DAHENG IMAGING | 大恒图像

HALCON BLOB分析

北京技术中心

中国大恒(集团)有限公司北京图像视觉技术分公司
www.daheng-imaging.com

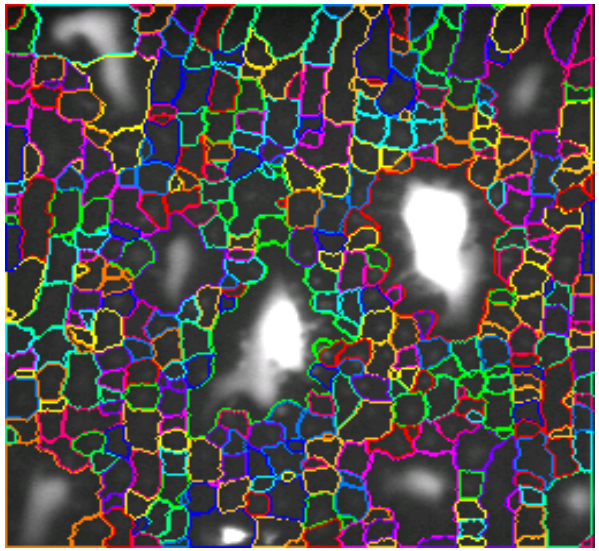
目录 Contents

01 分割图像

图像分割

分割方法

- threshold, fast_threshold, binary_threshold, dyn_threshold, histo_to_thresh, gray_histo
- local_threshold, var_threshold, watersheds, watersheds_threshold, regiongrowing, regiongrowing_mean

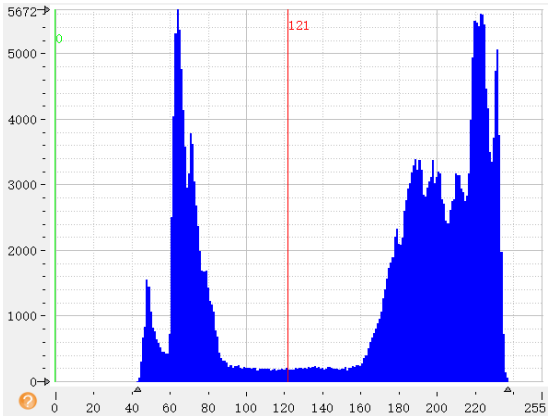


全局阈值 - threshold

- threshold 算子特点：简单、快速、使用频率最高
- 如果目标与背景之间存在灰度差，可优先选择threshold
- 对于一些困难的图像分割案例，可以对图像做阴影校正
- threshold 算子的定义:

$$R' = \{(r,c) \in R | g_{min} \leq g(r,c) \leq g_{max}\}$$

- 如果环境稳定，阈值可在离线状态下一次确定



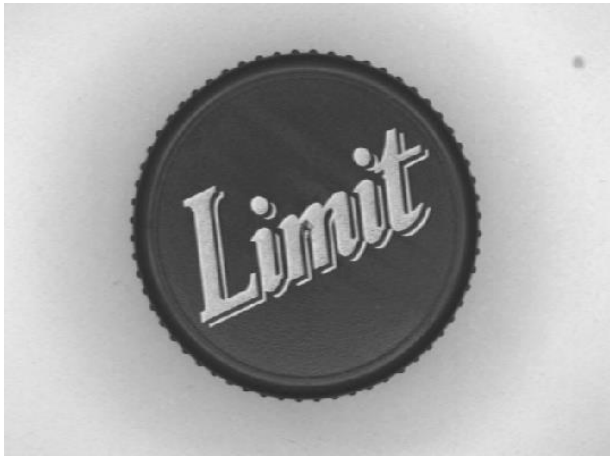
全局阈值 - threshold

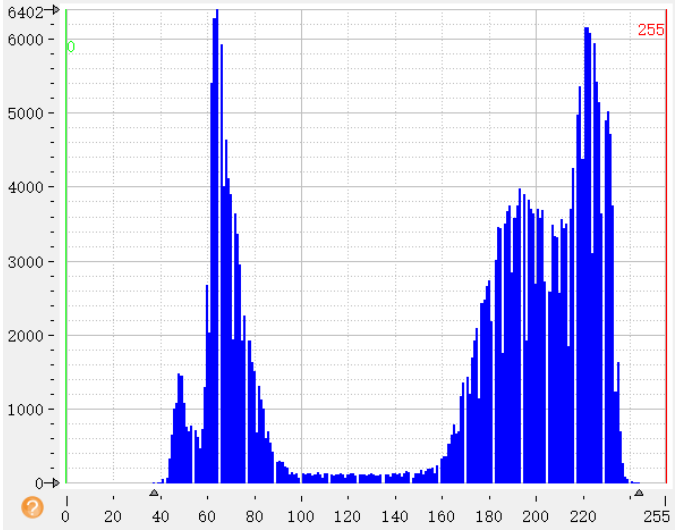
- 如果存在照明条件变化的情况，可通过直方图信息自动确定全局阈值
- 自动全局阈值分割方法
  - 计算直方图
  - 寻找出现频率最多的灰度值（最大值）
  - 在threshold中使用与最大值有一定距离的值作为阈值

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

全局阈值 - binary_threshold

- binary_threshold算子特点：自动确定全局阈值，并返回阈值。提供以下两种方法。
  - 'max_separability'
    - 最大类间方差法（OTSU，大津法）
  - 'smooth_histo'
    - 不断迭代地平滑直方图
    - 两个最大值之间的最小值是阈值
- 仅适合直方图具有双峰的图像





全局阈值 - binary_threshold

```
binary_threshold (Image, Region1, 'max_separability', 'dark',
   UsedThreshold1)
binary_threshold (Image, Region2, 'smooth_histo', 'dark',
   UsedThreshold2)
```

动态阈值 - dynamic_threshold

- 对于一些应用来说，确定一个全局阈值是不可能的，比如
  - 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

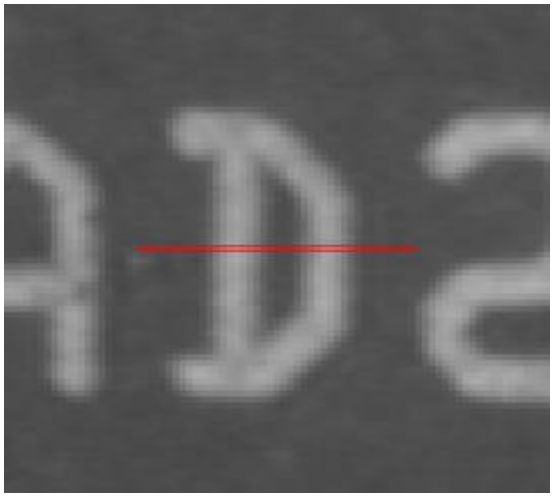
动态阈值 - dynamic_threshold

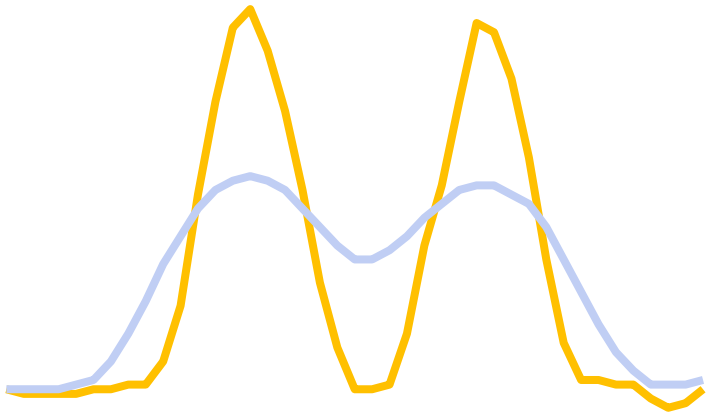
- dynamic threshold的定义：S 表示输入的平滑图像。

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

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

动态阈值 - dynamic_threshold

- 平滑图像的掩模尺寸确定了做能分割出来物体的最大尺寸
- 经验之谈
  - 平滑图像的掩模尺寸> 被提取物体的直径
  - 但也不能太大，如果平滑图像的掩模尺寸太大，那么相邻很近的物体将会连在一块
  - 推荐2×D+1（跟具体应用有关）
- 动态阈值也会返回沿着边缘的处理结果
- 应用：非均匀背景上的对象提取

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

动态阈值 - var_threshold

- 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 – 比较

原始图像：
带有不均匀的背景

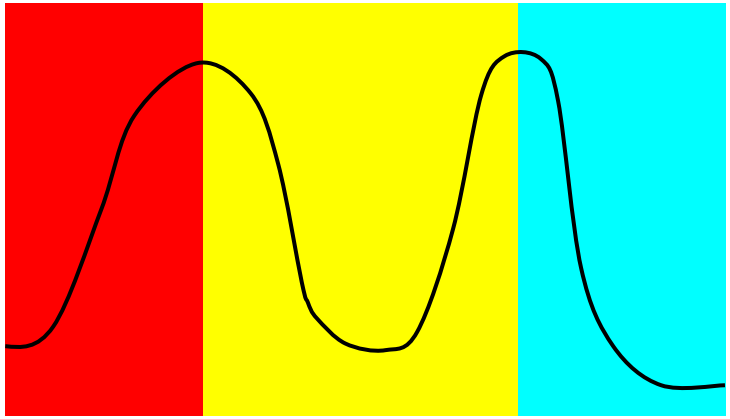
`dyn_threshold`：
在字符周围会有“电晕”

`var_threshold`：
每个字符有着更均匀的背景

图像分割

分水岭分割

- 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

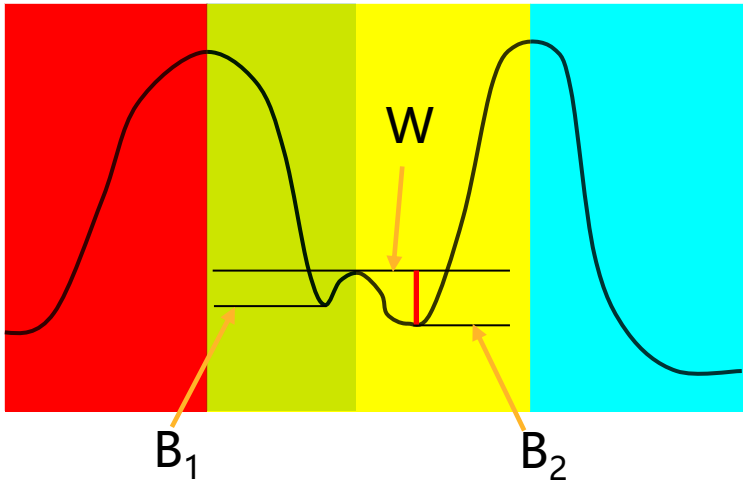
高级分水岭分割

- 如果对比度差异比较少，watersheds_threshold 可以使得相邻的盘地合并
- 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:

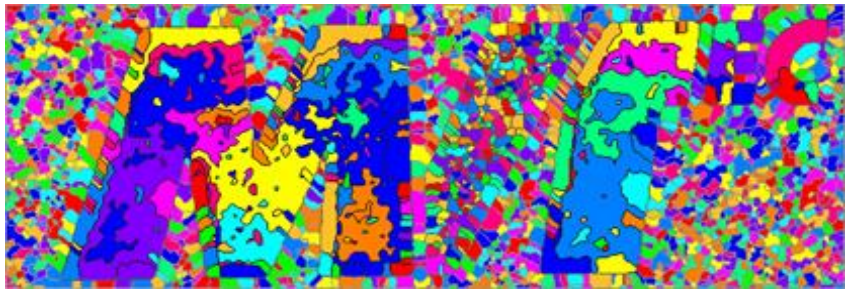
$$\max(W - B_1, W - B_2) < Threshold$$



比较: watersheds ↔ watersheds_threshold

watersheds

watersheds_threshold



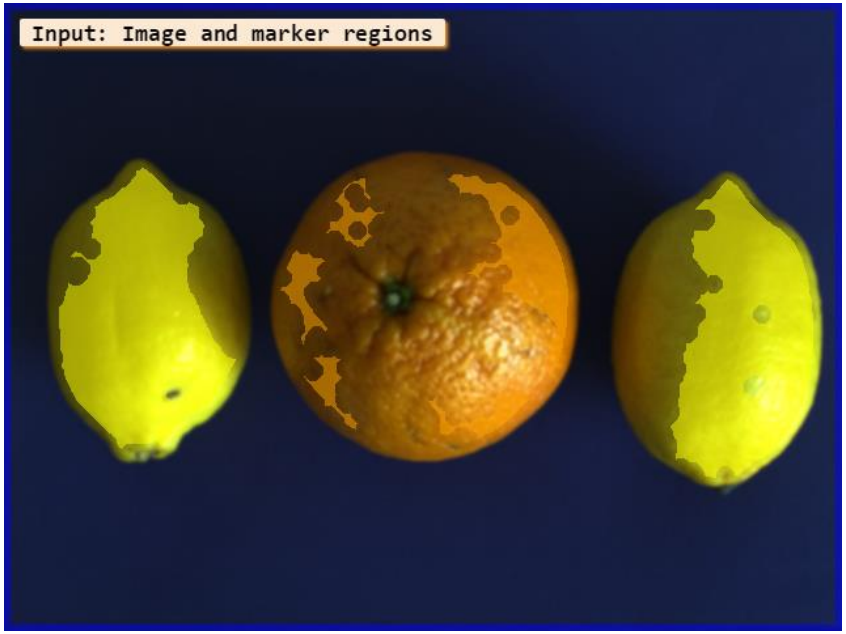
图像分割

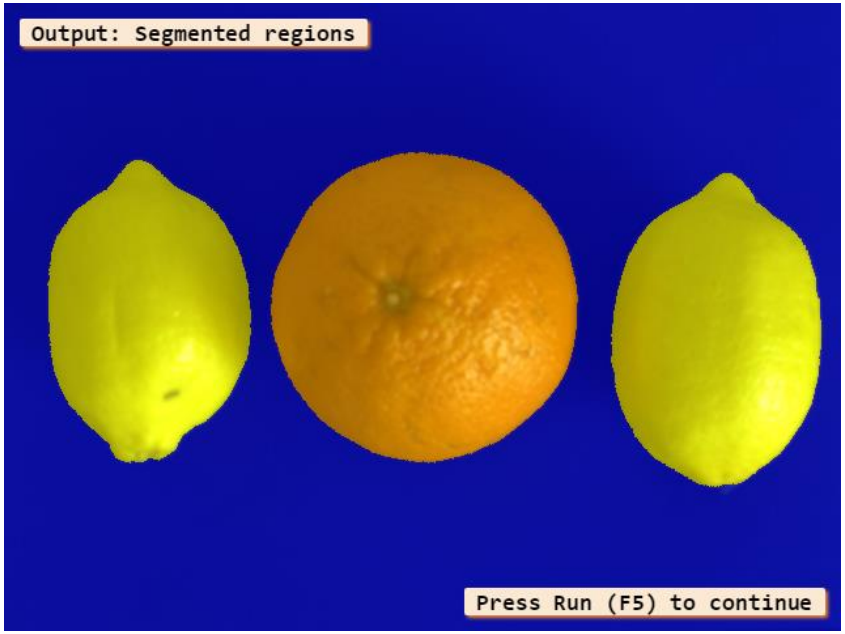
方法?

图像分割

DAHENG IMAGING | 大恒图像

watersheds_marker

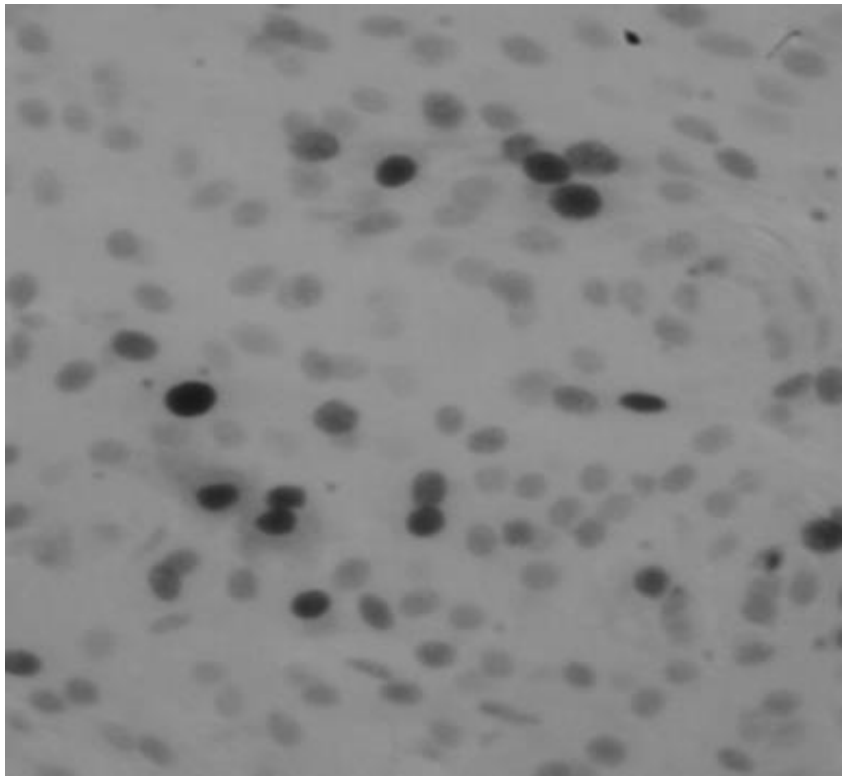


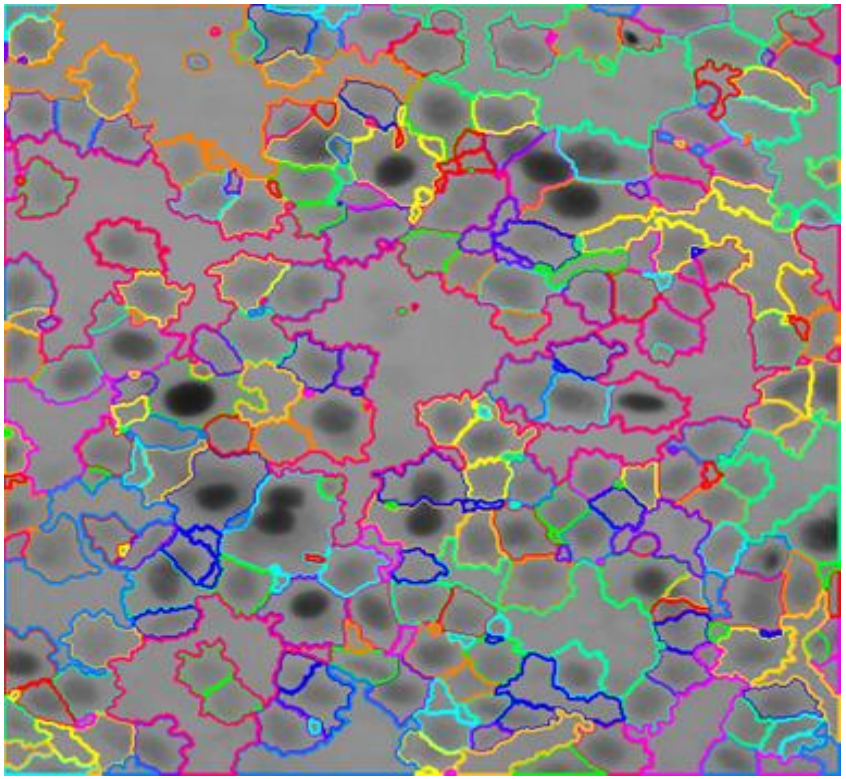


注：HALCON 19.11.0.0 Progress添加的功能

图像分割

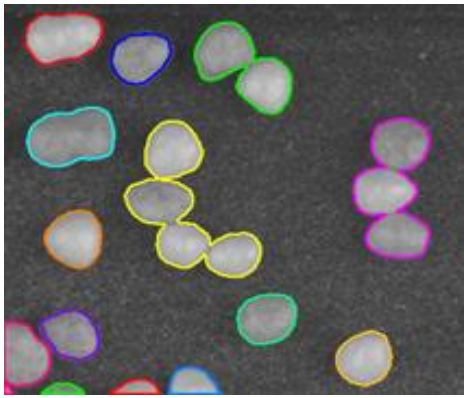
医学: 细胞分割

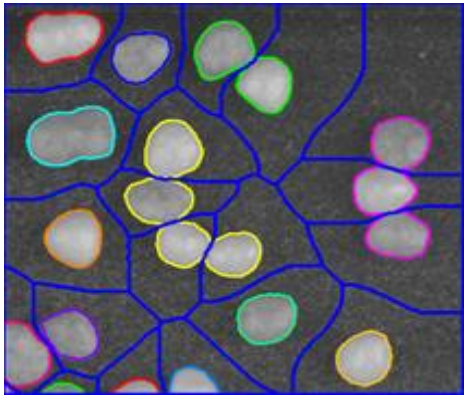


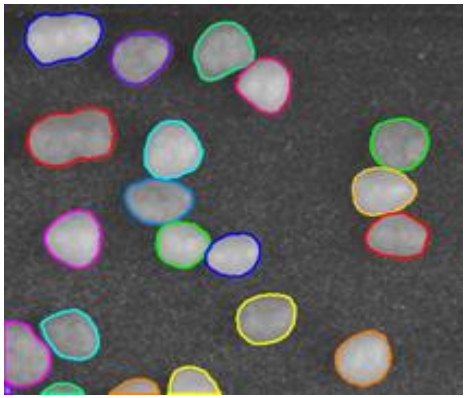


图像分割

制药:使用distance transformation来计算粒子个数





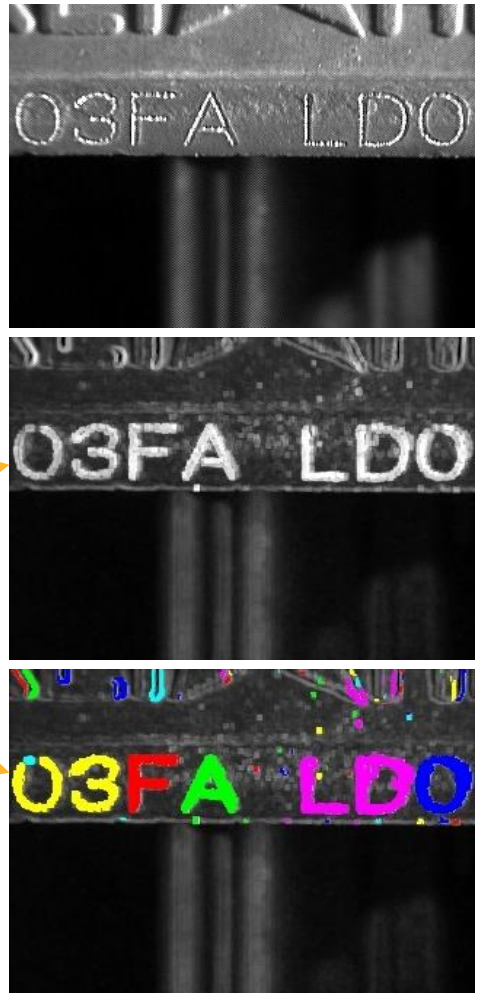


图像分割

分割: 金属表面

- 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)
```



图像分割

多通道分类

- 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

02 形态学处理

形态学处理

形态学相关操作

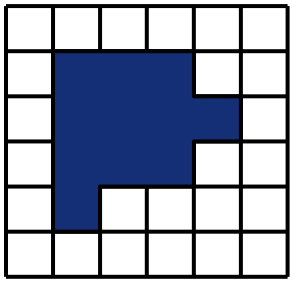
- 经典算子
  - Dilation
  - Erosion
  - Opening
  - Closing
- 所有的区域形态学处理能根据六个简单操作来定义
  - 并集(Union)
  - 交集(Intersection)
  - 补集(Complement)
  - 差集(Difference)
  - 平移(Translation)
  - 转置(Transposition)

并集(Union)

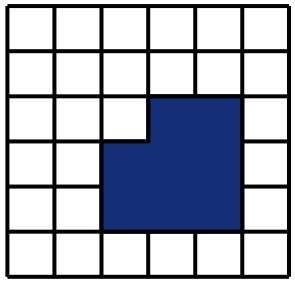
- 定义

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

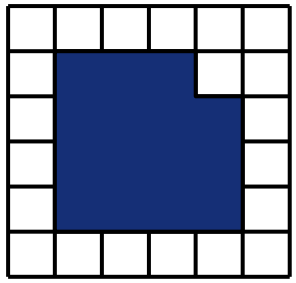
- 算子
  - union1: 把所有region合并成一个变量
  - union2: 把第二个参数里的所有区域统一到第一个参数的每一个区域中去
- 用法
  - 结合原有的形状来生成区域
  - 合并分割结果



R



S



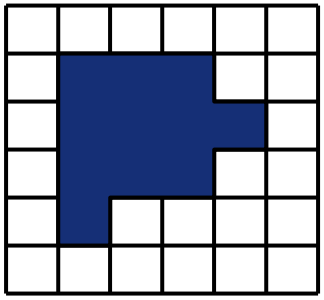
$R \cup S$

交集(Intersection)

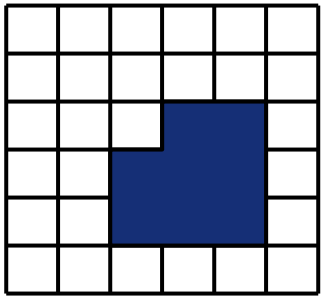
- 定义

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

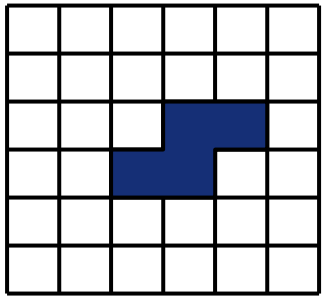
- 算子
  - intersection: 第一个参数中的每一个区域与第二个参数中的所有区域进行相交
- 用法
  - 结合原有形状来生成图像
  - 返回两区域共同点



R



S



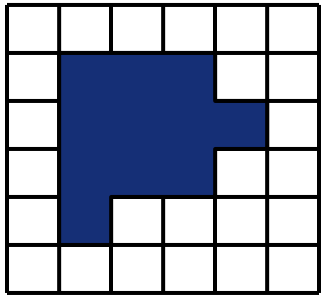
$R \cap S$

补集(Complement)

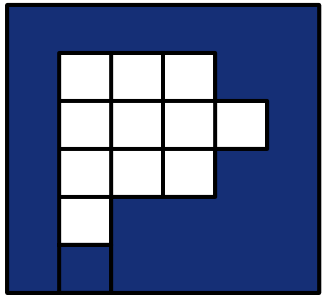
- 定义

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

- 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



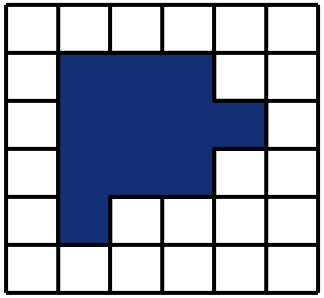
$\overline{R}$

差集(Difference)

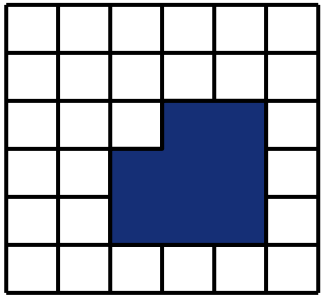
- 定义

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

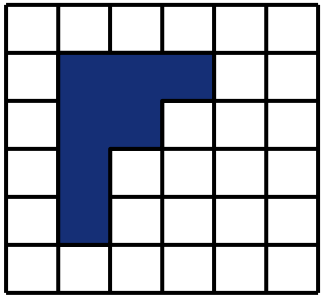
- 算子
  - difference: 去掉第二个参数与第一个参数共有的区域
- 用法
  - 结合原有形状来生成图像
  - 返回的点是在一个区域中出现但不在另一区域中



R



S



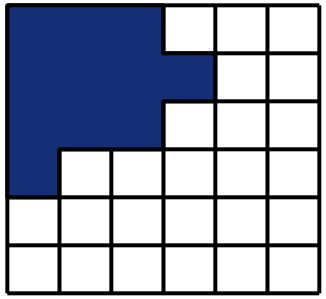
$R - S$

平移(Translation)

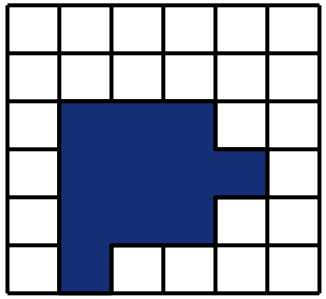
- 定义

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

- 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)
- 注意
  - 结果取决于系统标签'clip_region'



R

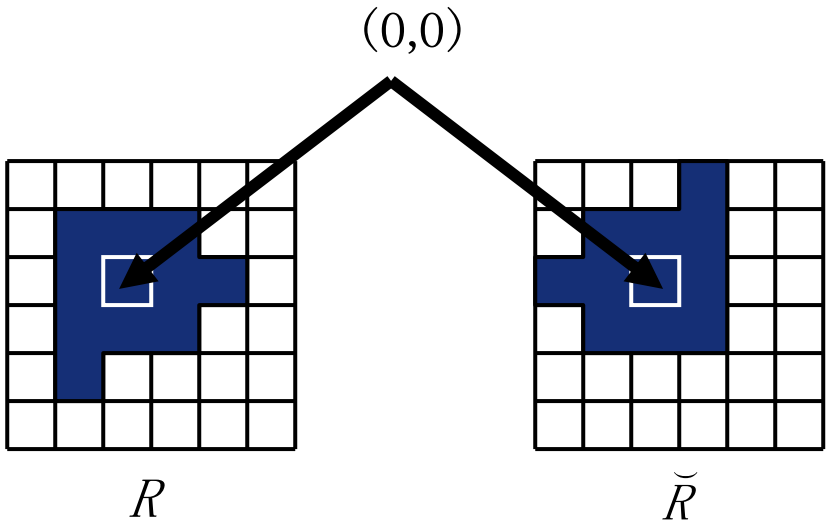


$R_{(2,1)}$

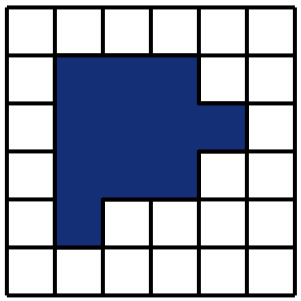
转置(Transposition)

- 定义

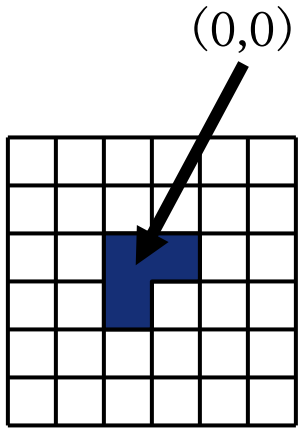
$$\breve{R} = \{-x \mid x \in R\}$$



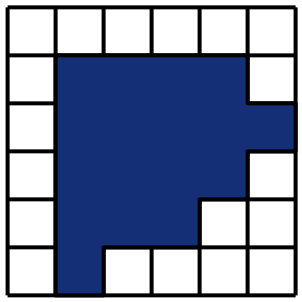
哪个是膨胀(Dilation)?



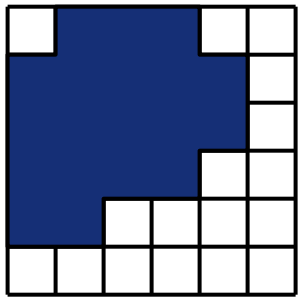
R



S

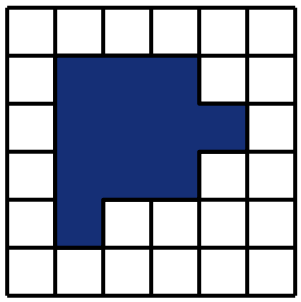


$R \oplus S$

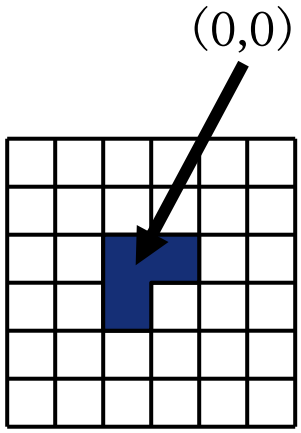


$R \oplus \check{S}$

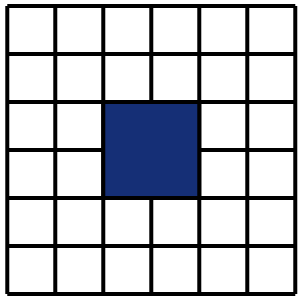
哪个是腐蚀(Erosion)?



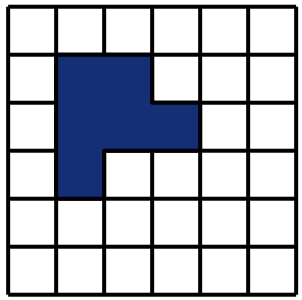
R



R



$R \ominus S$



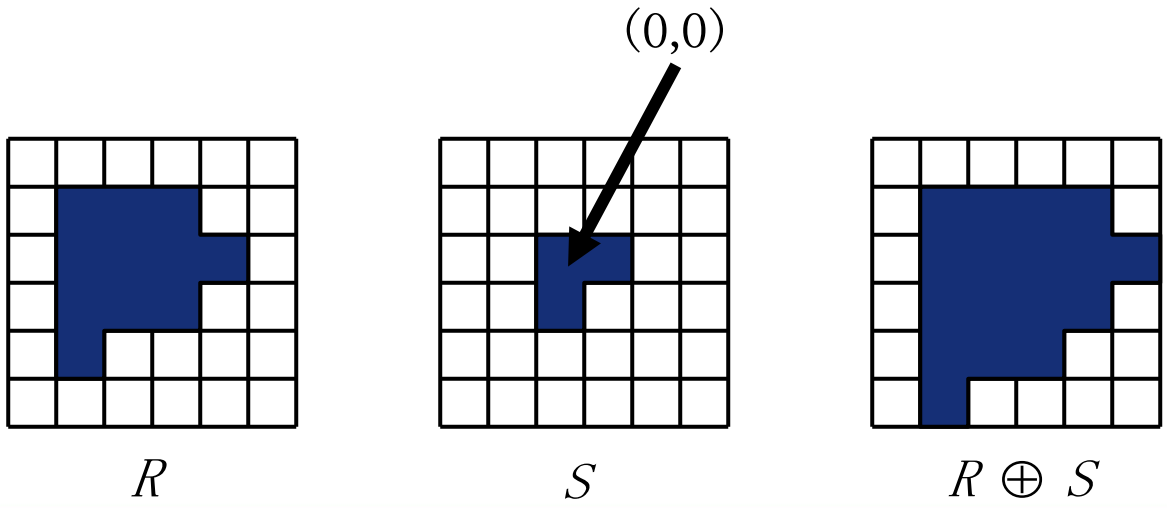
$R \ominus \check{S}$

闵可夫斯基加法（Minkowski addition）

- 定义：

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

取出S中所有的点，根据与从S中取出的点s所对应的向量来平移区域R，然后对全部平移得到的区域与R求并集



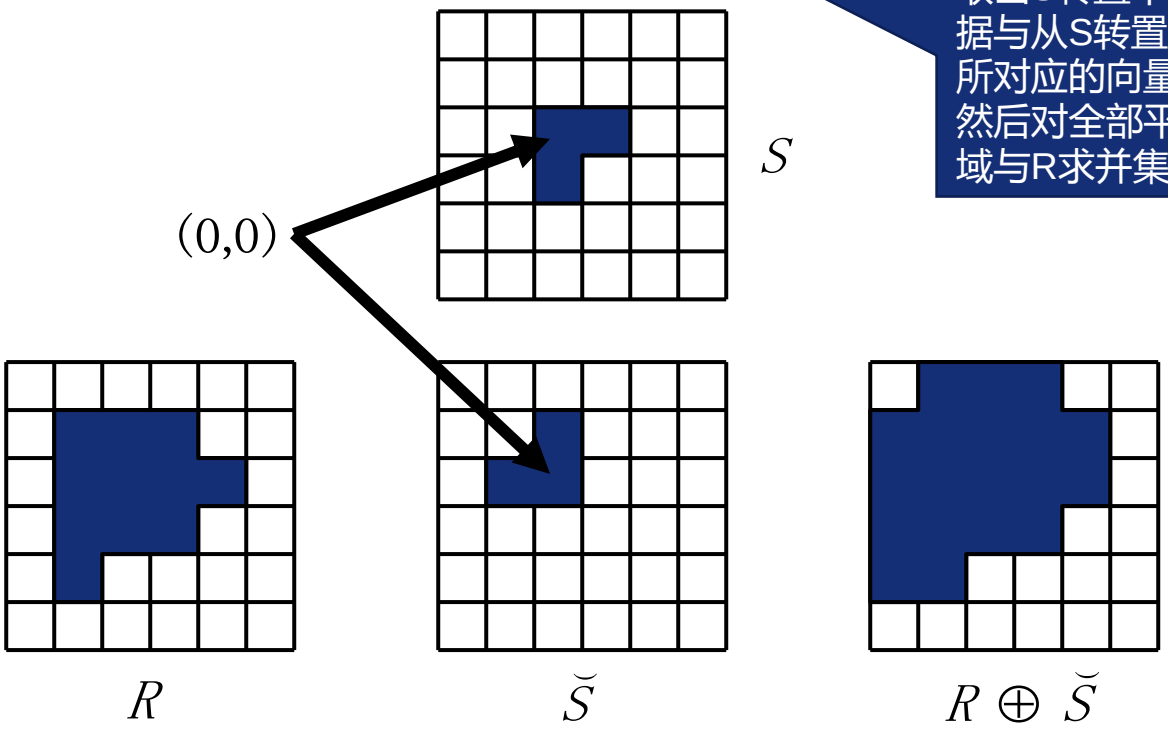
膨胀(Dilation)

- 定义

$$R \oplus \breve{S} = \left\{ t \mid R \cap S_t \neq \varnothing \right\}$$

$$= \bigcup_{s \in S} R_{-s}$$

取出S转置中所有的点，根据与从S转置中取出的点s所对应的向量来平移区域R，然后对全部平移得到的区域与R求并集

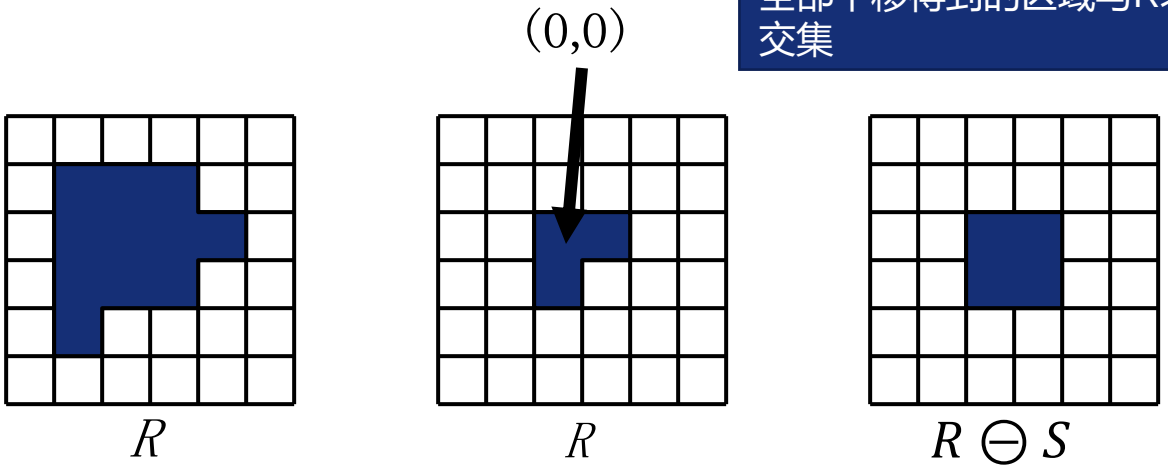


闵可夫斯基减法（Minkowski subtraction）

- 定义

$$R \ominus S = \bigcap_{s \in S} R_s = \{r | \forall\, s \in S : r - s \in R\} = \left\{ r \middle| \left(\check{S}\right)_t \subseteq R \right\}$$

取出S中所有的点，根据与从S中取出的点s所对应的向量来平移区域R，然后对全部平移得到的区域与R求交集

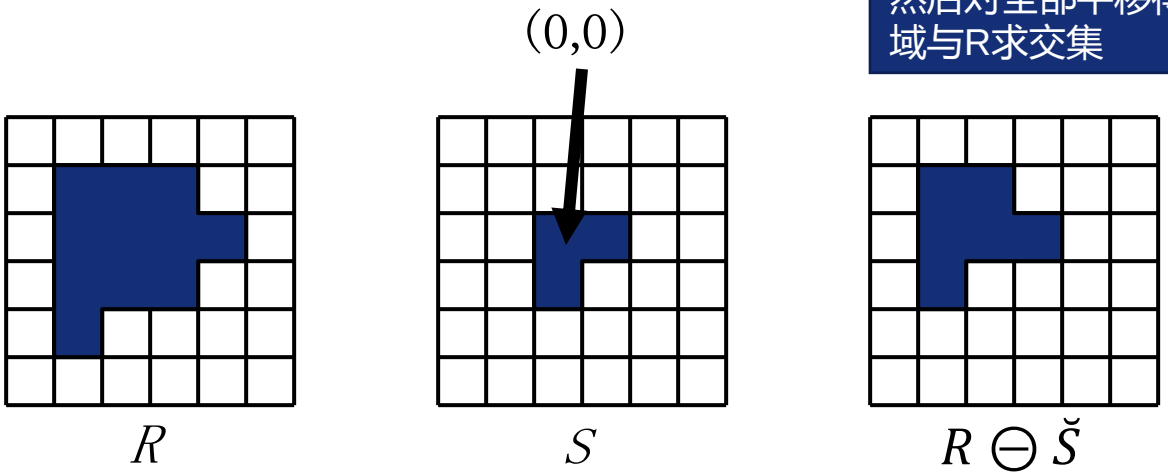


腐蚀(Erosion)

- 定义

$$R \ominus \check{S} = \bigcap_{s \in S} R_{-S} = \{t | S_t \subseteq R\}$$

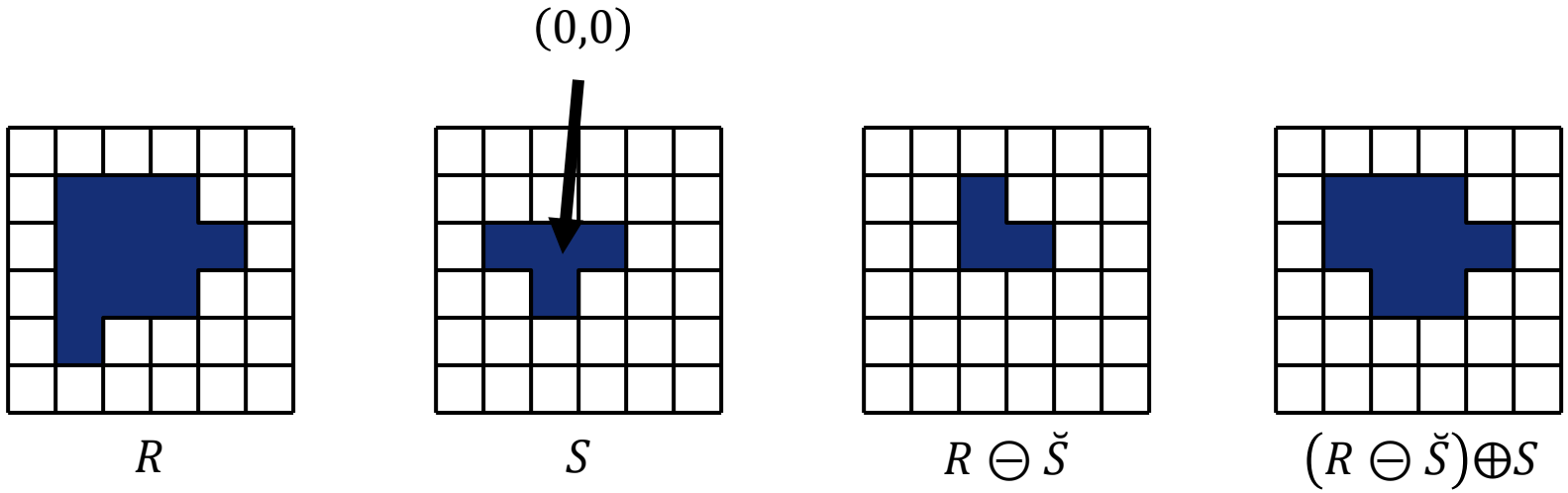
取出S转置中所有的点，根据与从S转置中取出的点s所对应的向量来平移区域R，然后对全部平移得到的区域与R求交集



开运算(Opening)

- 定义

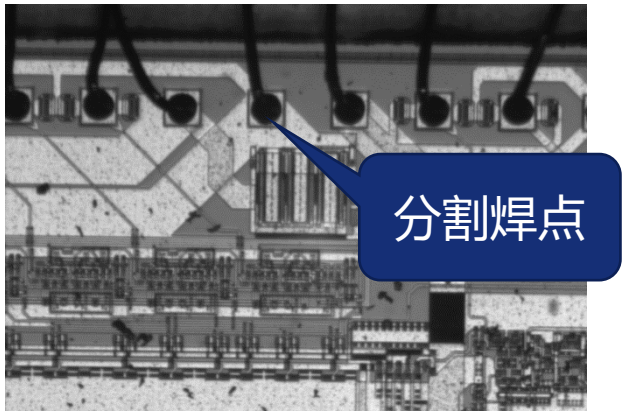
$$R \circ S = (R \ominus \check{S}) \oplus S = \bigcup_{S_t \subseteq R} S_t$$



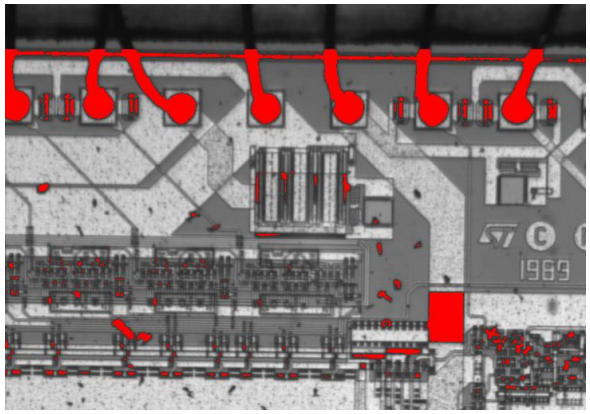
形态学处理

DAHENG IMAGING | 大恒图像

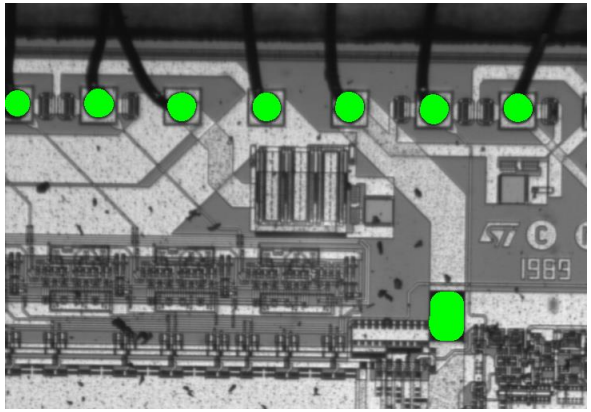
开运算(Opening) - 应用：噪声抑制，目标检测



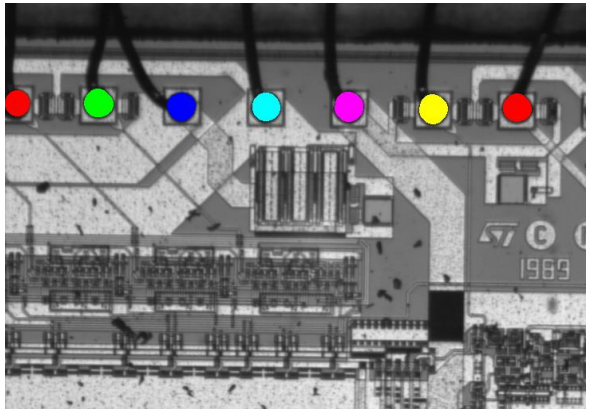
输入图像



分割



开运算

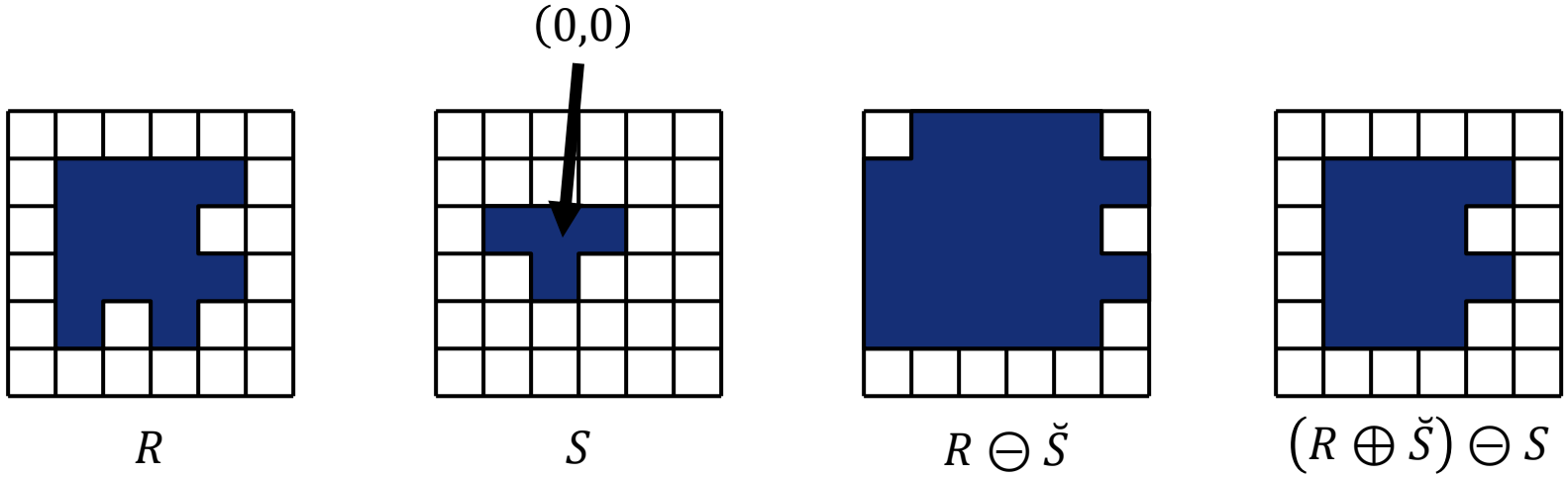


区域选择

闭运算(Closing)

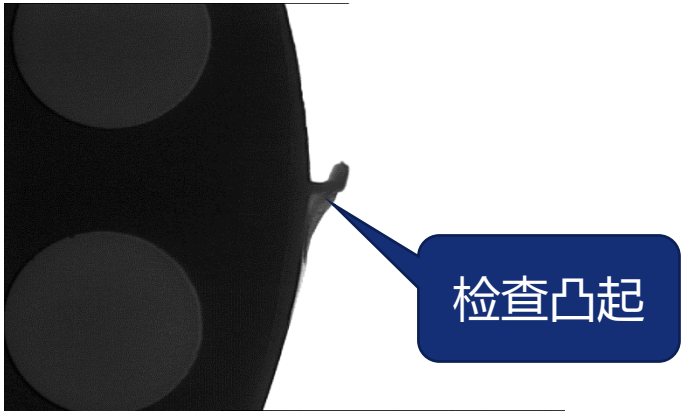
- 定义

$$R \cdot S = \left(R \oplus \check{S}\right) \ominus S = \overline{\bigcup_{S_t \subseteq \bar{R}} S_t}$$

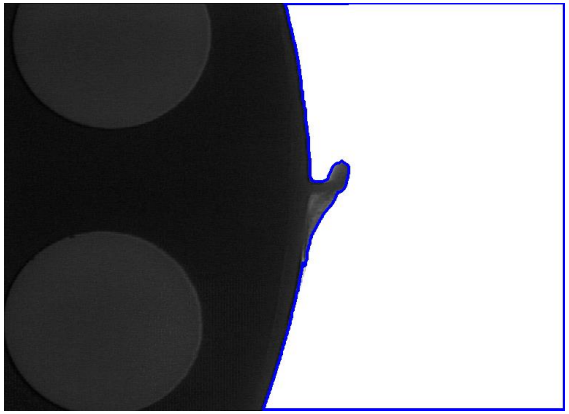


形态学处理

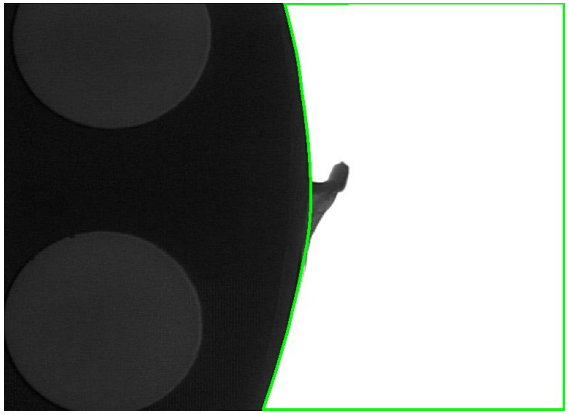
闭运算(Closing) - 应用：缺陷检测



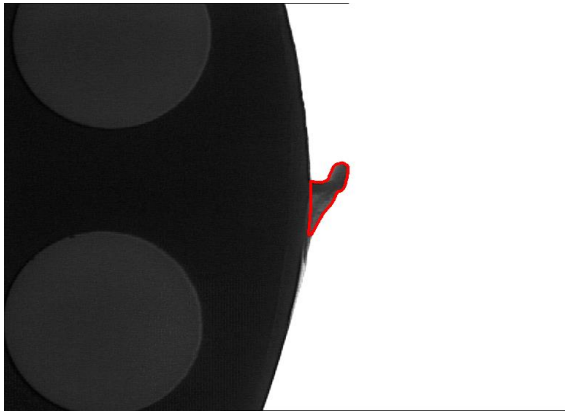
输入图像



分割



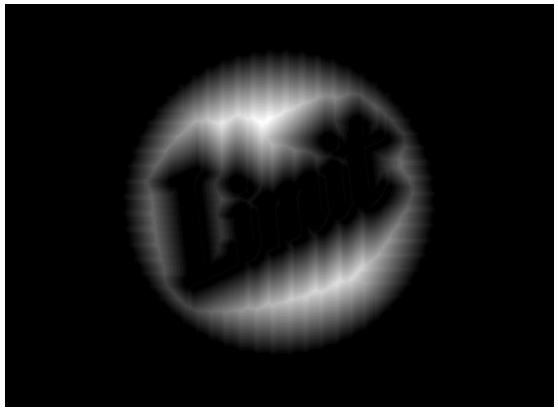
闭运算



缺陷检测

距离变换（Distance transformation）

Input region



City-block distance

Chessboard distance

Chamfer-3-4 distance

形态学处理

骨架(Skeleton)

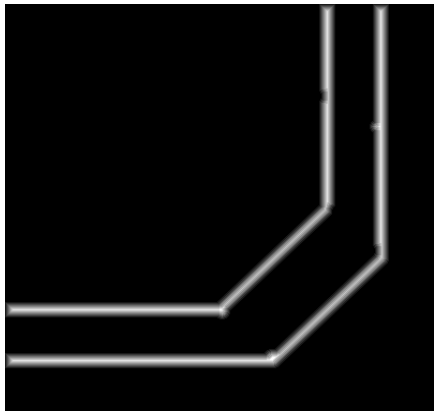
- 一个区域的骨架代表了区域的中心线。骨架上的这些点与一个区域同一位置上的边界距离都是一样的
- 骨架上的点也可以描述成一幅距离转换图像中的点，这些点到边界的垂直距离为最远距离。
- 骨架的计算通常是应用回归hit-or-miss 转换, 这种转换用来估计边界的点，这些点往往是不能成为skeleton的

- 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

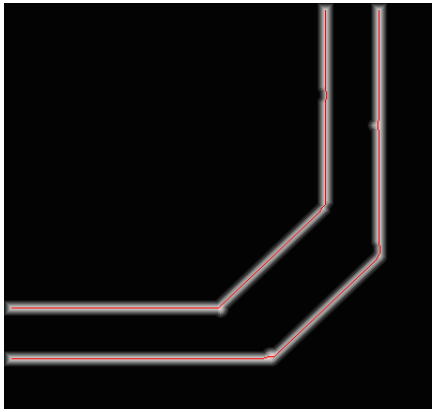
形态学处理

应用 - PCB电路检测

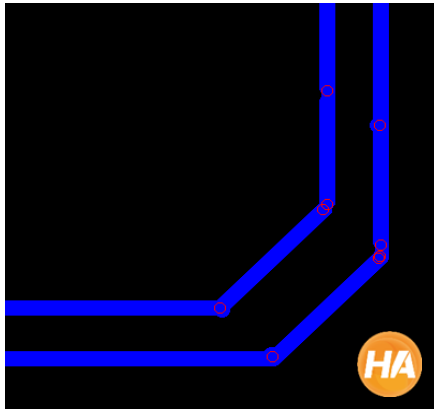
模拟PCB电路缺陷



距离变换



骨架



缺陷检测

形态学处理

总结

- 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

03 特征提取

特征提取

什么是特征提取?

- 通常，必须从分割结果中选出某些区域或轮廓。比如，为了去除分割结果中的不必要的部分，而且，通常感兴趣的是对物体进行测量。
- 或许，想对物体进行分类以确定物体的类型，比如在OCR中就需要类似的处理。
- 所有这些应用都需要从区域或轮廓中确定一个或多个特征量。
- 确定的特征量被称为特征，通常为实数。
- 确定特征的过程称为特征提取。

区域特征: Area and Moments

- 最简单的区域特征: area

$$a = |R| = \sum_{(r,c)\in R} 1$$

- moments定义 $(p \geq 0, q \geq 0)$:

$$m_{p,q} = \sum_{(r,c)\in R} r^p c^q$$

  - 面积是被称为区域的矩的广义特征中的一个特例，$m_{0,0}$就是区域的面积。

- 归一化的矩 $(p + q \geq 1)$:

$$n_{p,q} = \frac{1}{a} \sum_{(r,c)\in R} r^p c^q$$

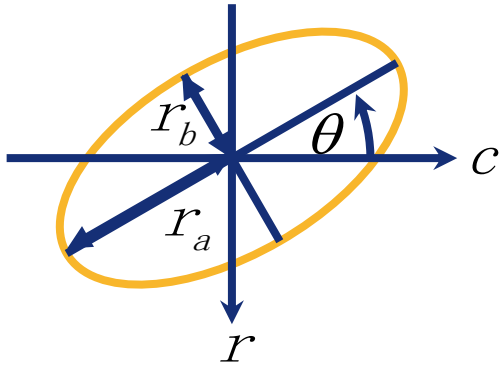
  - $(n_{1,0}, n_{0,1})$就是区域的重心，它能被用来描述区域的位置。

区域特征: Moments

- 归一化的矩是由图像中的位置决定的。如果要使特征不随图像中区域的位置变化而变化，可通过计算相对于区域重心的矩来实现。归一化中心矩（$p+q \geq 2$）：

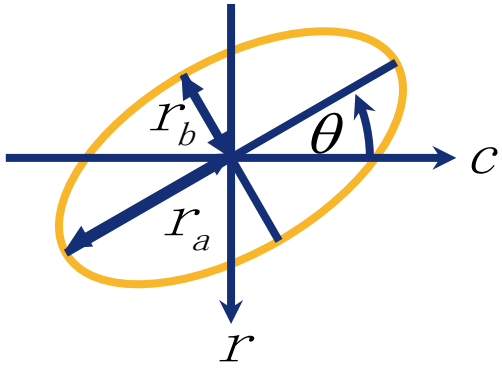
$$\mu_{p,q} = \frac{1}{a} \sum_{(r,c) \in R} (r - n_{1,0})^p (c - n_{0,1})^q$$

- 优点：
  - 归一化中心矩对区域平移、旋转、尺度变化具有不变性
- 应用：
  - 印刷体字符的识别
  - 飞机形状区分
  - 景物匹配
  - 染色体分析
  - 等等



区域特征: Moments

- 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”
- 应用: 寻找具有相同矩的椭圆区域
- 椭圆的参数:
  - 椭圆中心=区域重心
  - 椭圆长轴



$$r_a = \sqrt{2\left(\mu_{2,0} + \mu_{0,2} + \sqrt{\left(\mu_{2,0} - \mu_{0,2}\right)^2 + 4\mu_{1,1}^2}\right)}$$

  - 椭圆短轴

$$r_b = \sqrt{2\left(\mu_{2,0} + \mu_{0,2} - \sqrt{\left(\mu_{2,0} - \mu_{0,2}\right)^2 + 4\mu_{1,1}^2}\right)}$$

  - 椭圆角度

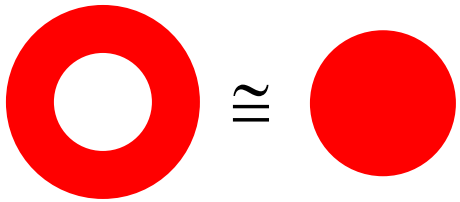
$$\theta = -\frac{1}{2}\arctan\frac{2\mu_{1,1}}{\mu_{0,2} - \mu_{2,0}}$$

区域特征: Moments

- 通过椭圆的参数，能推导出另一个非常有用的特征：各向异性（Anisometry）

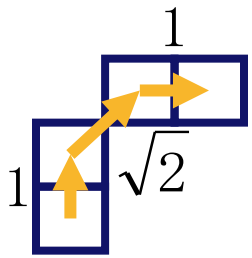
$$r_a / r_b$$

- 运算符：eccentricity和elliptic_axis
- 优点：区域缩放时，该特征量是保持恒定不变的，可以描述一个区域的细长程度
- 椭圆特征问题：
  - 只有$r_a \neq r_b$ 在才能确定区域方向，像正圆、正方形等对象无法确定方向
  - 区域中的孔会对特征值（r_a / r_b）产生一定的的影响



区域特征: 其它特征

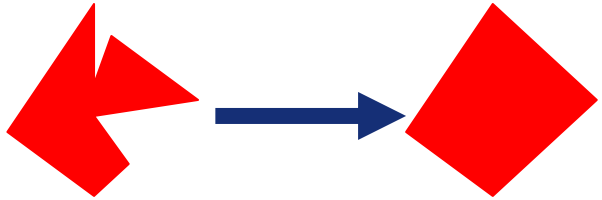
- Contlength：区域边界的长度
  - 水平线段和垂直线段的欧几里得距离都是1
  - 对角线段的距离是$\sqrt{2}$



- Compactness：区域紧密性
  - 区域紧密性的度量$c = l^2/(4\pi a)$
  - l 为轮廓长度，a是区域的面积
  - 所有圆形的紧密性特征值都是1，而其他区域的紧密性特征值更大。

区域特征: 其它特征

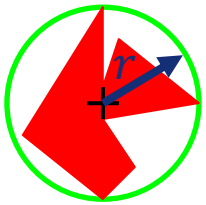
- Convexity：区域面积与凸包面积的比例



- Circularity：区域面积与最小外接圆的比例

$$C = \frac{a}{\pi r^2}$$

- Roundness：圆度，与Circularity计算方式不同



$$d = \frac{1}{a}\sum \|r - r_i\|$$

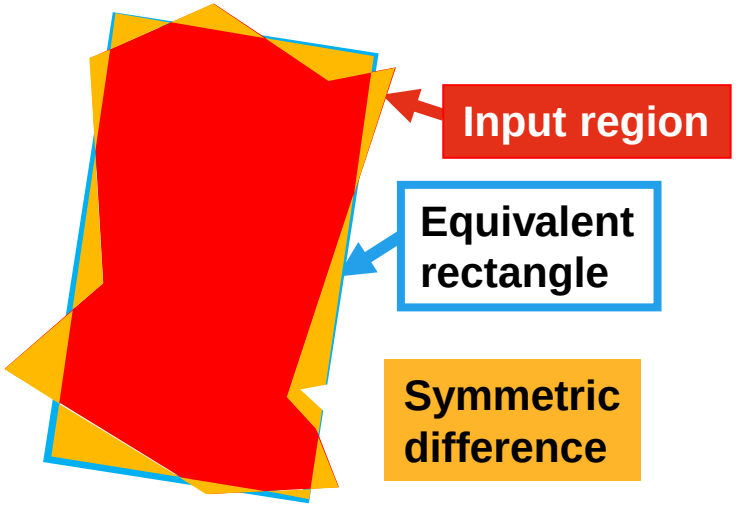
$$\sigma^2 = \frac{1}{a}\sum (\|r - r_i\| - d)^2$$

$$R = 1 - \sigma/d$$

特征提取

区域特征: 其它特征

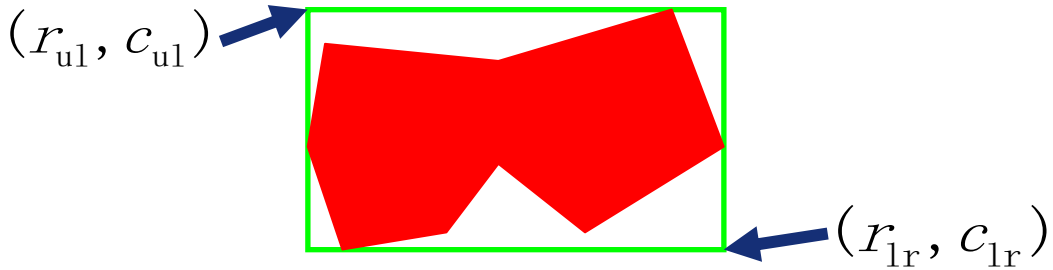
- 除了圆和椭圆之外，典型的标准形状是矩形。
- 经典形状特征（如roundness, circularity或compactness）不适合选择矩形。
- HALCON提供了一个专为选择/评估矩形而设计的功能。
- Rectangularity



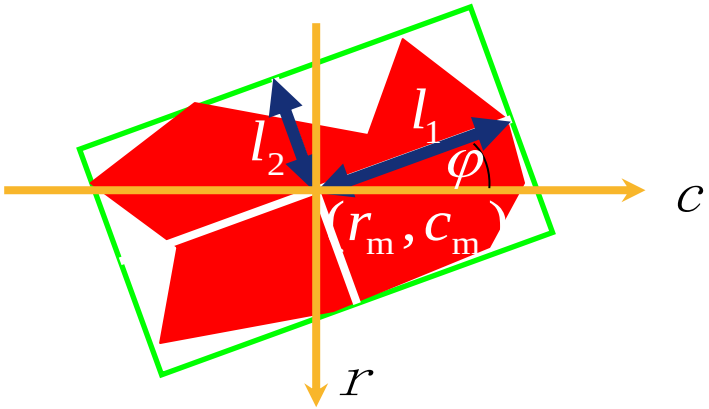
- **注意：**对于无法使用二阶矩计算方向的区域（如，正方形），计算出的矩形度可能比真实值小最多10%

区域特征: 区域最小外接矩形

- 平行于主轴的最小外接矩形：smallest_rectangle1：

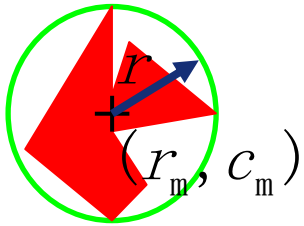


- 定义：$r_{ul} = \min r$, $r_{lr} = \max r$, $c_{ul} = \min r$, $c_{lr} = \max r$, $(r, c) \in R$
- 任意方向的最小外接矩形(复杂但快速的方法)：smallest_rectangle2:

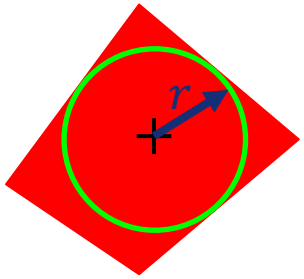


区域特征: 区域最小外接圆和最大内接圆

- smallest_circle：计算最小外接圆



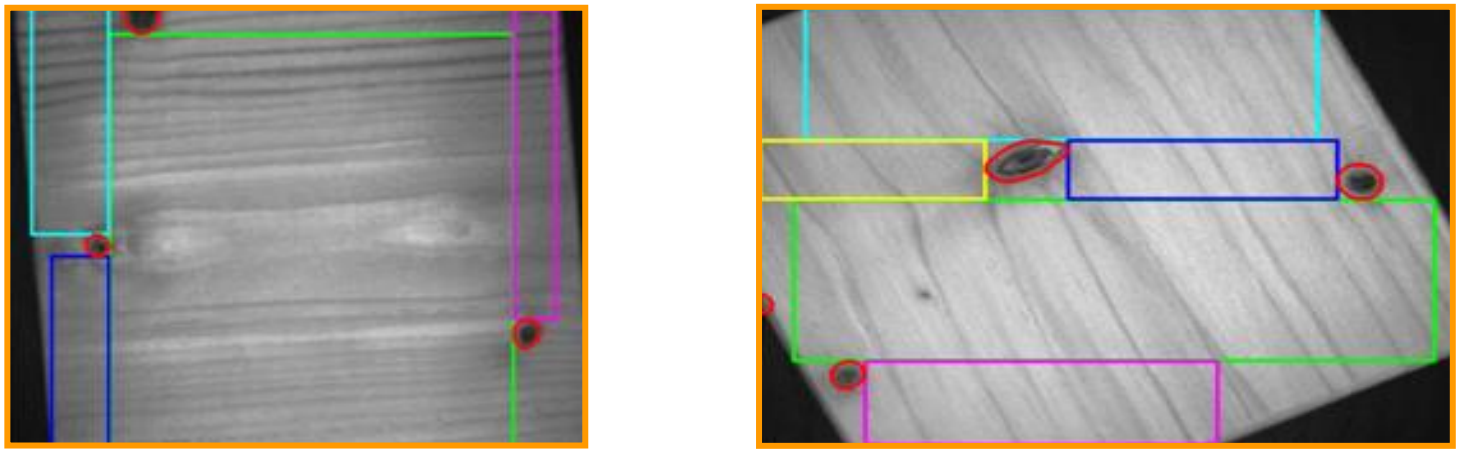
- inner_circle：计算最大内接圆



特征提取

区域特征: 区域最大内接矩形

- 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

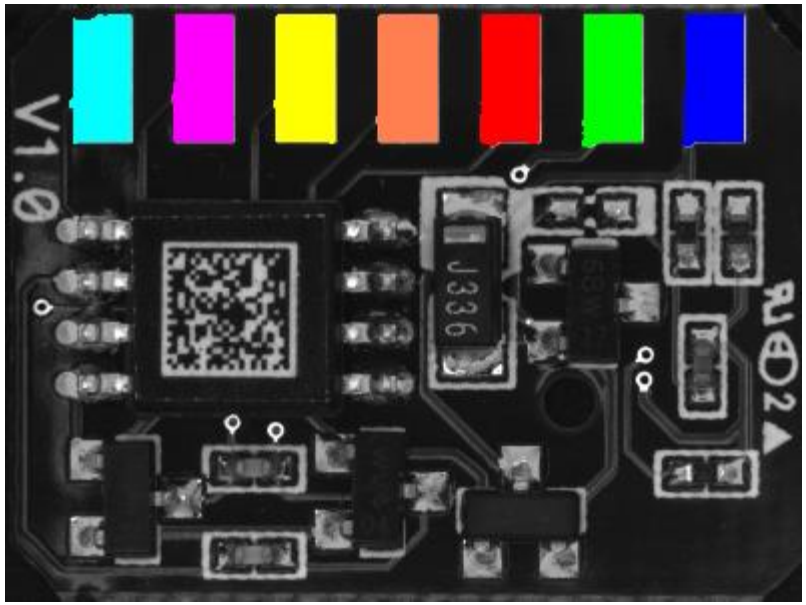


示例应用: 木头上节点

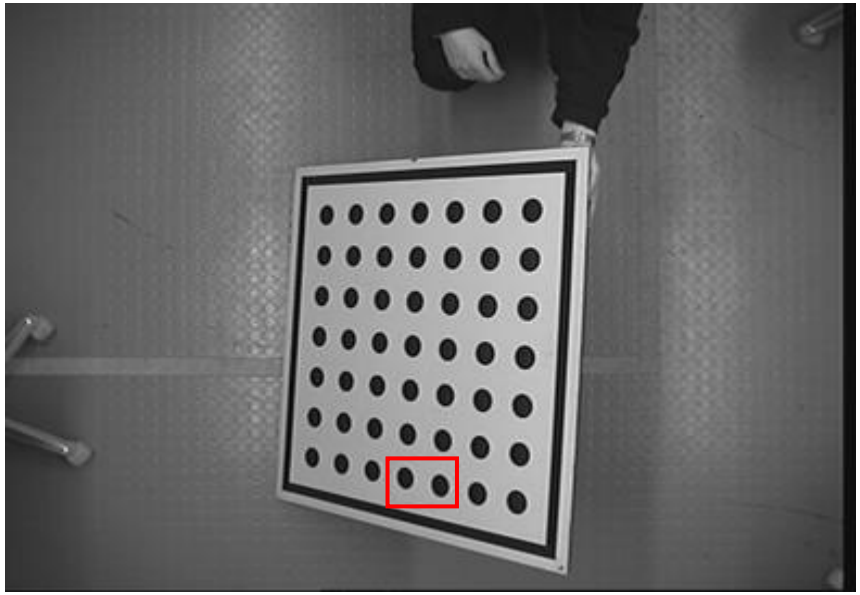
特征提取

区域选择

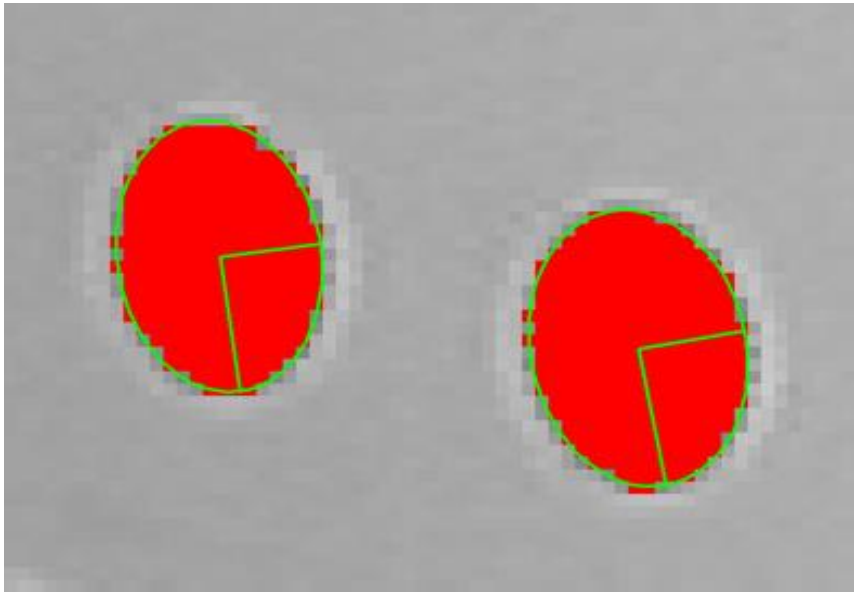
```
dev_close_window()
dev_open_window (0, 0, 400, 300, 'black', WindowHandle)

read_image (Image, 'printer_chip/printer_chip_01')
threshold (Image, Region, 128, 255)
connection (Region, ConnectedRegions)
select_shape (ConnectedRegions, SelectedRegions, ['area','rectangularity'], 'and', [27000,0.8], [31000,1])
dev_clear_window()
dev_display (Image)
dev_display (SelectedRegions)
```



区域特征: Moments - 应用



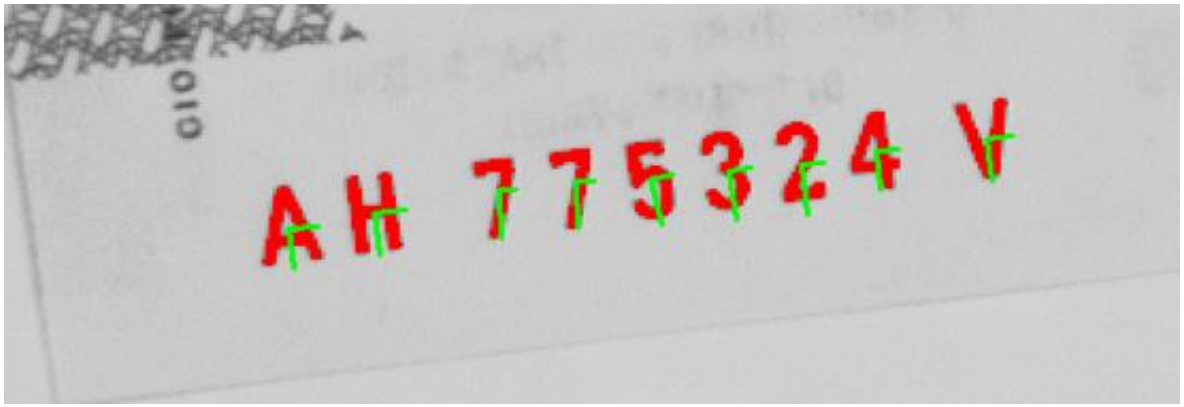
输入图像



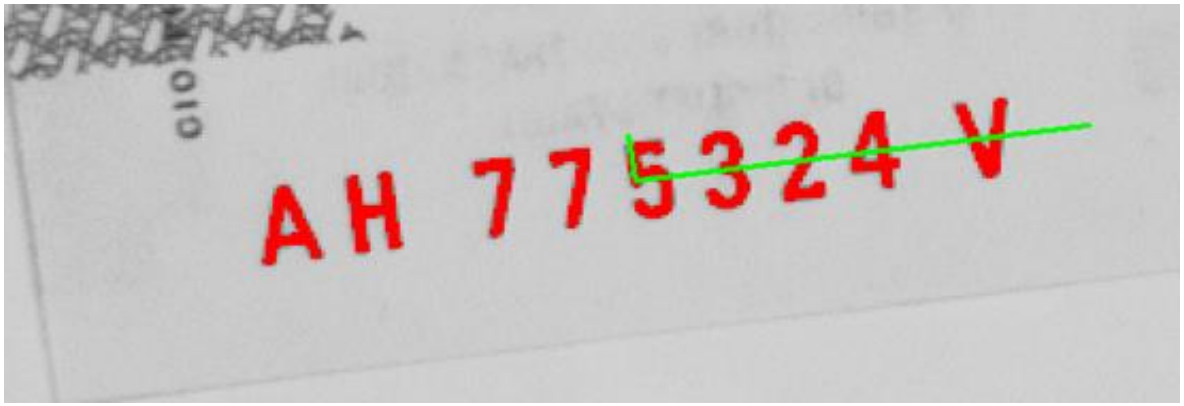
具有相同矩的椭圆和区域

特征提取

区域特征: Moments - 应用



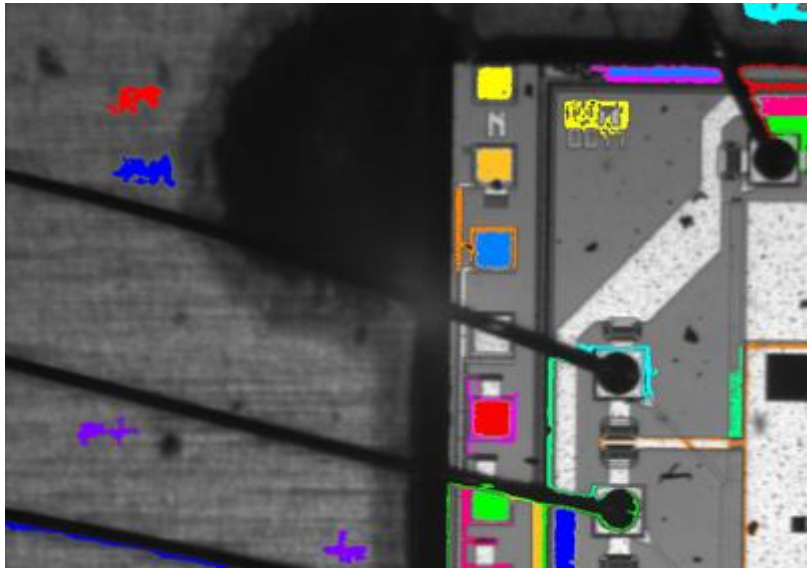
单个字符的方向



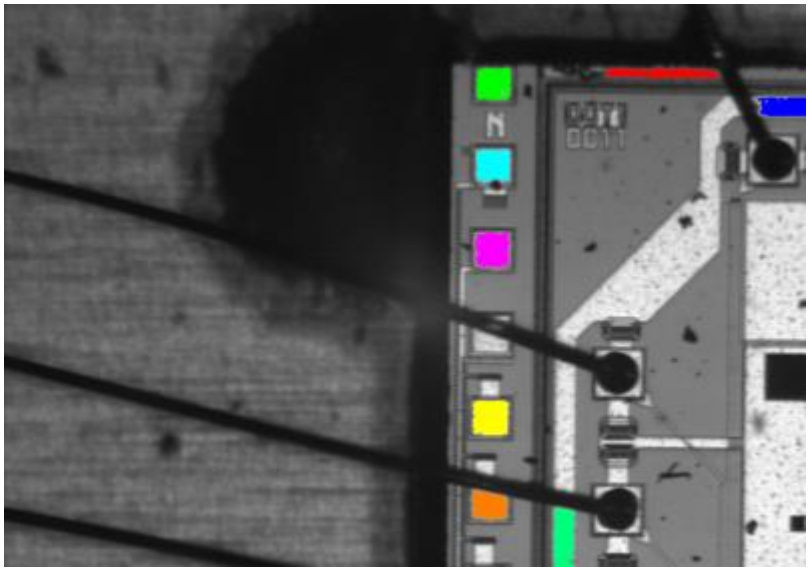
所有区域的方向，比如用于旋转校正

特征提取

rectangularity: 应用

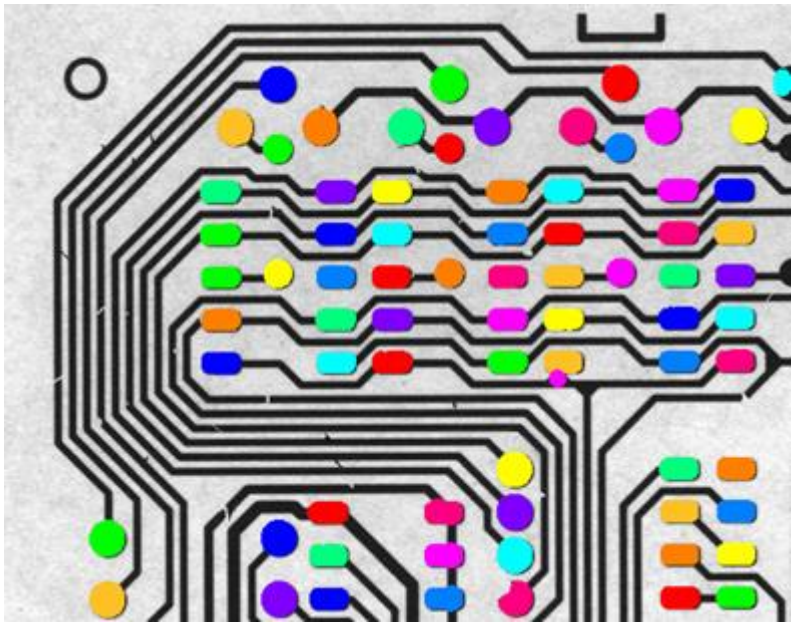


图像分割

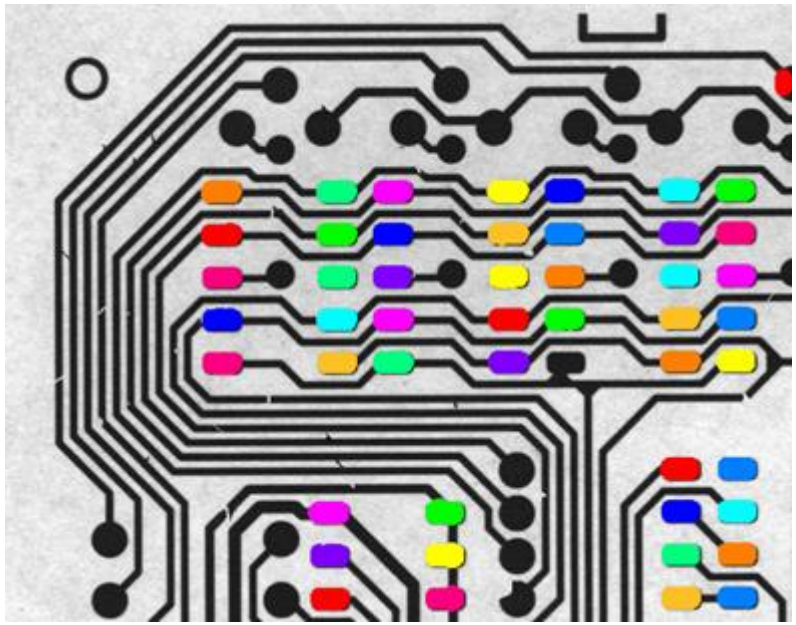


选择区域

rectangularity: 应用



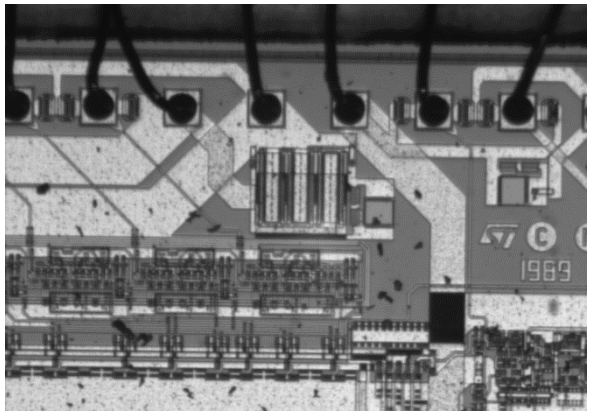
图像分割



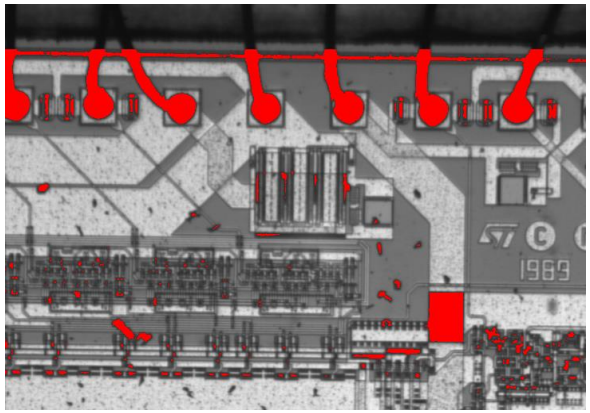
选择区域

特征提取

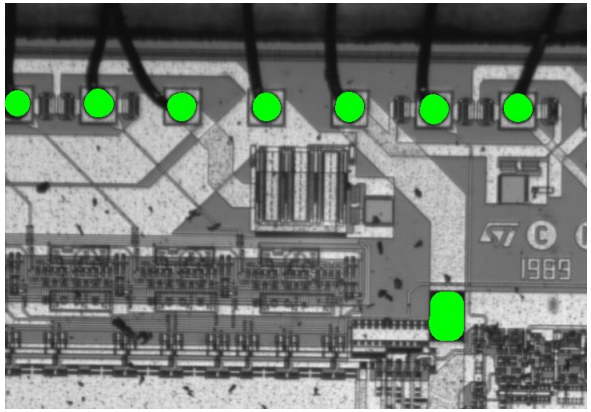
circularity: 应用



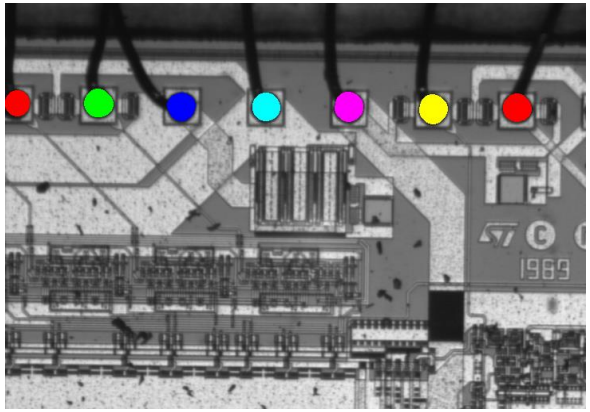
输入图像



图像分割



图像开运算



区域选择

DAHENG IMAGING | 大恒图像

中国大恒(集团)有限公司北京图像视觉技术分公司

China Daheng Group, Inc. Beijing Image Vision Technology Branch