# 1 解决思路

1. 计算标定板中两个相邻圆点，圆心距dx，单位是像素跑pixel，和实际物理尺寸2mm对应起来
2. 计算每个轮廓的周长x\_i
3. 一个比值，就可以计算实际轮廓长度

# 2 代码实现

1. 计算圆心距

在计算标定板圆心距.txt中，用halcon软件实现。实现效果如下

圆心距是在58.9037左右

dev\_update\_off ()

\*读取图片

read\_image (Image, 'Pic2\_1.bmp')

\*

\* 关闭窗口

dev\_close\_window ()

get\_image\_size (Image, Width, Height)

dev\_open\_window (0, 0, Width, Height, 'black', WindowHandle)

\*

\* Segment a region containing the edges

\*基于全局阈值的图像快速阈值化

fast\_threshold (Image, Region, 0, 120, 7)

\*获取一个边界区域

boundary (Region, RegionBorder, 'inner')

\*通过区域的最小外接矩形，从矩形的 Top,Bottom,Left,Right四个方向裁剪区域，得到所需要的区域

clip\_region\_rel (RegionBorder, RegionClipped, 5, 5, 5, 5)

\*用圆形结构元素扩张一个区域。

dilation\_circle (RegionClipped, RegionDilation, 2.5)

\*获得特定区域Region位置的图像

reduce\_domain (Image, RegionDilation, ImageReduced)

\*

\*使用动态阈值分割图像

connection(Region, ConnectedRegions)

\*\* 图像矫正(第一个标定板，是正的，就不用校正)

\* 计算图像偏转角度

\*orientation\_region(ConnectedRegions,Phi)

\*area\_center (ConnectedRegions, Area, Row, Column)

\* 取图像变换到水平位置的二维矩阵

\*vector\_angle\_to\_rigid(Row,Column,Phi,Row,Column,0,HomMat2D)

\* 将二维矩阵作用到零件图像将其转换至水平位置

\*affine\_trans\_image(ImageReduced,ImagTrans,HomMat2D,'cons-tant','false')

\*提取亚像素精密边缘轮廓

edges\_sub\_pix (ImageReduced, Edges, 'canny', 2, 20, 60)

\*dev\_display (ImageReduced)

\*将一个XLD轮廓分割为直线段、圆（圆弧)、椭圆弧

segment\_contours\_xld (Edges, ContoursSplit, 'lines\_circles', 5, 4, 3)

\* 计算提取的轮廓数量

count\_obj (ContoursSplit, Number)

\*\*\*\*\*\*\*\*\*\*\*\*\*求第一个圆心坐标

select\_obj (ContoursSplit, SingleSegment, 2)

get\_contour\_global\_attrib\_xld (SingleSegment,'cont\_approx', Attrib)

\* 获取每段轮廓的属性， 当某段轮廓的属性 Attrib = 1 也就是圆弧段, 0 的近似为椭圆弧，属性为-1 的轮廓近似为线段

if (Attrib=1)

\* 通过合适的圆拟合圆弧轮廓段并获得对应圆心的图像坐标(Row,Column)及半径 Radius

fit\_circle\_contour\_xld (SingleSegment,'atukey',-1,2,0,5,2,Row,Column,Radius,StartPhi,EndPh,PointOrder)

\*拟合提取的亚像素边缘的外接矩形并返回外接矩形的相关参数

fit\_rectangle2\_contour\_xld(SingleSegment,'regression',-1,0,0,3,2,Row,Column,Phi,Length11,Length21,PointOrder)

\*生成对应的圆

gen\_ellipse\_contour\_xld (ContEllipse, Row,Column, 0, Radius,Radius, 0, rad(360),'positive',1.0)

endif

\*\*\*\*\*\*\*\*\*\*\*\*\*求第二个圆心坐标

select\_obj (ContoursSplit, SingleSegment2, 13)

get\_contour\_global\_attrib\_xld (SingleSegment2,'cont\_approx', Attrib2)

\* 获取每段轮廓的属性， 当某段轮廓的属性 Attrib = 1 也就是圆弧段, 0 的近似为椭圆弧，属性为-1 的轮廓近似为线段

if (Attrib2=1)

\* 通过合适的圆拟合圆弧轮廓段并获得对应圆心的图像坐标(Row,Column)及半径 Radius

fit\_circle\_contour\_xld (SingleSegment2,'atukey',-1,2,0,5,2,Row1,Column1,Radius1,StartPhi,EndPh,PointOrder)

\*拟合提取的亚像素边缘的外接矩形并返回外接矩形的相关参数

fit\_rectangle2\_contour\_xld(SingleSegment2,'regression',-1,0,0,3,2,Row1,Column1,Phi,Length12,Length22,PointOrder)

\*生成对应的圆

gen\_ellipse\_contour\_xld (ContEllipse1, Row1,Column1, 0, Radius1,Radius1, 0, rad(360),'positive',1.0)

endif

\* 最后通过算子 distance\_pp 算出圆弧圆心间的距离

distance\_pp(Row,Column,Row1, Column1, Distance)

PointImageDis[0]:=Distance

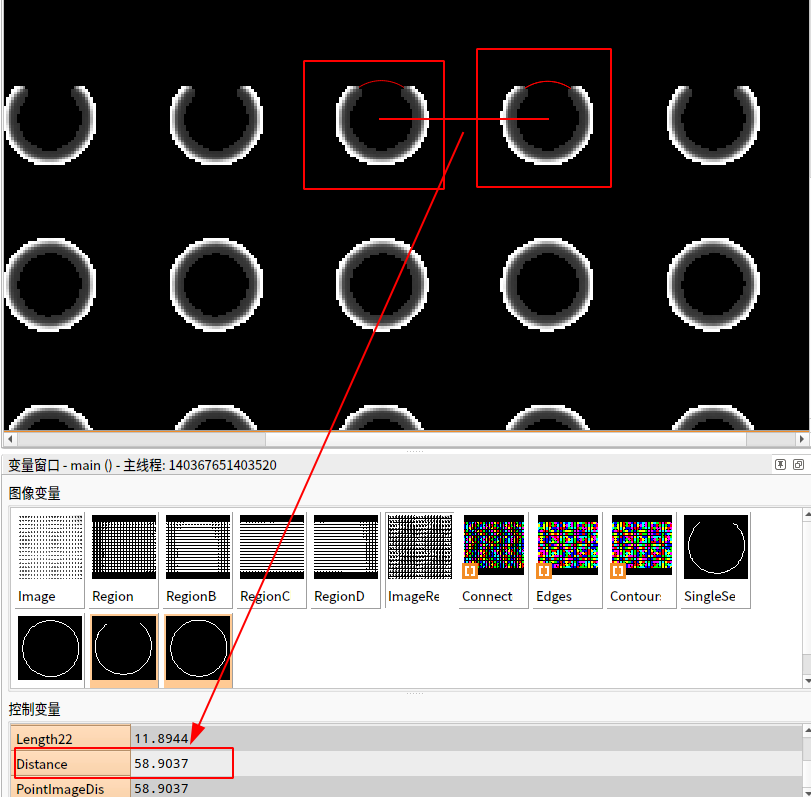
read\_image (Image2, 'Pic2\_4.bmp')

rgb1\_to\_gray (Image2, GrayImage)

fast\_threshold (GrayImage, Region2, 150, 255, 20)

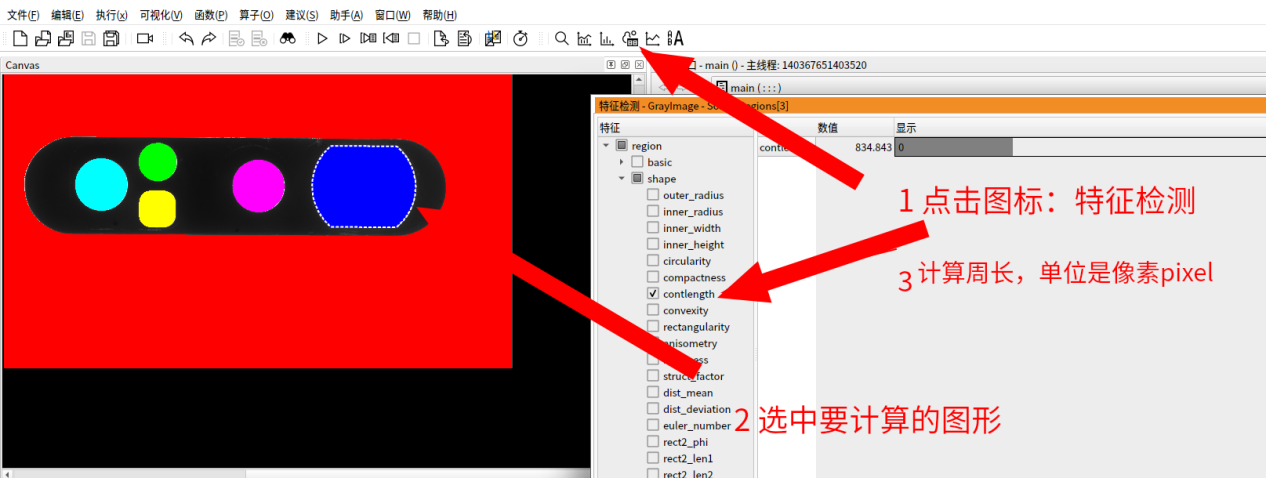
connection (Region2, ConnectedRegions2)

sort\_region (ConnectedRegions2, SortedRegions, 'first\_point', 'true', 'row')



1. 计算轮廓周长

代码在计算轮廓周长.txt中



第一步，运行程序

第二步，点击“特征检测图标”

第三步：点击要计算轮廓的图形

第四步：勾选在特征检测的shape-->countlength，计算轮廓周长，显示在“数值”处

\*

read\_image (Image2, 'Pic2\_4.bmp')

rgb1\_to\_gray (Image2, GrayImage)

fast\_threshold (GrayImage, Region2, 150, 255, 20)

connection (Region2, ConnectedRegions2)

sort\_region (ConnectedRegions2, SortedRegions, 'first\_point', 'false', 'row')

# 3 实验数据

|  |  |
| --- | --- |
| 轮廓 | 像素长度值 |
| 1 | 4532 |
| 2 | 437.789 |
| 3 | 320.392 |
| 4 | 330.309 |
| 5 | 436.617 |
| 6 | 834.843 |

