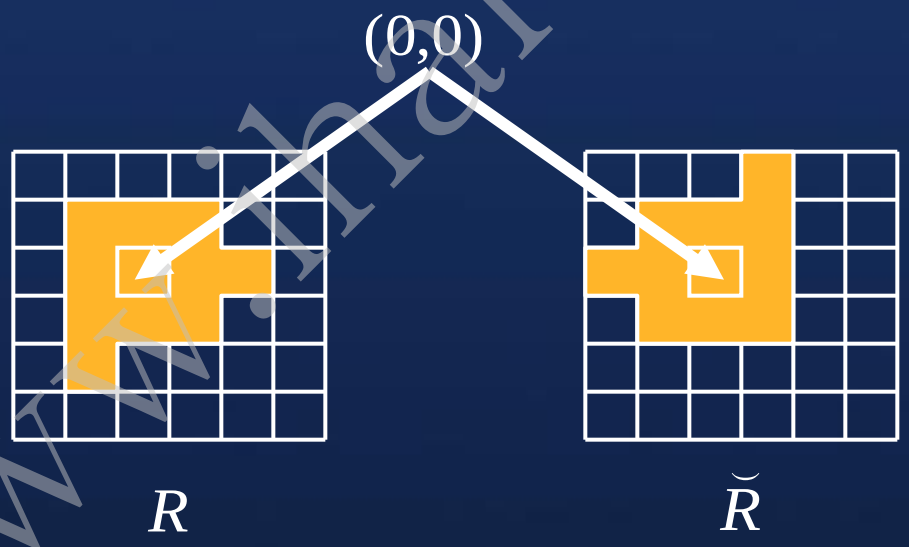
定义：

$$\breve{R} = \{-x \mid x \in R\}$$

$$\breve{\breve{R}} = R$$

$$R \subseteq S \Rightarrow \breve{R} \subseteq \breve{S}$$

$$\left|Z(\breve{R})\right| = \left|Z(R)\right|$$

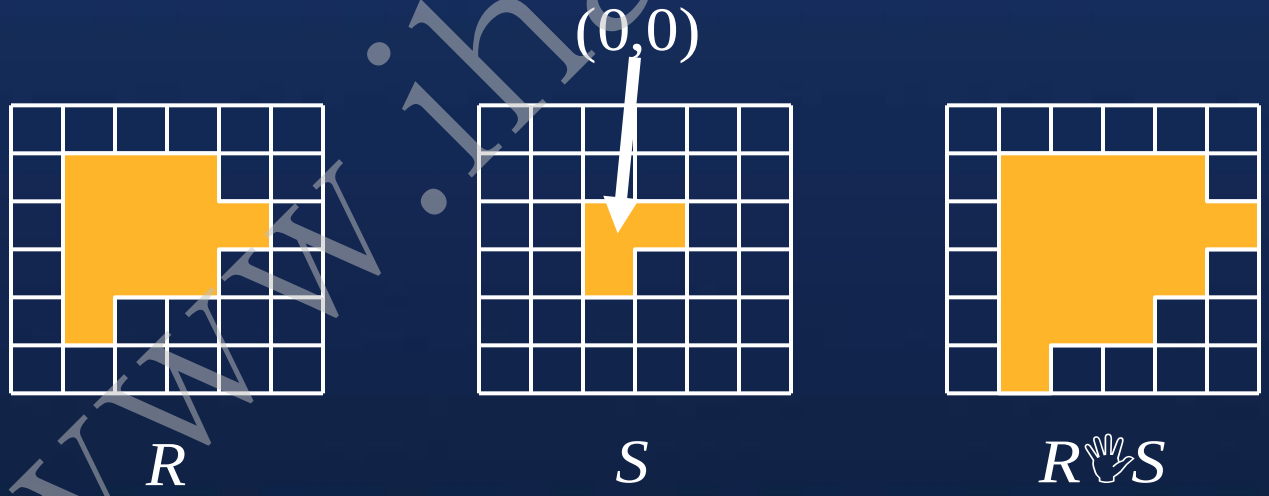


Minkowski-Addition ——形态学处理

定义:

$$
\begin{aligned}
R \oplus S &= \{r + s \mid r \in R, s \in S\} \\
&= \{t \mid R \cap (\check{S})_t \neq \varnothing\} \\
&= \bigcup_{s \in S} R_s \\
&= \bigcup_{r \in R} S_r
\end{aligned}
$$

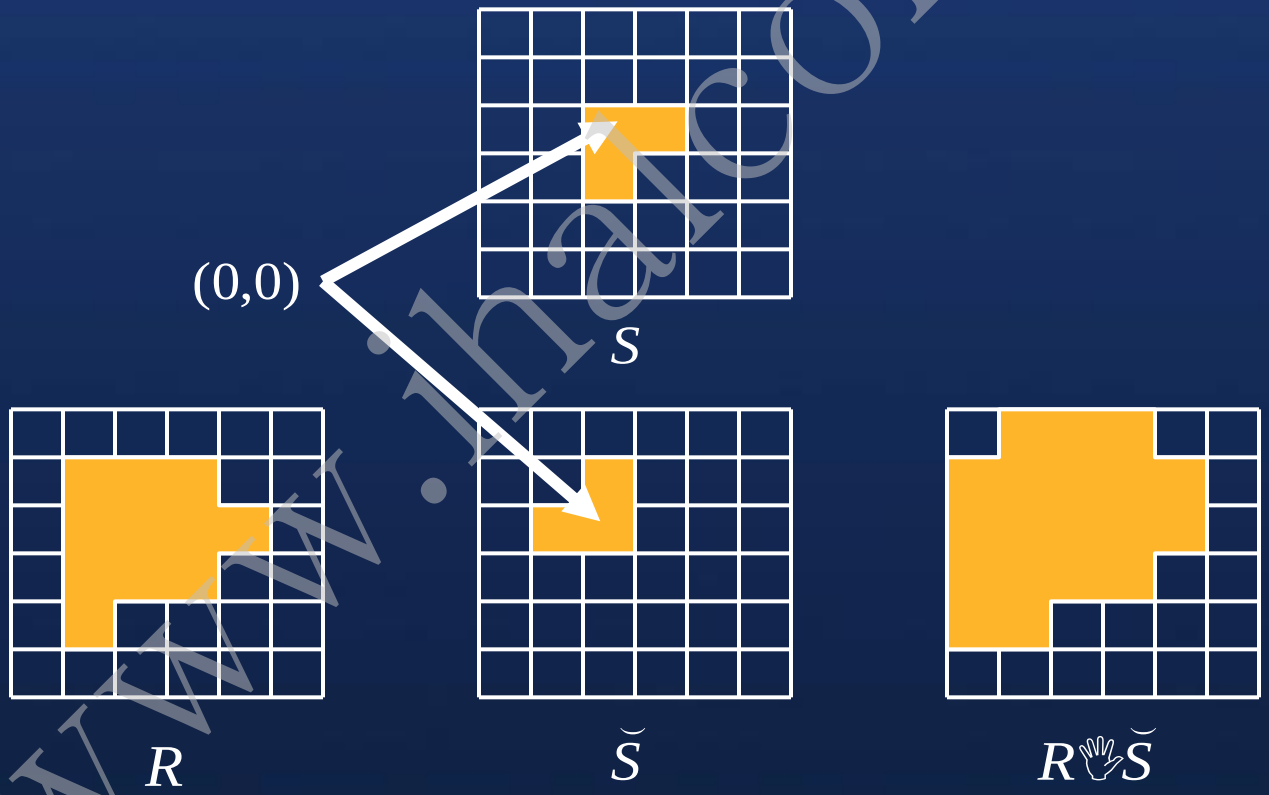
S: 结构元素



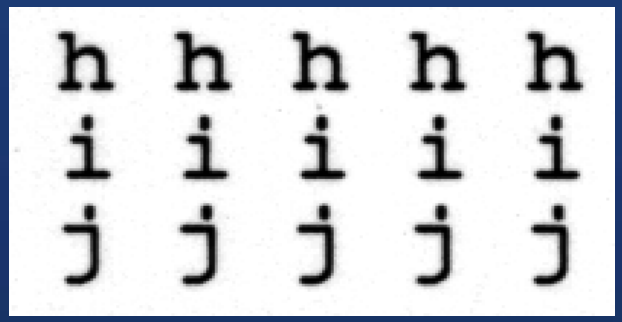
Dilation ——形态学处理

定义:

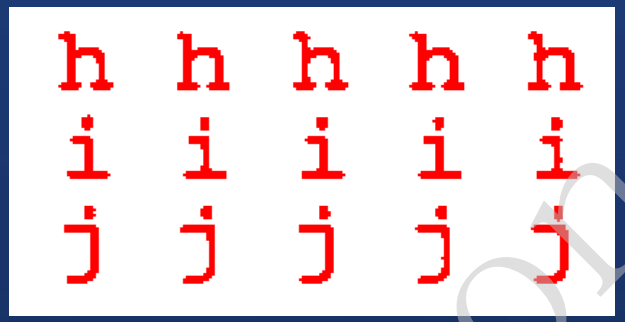
$$R \oplus \check{S} = \{t \mid R \cap S_t \neq \varnothing\}$$
$$= \bigcup_{s \in S} R_{-s}$$



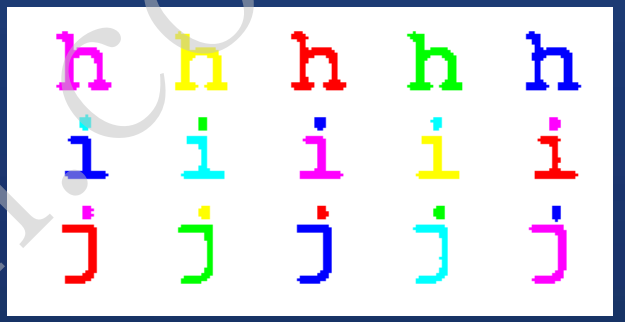
分割字符——形态学处理



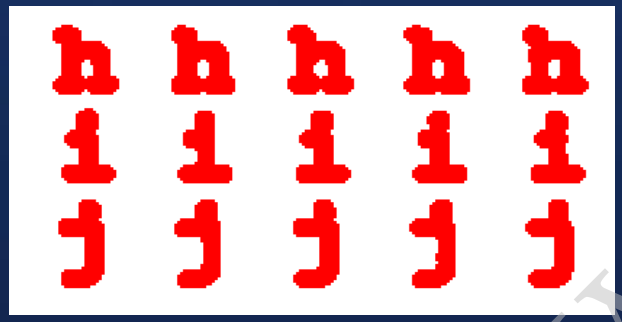
Input image



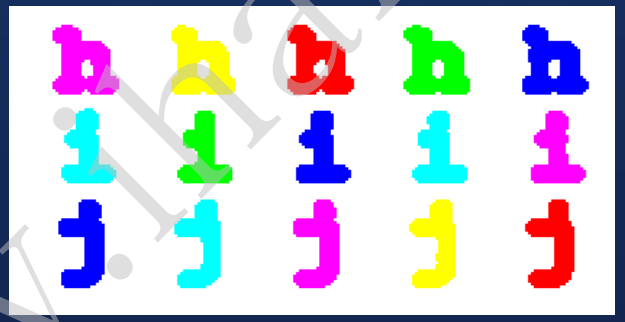
Segmented region



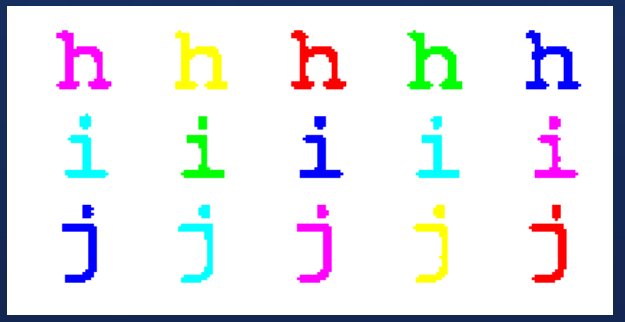
Connected components
(undesirable decomposition)



Segmented region
after applying dilation
(circle, diameter 5)

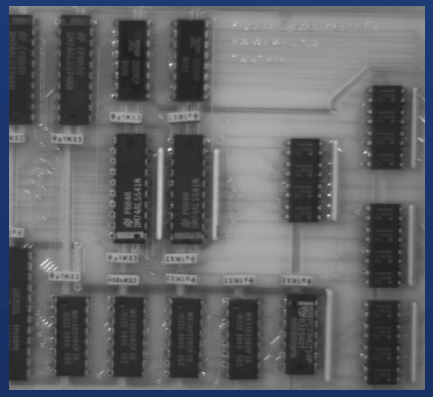


Connected components
(correct decomposition)

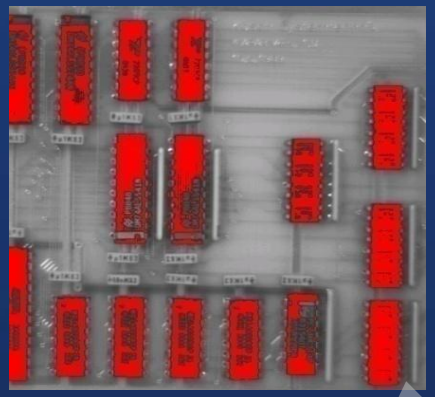


Intersection between
connected components
and segmented region

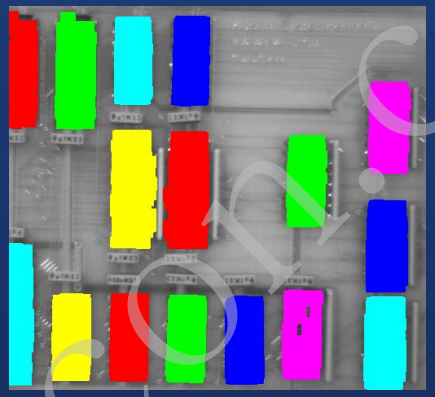
ICs 和Pins分割——形态学处理



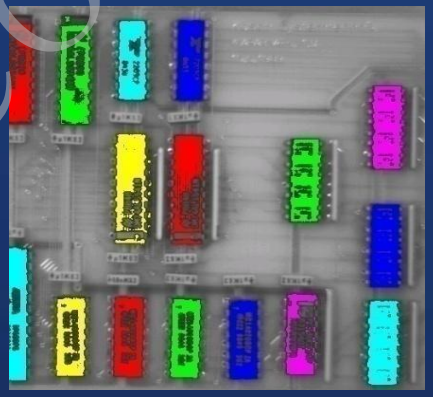
Input image



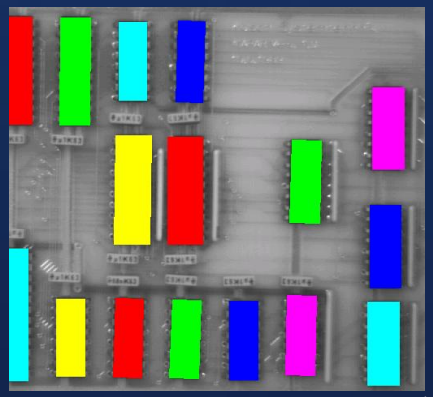
Segmentation



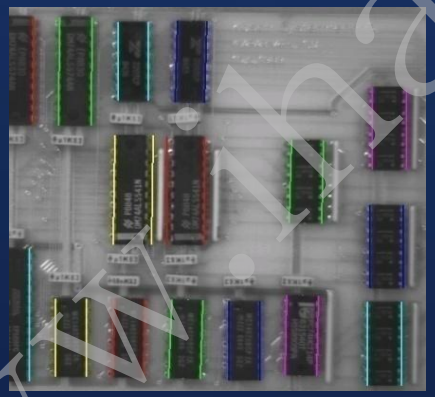
Dilation + connection



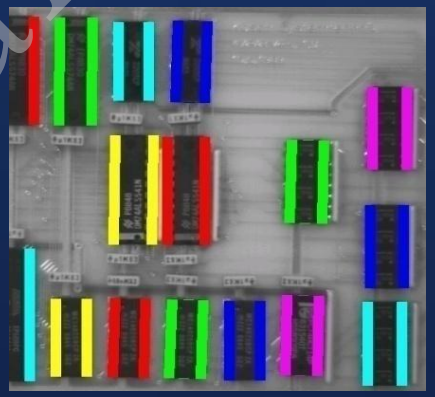
Intersection with segmentation



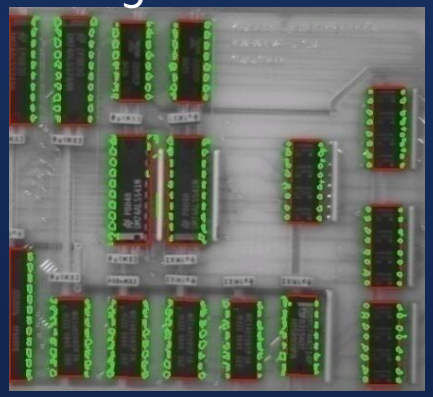
Rectangles



Diagonal border (dilation + cut)



Search area (dilation with diag. rectangle)

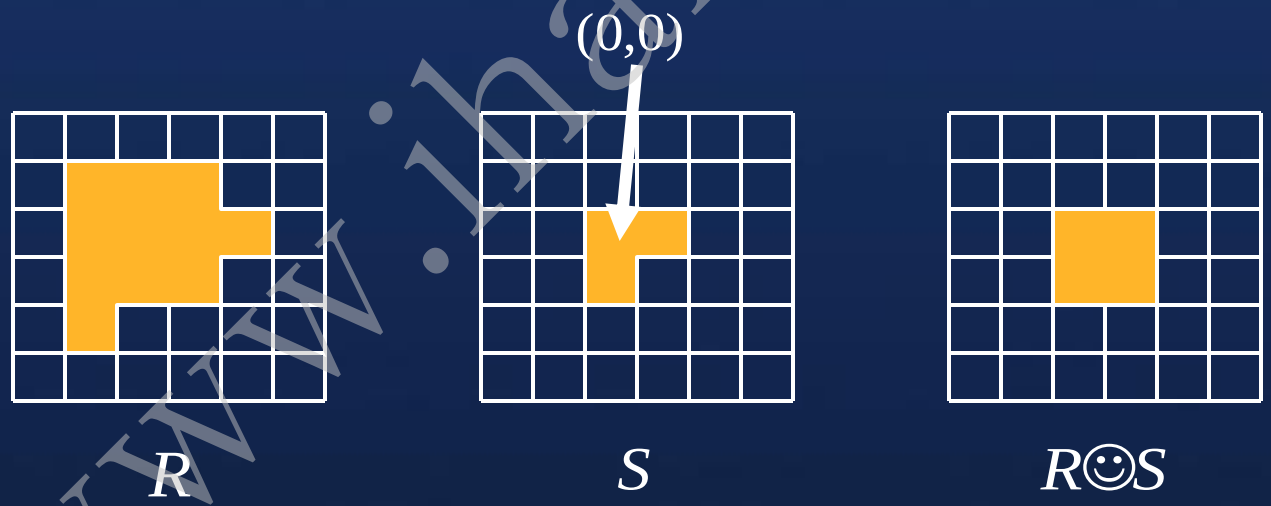


ICs and Pins

Minkowski-Subtraction ——形态学处理

定义:

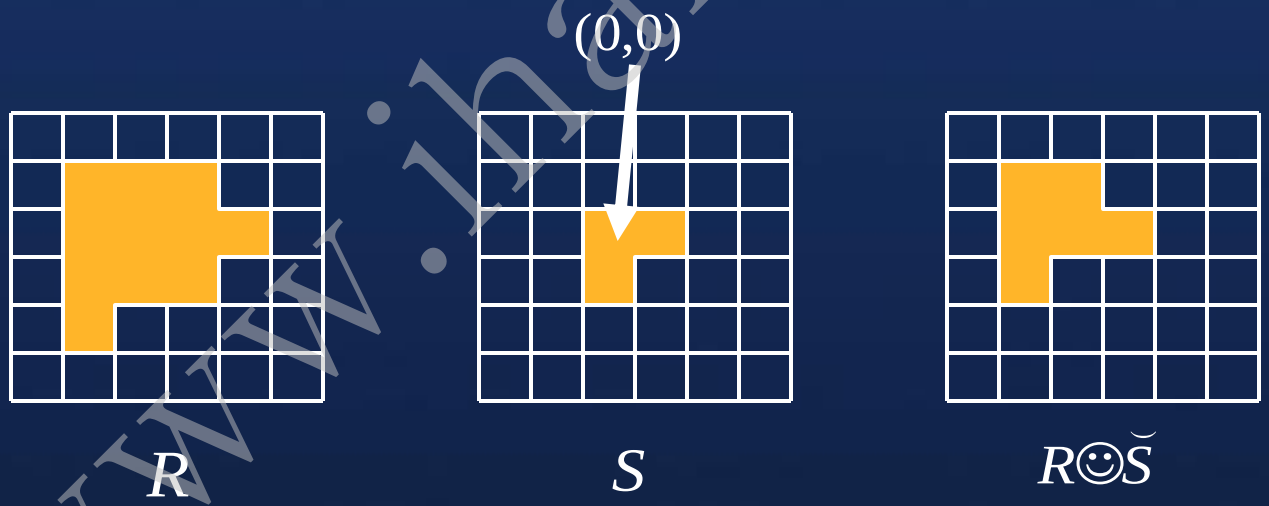
$$R \quad S = \bigcap_{s \in S} R_s$$
$$= \{r \mid \forall s \in S : r - s \in R\}$$
$$= \{t \mid (\check{S})_t \subseteq R\}$$



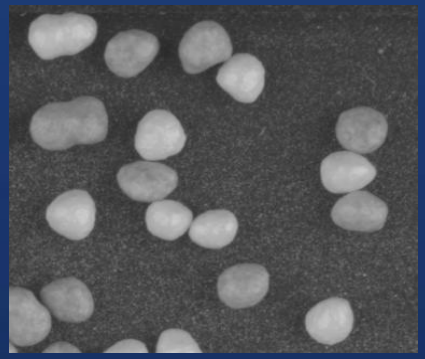
Erosion ——形态学处理

Definition:

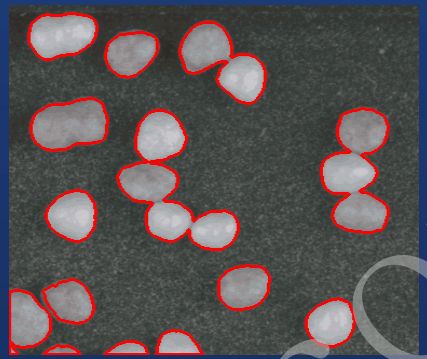
$$
\begin{aligned}
R \ominus \check{S} &= \bigcap_{s \in S} R_{-s} \\
&= \{t \mid S_t \subseteq R\} \\
&= \{r \mid r + s \in R \ \ \forall s \in S\} \\
&= \{x \mid \forall s \in S \ \ \exists r \in R : x = r - s\}
\end{aligned}
$$



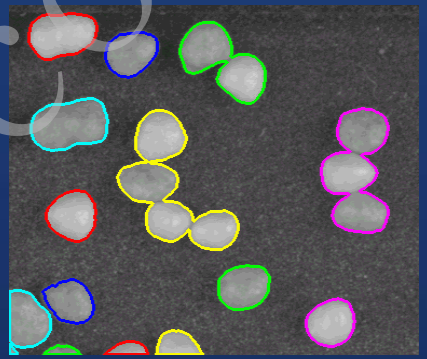
分割目标——形态学处理



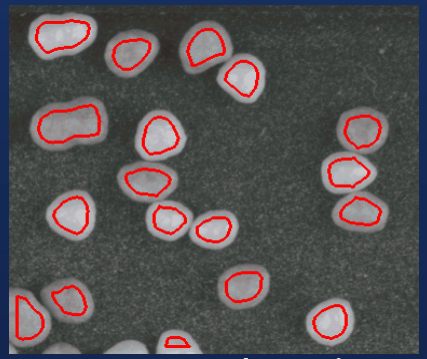
Input image



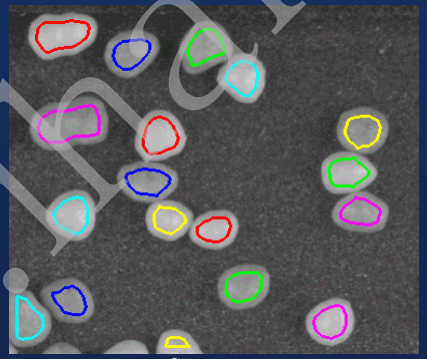
Segmented region



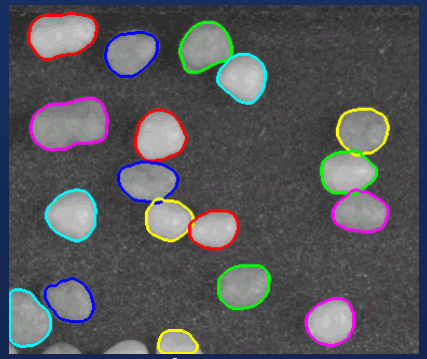
Connected components
(undesirable decomposition)



Segmented region
after applying erosion
(circle, diameter 15)

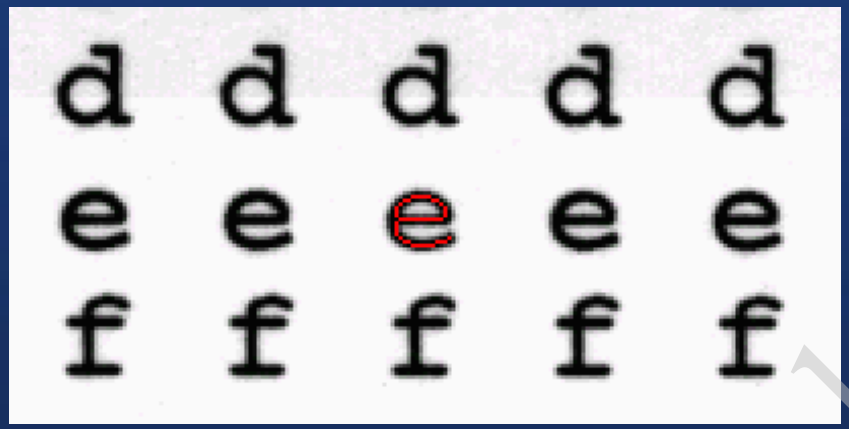


Connected Components
(correct decomposition)

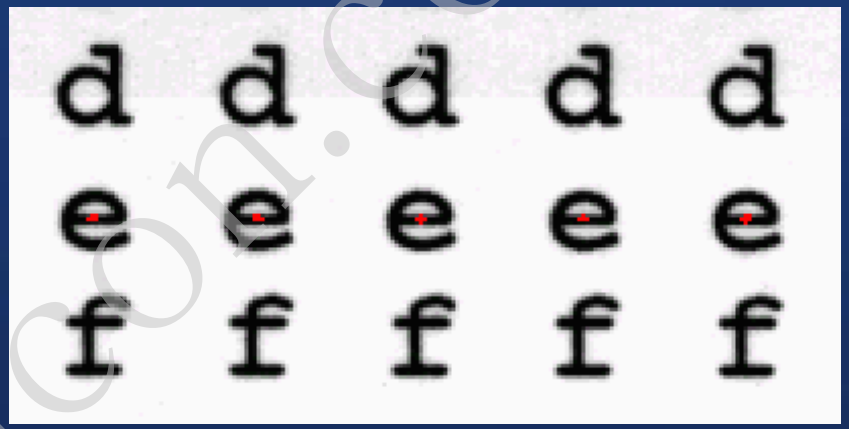


Connected Components
after applying dilation
(circle, diameter 15)

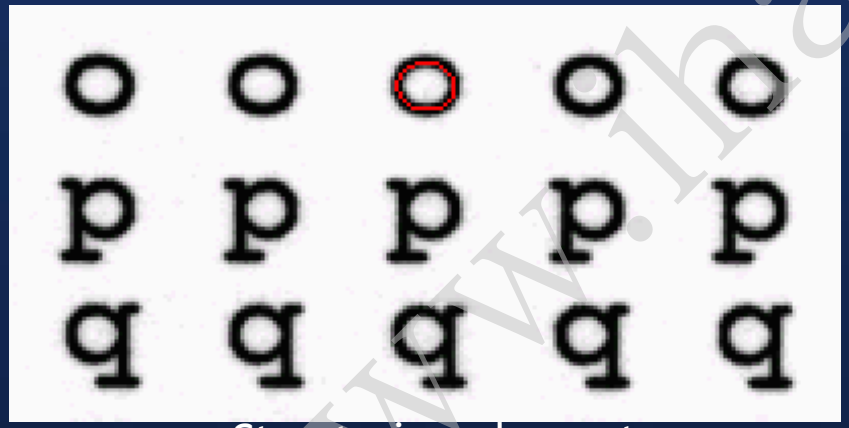
检测目标——形态学处理



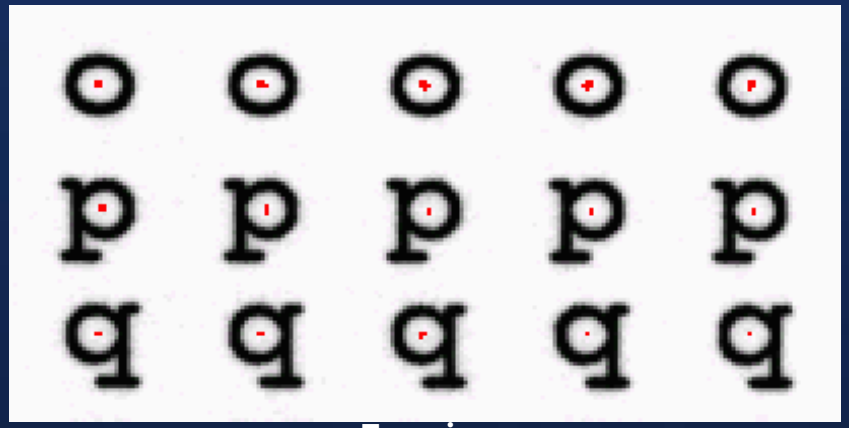
Structuring element



Erosion



Structuring element



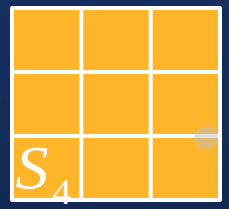
Erosion

Boundary ——形态学处理

区域边界的计算要求比较复杂

可以通过以下简单算法获得边界:

- 内部边界: $\partial R = R \setminus (R \ominus S)$
- 外部边界: $\partial R = (R \oplus S) \setminus R$
- 结构元素:



- $S_8 \Rightarrow$ 8领域来边界近似
- $S_4 \Rightarrow$ 4领域来边界近似

Boundary ——形态学处理

Input region


Inner boundary
4 neighborhood


Inner boundary
8 neighborhood


Region contour


Outer boundary
4 neighborhood


Outer boundary
8 neighborhood

Hit-or-Miss-Transformation ——形态学处理

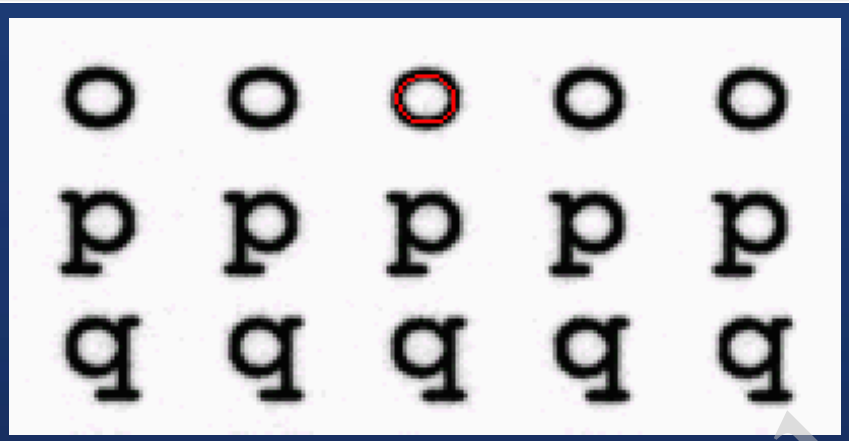
- 腐蚀操作可以被 用作检测目标 ，但是由于没有考虑背景，误报率比较高，
- hit-or-miss 变换可以对背景建立模型
- R 作为单个区域， $S=(S^{\mathrm{f}},S^{\mathrm{b}})$ 含有两个区域，同时 $S^{\mathrm{f}}\cap S^{\mathrm{b}}=\varnothing$
- hit-or-miss 变换定义如下:

$$\begin{aligned} R \otimes S &= (R \ominus \check{S}^{\mathrm{f}}) \setminus (R \oplus \check{S}^{\mathrm{b}}) \\ &= \left\{ t \mid S_t^{\mathrm{f}} \subseteq R \wedge S_t^{\mathrm{b}} \subseteq \overline{R} \right\} \\ &= (R \ominus \check{S}^{\mathrm{f}}) \cap (\overline{R} \ominus \check{S}^{\mathrm{b}}) \end{aligned}$$

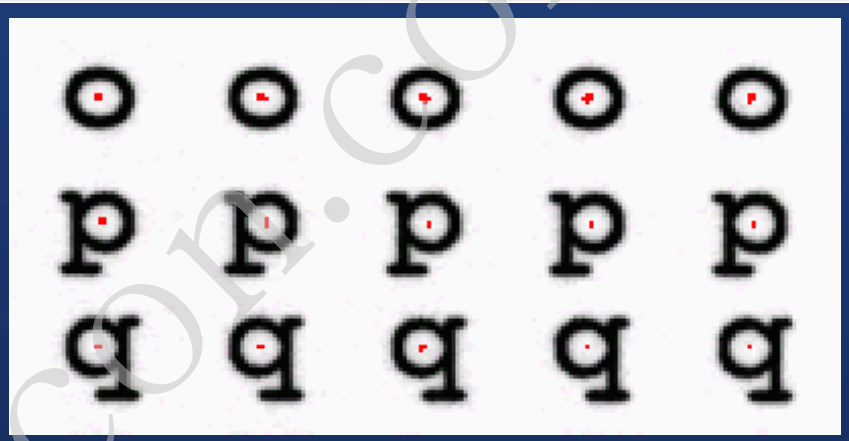
Duality: $R \otimes S = \overline{R} \otimes S'$, with $S=(S^{\mathrm{f}},S^{\mathrm{b}})$ and $S'=(S^{\mathrm{b}},S^{\mathrm{f}})$

- 这种变换的语意：所有在前景中的点必须位于区域 S^{f}，所有属于背景得点必须位于背景区域 S^{b}（前景的补集），目的是把测试点加入到输出区域 S^{b}。

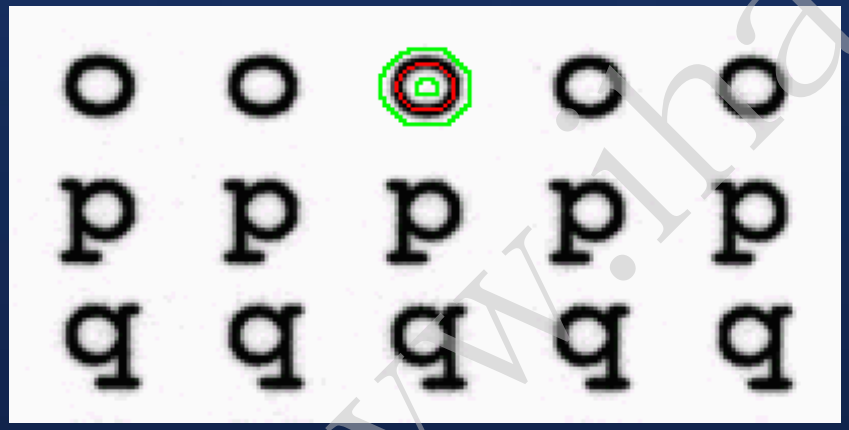
目标检测——形态学处理



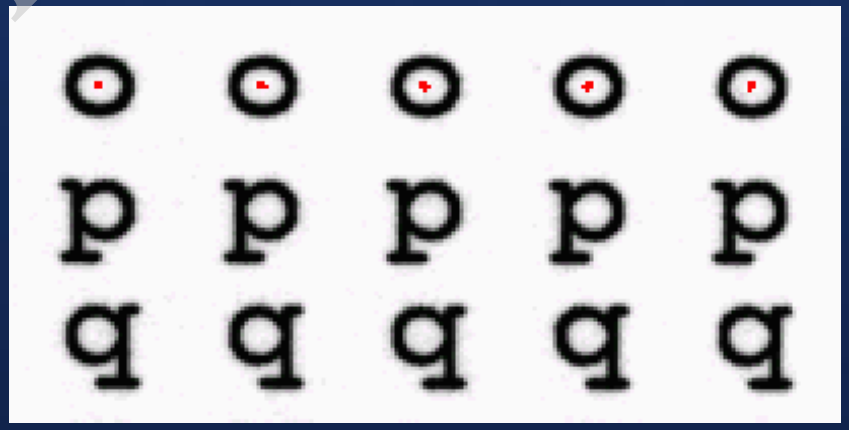
Structuring element



Erosion



Structuring elements

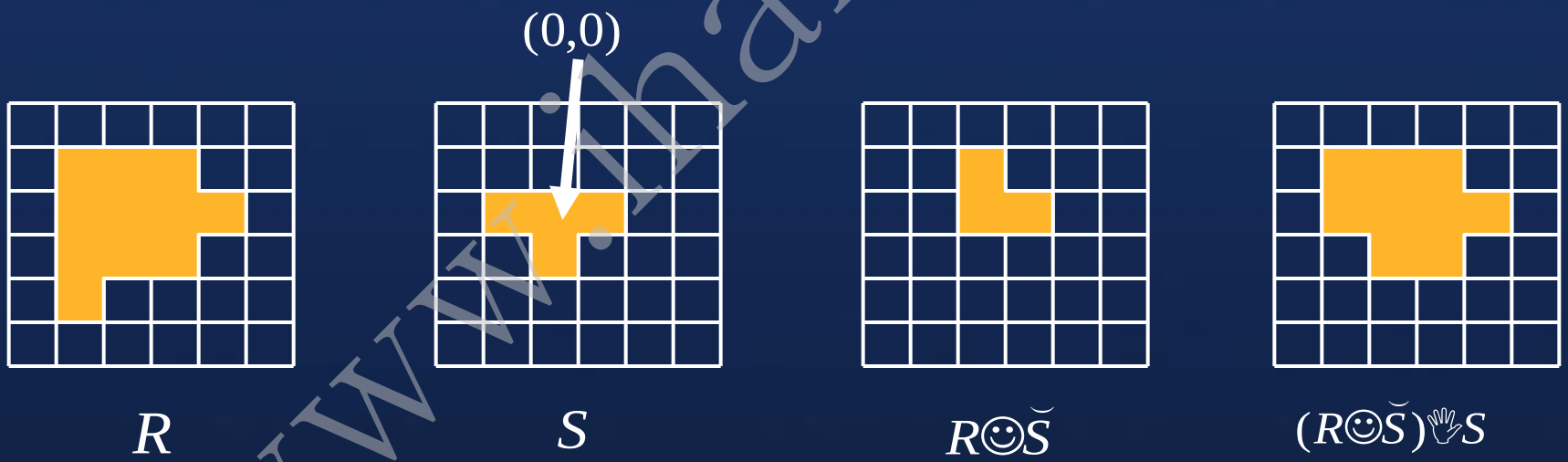


Hit-or-miss transformation

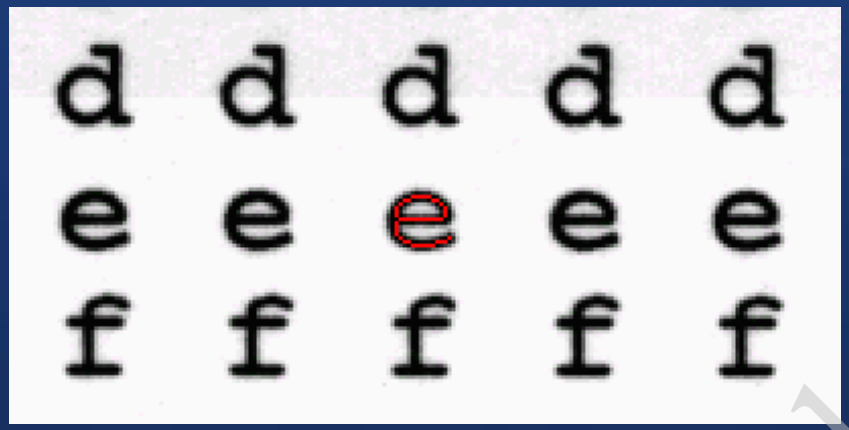
Opening ——形态学处理

定义:

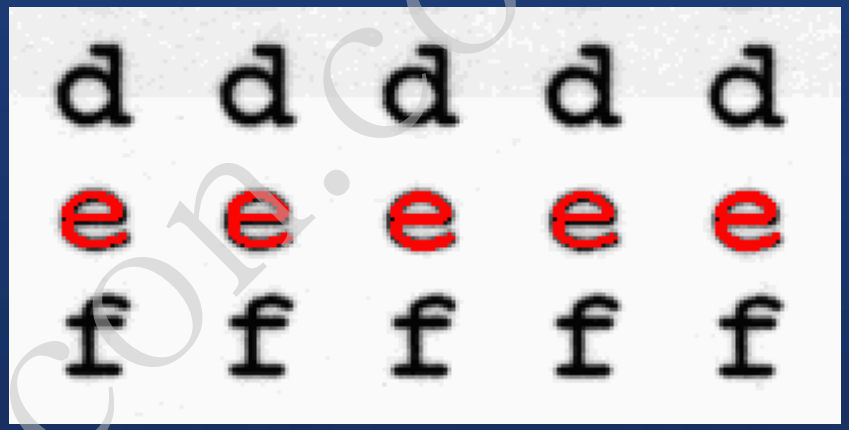
$$
\begin{aligned}
R \circ S &= (R \circledcirc \check{S}) \circledwedge S \\
&= \{r \in R \mid \exists t : r \in S_t \wedge S_t \subseteq R\} \\
&= \bigcup_{t \in R \circledcirc \check{S}} S_t \\
&= \bigcup_{S_t \subseteq R} S_t
\end{aligned}
$$



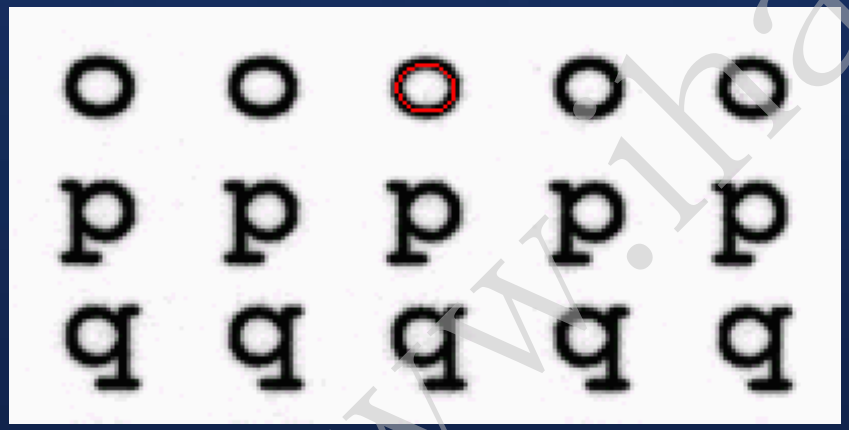
目标检测——形态学处理



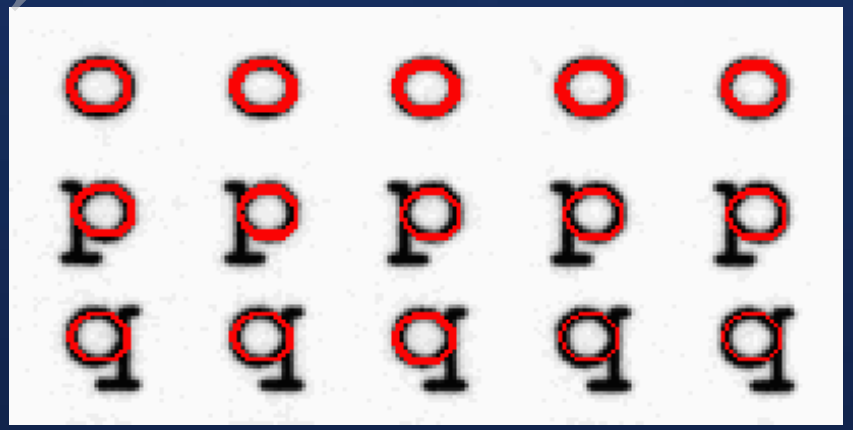
Structuring element



Opening

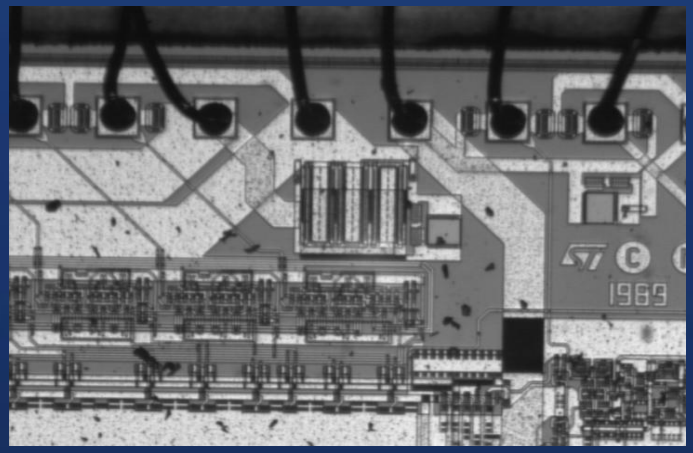


Structuring element

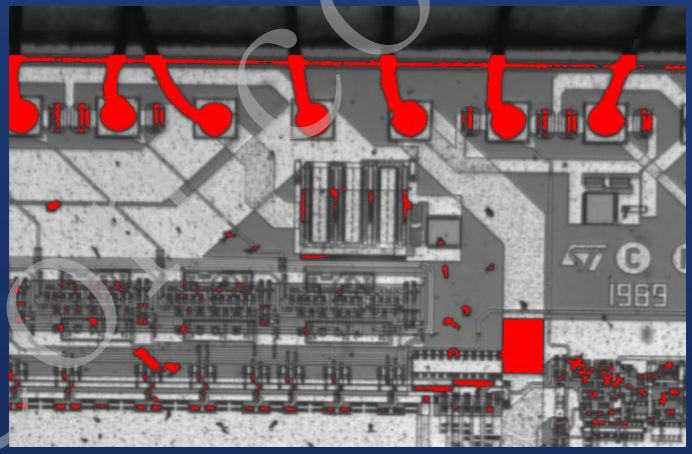


Opening

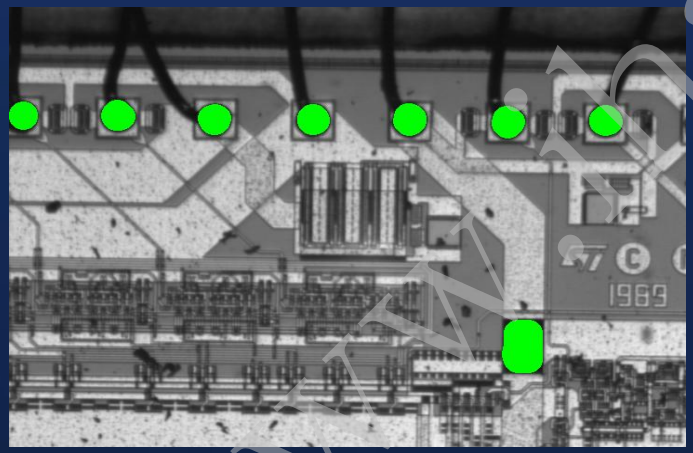
噪声抑制，目标检测——形态学处理



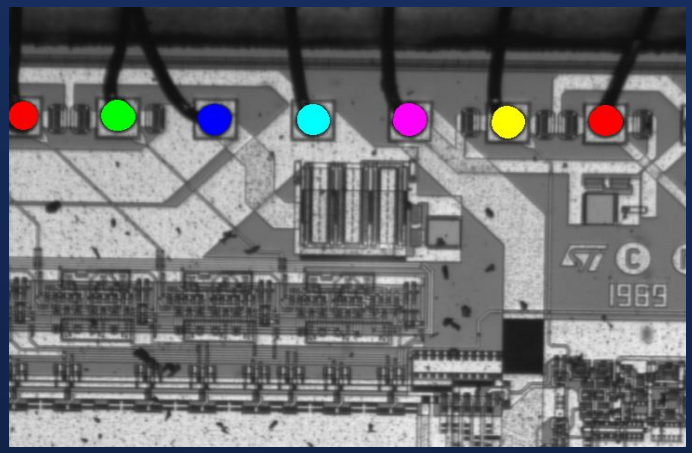
Input image



Segmentation



Opening



Selected balls

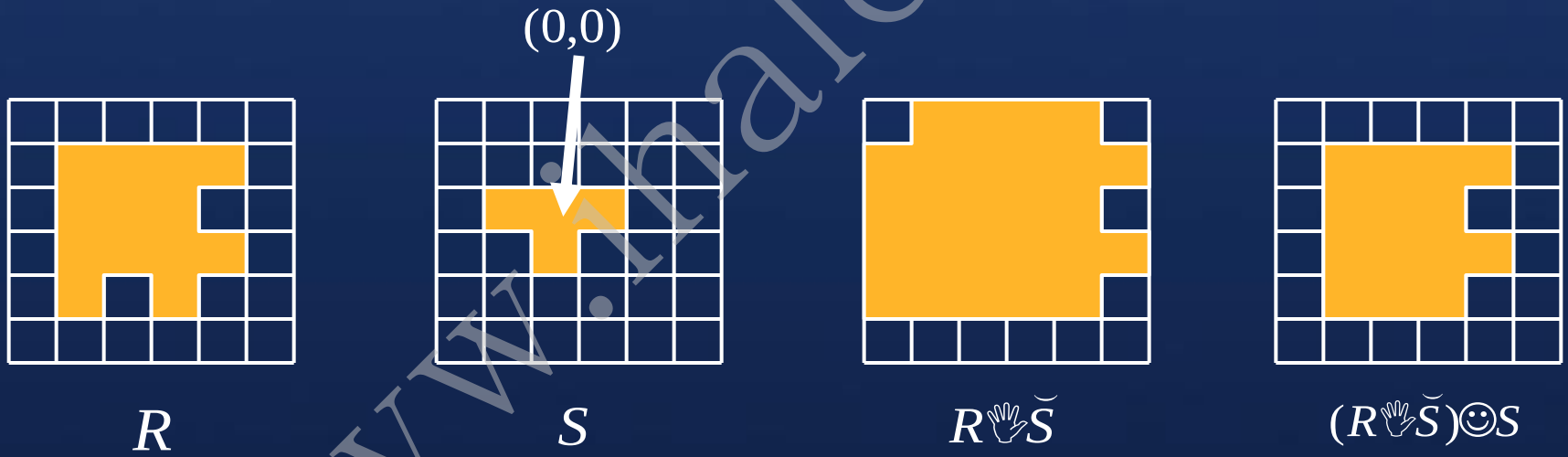
Closing ——形态学处理

定义:

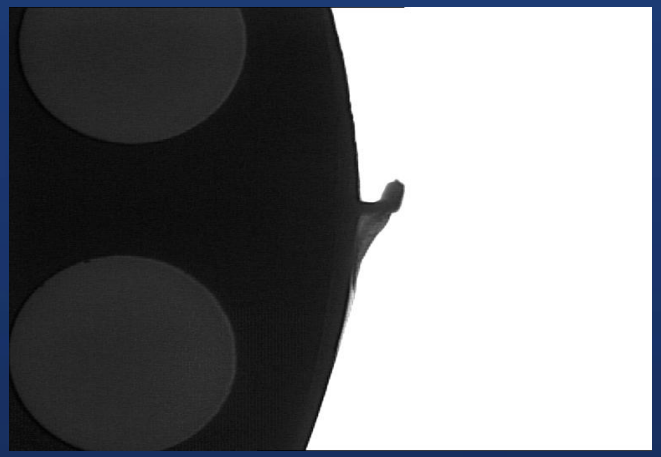
$$R \bullet S = (R \oplus \check{S}) \ominus S$$

$$= \overline{\bigcup_{S_t \subseteq \overline{R}} S_t}$$

$$= \bigcap_{R \subseteq \overline{S_t}} \overline{S_t}$$

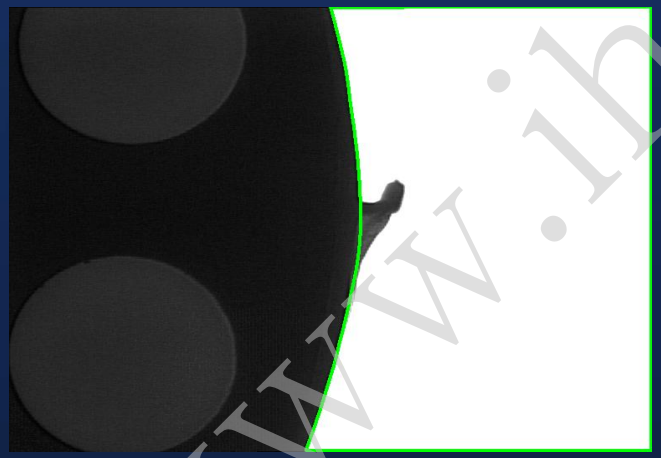


缺陷检测——形态学处理

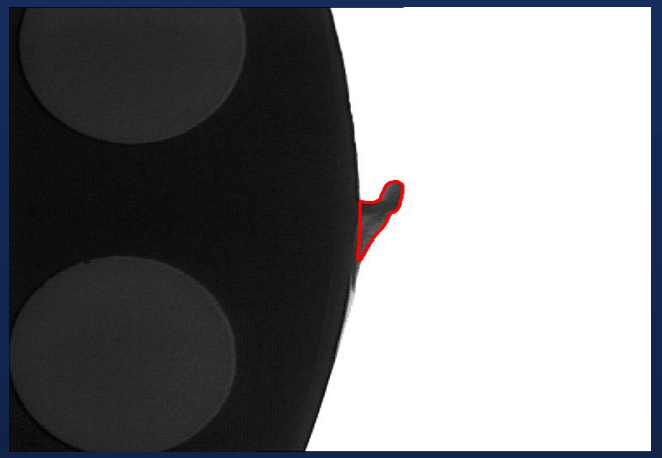


Input image

Segmentation

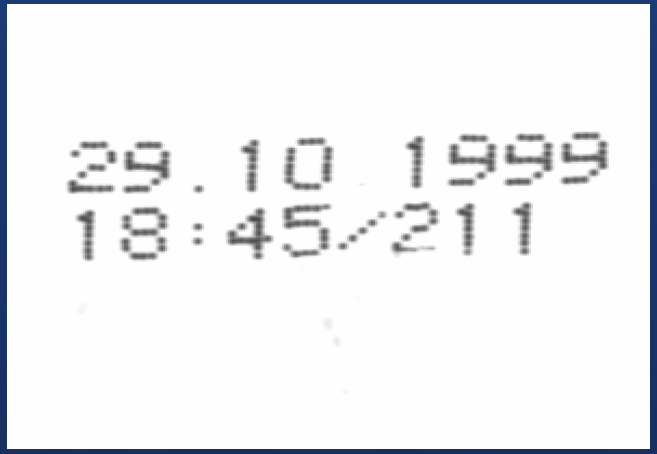


Closing



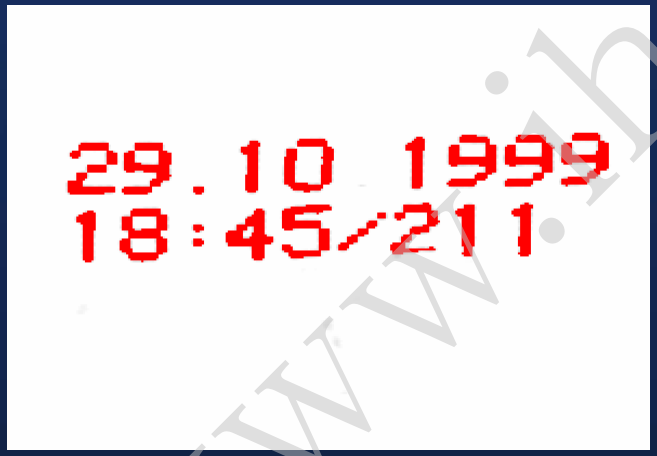
Detected defect

点印刷分割——形态学处理

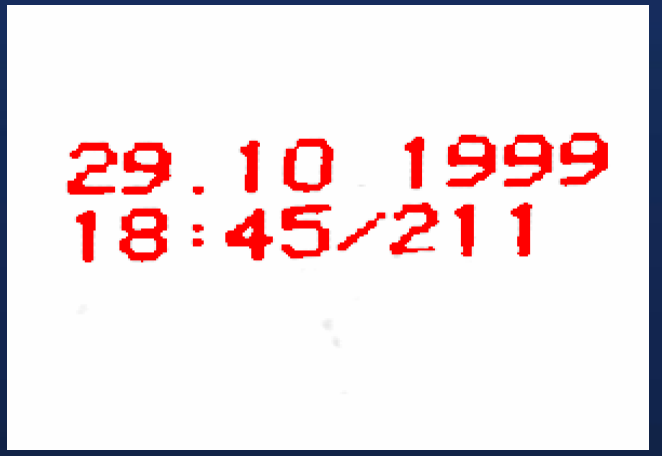


Input image

Segmentation



1. Closing with vertical rectangle



2. Closing with diagonal rectangle

Skeleton ——形态学处理

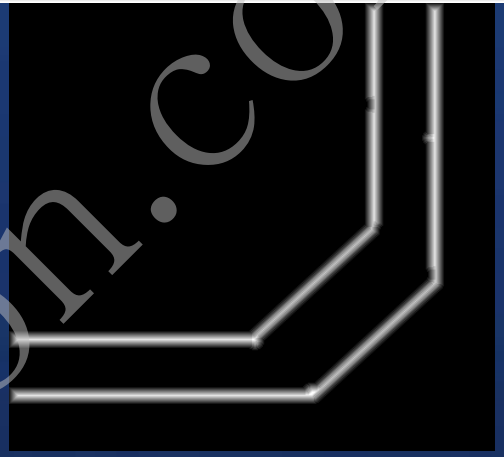
- 一个区域的骨架代表了区域的中心线。骨架上的这些点与一个区域同一位置上的边界距离都是一样的
- 骨架上的点也可以描述成一幅距离转换图像中的点，这些点到边界的垂直距离为最远距离。
- 骨架的计算通常是应用回归hit-or-miss 转换, 这种转换用来估计边界的点，这些点往往是不能成为skeleton的
- 骨架也可以用来检测物体的宽度，比如结合距离变换来检测PCB板上的引导电路

常用方法:

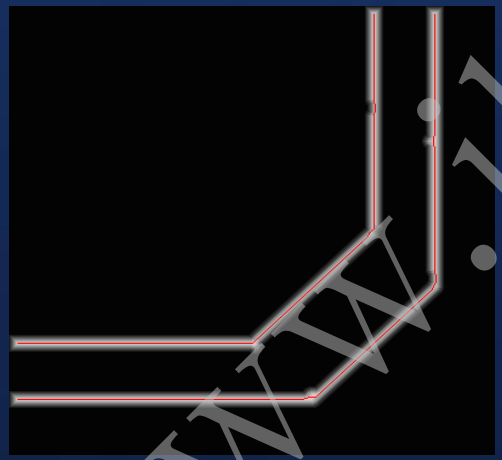
- 电路板电路的分割
- 距离变化和骨架的计算
- 距离变化包含了物体的宽度，这些宽度信息包含在骨骼上那些点的灰度值

PCB 检测——形态学处理

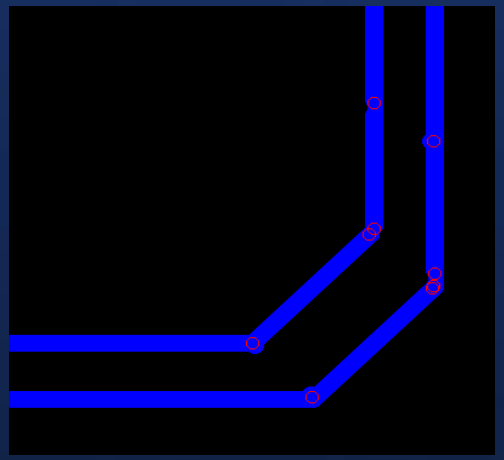
Simulated conducting paths with defects



Distance t变换



Skeleton和distance 变化



检测缺陷

总结——形态学处理

记住重要的几点:

- 算子集(union, intersection, difference, complement)
- dilation, erosion, opening, closing的定义
- dilation, erosion, opening, closing, skeleton的效果
- 在分割时erosion, dilatation 和connected components 的交叉使用

特征提取

- 特征描述了区域的“特征属性”
- 区域特征(形状特征) 描述区域的几何特征，这些特征是不受“相关的”灰度值影响的
- 灰度值特征使用了给定图像的灰度值，通常这个图像来自分割出来的区域
- 常常应用在
  - 分割出来一些区域之后，选择出满足条件的区域
  - 区域分类, 比如OCR
  - 测量
  - 质量检测

区域特征

灰度特征

区域特征: Area和Moments ——特征提取

简单的区域特征: area $a = |R| = \sum_{(x,y)\in R} 1$

Moments定义 ($p \geq 0,\ q \geq 0$): $m_{p,q} = \sum_{(x,y)\in R} x^p y^q$

$m_{0,0}$ 是 a 归一化矩($p+q \geq 1$) $n_{p,q} = \frac{1}{a} \sum_{(x,y)\in R} x^p y^q$

$(n_{1,0}, n_{0,1})$ 是区域的中心重心

区域特征: Moments ——特征提取

- 一阶归一化矩 $(n_{1,0}, n_{0,1})$ 提供了区域位置的信息(重心)
- 二阶中心矩确定了区域的“方向”
- 方法: 寻找具有相同矩的椭圆区域

椭圆的参数:

- 重心=区域中心
- 长半轴的长度

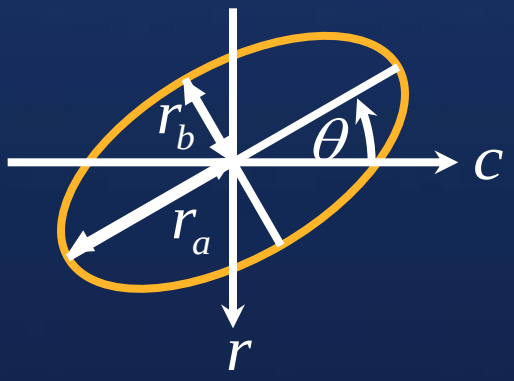
$$r_a = \sqrt{2}\sqrt{\mu_{rr} + \mu_{cc} + \sqrt{(\mu_{rr} - \mu_{cc})^2 + 4\mu_{rc}^2}}$$

- 短半轴的长度

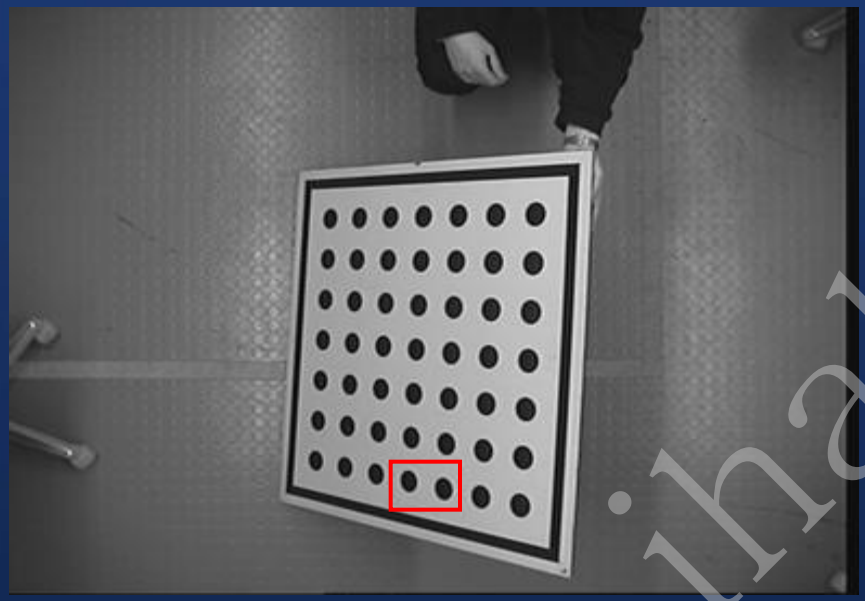
$$r_b = \sqrt{2}\sqrt{\mu_{rr} + \mu_{cc} - \sqrt{(\mu_{rr} - \mu_{cc})^2 + 4\mu_{rc}^2}}$$

- 角度

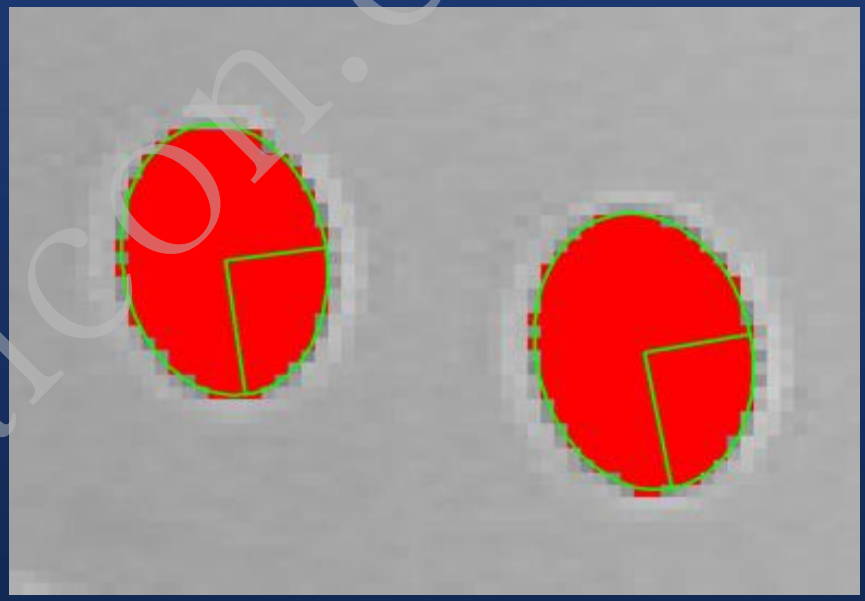
$$\theta = -\frac{1}{2}\operatorname{atan2}(2\mu_{rc}, \mu_{cc} - \mu_{rr})$$



区域特征: Moments ——特征提取

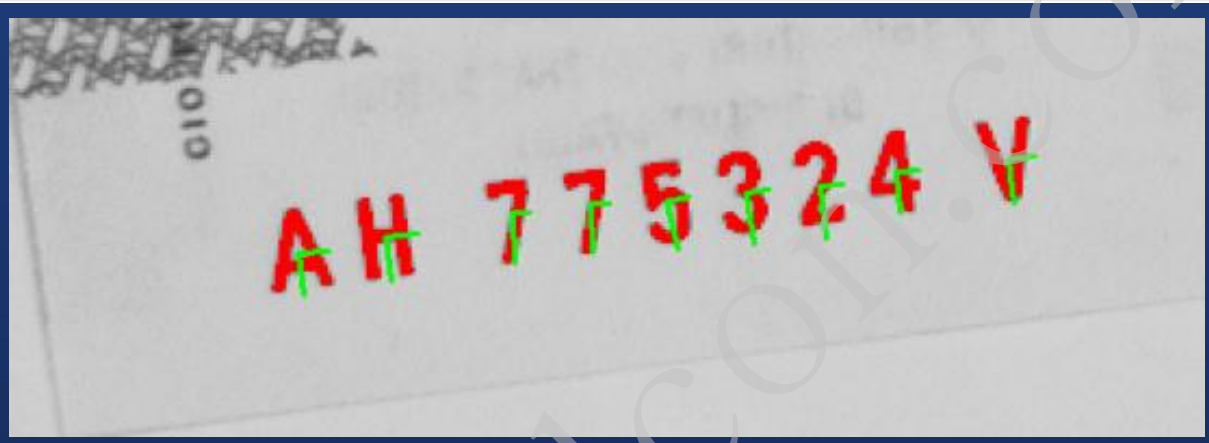


输入图像

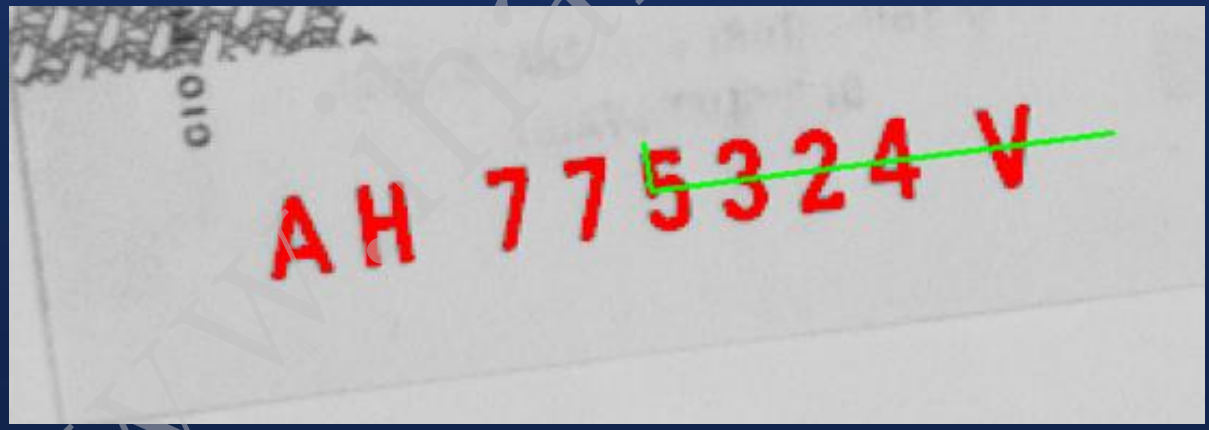


具有相同矩的椭圆和区域

区域特征: Moments ——特征提取



单个字符的方向



所有区域的方向，比如用于旋转校正

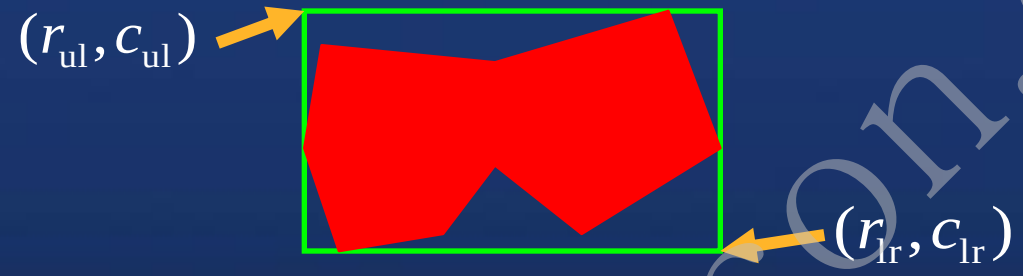
区域特征: Moments ——特征提取

- 椭圆长短轴相除:Anisometry r_a / r_b
- 算子eccentricity 和 elliptic_axis
- 优点: 尺寸不会改变
- 椭圆特征的问题:
  - 只有 $r_a \neq r_b$ 在才能确定方向
  - 像正方形之类的物体，方向确定不了
  - 不能体现区域里的孔洞产生的一些直观但相反的影响:

  - 方向只能确定180度模
- 解决180度问题的方法：使用最远点来使得方向明显 (orientation_region)
- 椭圆特征是亚像素精度

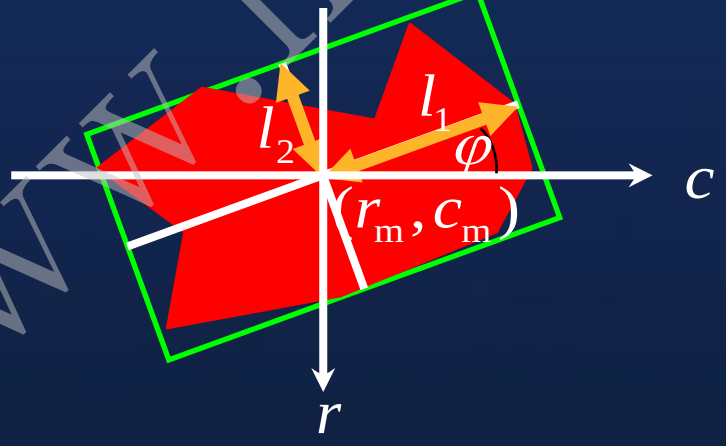
区域特征: Smallest Rectangle ——特征提取

平行于主轴的最小矩形smallest_rectangle1：



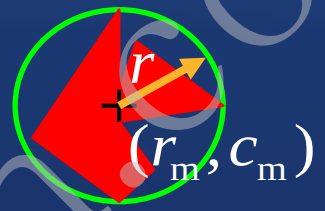
定义：$r_{ul} = \min_{(r,c)\in R} r$, $r_{lr} = \max_{(r,c)\in R} r$ ； c_{ul}, c_{lr} 类似

任意方向的最小矩形(复杂但快速的方法):smallest_rectangle2:



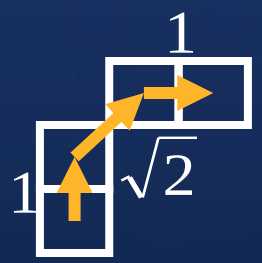
区域特征: 其它特征——特征提取

smallest_circle:



convexity:区域面积与突包面积的比例

contlength: 区域边界的长度



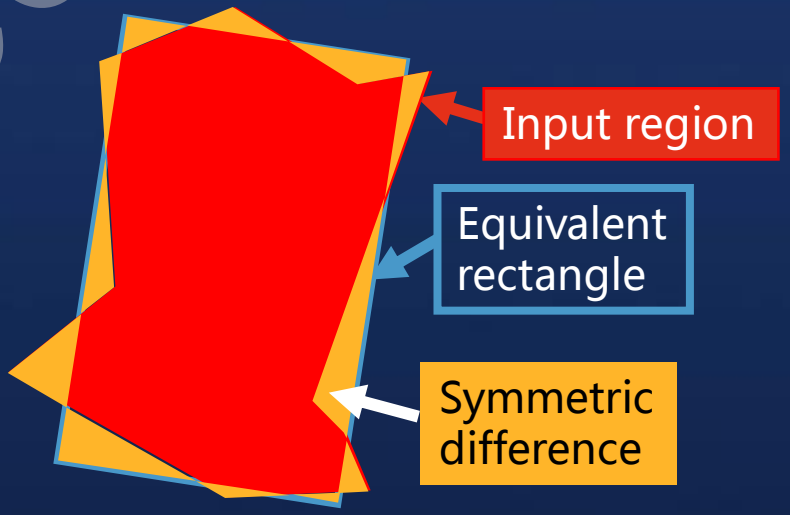
compactness: l 为轮廓长度，a 是区域的面积

$$k = \frac{l^2}{4\pi a}$$

- $k \geq 1$； $k = 1$是圆形区域

区域特征rectangularity ——特征提取

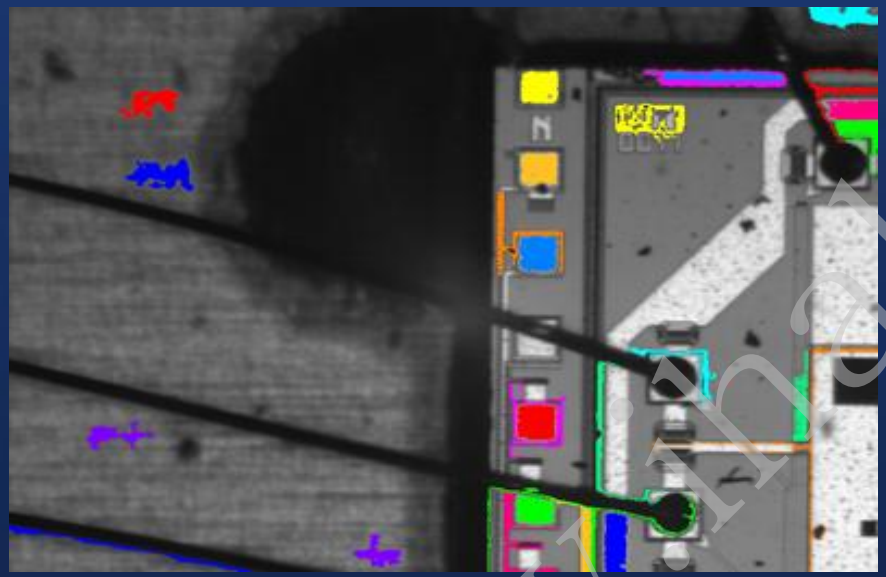
- 除了圆和椭圆外，经典形状还有矩形
- 经典的形状特征比如`roundness`, `circularity`, 或者 `compactness` 都不适用于选出矩形
- HALCON提供了用来专门选取/评估矩形的特征



注意：对于不能计算二阶矩的区域(比如正方形)`rectangularity`或许被低估了10%

rectangularity: 示例 ——特征提取

选择die上的矩形垫片

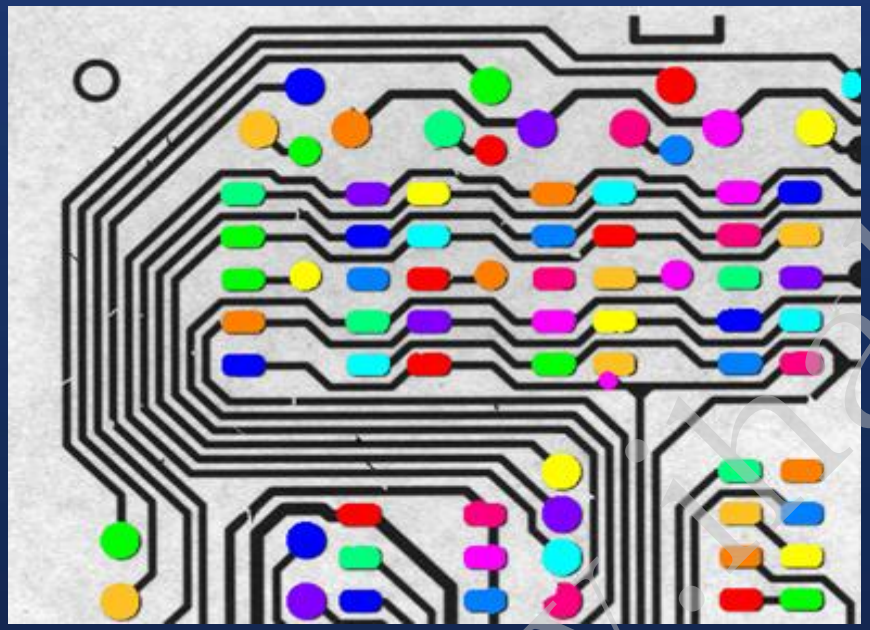


Raw segmentation

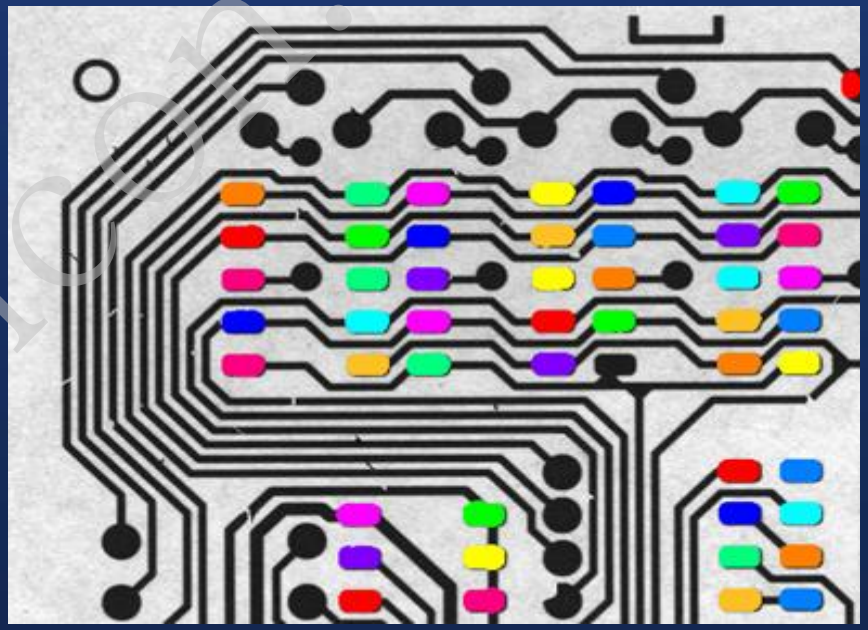
Selected rectangles

rectangularity: 示例——特征提取

选择PCB板上矩形垫片



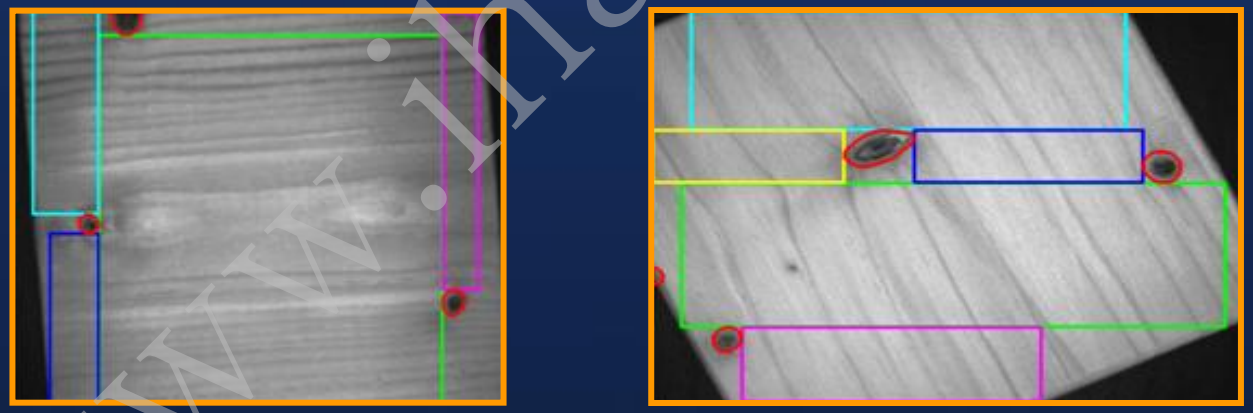
Raw segmentation



Selected rectangles

inner_rectangle1 ——特征提取

- HALCON提供了一个算子可以用来确定一个区域的最大内接矩形
- 这件看似简单的任务非常具有挑战性，因为矩形的四个参数必须确定(位置和x/y方向)
- 这个新的函数使得以下应用成为了可能，比如表面检测、木头和木材，皮革等
- 这个算子可以容易地反复提取出n个最大内接矩形



示例应用: 木头上节点

Select_shape ——特征提取

```
select_shape(Regions : SelectedRegions : Features,
  Operation, Min, Max : )
```

区域特征

灰度特征

灰度值特征——特征提取

简单灰度值特征：区域的平均灰度值

$$\bar{g} = \frac{1}{a} \sum_{(x,y)\in R} g_{x,y}$$

区域的最小和最大灰度值:

$$g_{\min} = \min_{(x,y)\in R} g_{x,y}, \quad g_{\max} = \max_{(x,y)\in R} g_{x,y}$$

α-直方图的分位数

$$g_\alpha = \min\{g : d_g \geq \alpha\} = \max\{g : d_g \leq \alpha\}, \qquad h_g = \frac{n_g}{n}, \qquad d_g = \sum_{j=0}^{g} h_j$$

α = 0.5: 灰度值的中值

用法：照明校正

灰度值的方差和标准偏差:

$$s^2 = \frac{1}{a-1} \sum_{(x,y)\in R} (g_{x,y} - \bar{g})^2, \quad s = \sqrt{s^2}$$

灰度值矩——特征提取

区域矩的广义性：使用单个点相应的灰度值对该点加权:

$$m_{p,q} = \sum_{(x,y)\in R} g_{x,y} x^p y^q$$

$a = m_{0,0}$ 面积类似

归一化矩　$p+q \geq 1$

$$n_{p,q} = \frac{1}{a} \sum_{(x,y)\in R} g_{x,y} x^p y^q$$

$(n_{1,0}, n_{0,1})$ 是该区域的灰度值重心

当 $p+q \geq 2$ 为中心矩

$$\mu_{p,q} = \frac{1}{a} \sum_{(x,y)\in R} g_{x,y} (x-n_{1,0})^p (y-n_{0,1})^q$$

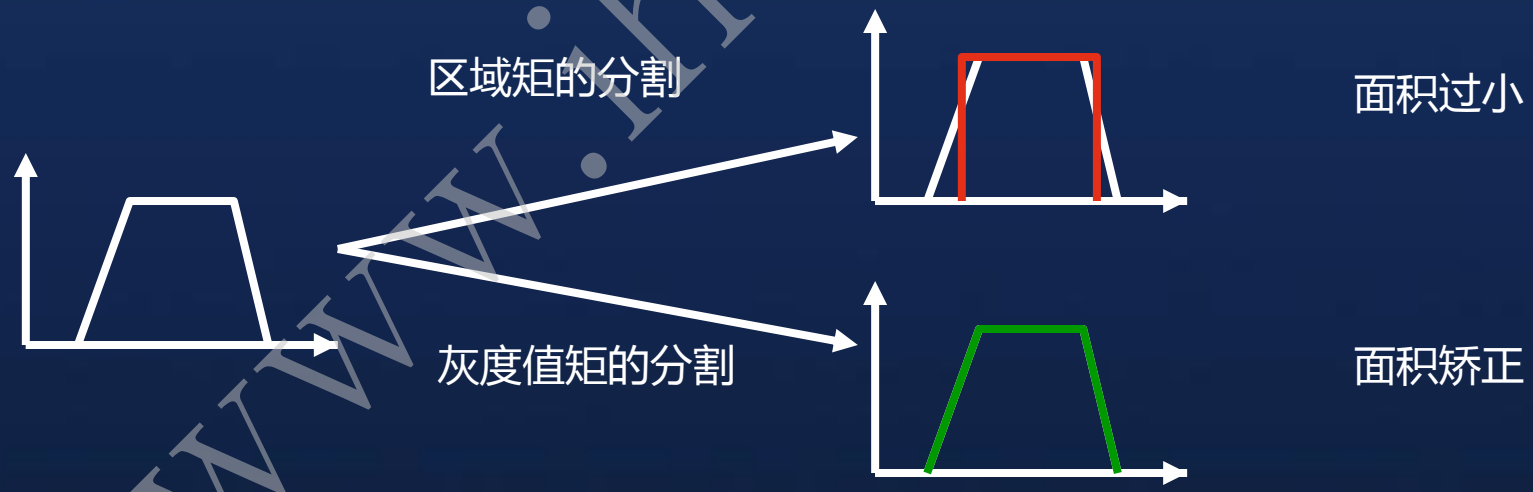
灰度值矩——特征提取

如果灰度值是该区域的示性函数，那么灰度值矩和区域矩具有相同的结果

i.e., $g_{x,y} = \chi_R(x,y) = 1$ 如果 $(x,y) \in R$

与区域矩的差异

- 区域矩有一个硬过渡
- 通过加权灰度值，矩会有一个软过渡，所以目标边界的灰度值转换可以被很好的检测。



区域- vs. 灰度值Moments ——特征提取

- 区域特征（重心，椭圆参数）精度正比于区域面积
- ⇨ 对于小的区域而言精度不是很高
- 对于高质量的图像（高填充因子，线性灰度值响应），过渡面积（边界，来源于传感器的平均）提供的信息大大增加了精度。
- 由于过渡区域相对于区域面积比较小，所以大区域精度的提高也是有限的。
- ⇨ 灰度值特征主要是在小的区域比较有效。
- 需要记住的是灰度值矩的计算时间大于普通矩

$$O(a) \Leftrightarrow O(\sqrt{a})$$

灰度值特征用法：评估噪声——特征提取

- 确定一个摄像机的品质
- 确定一块采集卡的品质
- 确定摄像机的电流噪声 (取决于电流设置比如增益和偏移等)
- 确定图像处理算子的参数，比如确定边缘滤波的最小对比度
- 确定图像中纹理噪声的幅值

确定出来的值用于

- 不断地告知算子关于系统的电流状态
- 自适应调节图像处理方法中的参数
- 生产过程中品质的保证

评估噪声: estimate_noise ——特征提取

从单幅图像中评估噪声的标准偏差

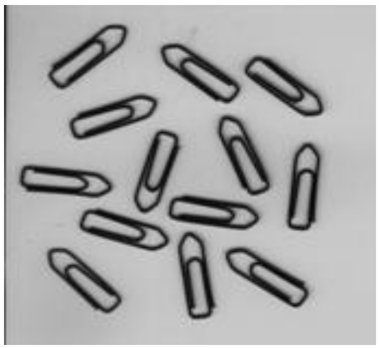
可以从以下四种方法中选取一种

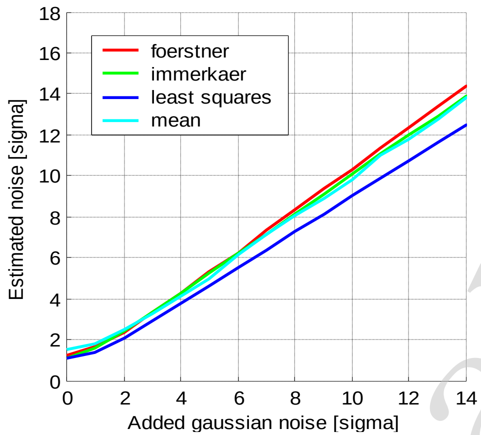
- 'foerstner'：反复寻找图像中的均匀区域，并计算均匀区域中灰度值的标准偏差
- 'immerkaer'：对图像使用一个特殊滤波掩码. 噪声可以从滤波输出中计算出来，这取决于图像的噪声
- 'least_squares' 和 'mean'：都是在第一步计算图像中均匀像素的最大百分比
  - 'least_squares'：对于每个均匀像素而言，一个灰度值平面适应于3x3领域
    - ⇨ *noise = gray values – fitted plane*
  - 'mean'：噪声可以被消除，通过在均匀图像区域中应用均值滤波器
    - ⇨ *noise = gray values – mean image*

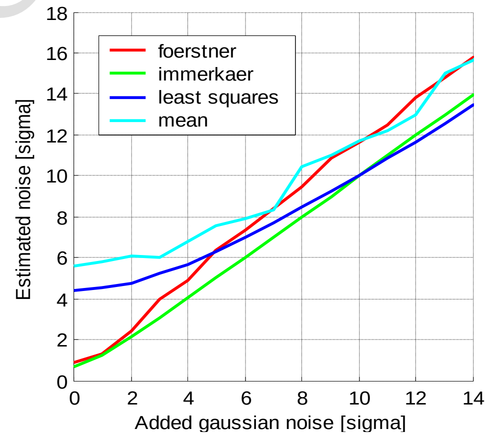
总而言之，'foerstner' 返回值精度最高，然而 'immerkaer' 是计算速度最快

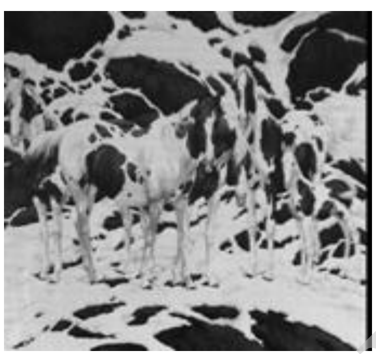
评估噪声: estimate_noise ——特征提取

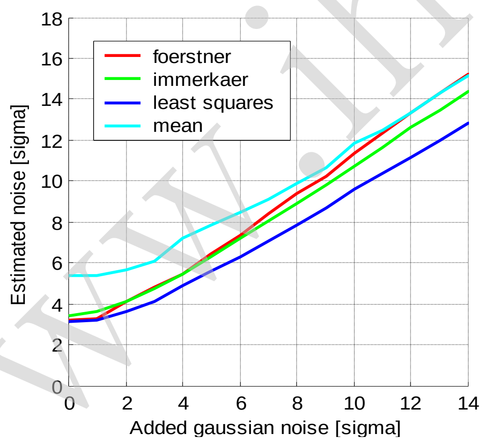
结果示例

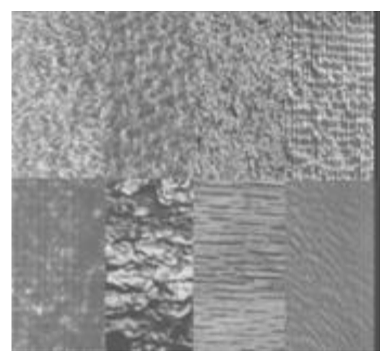


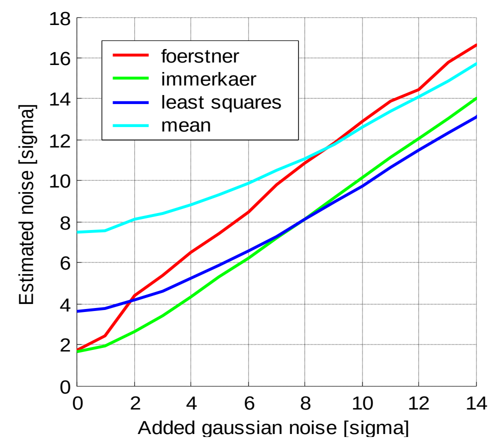












Select_gray ——特征提取

```
select_gray(Regions, Image : SelectedRegions :
  Features, Operation, Min, Max : )
```

回顾

练习.....