大恒图像
IMAVISION

中国大恒（集团）有限公司
北京图像视觉技术分公司

HALCON中的定位方法

大恒图像深圳办 技术部经理 偏召华

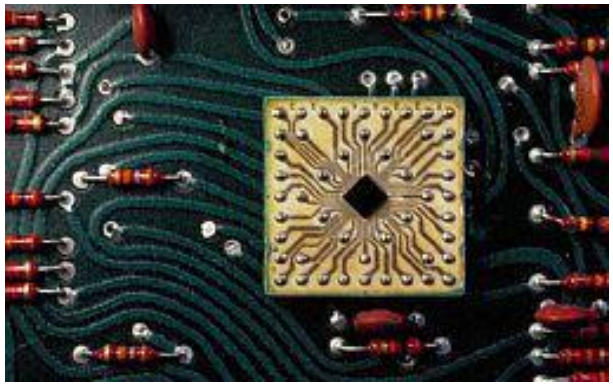
概要

- 基本介绍
- 方法介绍
  - 基于形状的匹配
  - 基于组件的匹配
  - 基于互相关匹配
  - 变形匹配
  - 三维匹配
- 总结

什么是匹配

- 在图像中找到物体
  - 已知
    - 模板图像
    - 搜索图像
    - 转换类型
  - 待定
    - 模板物体在模板图像和搜索图像中的关系

参考图片

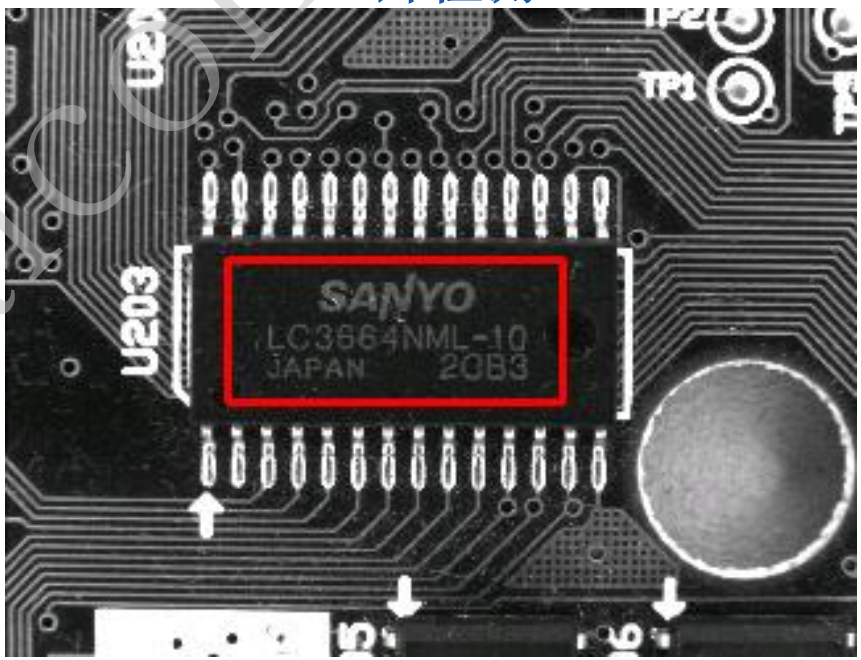


搜索图像

匹配的典型应用

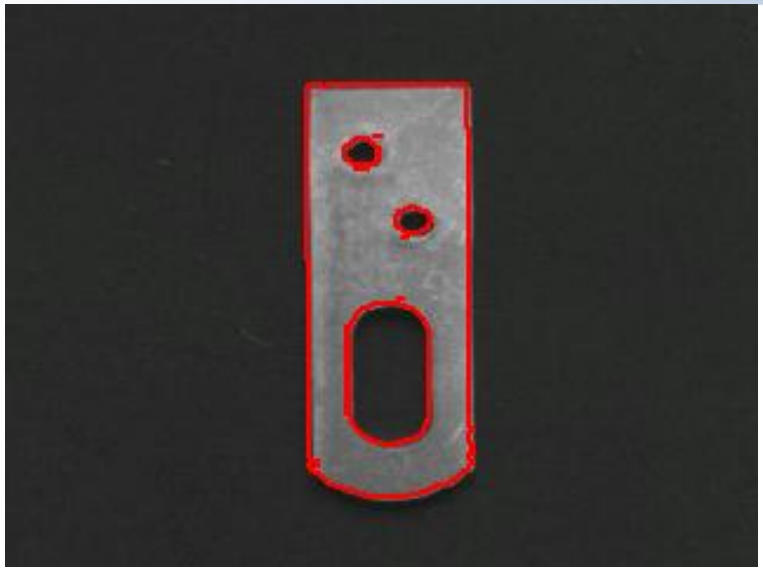
印刷检测

芯片检测

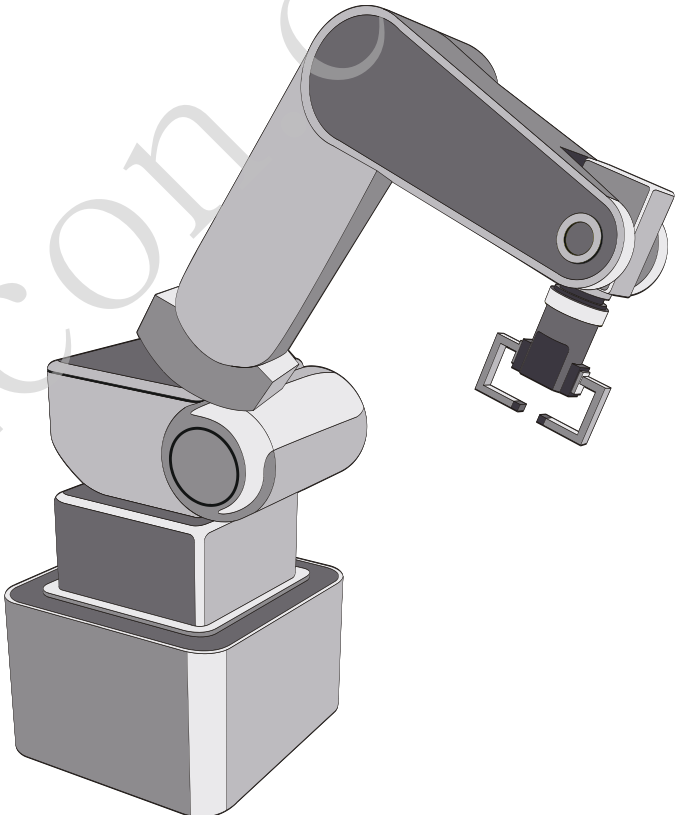


瓶盖检测

匹配的典型应用



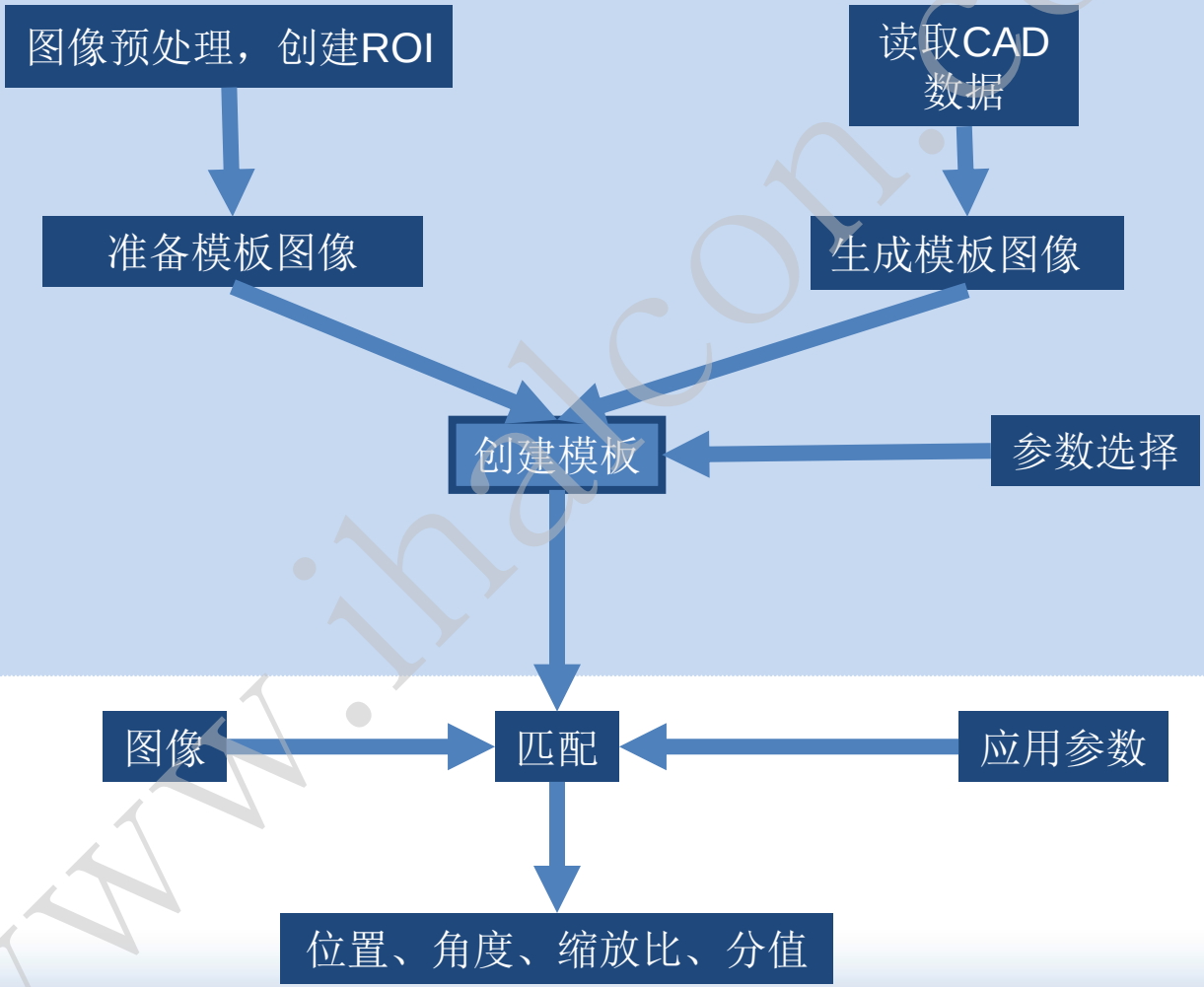
加工件检测



机械手定位

基于形状的匹配

形状匹配流程



创建模板

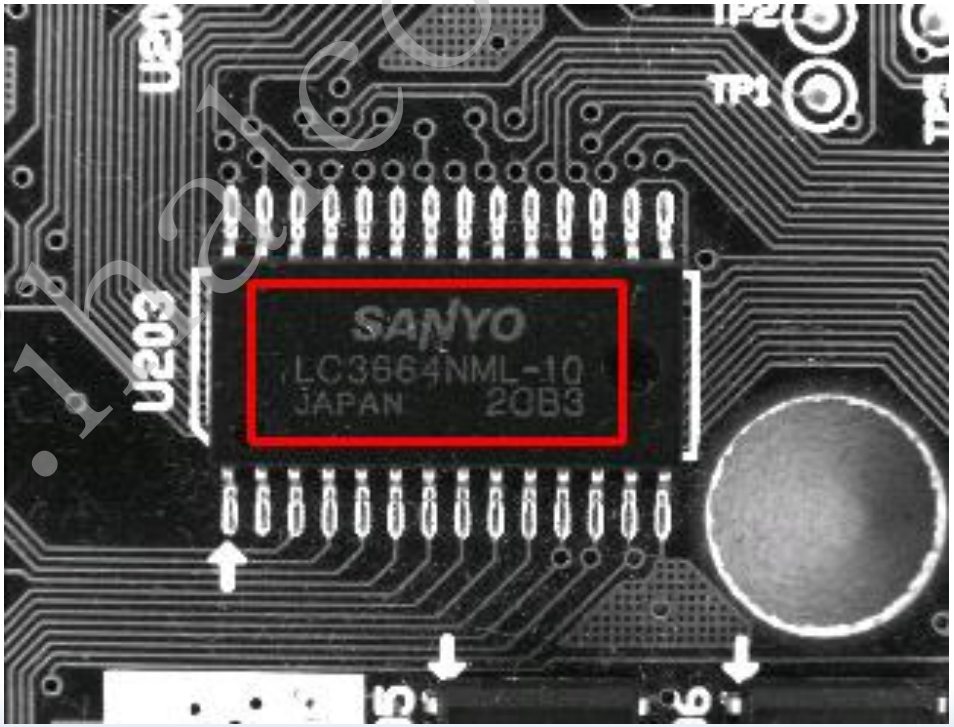
查找模板

创建ROI

- 使用Halcon算子可以方便的设置ROI
- 标准形状
  - draw_rectangle1/2
  - draw_circle
  - draw_ellipse
  - draw_line
- 任意形状
  - draw_region
  - draw_polygon
- 生成标准ROI
  - gen_rectangle1/2
  - gen_circle
  - gen_ellipse
  - gen_region_line
- 通过XLD创建AOI
  - gen_region_contour_xld
  - gen_region_polygon_xld

生成ROI

```
read_image(Image,'board/board-01.tif')
get_image_size(Image,Width,Height)
draw_rectangle2(Window,Row,Column,Phi,Length1,Length2)
gen_rectangle2(ROI,Row,Column,Phi,Length1,Length2)
```



ROI修正

- 修正函数
  - erosion_* 减小ROI
  - dilation_* 扩大ROI
  - shape_trans 形状转换
  - boundary 像素级边界
  - move_region 移动区域到新位置
- 组合
  - Intersection 交集
  - Difference 差集
  - Union2 并集

生成ROI

```
dev_display(Image)
draw_circle(Window,Row1,Column1,Radius1)
draw_circle_mod(Window,Row1,Column1,Radius1+10,
                Row2,Column2,Radius2)
gen_circle(CircleSmall,Row1,Column1,Radius1)
gen_circle(CircleLarge,Row2,Column2,Radius2)
difference(CircleLarge,CircleSmall,DoughnutROI)
dev_display(DoughnutROI)
```

直接创建模板

- create_shape_model(
  - Template , // 模板图像
  - NumLevels, // 图像金字塔
  - AngleStart, // 起始角度
  - AngleExtent, // 角度范围
  - AngleStep, // 角度步长
  - Optimization, // 优化算法
  - Metric, // 极性
  - Contrast, // 对比度
  - MinContrast, // 最小对比度
  - ModelID // 模板ID
  -)
- create_scaled_shape_model
- create_aniso_shape_model

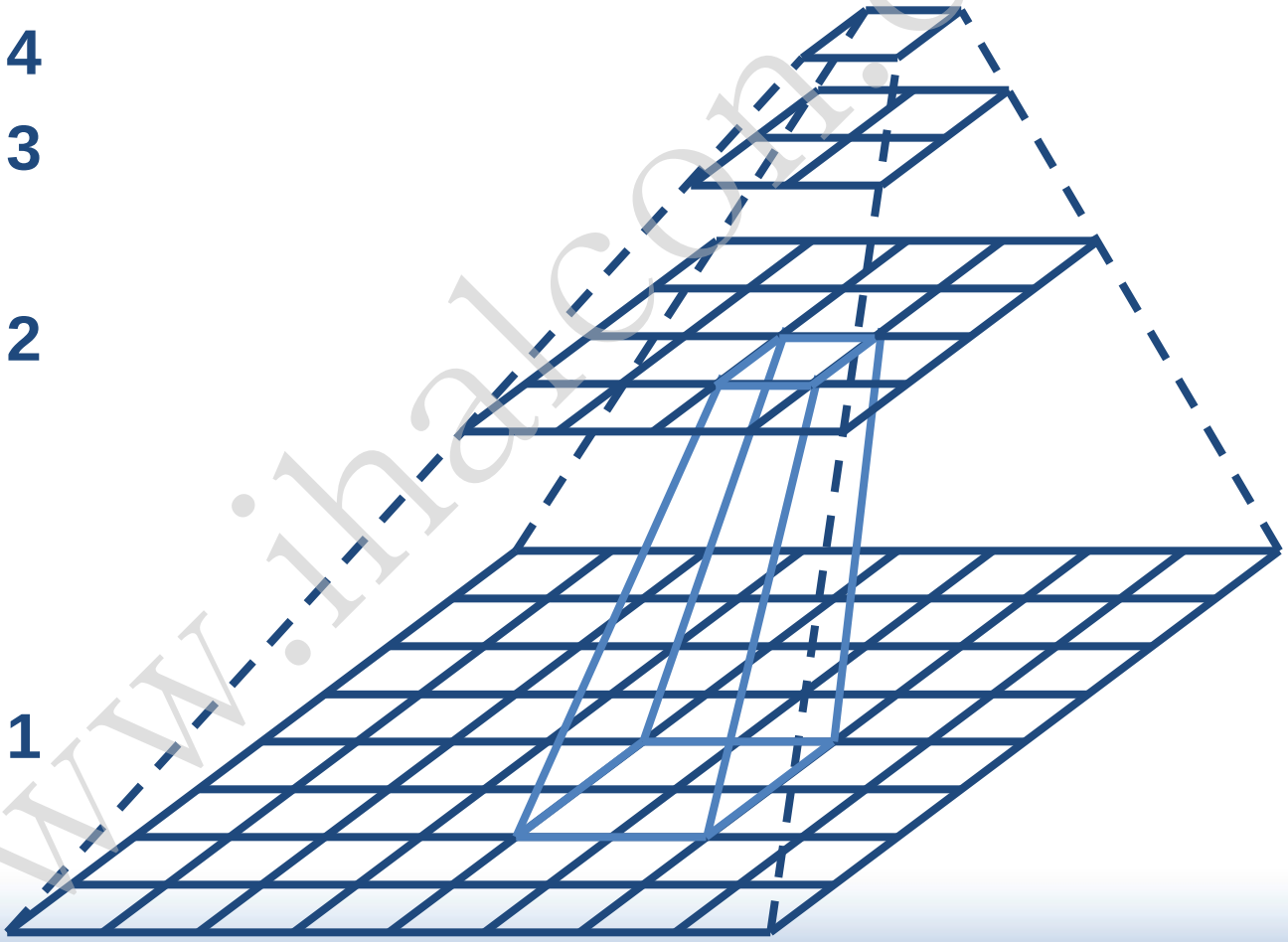
模板参数-NumLevels

Level 4

Level 3

Level 2

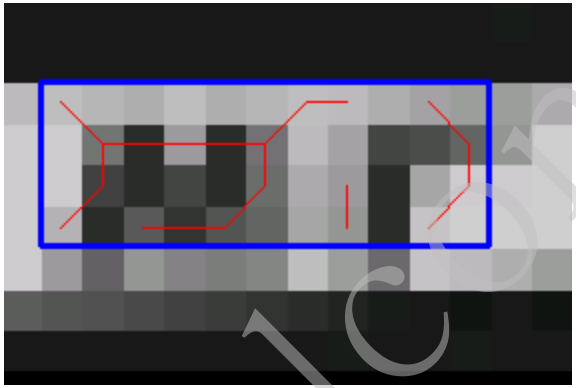
Level 1



模板参数-NumLevels

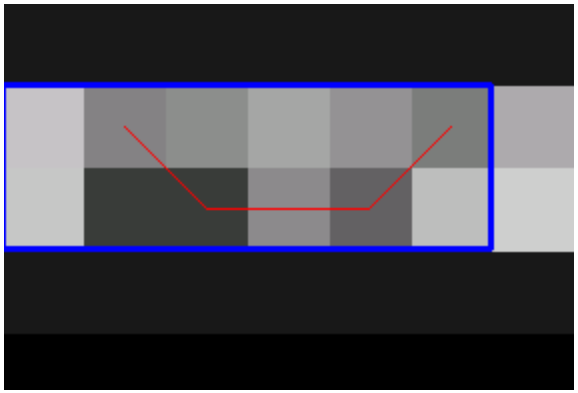
	Level
	4
	3
	2
	1

模板参数-NumLevels



Level 6

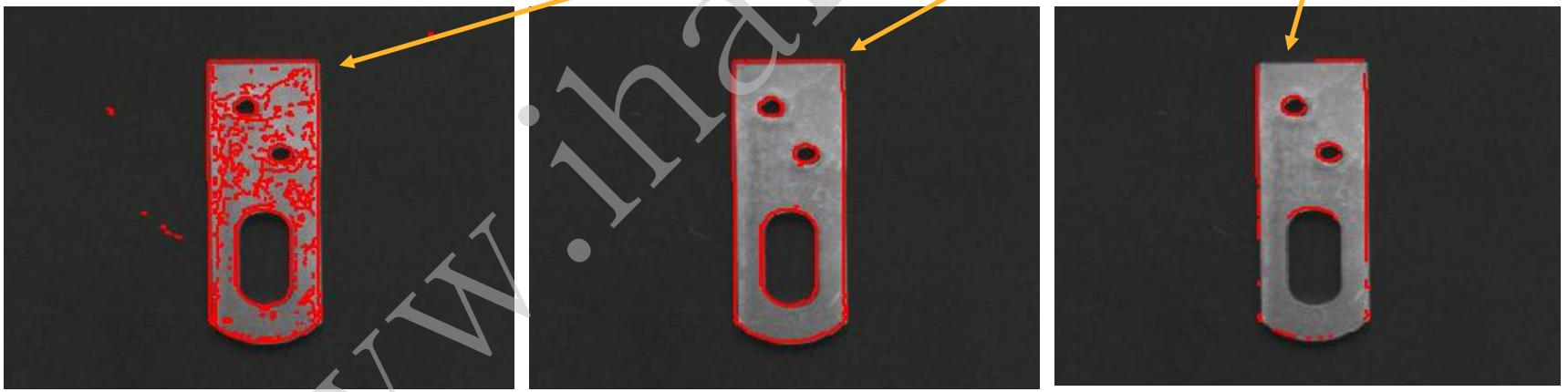
Level 1



Level 7 (太高)

模板参数-Contrast

```
* 创建ROI
* 取图
inspect_shape_model(Image,ModelImage,ModelRegion,1,Contrast)
dev_display(Image)
dev_display(ModelRegion)
```

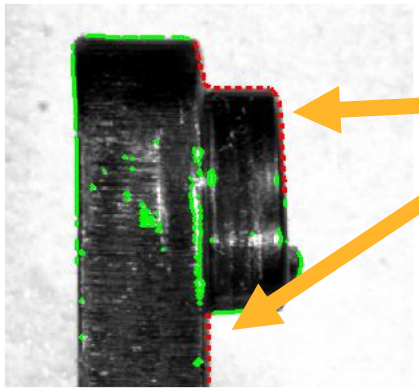


对比图太低　　合适的对比度　　对比度太高

模板参数-Contrast

- **参数Contrast不仅仅是对比度，根据数组元素数量不同，其意义不同**
  - 1个元素时：128，对比度，直接提取边缘
  - 2个元素时：[100, 128]，表示使用磁滞分割来提取边缘
  - 3个元素时：[100, 128, 10]，前两个参数同2，最后一个参数表示所提取边缘的最小长度为10.

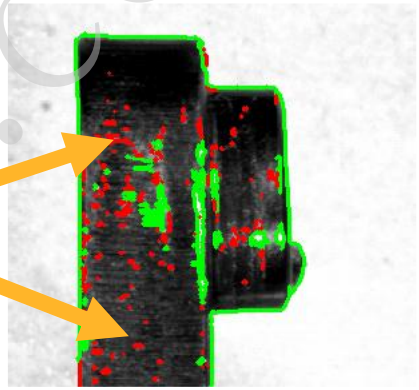
模板参数-Contrast



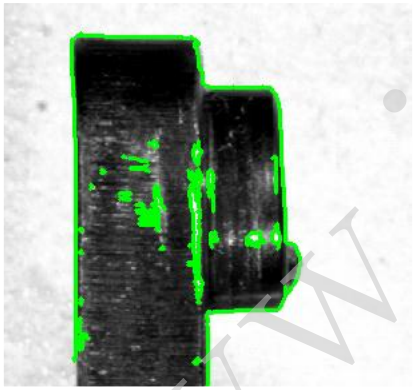
丢失边缘

对比度太高

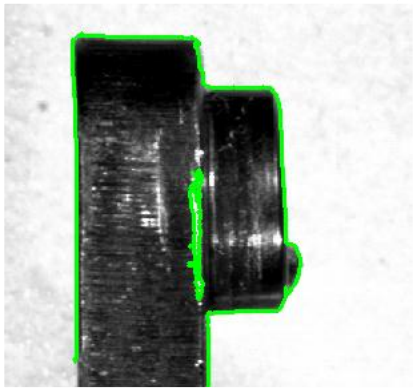
边缘过多



对比度太低



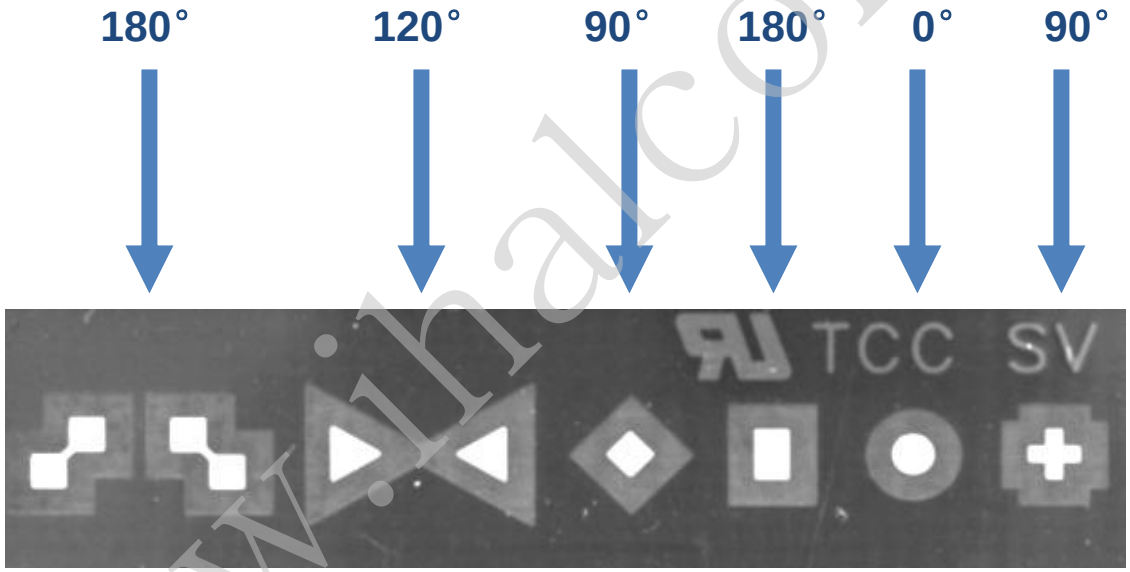
使用磁滞分割



使用边缘选择

模板参数-AngleExtent

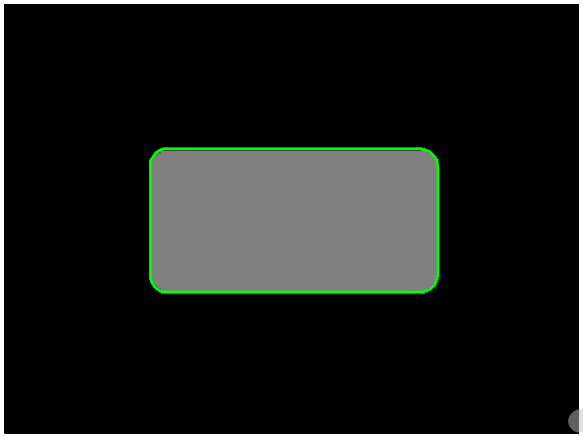
- 对称性和角度范围选择



模板参数-AngleStep

- 弧度非角度
- 选择标准
  - 模板越大，角度步长越小
  - 要求越精确，步长越小
- 速度和内存
  - 步长越小，占用内存越多，定位速度越慢
- 如果没有特殊要求，可选”auto”让系统做最佳选择

模板参数-Scale



合成形状模板

例子

模板参数-Optimization

- 一些模板包含了太多像素点，这导致
  - 模板过大
  - 增加执行时间
  - 增加了内存需求
- 参数Optimization用来减少这些点
  - none　　不减少像素
  - point_reduction_low　　大约一半点
  - point_reduction_medium　　大约1/3
  - point_reduction_high　　大约1/4
- 减少点可能导致的问题
  - 可能导致无法创建高层金字塔
  - 有可能会降低结果的精度和准确度
- 原则
  - 边缘较多时才减少

模板参数-Optimization

- 相同环境下，Optimization取值不同时的运行时间对比
  - none 14.53 ms
  - point_reduction_low 12.53 ms
  - point_reduction_medium 11.39 ms
  - point_reduction_high 10.67 ms

模板参数-Optimization

- 除了减少像素，该参数也可以控制模板的创建方式，来选择内存优先还是速度优先
- 第二个值可选下面两个
  - 'pregeneration'
    - 模板预先创建，牺牲内存来换取查找速度
  - 'no_pregeneration'
    - 在查找时才创建必须数据，占用内存少
- 如果系统中所有选择相同，可以
  - set_system('pregenerate_shape_models','true'/'false')
- 如果没有设置，默认为
  - set_system('pregenerate_shape_models','false')

模板参数-Optimization

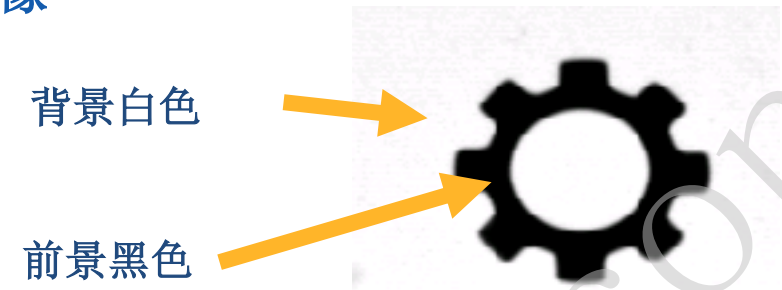
相同环境下，Optimization第二个参数取值不同时的运行时间对比

Example	Create No-Pre	Create Pre	Find No-Pre	Find Pre
find_scaled_shape_model.dev	155ms	38s	72ms	69ms
first_example_shape_matching.dev	72ms	13s	61ms	72ms
multiple_models.dev	100ms	7.4s	60ms	60ms
multiple_scales.dev	96ms	13s	40ms	44ms
print_check.dev	113ms	1.1s	13ms	13ms

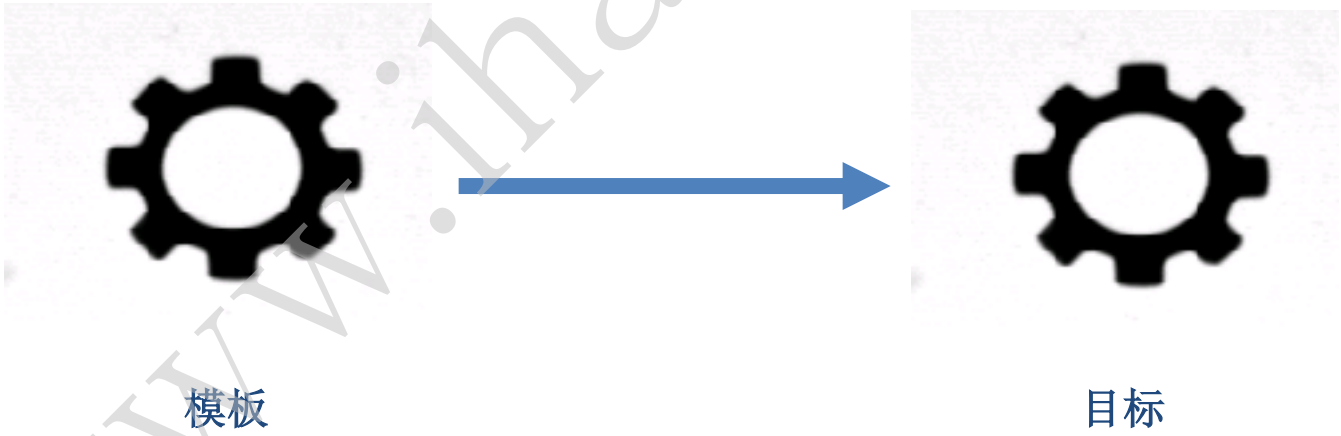
- 因此，建议当内存较大，就选预创建的方式，如果CPU速度快，就可以选另外方式。

模板参数-Metric

- 照明与成像

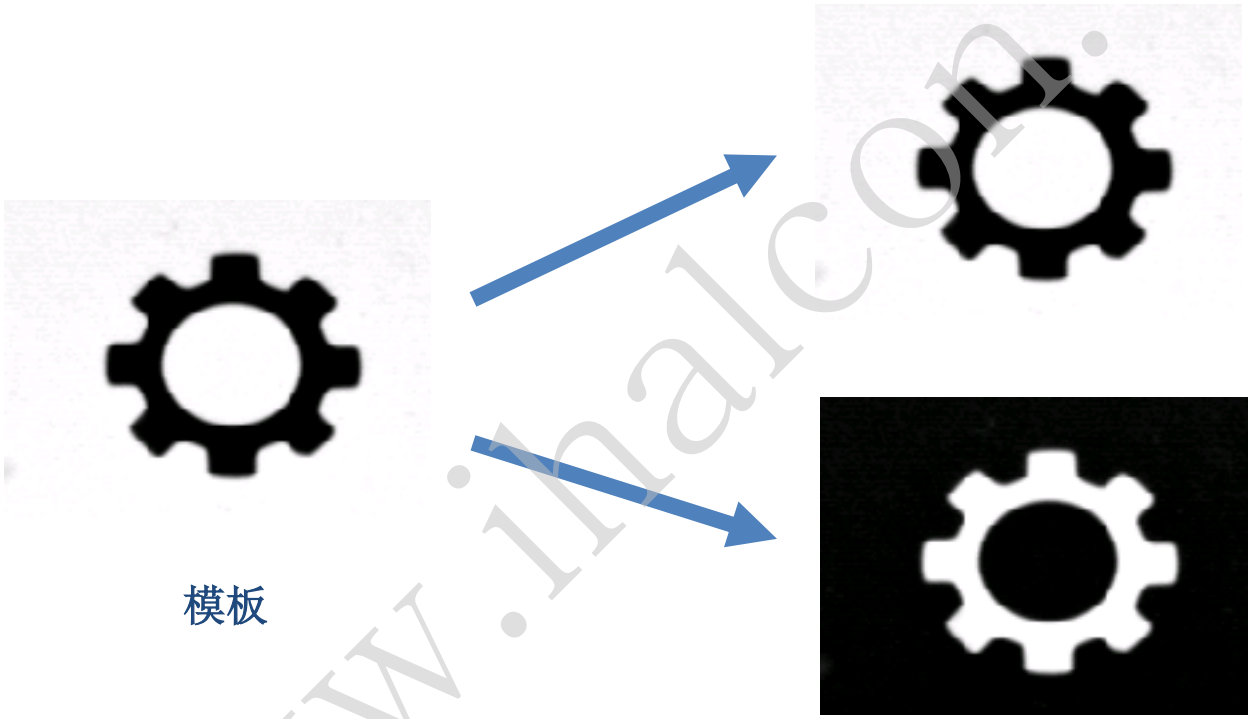


- 极性模式：use_polarity



模板参数-Metric

- 极性模式：ignore_global_polarity

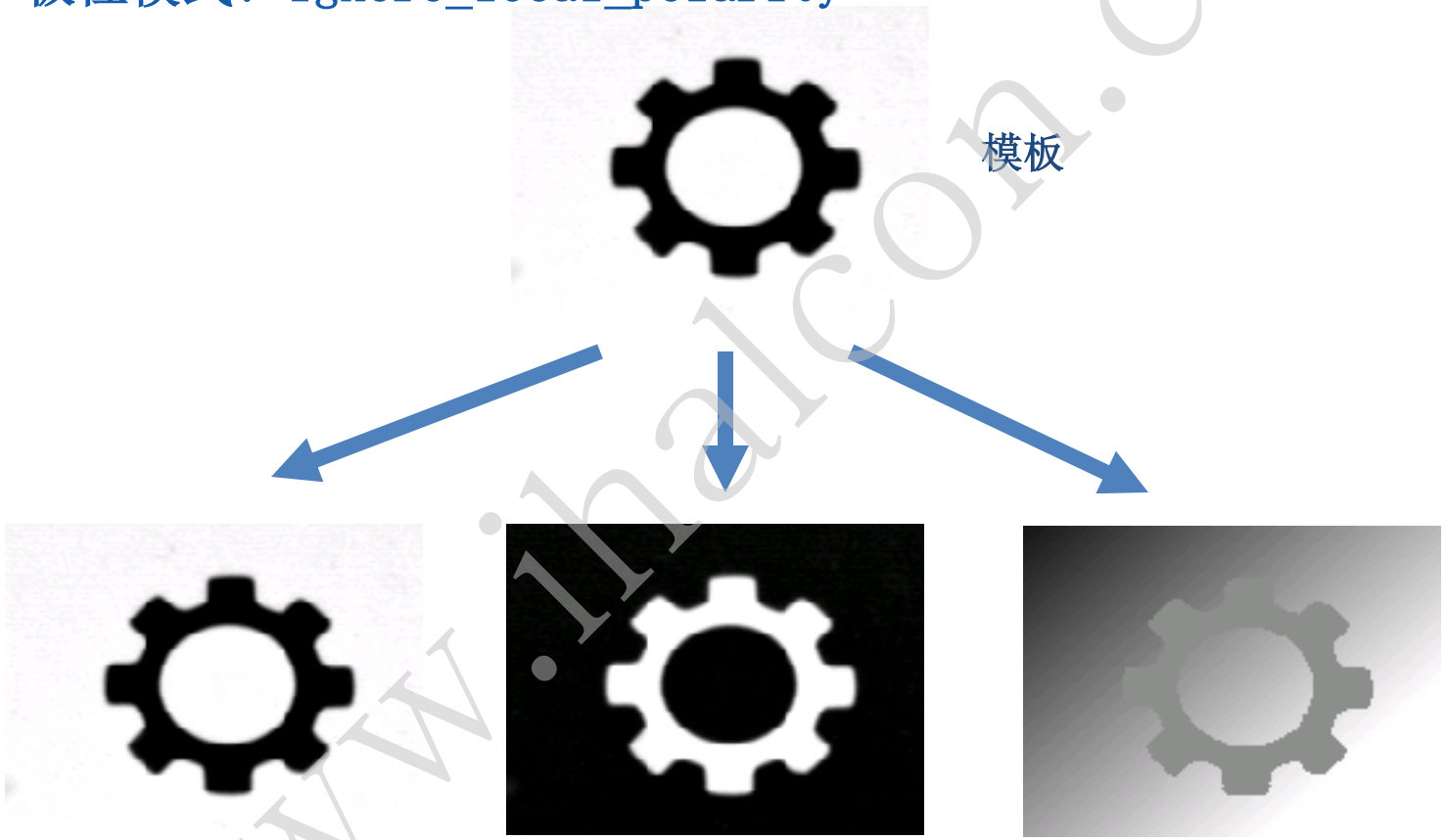


模板

目标

模板参数-Metric

- 极性模式：ignore_local_polarity



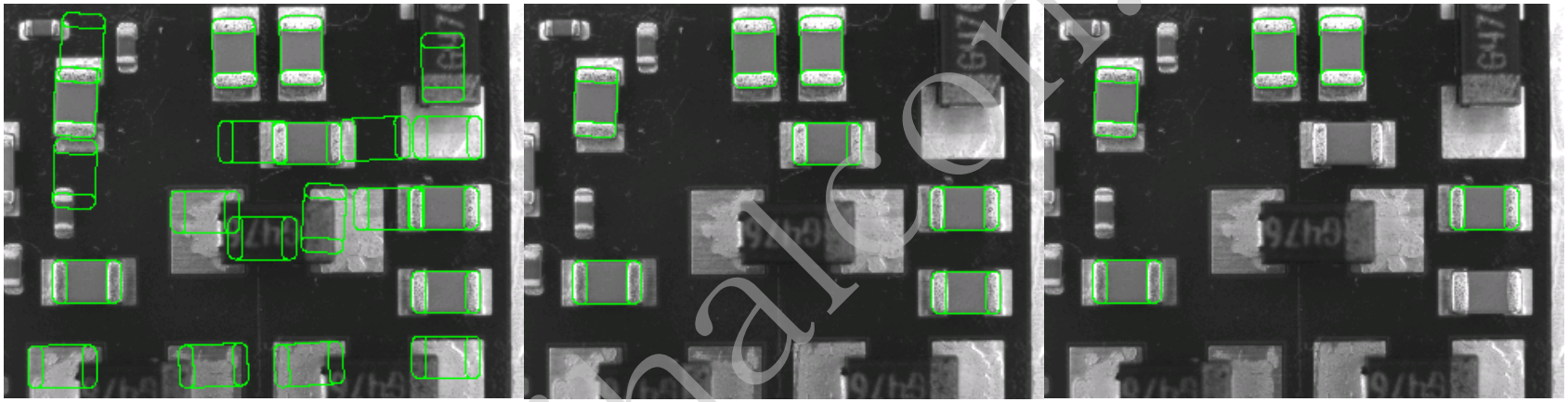
模板参数-MinContrast

- **MinContrast参数是被查找图片的最小对比度**
- **不是模板图片的所有边缘都是“有益”的，下面因素经常会产生“有害”边缘**
  - 噪声
  - 纹理
- **这些多余的便可导致**
  - 定位不准或找错
  - 错误的分值
  - 稍微增大查找时间
- **参数MinContrast是在查找模板的时候，来减少“有害”边缘的。它的值可通过下面方法得到**
  - estimate_noise函数
  - inspect_shape_model函数
  - 通过助手判断

模板参数-程序获取

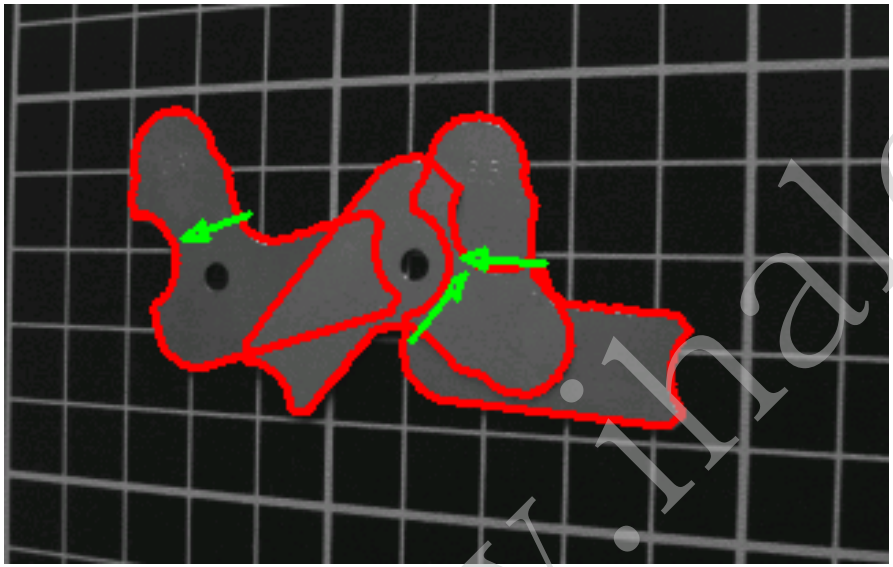
- determine_shape_model_params(Template, 模板
  'auto', 金字塔层数
  0, 起始角度
  rad(360), 角度范围
  0.9, 缩小范围
  1.1, 放大范围
  'auto', 减少像素的方法
  'use_polarity', 极性
  'auto', 对比度
  'auto', 最小对比度
  'all', Which values
  ParameterName, Name of values
  ParameterValue) Values

查找参数-Score



Too low	Optimal	Too high
Minimum score: 0.35	Minimum score: 0.6	Minimum score: 0.75
Smallest score: 0.41	Smallest score: 0.74	Smallest score: 0.87

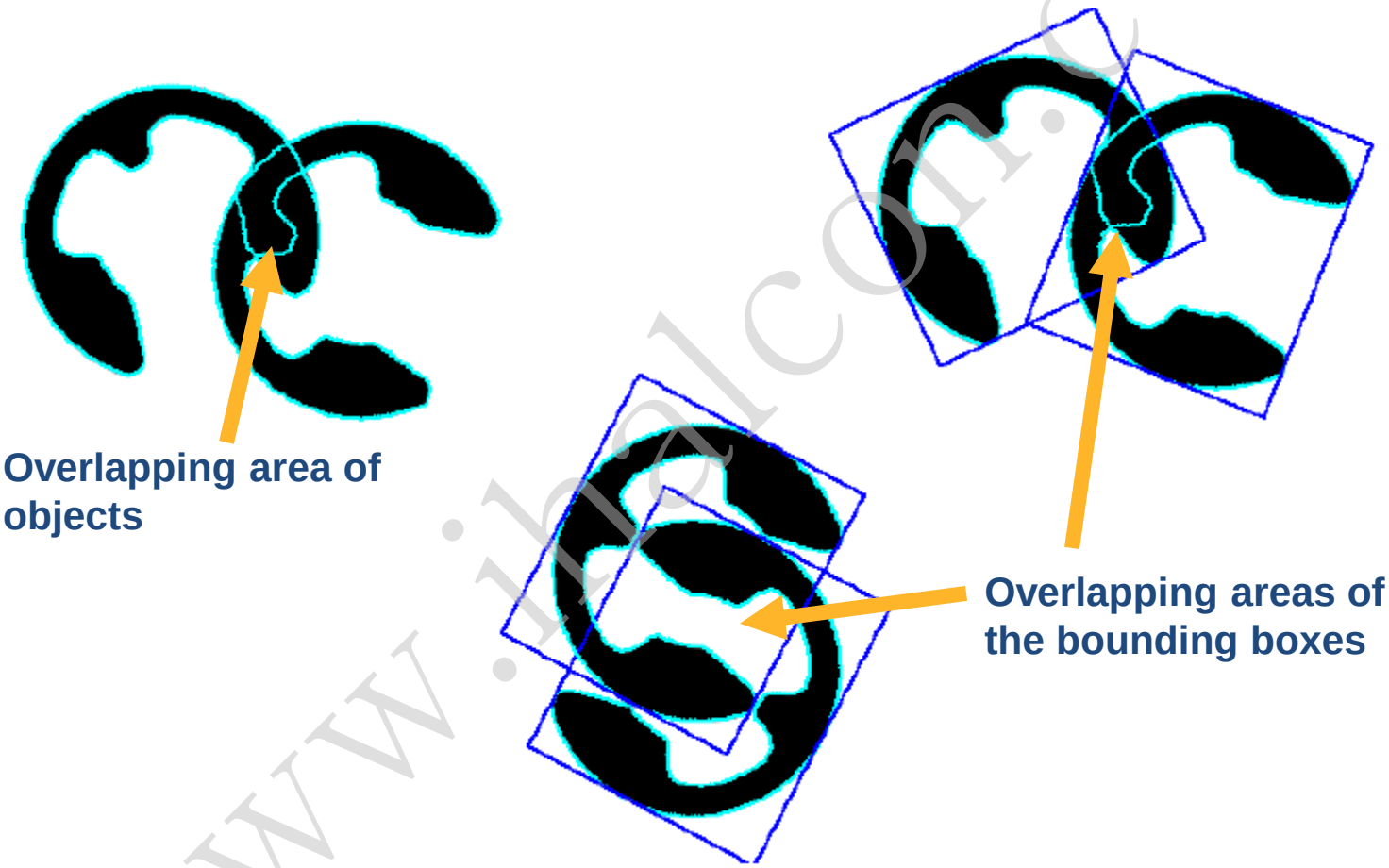
查找参数- Maximum Overlap



Edges of the model

Bounding box of the model

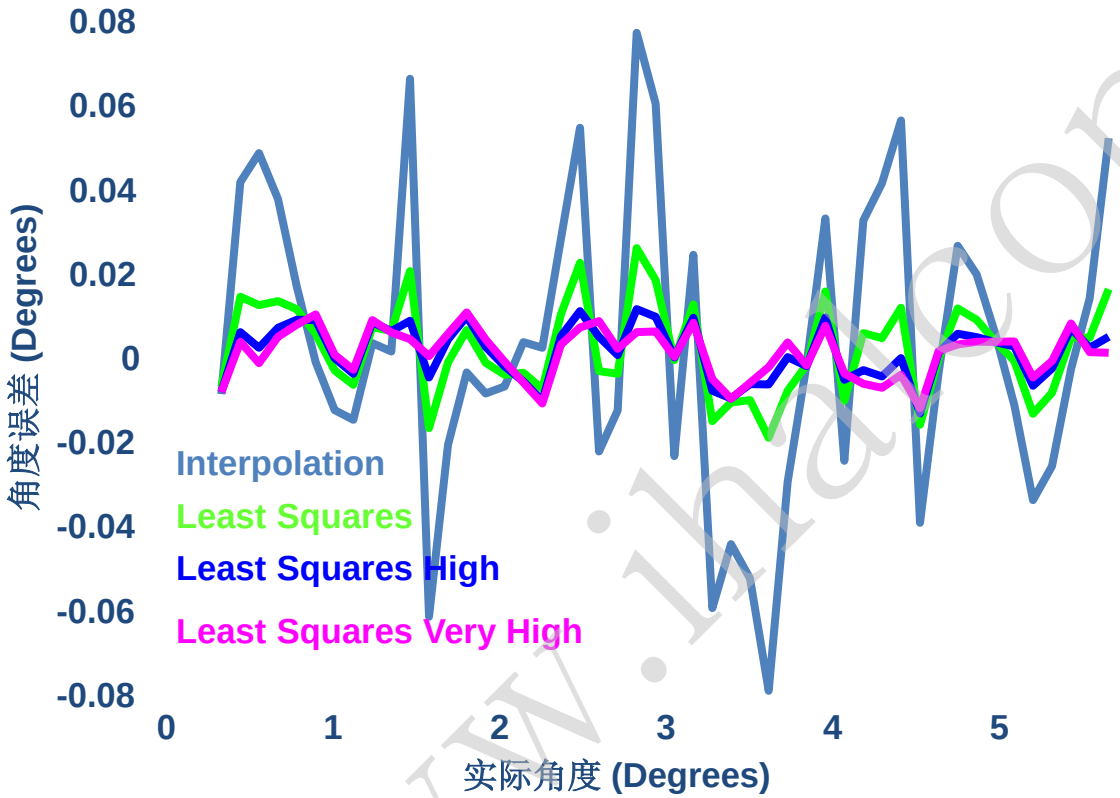
查找参数- Maximum Overlap



查找参数-SubPixel

- 目标位置的精度可以通过“sub pixel”来设置
- 精度控制模型
  - 'none': 不使用亚像素，最大误差为半个像素
  - 'interpolation': 差值的亚像素精度
  - 'least_squares','least_squares_high', 'least_squares_very_high':最小二乘法亚像素精度
- 不同模式对运行时间的影响
  - 例外: 'none' and 'interpolation' 时间相同
  - 最小二乘法时间比较长
- 该参数可影响以下结果
  - Position, Angle, Scaling

查找参数-SubPixel



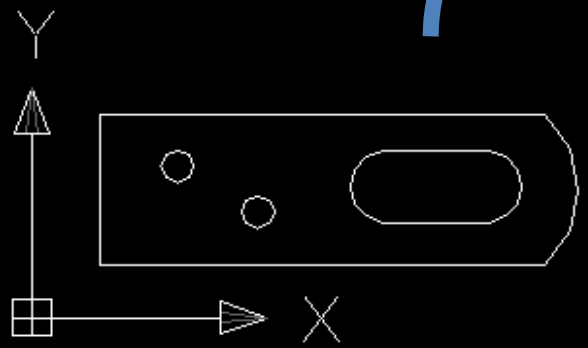
亚像素模型	运行时间	最大角度误差
Interpolation	100%	0.079°
Least Squares	120%	0.025°
Least Squares High	131%	0.014°
Least Squares Very High	142%	0.013°

查找参数-Greediness

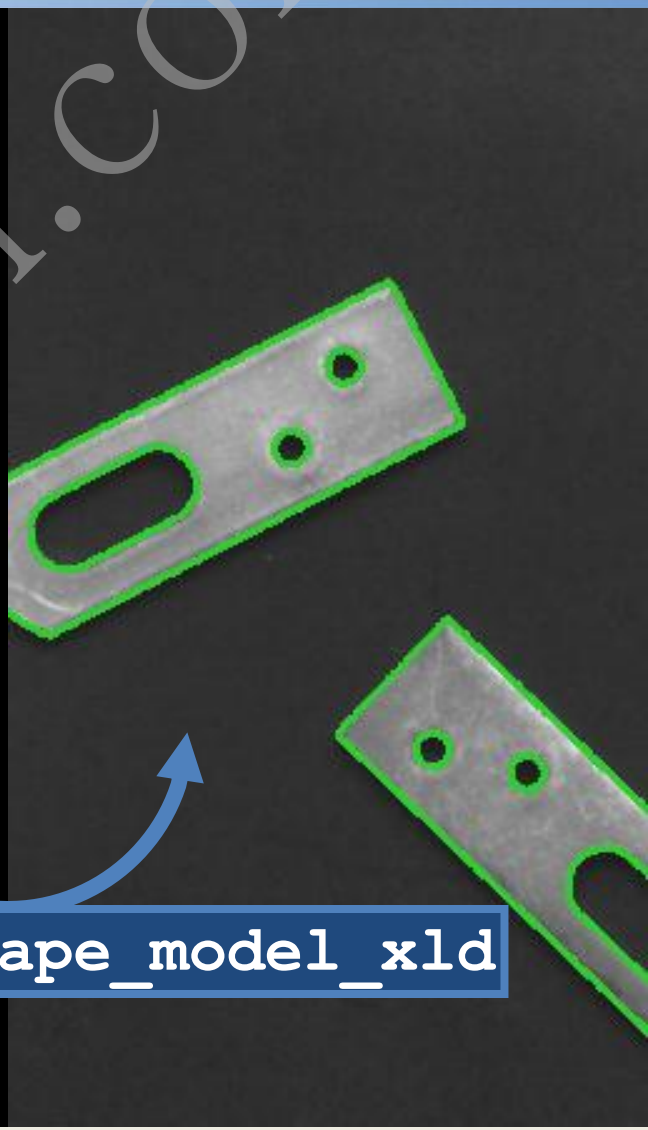
- 该参数是用来做定位加速的
- 值越小，速度越慢
- 值越高，找丢目标的可能越大
- 建议取值：0.7 — 0.9

通过Dxf文件创建模板

read_contour_xld_dxf



create_shape_model_xld

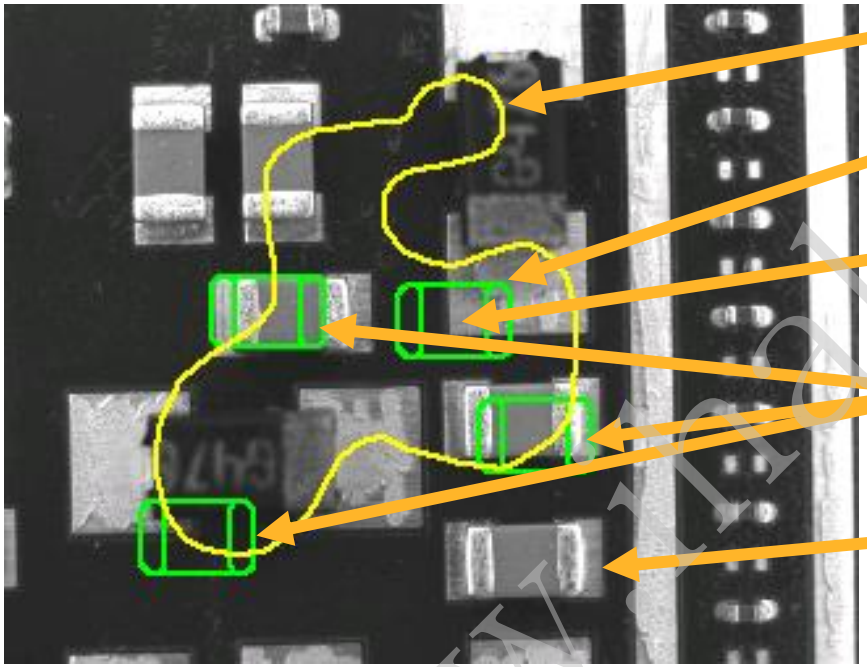


dxf文件　　轮廓　　形状模板

使用dxf创建函数

- 通过像素轮廓可以直接创建模板
  - create_shape_model_xld
  - create_scaled_shape_model_xld
  - create_aniso_shape_model_xld

查找ROI



Domain (search ROI)

Model

Default reference point

Locations still inside the domain

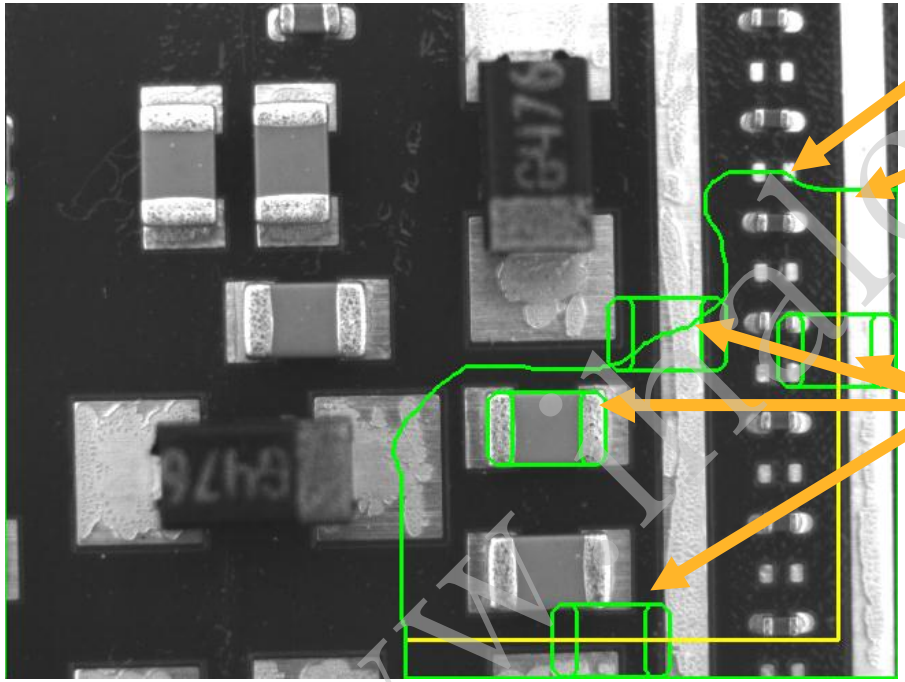
Outside the domain

边界处理

- HALCON提供了两种边界处理方法
- set_system('border_shape_models','false')
  - 模板必须在roi内
  - 靠近边缘部分会被裁减
- set_system('border_shape_models','true')
  - 模板可以部分在ROI外面
  - 注意：分值会降低

边界处理

set_system('border_shape_models','false')



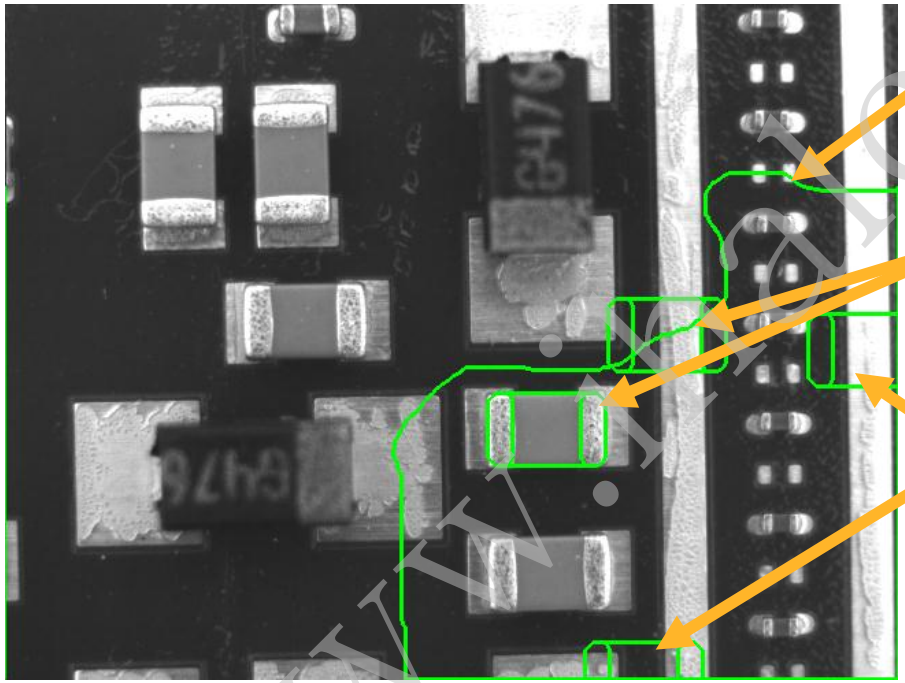
Domain (search ROI)

Clipped area near the border

Locations still inside the domain

边界处理

set_system('border_shape_models','true')



搜索区域

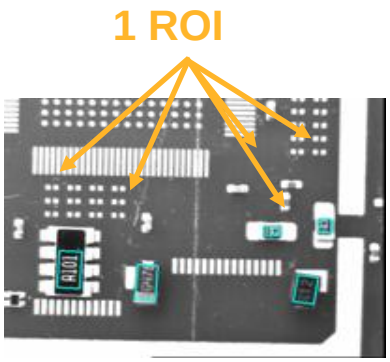
物体完全在ROI内
(最大分值100%)

跨边界

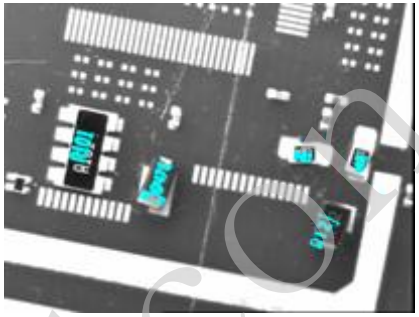
ROI (Maximum Score <100%)

基于组件的匹配

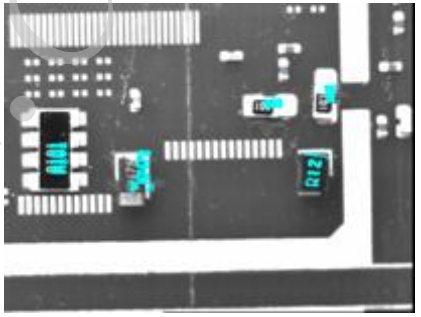
适合情况



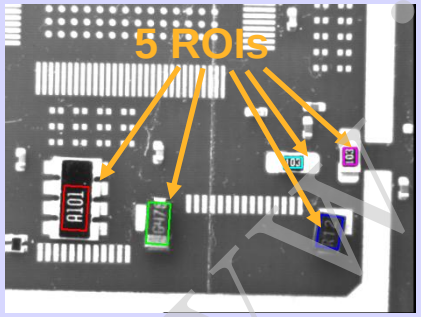
1 shape model



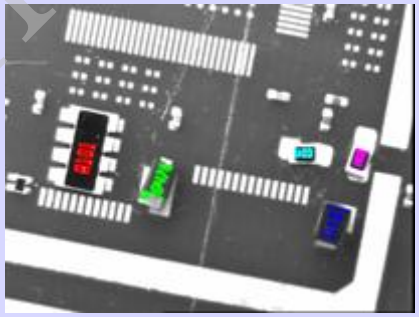
Score = 0.4



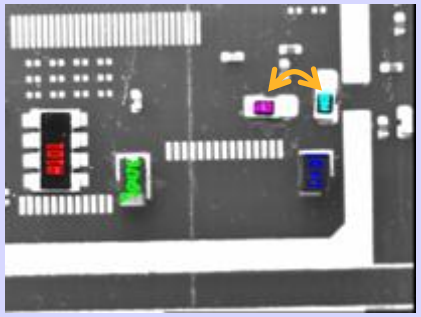
Score = 0.3



5 shape models



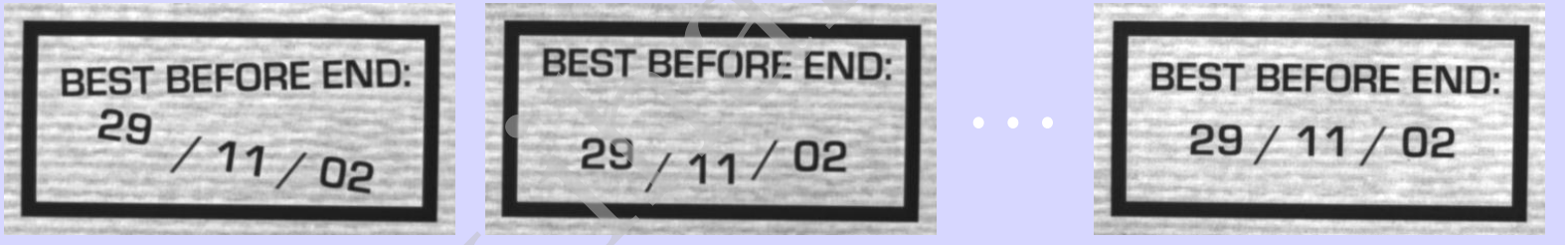
速度慢

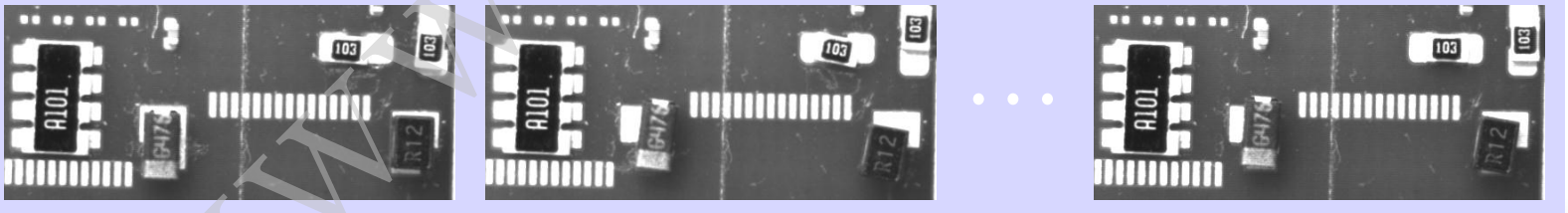


错配

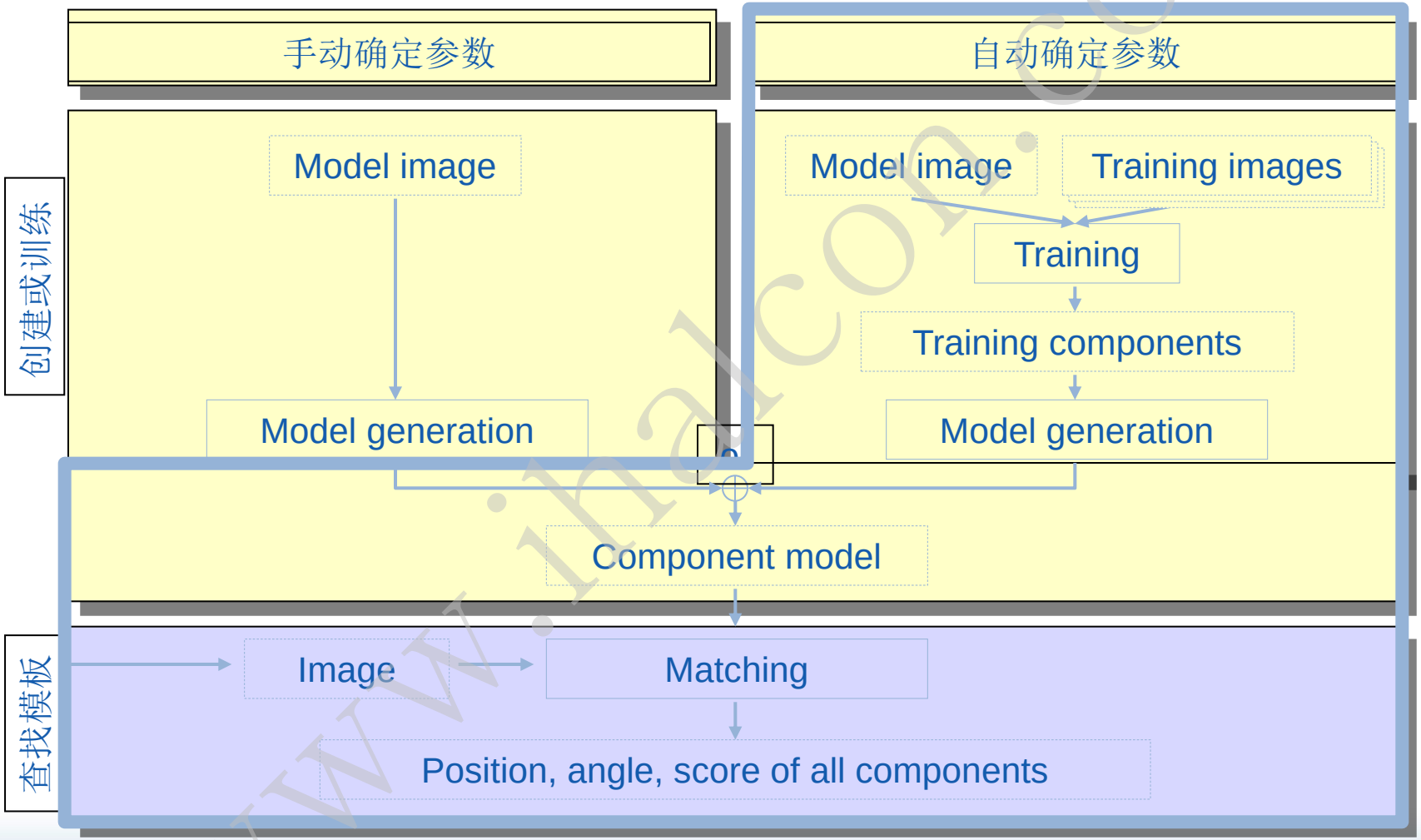
基于组件的匹配

- 组合物体要包含几个刚性组件
- 组件之间存在一定的位置关系





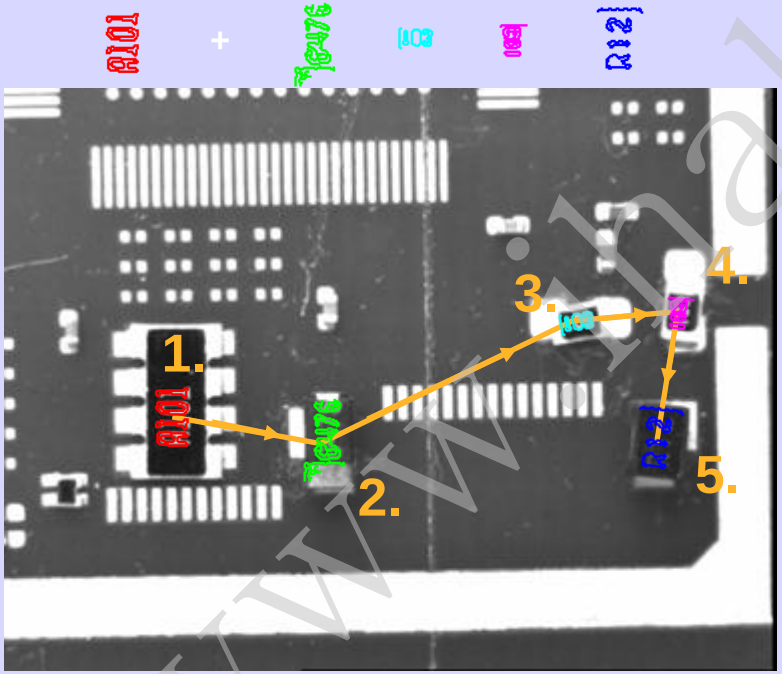
流程



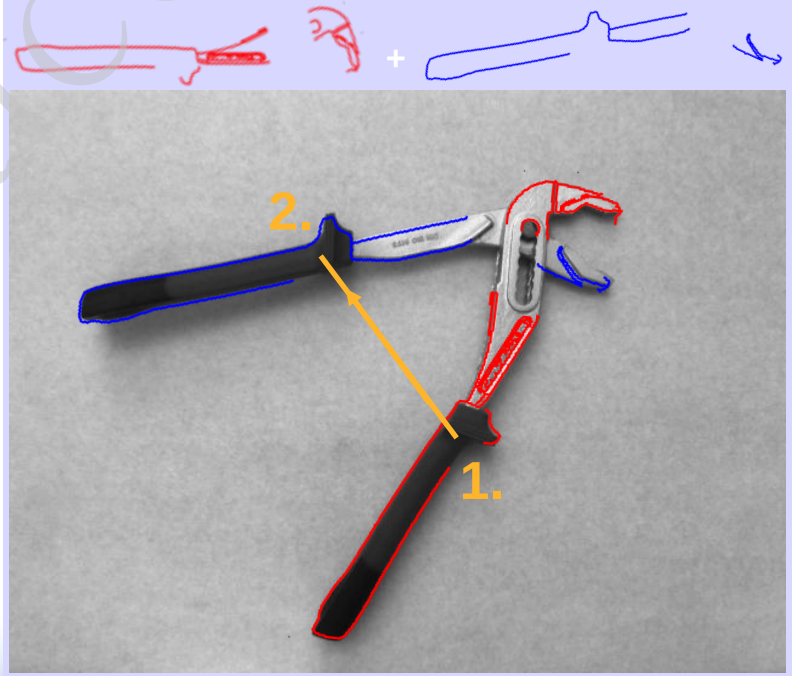
基于组件的匹配

- 基于组件的匹配是形状匹配的扩展算法
- 只有一个组件会在整个ROI区域搜索
- 其余组件会根据组件之间的关联关系去小范围搜索

Component model consists of 5 shape models:



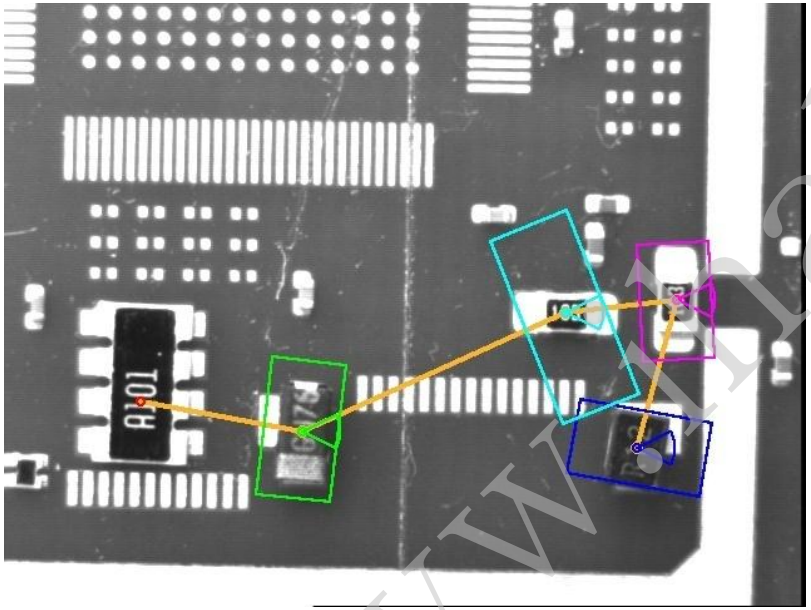
Component model consists of 2 shape models:



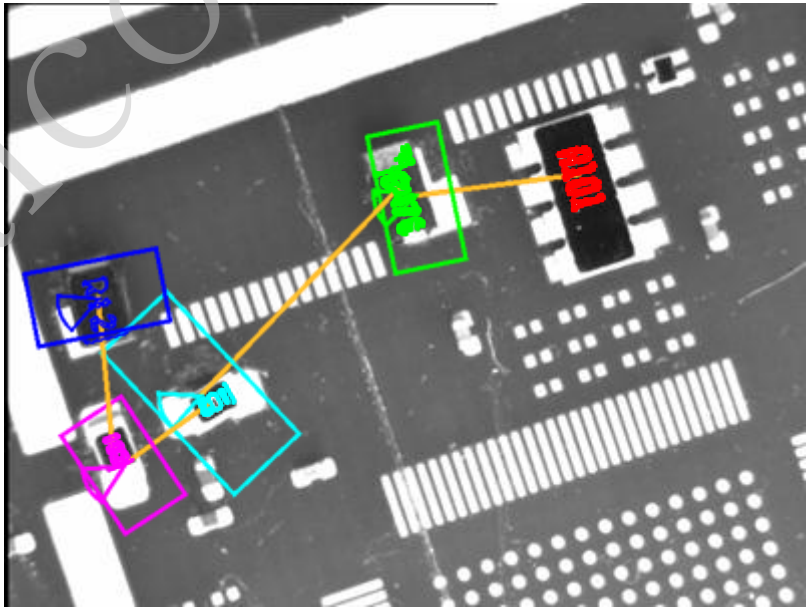
组件间的关系

- 全图像内搜寻 root component
- 根据root component的位置确定其它组件位置

Component model

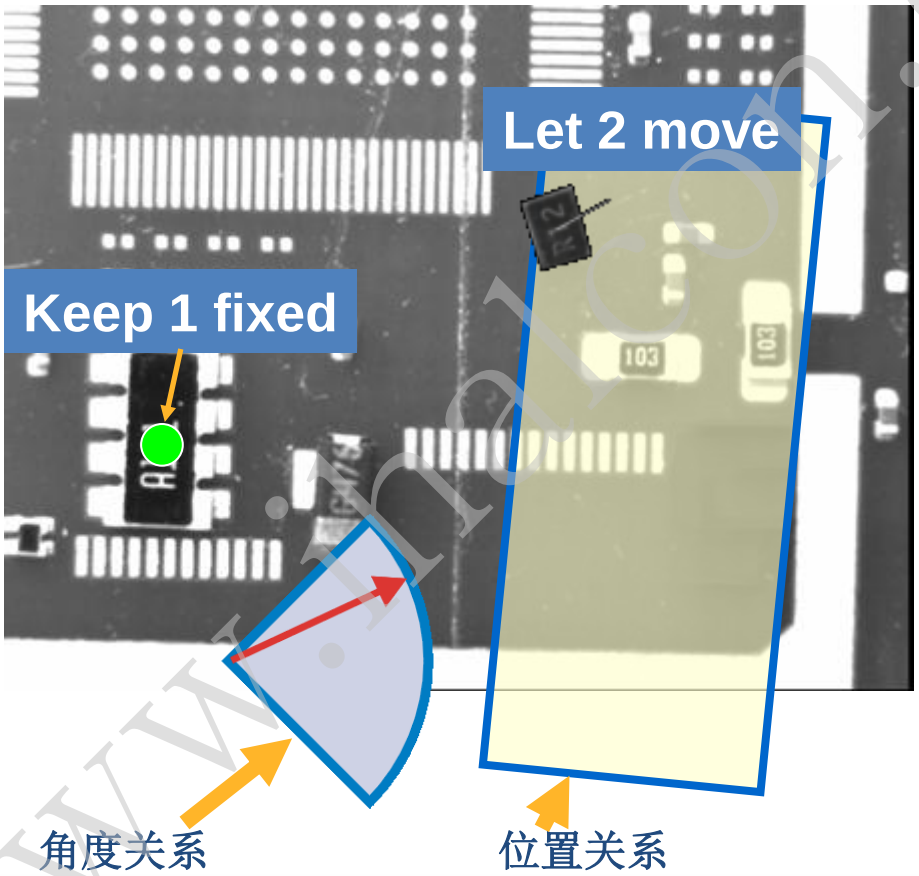


Search image



组件间的关系

组件1和组件2之间的关系

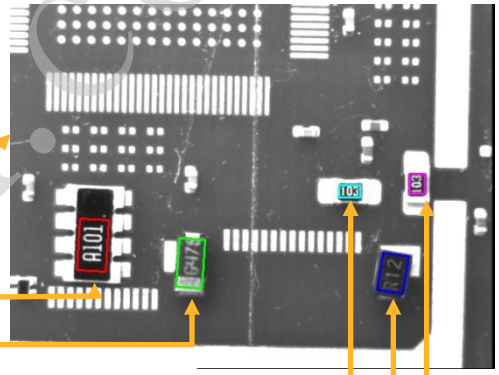


主要函数

- 四个主要函数
  - 训练函数
    - train_model_components
  - 创建组件模板函数
    - create_component_model
    - create_trained_component_model
  - 搜索组件模板函数
    - find_component_model

创建模板

- 确定组件区域
- 创建组件模板




```
read_image(ModelImage,'modules/modules_model')

gen_rectangle2(ComponentRegions,318,109,-1.62,34,19)
gen_rectangle2(Rectangle2,342,238,-1.63,32,17)
gen_rectangle2(Rectangle3,355,505,1.41,25,17)
gen_rectangle2(Rectangle4 247,448,0,14,8)
gen_rectangle2(Rectangle5,237,537,-1.57,13,10)
ComponentRegions := [ComponentRegions,Rectangle2]
ComponentRegions := [ComponentRegions,Rectangle3]
ComponentRegions := [ComponentRegions,Rectangle4]
ComponentRegions := [ComponentRegions,Rectangle5]

create_component_model(ModelImage,ComponentRegions,20,20,rad(25),0,
                       rad(360),15,40,15,10,0.8,[4,3,3,3,3],0,'none',
                       'use_polarity','true',ComponentModelID,RootRanking)

write_component_model(ComponentModelID,Filename)
```

查找模板

- 查找组件模板

```
read_component_model(Filename,ComponentModelID)
while(true)
  grab_image(SearchImage,FGHandle)
  find_component_model(SearchImage,ComponentModelID,RootRanking[0],0,
                       rad(360),0.5,0,0.5,'stop_search','search_from_best',
                       'none',0.8,'interpolation',0,0.8,ModelStart,
                       ModelEnd,Score,RowComp,ColumnComp,AngleComp,
                       ScoreComp,ModelComp)
  dev_display(SearchImage)
  for Match := 0 to |ModelStart|-1 by 1
    get_found_component_model(FoundComponents,ComponentModelID,ModelStart,
                              ModelEnd,RowComp,ColumnComp,AngleComp,
                              ScoreComp,ModelComp,Match,'false',
                              RowCompInst,ColumnCompInst,AngleCompInst,
                              ScoreCompInst)
    dev_display(FoundComponents)
  endfor
endwhile
clear_component_model(ComponentModelID)
```

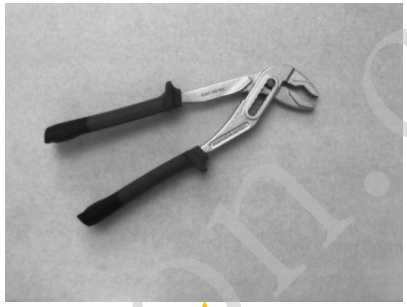
寻找模板

找root model 以加快速度

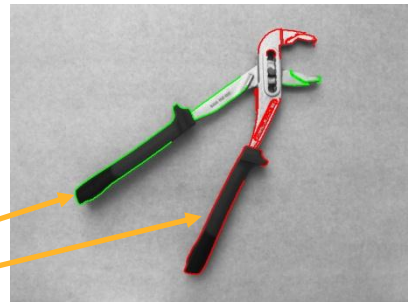
可视化

训练模板






```
TrainingImages := []
for i := 1 to 4 by 1
  read_image(TrainingImage,'pipe_wrench_training_'+i)
  TrainingImages := [TrainingImages,TrainingImage]
endfor
```



```
train_model_components(ModelImage,InitialComponentRegions,TrainingImages,
                       ModelComponents,22,60,30,0.65,-1,1,rad(60),'speed',
                       'rigidity',0.2,0.4,ComponentTrainingID)

write_training_components(ComponentTrainingID,FileNameTraining)
```

训练模板

- create_component_model
  - 手动
  - 用户自己确定位置关系
- create_trained_component_model
  - 自动
  - 通过函数train_model_components确定相关位置关系

```
read_training_components(FileNameTraining,ComponentTrainingID)
```

训练结果 (包含相互位置关系)

```
create_trained_component_model(ComponentTrainingID,-rad(30),rad(60),10,
                               0.55,4,0,'none','use_polarity','false',
                               ComponentModelID,RootRanking)

write_component_model(ComponentModelID,FileName)
clear_training_components(ComponentTrainingID)
```

- 与手动方式一样的匹配函数：find_component_model

train_model_components

```
train_model_components(
    ModelImage,                 // 模板图片
    InitialComponents,          // 根组件
    TrainingImages,             // 训练图片
    ModelComponents,            // 组件模板
    ContrastLow, ContrastHigh,      // 同形状模板
    MinSize, MinScore,              // 同形状模板
    SearchRowTol, SearchColumnTol, // 位置变化范围
    SearchAngleTol,                 // 角度变化范围
    TrainingEmphasis,           // 速度或鲁棒性优先
    AmbiguityCriterion,         // 模糊匹配规则
    MaxContourOverlap,          // 轮廓重合
    ClusterThreshold,           // 根组件归类阈值
    ComponentTrainingID         // 模板ID
    )
```

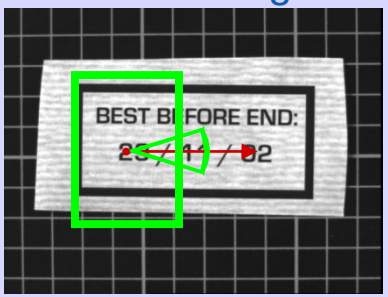
训练模板参数- TrainingImages

- 参数TrainingImages 包含组件运动关系的一系列图像
- 没有相对运动关系的组件合成一个刚性组件
- 训练图像的数量取决于组件运动的复杂性
- 每一个训练图像至少含有一个组件

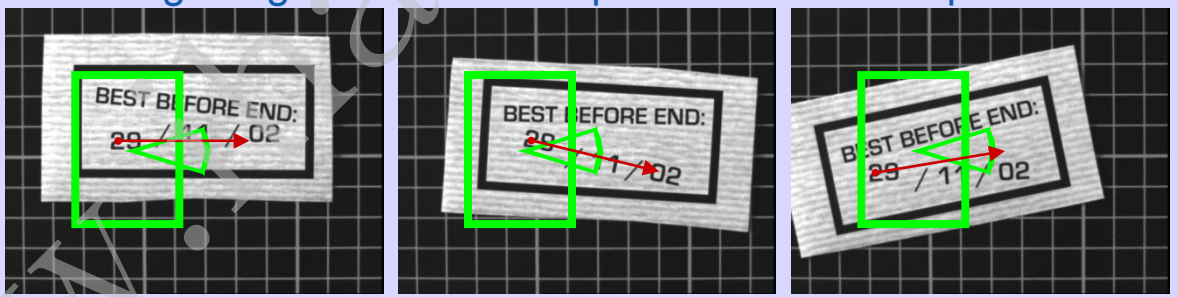
训练模板参数-范围

- 参数SearchRowTol，SearchColumnTol，和 SearchAngleTol 描述了在row，column和angle上的变化范围。
- 相对于模板图像中的初始组件而言
- 默认值 -1，表示无限制

Model image



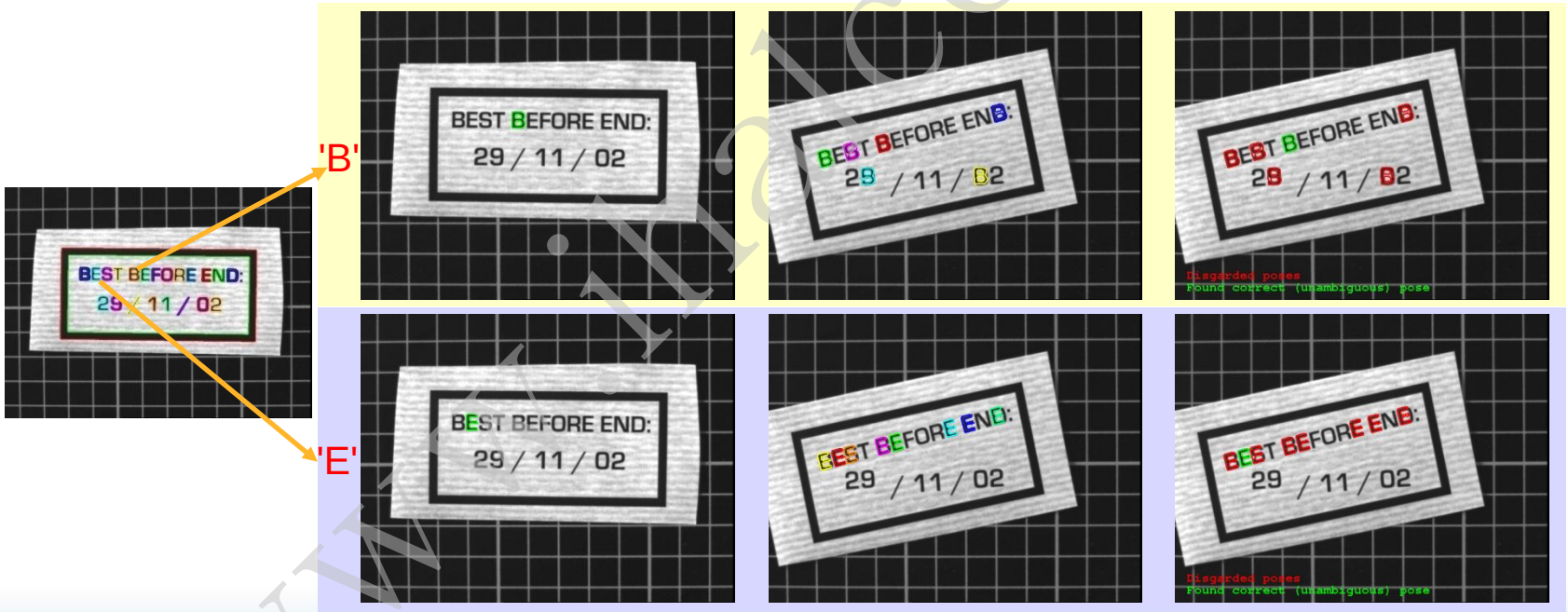
Training images and search space for initial component “2”



e.g., SearchRowTol = 120, SearchColumnTol = 80, SearchAngleTol = 20°

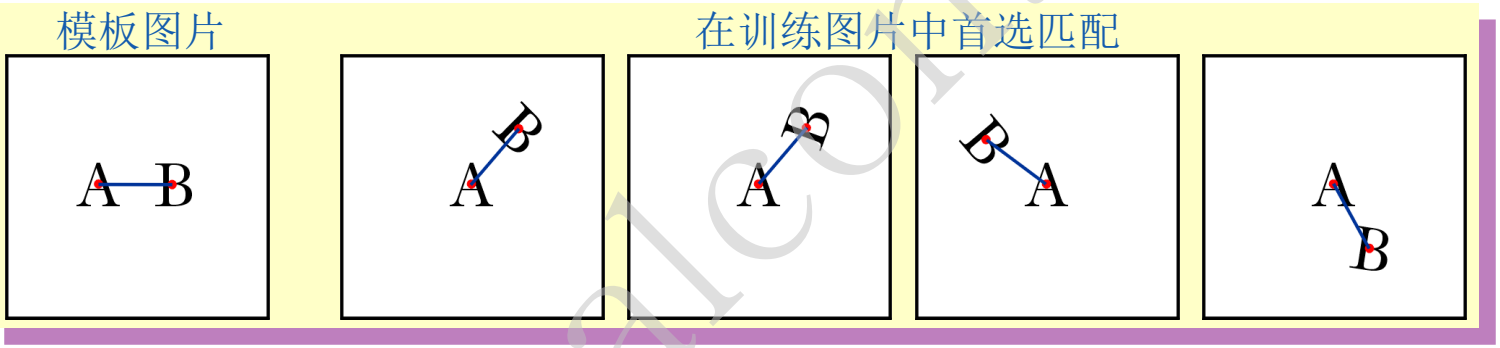
训练模板参数-模糊匹配

- 匹配模糊情况
  - 对称的初始组件
  - 相似的初始组件
- 模糊匹配会影响组件间的相互关系

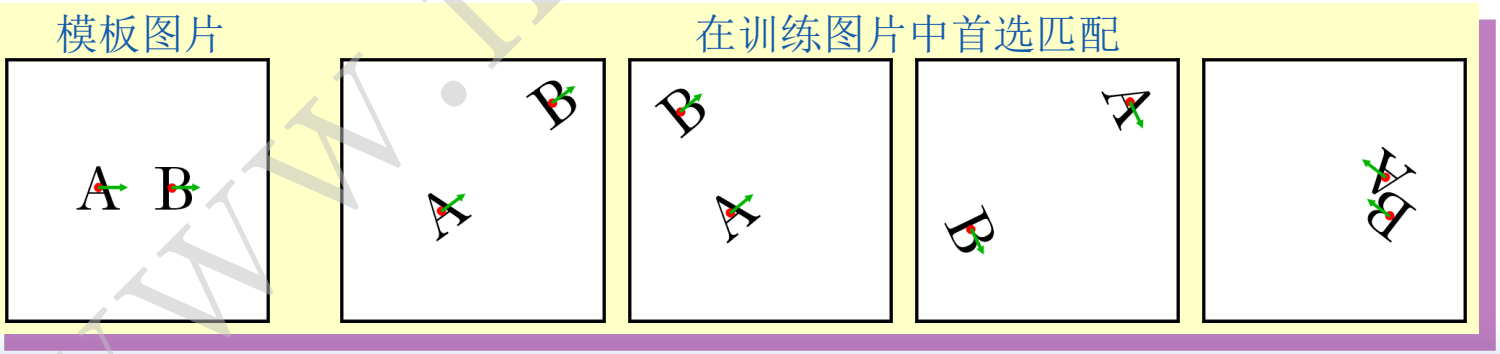


训练模板参数-模糊匹配

- 参数AmbiguityCriterion 提供了四种方法来解决模糊匹配问题
  - 'distance'

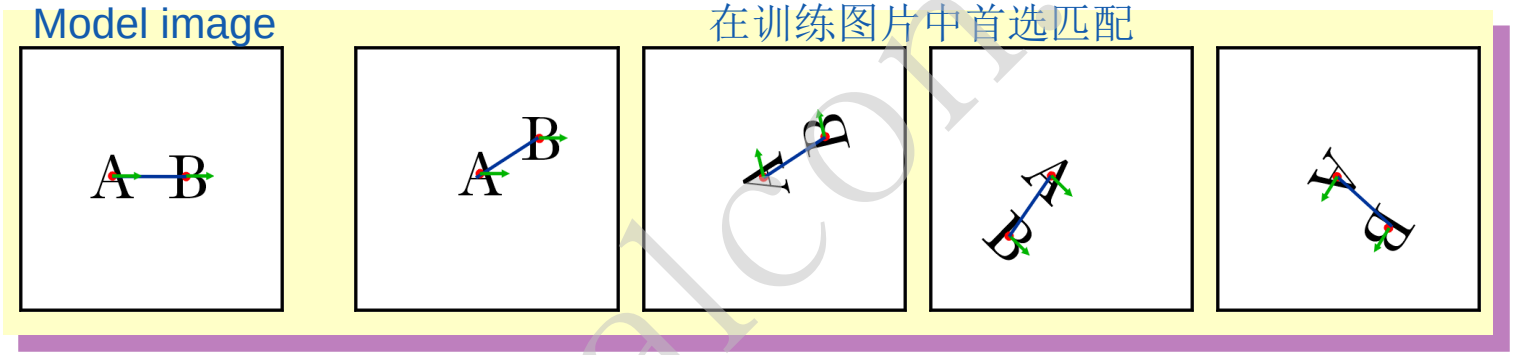


  - 'orientation'

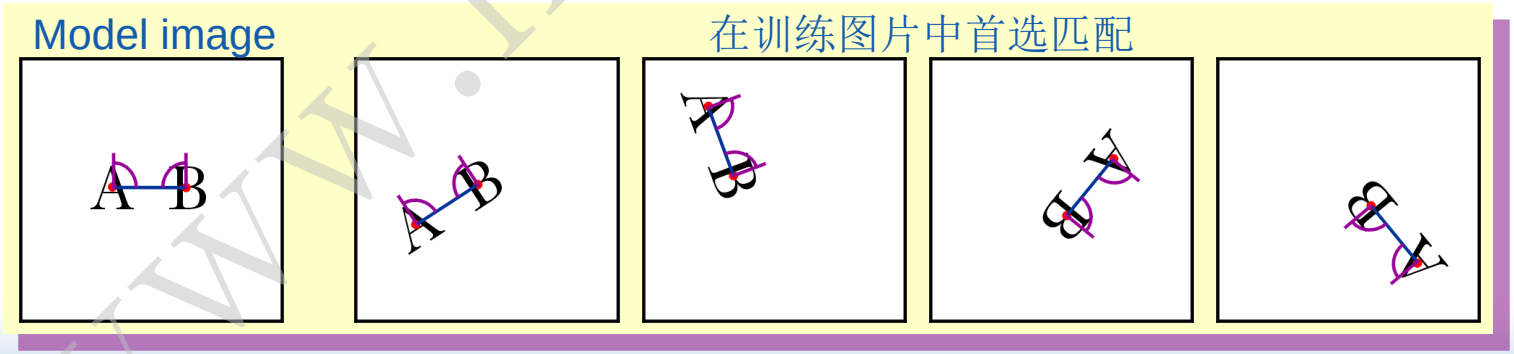


训练模板参数-模糊匹配

- 参数AmbiguityCriterion 提供了四种方法来解决模糊匹配问题
  - 'distance_orientation'



  - 'rigidity'（大多数情况下用此参数）



训练模板参数-最大轮廓交叠

- 通过设值参数MaxContourOverlap来减少错误的匹配
- 交叠:两个轮廓区域交叠的比例

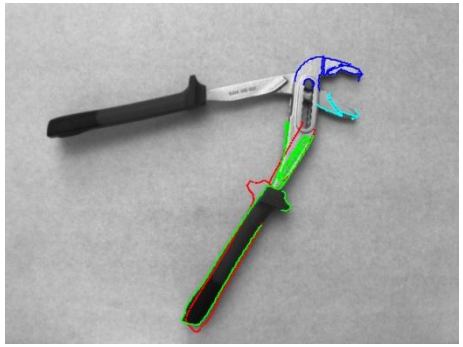
Model image

MaxContourOverlap = 1

MinScore = 0.4

MaxContourOverlap = 0.2 (default)

Training image (solved ambiguities)



create_trained_component_model

- create_trained_component_model(
  ComponentTrainingID,
  AngleStart, AngleExtent,
  MinContrastComp,MinScoreComp,NumLevelsComp,
  AngleStepComp,
  OptimizationComp, MetricComp, PregenerationComp,
  ComponentModelID,
  RootRanking
)

find_component_model

- 图像中找到模板
- 组件模板必须提前创建完成
  - **create_component_model** or **create_trained_component_model**
- 参数
  - **ComponentModelID**: Handle of the component model
  - **RootComponent**: Index of the root component that should be used
  - **AngleStartRoot**: Smallest rotation of the root component
  - **AngleExtentRoot**: Extent of the rotation of the root component
  - **MinScore**: Minimum score of the instances of the component model to be found
  - **NumMatches**: Number of instances of the component model to be found
  - **MaxOverlap**: Maximum overlap of component model instances
  - **IfRootNotFound**: Behavior if the root component is missing
  - **IfComponentNotFound**: Behavior if a component is missing

find_component_model

参数

- **PosePrediction**: Pose prediction of components that are not found
- **MinScoreComp**, **SubPixelComp**, **NumLevelsComp**, **GreedinessComp**: Shape-model-specific (see **find_shape_model**)
- **ModelStart**: Start index of each component model instance in the tuples describing the component matches
- **ModelEnd**: End index of each component model instance in the tuples describing the component matches
- **Score**: Score of the found instances of the component model
- **RowComp**: Row coordinate of the found component matches
- **ColumnComp**: Column coordinate of the found component matches
- **AngleComp**: Rotation angle of the found component matches
- **ScoreComp**: Score of the found component matches
- **ModelComp**: Index of the found components

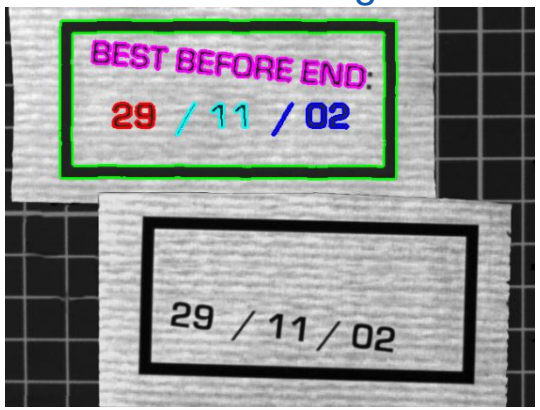
模板匹配 - 丢失根组件

- 首先需要找到根组件，然后寻找其它组件
- 有些情况根组件丢失

Model image

5 model components

Search image



Root component

Search tree

1 instance not found

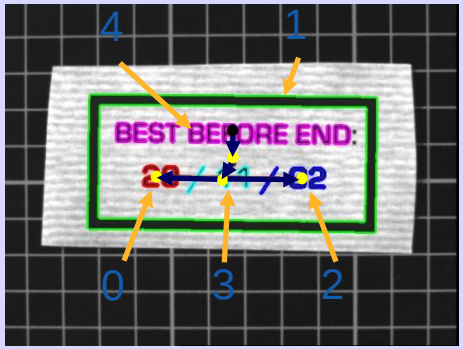
模板匹配 - 丢失根组件

- 通过设置参数IfRootNotFound解决根组件丢失的情况，可设参数
  - 'stop_search' (default)
  - 'select_new_root'
- 选取 'select_new_root' 情况，重复四个步骤
  1. 搜寻根组件
  2. 找到根组件的情况下，继续寻找其它的相关组件
  3. 找不到根组件的情况下选取下一个组件做为根组件
  4. 回到步骤 1

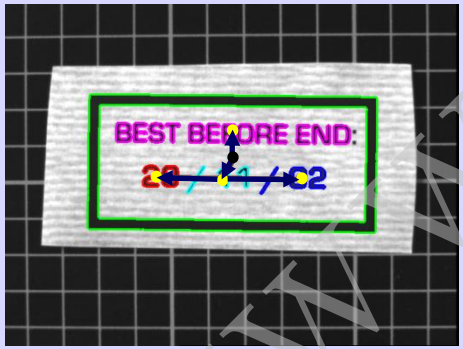
模板匹配 - 丢失根组件

- 使用RootRanking = [4, 1, 3, 0, 2]

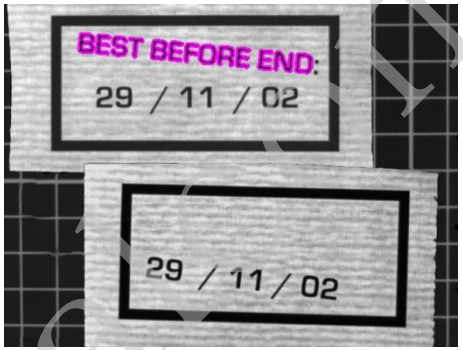
Search tree of component 4



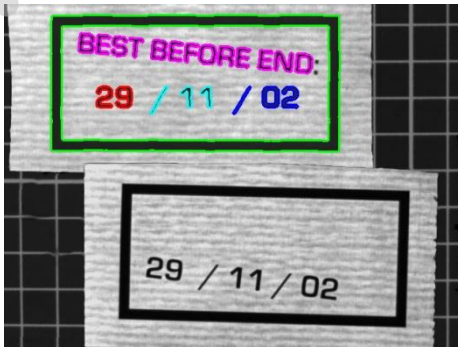
Search tree of component 1



Step 1: Search root component 4



Step 2: Start relative search



Step 3: Search root component 1

Step 4: Start relative search

模板匹配 - 丢失中间组件

- 丢失中间组件时如何找到其余组件

Model components and search tree

Search image with one component missing

→ The relative search for "29" and "/ 02" could not be started because the pose of "/ 11" is not known

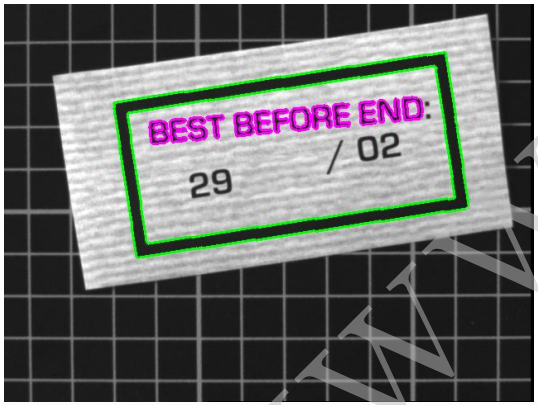
模板匹配 - 丢失中间组件

- 通过设置参数IfComponentNotFound来动态适应搜索顺序
- 三个可选参数
  - 'prune_branch' (default)
    - No adaptation of the search tree is performed
    - Such components are not searched at all
  - 'search_from_upper'
    - Such components are searched relative to the pose of the predecessor of the (missing) predecessor in the search tree
  - 'search_from_best'
    - Such components are searched relative to the pose of the already found component from which the relative search is fastest

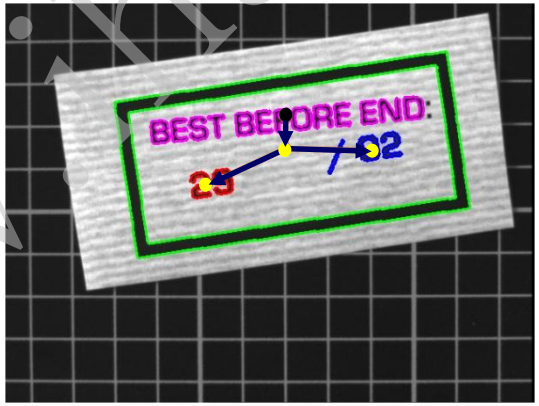
模板匹配 - 丢失中间组件

初始的搜索顺序

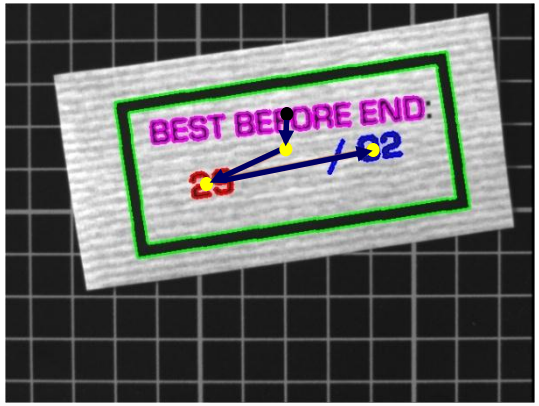
'prune_branch'



'search_from_upper'



'search_from_best'



基于灰度的匹配

互相关匹配

- 模板与图像中最原始的灰度值进行匹配。
- 相似计算
  - 模板与图像之间差值的绝对值，受光照变化明显

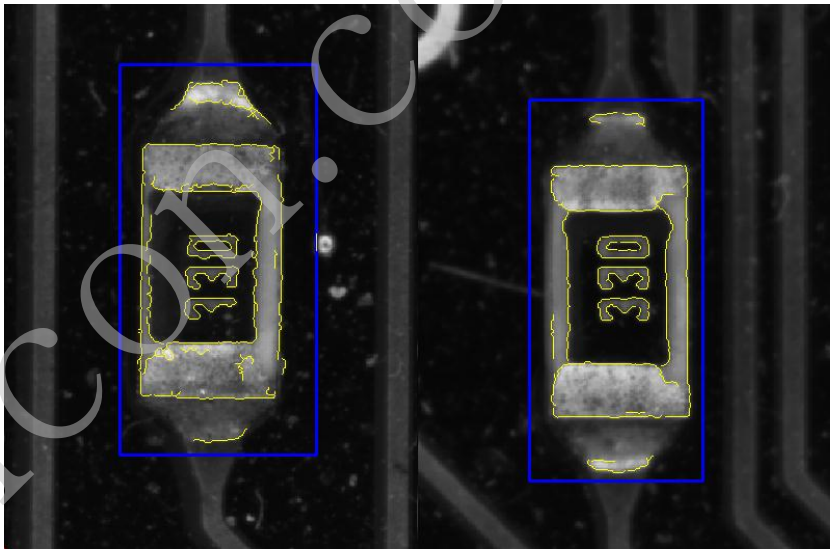
$$a(x,y)=\frac{1}{n}\sum_{(u,v)\in R}\left|t(u,v)-f(x+u,y+v)\right| \qquad n=|R|$$

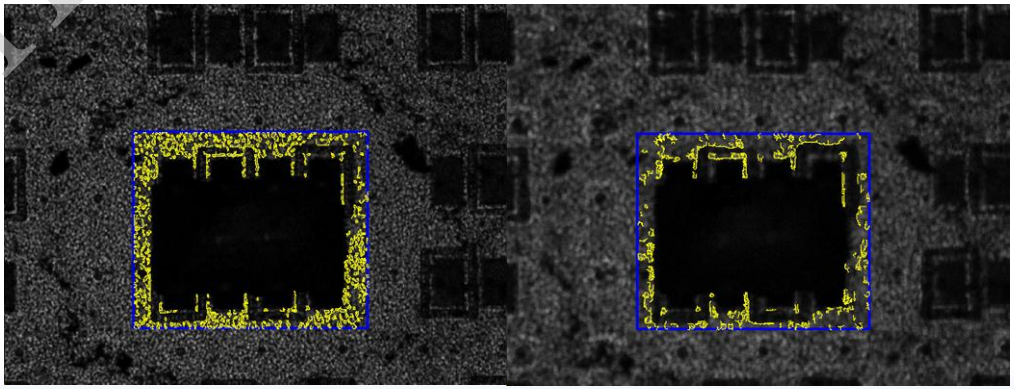
  - 归一化互相关系数(NCC)，受光照变化不明显

$$c(x,y)=\frac{1}{n}\sum_{(u,v)\in R}\frac{t(u,v)-m_t}{\sqrt{s_t^2}}\cdot\frac{f(x+u,y+v)-m_f(x,y)}{\sqrt{s_f^2(x,y)}}$$

NCC(Normalized Cross Correlation Matching)匹配

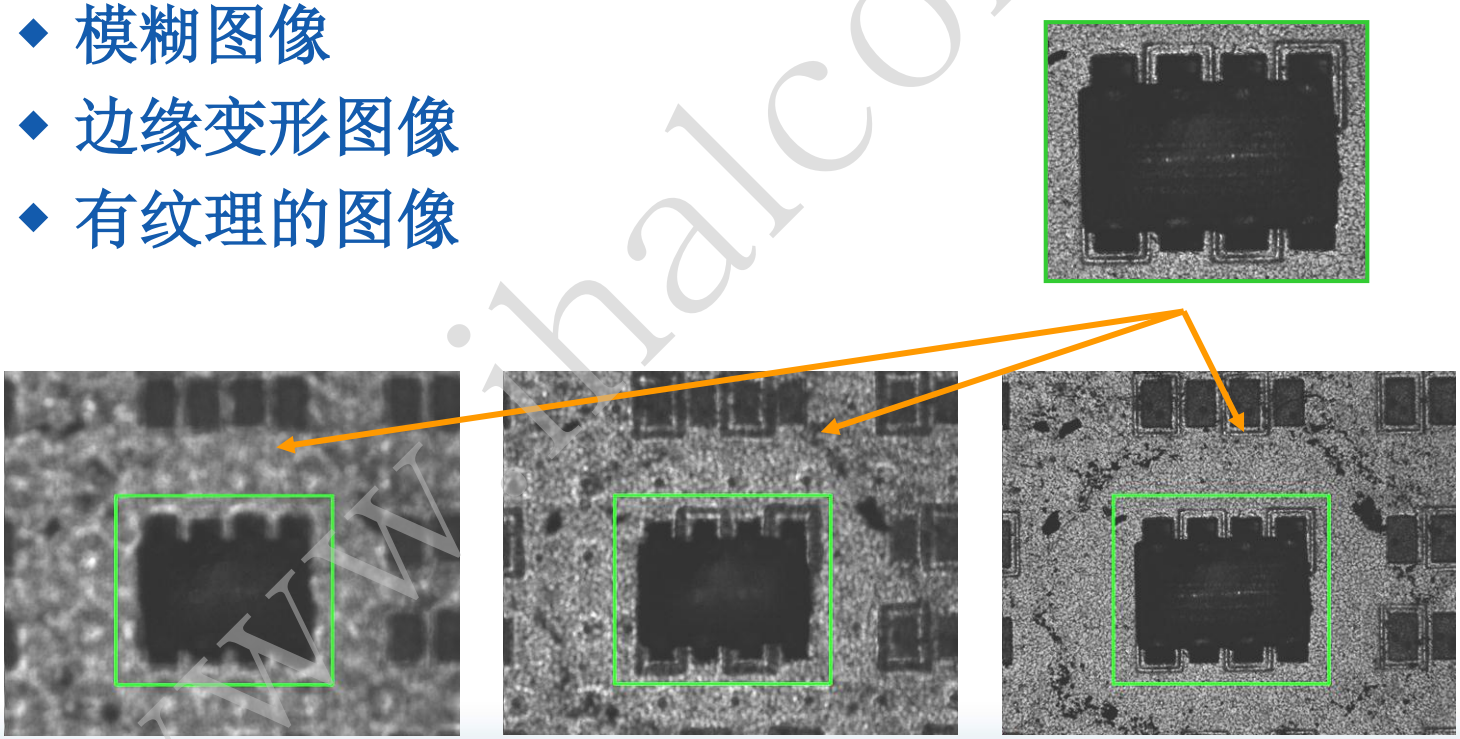
- 对于如下情况，使用形状匹配比较困难，而NCC可以解决
  - 物体有轻微变形
  - 图像模糊、边缘不清的图片
  - 图片有纹理





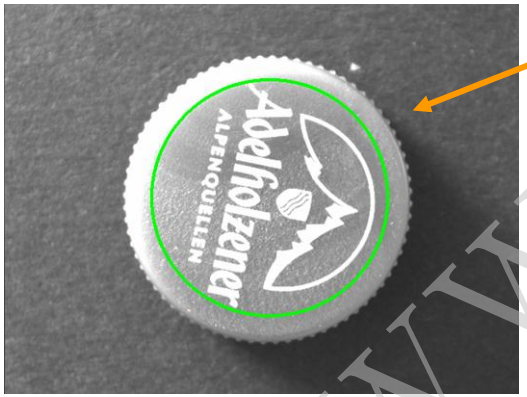
NCC匹配介绍

- 特点
  - 快速、基于灰度的匹配
  - 鲁棒性
  - 模糊图像
  - 边缘变形图像
  - 有纹理的图像



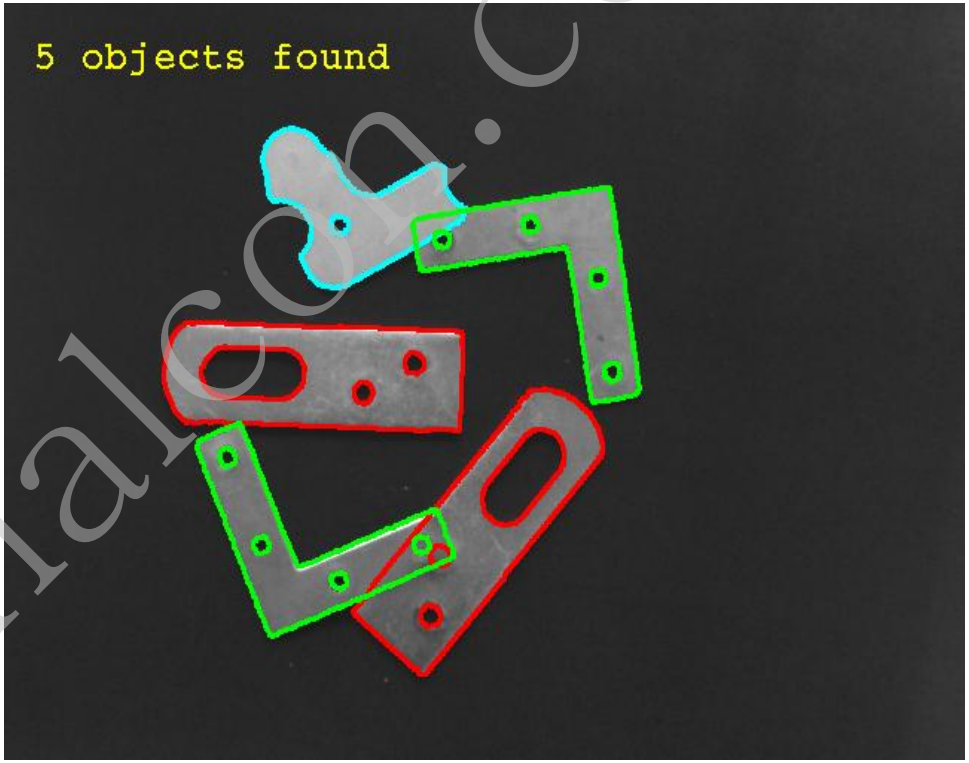
NCC匹配介绍

- NCC匹配支持光照变化的情况。



NCC与形状匹配的比较

- 两种方法的参数类似
- NCC优点
  - 纹理
  - 对焦不清
  - 形状轻微变形
- 形状匹配优点
  - 精度高
  - 支持X/Y方向缩放
  - 支持物体遮挡
  - 支持多模板
  - 支持非线性光照变化



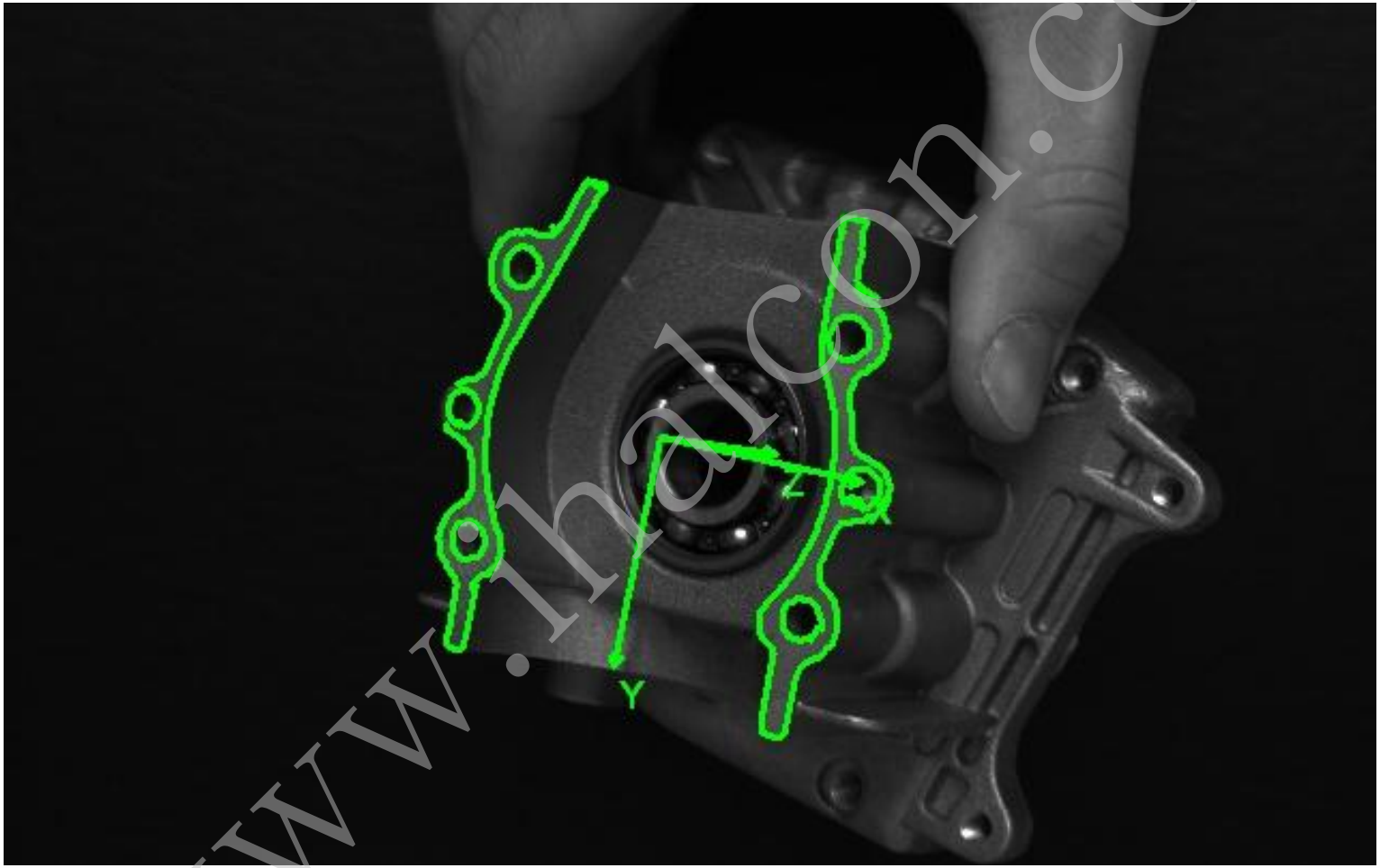
形状匹配的多模板匹配

NCC匹配相关函数

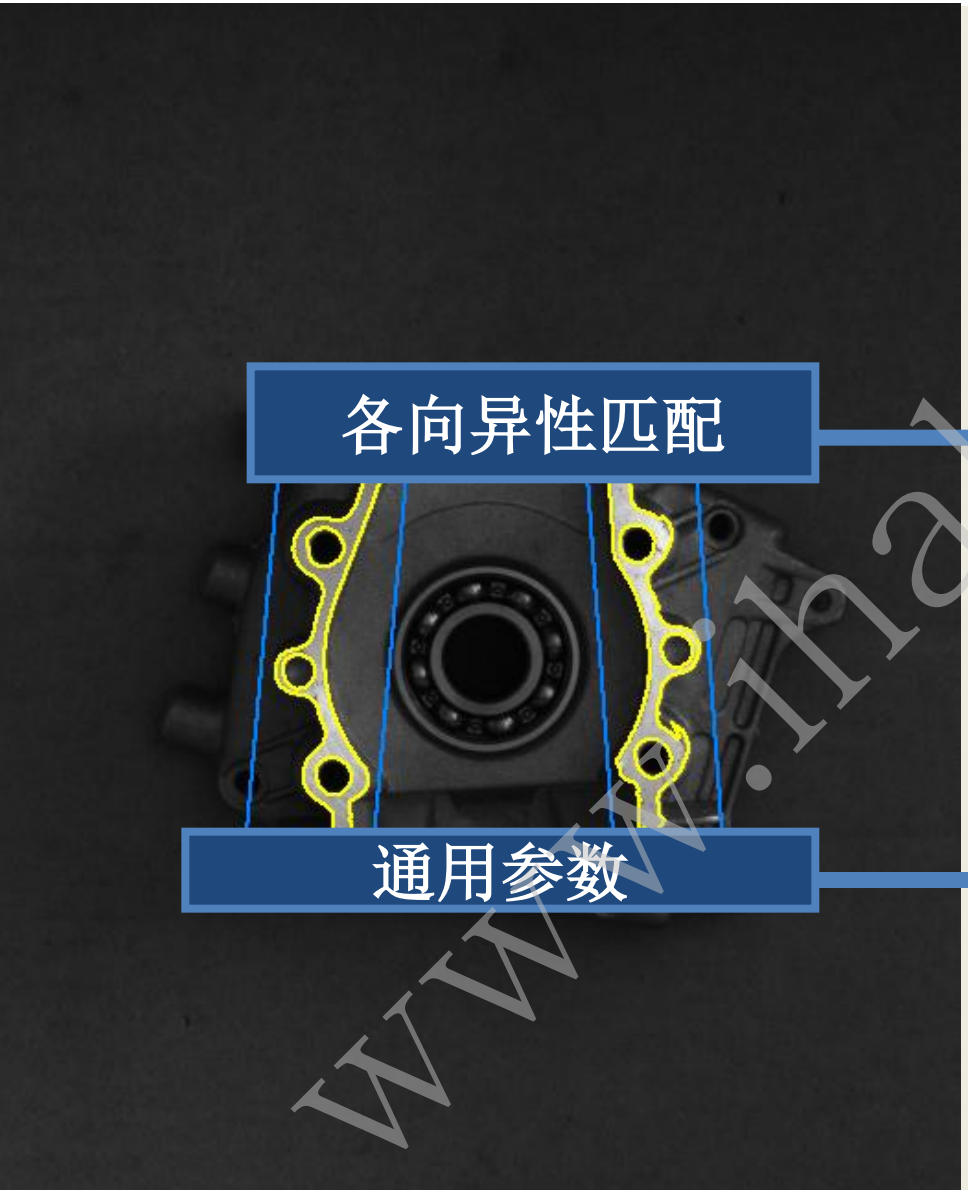
- 创建
  - create_ncc_model
- 查找
  - find_ncc_model
- 读写
  - read_ncc_model
  - write_ncc_model
- 内存清除
  - clear_ncc_model
  - clear_all_ncc_models
- 其他
  - get_ncc_model_params
  - get_ncc_model_origin
  - set_ncc_model_origin
  - determine_ncc_model_params

变形模板

变形匹配



创建平面变形模板

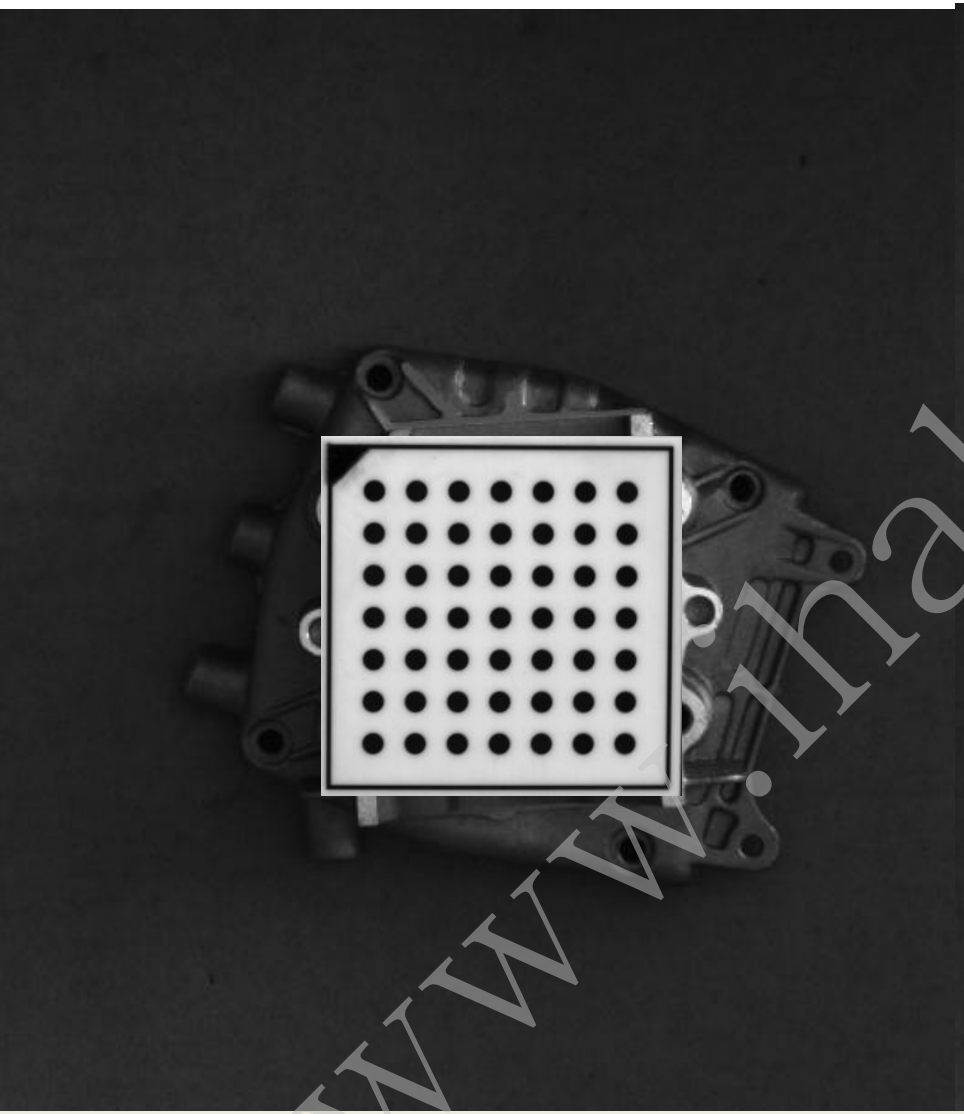



```
create_planar_uncalib
  _deformable_model(
  ▸ Template::
  ▸ NumLevels,
  ▸ AngleStart,
  AngleExtent,
  AngleStep,
  ▸ ScaleRMin, ▸
  ScaleRMax, ▸
  ScaleRStep,
  ▸ ScaleCMin, ▸
  ScaleCMax, ▸
  ScaleCStep,
  ▸ Optimization,
  ▸ Metric,
```

创建平面变形模板


```
create_planar_calib_d
  eformable_model(
  ▸ Template::
  ▸ CamParam,
  ▸ ReferencePose,
  ▸ NumLevels,
  ▸ AngleStart,
  AngleExtent,
  AngleStep,
  ▸ ScaleRMin, ▸
  ScaleRMax, ▸
  ScaleRStep,
  ▸ ScaleCMin, ▸
  ScaleCMax, ▸
  ScaleCStep,
```

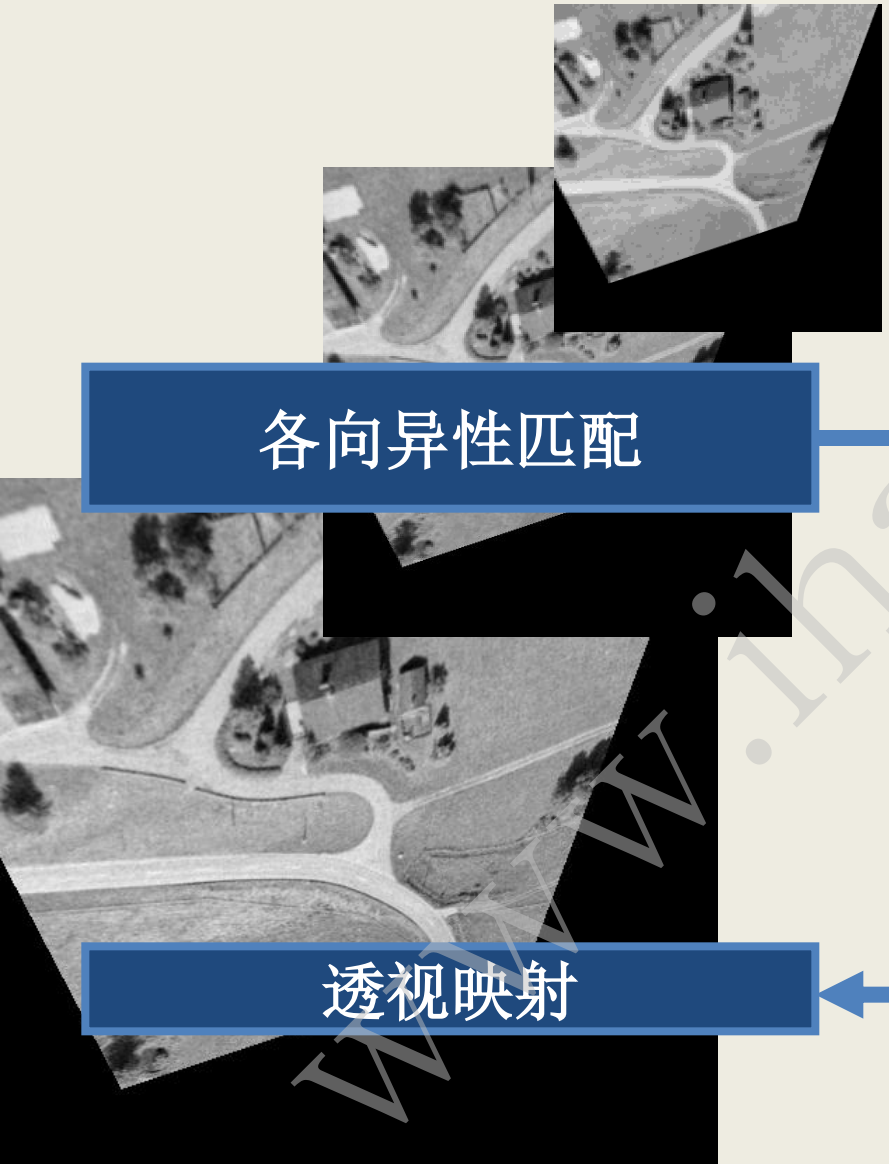
计算模板位姿



a) 使用标定板

b) 使用物体特征

找到变形模板

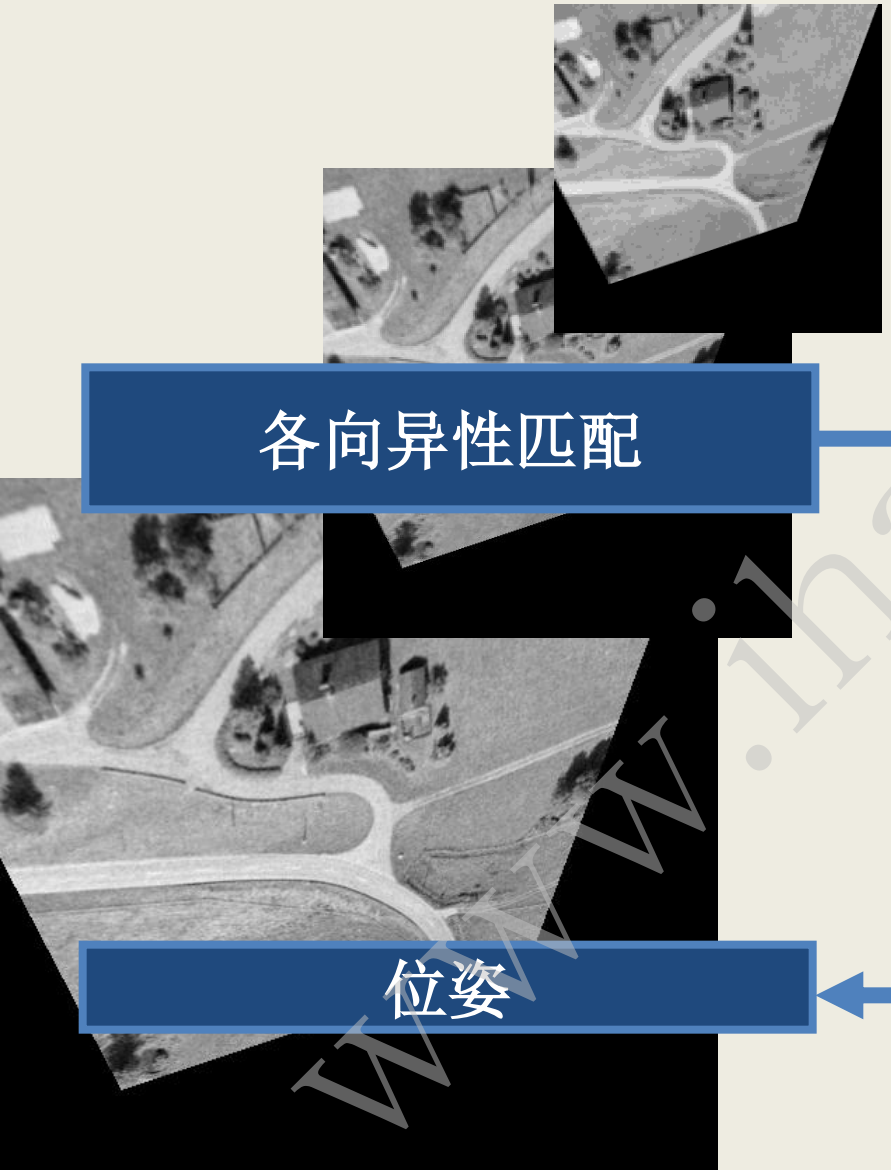


各向异性匹配

透视映射

```
find_planar_uncalib_def
  ormable_model(
  ▸ Image::
  ▸ ModelID,
  ▸ AngleStart,
  AngleExtent,
  AngleStep,
  ▸ ScaleRMin, ▸
  ScaleRMax, ▸
  ScaleRStep,
  ▸ ScaleCMin, ▸
  ScaleCMax, ▸
  ScaleCStep,
  ▸ MinScore,
  ▸ NumMatches
```

找到变形模板



各向异性匹配

位姿

```
find_planar_calib_def
  ormable_model(
  ▸ Image::
  ▸ ModelID,
  ▸ AngleStart,
  AngleExtent,
  AngleStep,
  ▸ ScaleRMin, ▸
  ScaleRMax, ▸
  ScaleRStep,
  ▸ ScaleCMin, ▸
  ScaleCMax, ▸
  ScaleCStep,
  ▸ MinScore,
  ▸ NumMatches
```

谢谢