# 1.椭圆定位

1.1功能说明

【椭圆定位】模块在矩形ROI中寻找指定颜色、半径范围和方位的椭圆，通过椭圆中心和倾斜角构建定位坐标系。

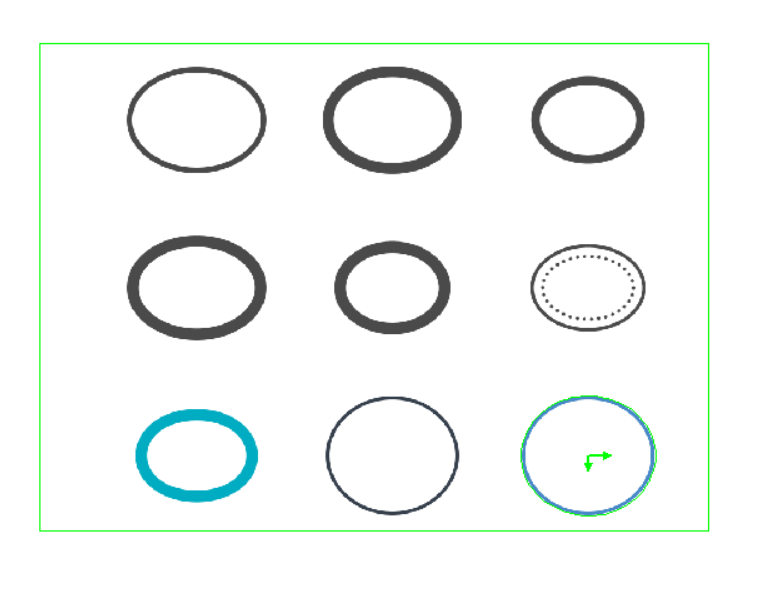


图4-2-121 【椭圆定位】例



图4-2-122 【椭圆定位】的参数

## 1.2参数说明

【颜色变化】设置椭圆的颜色；

【白色椭圆】椭圆内侧相对外侧是白色；

【黑色椭圆】椭圆内侧相对外侧是黑色；

【两者皆可】以上两者均可

椭圆是基于边缘轮廓分析得到的，以下两个参数设置边缘检出参数

【边缘尺度】边缘尺寸设置边缘检出的尺度。当图像噪声大或者边缘模糊的时候，需要增大边缘尺度。

【边缘阈值】边缘检出阈值

【半径范围】设置椭圆半长轴和半短轴的范围，用来缩小椭圆搜索的范围；

【圆度范围】圆度定义为椭圆短轴长度与长轴长度之比，用来缩小椭圆搜索的范围。

找到椭圆后需要对结果进行过滤，

【距离阈值】当边缘点和椭圆的距离小于此参数时，边缘点在椭圆上。

【检出阈值】椭圆上的边缘点覆盖椭圆一圈的比例就是椭圆的得分100分制。大于此阈值的才

被检出。可以用来过滤椭圆弧。

【挑选类型】用来挑选得到唯一的一个椭圆来定位。

1.3属性表示方式

Mat  src          ：输入图像  
scanArea scanArea       :扫描区域   
int colorChange       ：颜色变换，0为两者皆可，1为白到黑，-1为黑到白  
int edgeDimension       : 边缘尺度  
int edgeThreshold       ：边缘阈值  
int maxRadius        :最大半径  
int minRadius        :最小半径  
int maxRoundness      :最大圆度  
int minRoundness       :最小圆度  
int distanceThreshold   :距离阈值  
int score          :找圆得分(检出阈值)  
int selectType        :挑选类型，0为最小椭圆，1为最大椭圆  
typedef struct scanArea{ :自定义的扫描区域的表示方法  
    Point2f p1;  
    Point2f p2;  
    Point2f p3;  
    Point2f p4;  
}scanArea;  
  
typedef struct Ellipse{ :自定义的检出的椭圆的表示方法  
    Point2f center;  
    float radius1;  
    float radius2;  
    float angle;

}Ellipse;

## 1.4主要方法

# 1.5解决思路

1. 对原图片转化为灰度图进行滤波和边缘检测处理；
2. 用findcontours（）方法进行轮廓检测；
3. 根据ROI区域将非ROI区域的轮廓清除；
4. 对ROI区域轮廓求最小外接矩形；
5. 根据椭圆面积公式和所求的最小外接矩形判断轮廓是否是椭圆；
6. 根据给出的参数要求选择满足条件的唯一椭圆；

需要解决的主要问题：

1. 非ROI区域的轮廓清除
2. 椭圆的颜色判断方法

# 2.边定位（离散）

2.1功能说明

在有规律排列的一行序列化矩形ROI中，在每个子ROI中寻找一个边缘点，然后利用找到的边缘点拟合直线段。返回的定位坐标系原点为检出直线段的中点，角度固定为0.0。

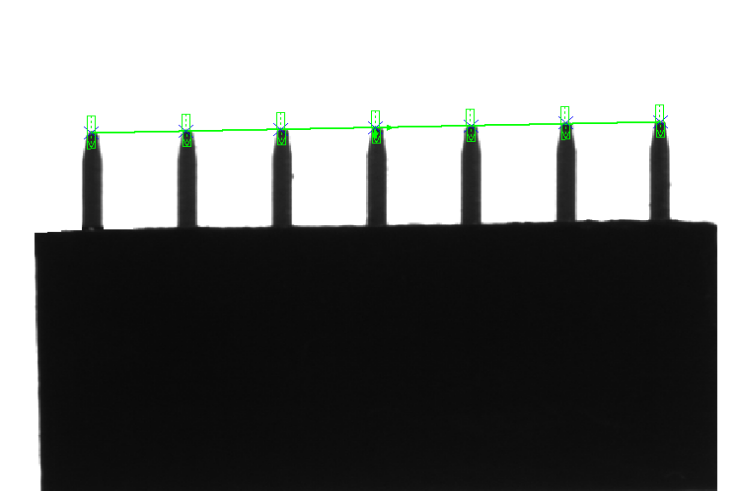


图4-2-111 添加后的【边定位（离散）】模块

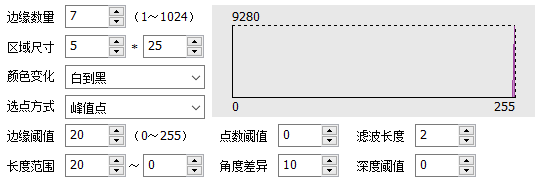


图4-2-112 【边定位（离散）】模块参数

## 2.2参数说明

【边定位（离散）】模块参数，

【边缘数量】序列化ROI中子ROI的数量就是检出边缘点的数量，即在每个子ROI中检出一个边缘点；

【区域尺寸】表示序列化ROI中每个子ROI的长度和宽度；

【颜色变化】边缘点颜色变化参数。边缘的扫描方向由序列化ROI确定，用户可以在图像上调整。子ROI中显示了每个子矩形ROI中边缘扫描方向；

【选点方式】扫描线上选择满足颜色变化需求并且强度大于阈值的峰值点或者选择第一个点；

【边缘阈值】，图像上检出边缘点时的阈值，参数范围为0到255。默认值20。【检出阈值】也可以通过【直方图显示窗口】中调整蓝色竖直线来调节。

【点数阈值】找到的边缘点数量少于此阈值时直接报错。

【扫描线宽度】边缘扫描线的宽度，等于子矩形ROI的宽度；

【滤波宽度】差分滤波器的半长度。

【深度阈值】此参数仅用户选择寻找第一个满足上述要求的边缘点时有效。大于设定阈值的边缘点与下一个大于同样阈值的反色边缘点的距离定义为此边缘点的深度。深度小于深度阈值的边缘点被认为是噪声，此时算法将跳过继续寻找满足要求的边缘点。

【长度范围】找到的边缘点拟合得到直线段之后，此直线段长度范围。最大值为0时表示不限制。

【角度差异】找到的边缘点拟合得到直线段之后，此直线段与【边定位】ROI的角度差异最大值，参数范围为0-180度，默认值为10度。

## 2.3属性表示方式

Mat  src         ：输入图像   
int selectPointCount    : 选点数量  
int blurWidth       ：扫描线宽度  
int roiHeight       : 子ROI高度  
int edgeThreshold    ：边缘阈值  
int selectionMethod   ：选点方式，0为峰值点，1为最先找到点  
int colorChange      ：颜色变换，0为黑到白，1为白到黑  
int blurLength      ：差分滤波器长度  
int depthThreshold     : 深度阈值  
int pointnumThreshold:  ：点数阈值  
int minlength       : 最小长度  
int maxlength       :最大长度  
int angleDiffer      :角度差异  
scanArea  detectArea   :扫描区域  
scanArea  templateArea  :模板区域  
float step         ：选子ROI时间距                 
  
typedef struct Line{ 自定义检出线段格式  
    Point2f p1;  
    Point2f p2;  
    Point2f center;  
}Line;  
typedef struct scanArea{ 自定义扫描区域格式  
    Point2f p1;  
    Point2f p2;  
}scanArea;

## 2.4主要方法

## 2.5解决思路

1. 寻找与扫描区域同起点同长度的水平线段；
2. 跟据参数要求在水平线上做出所有子ROI区域；
3. 根据旋转找到真实的子ROI区域；
4. 在所有子ROI区域通过扫描线进行扫描，找到满足条件的边缘点；
5. 对边缘点进行线性拟合，得到边缘；

### 3.圆检出

## 3.1功能说明

【圆检出】模块在【有向圆环ROI】中寻找圆，以确认是否有满足参数范围的圆的存在。

在【视觉模块列表】中双击【圆检出】后，如图4-4-51所示，【图像及数据显示区域】中会出现一个【有向圆环ROI】，表示【圆定位】的ROI区域和【圆定位】的扫描方向。拖动红色环状ROI到达预定位置，通过拖动ROI上内外两个半径以使得扫描方向为期望状态后，点击【添加】后将此模块添加到【处理流程】中，如图4-4-52所示。

如图4-4-52所示，检出的圆用绿色圆环显示出来。

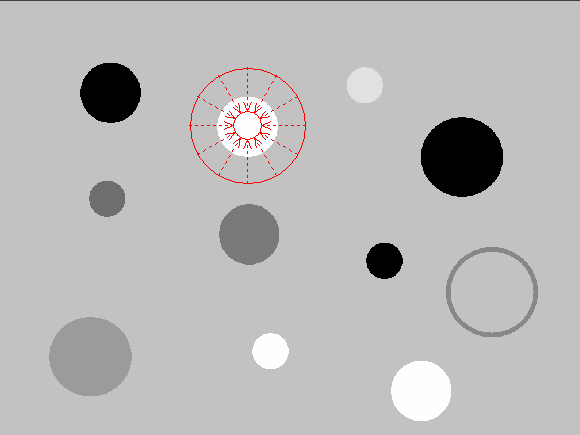


图4-4-51添加【圆检出】及ROI设定

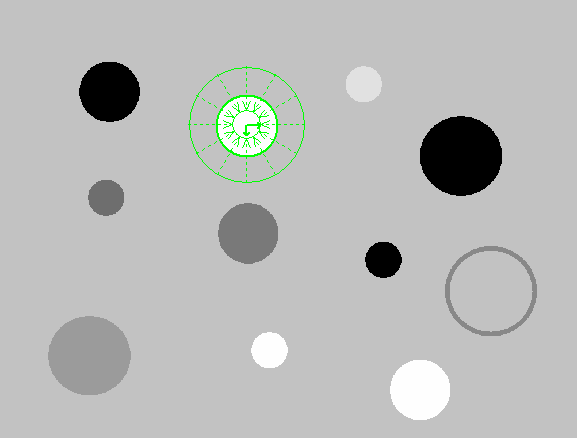


图4-4-52添加后的【圆检出】模块

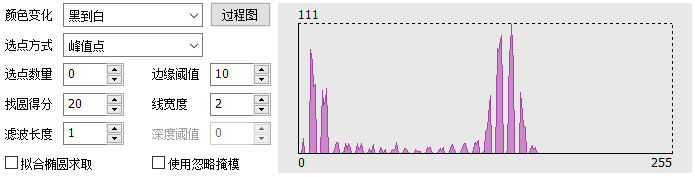


图4-4-53【圆检出】的参数

## 3.2参数说明

圆检出算法流程如下：

1）在用户设定的ROI中生成一组同心的扫描线；

2）每条扫描线上计算边缘点（详见4.1.6节）返回最多一个边缘点；

3）利用所有扫描线上的边缘点提取圆；

如图4-4-53所示，【圆检出】的参数有：

【颜色变化】沿着ROI设定的扫描方向进行扫描，寻找从【黑到白】或者【白到黑】的边缘点，二选一。

【选点方式】沿着ROI设定的扫描方向进行扫描，寻找【峰值点】或者【最先找到点】作为边缘点。【峰值点】指扫描方向上边缘最强且超过【检出阈值】的点，【最先找到点】指扫描方向上最先找到的边缘强度超过【检出阈值】的点。

【选点数量】沿着ROI设定的扫描方向等间隔设置扫描线进行扫描，扫描线的数量。数量为0时表示自动选择，一般为逐像素设置扫描线；数量大于0时表示实际的扫描线数量。

【边缘阈值】，图像上检出边缘点时的阈值，参数范围为0到255。默认值20。【检出阈值】也可以通过【直方图显示窗口】中调整蓝色竖直线来调节。

【线宽度】为了抑制噪声，在扫描方向上扫面边缘点时，可以选择一定的线宽，综合扫描方向上多个像素的信息寻找边缘，边缘位置会更稳定一点。参数范围为非负整数，默认值为3。

【滤波宽度】差分滤波器的半长度。

【深度阈值】此参数仅用户选择寻找第一个满足上述要求的边缘点时有效。大于设定阈值的边缘点与下一个大于同样阈值的反色边缘点的距离定义为此边缘点的深度。深度小于深度阈值的边缘点被认为是噪声，此时算法将跳过继续寻找满足要求的边缘点。

【找圆得分】根据边缘点落在圆上点的比例可计算得到圆的得分，用来赛选某些虚假的圆；

【使用忽略掩膜过滤边缘点】选中后可使用忽略掩膜过滤部分边缘点；

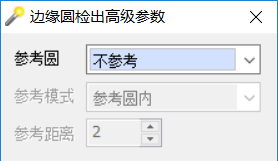


图4-4-54 【圆检出】高级参数

圆检出高级参数如图4-4-54所示，

【参考圆】默认不参考。当启用参考的时候，可以选择在参考圆的内部或者内部找圆，从而限定找圆的范围。当图像中多个圆靠近的时候，使用参考圆可以逐个找到一层一层的圆，从而准确定位需要的圆。

【参考模式】

【参考圆内】参考圆外的边缘点将被跳过；

【参考圆外】参考圆内的边缘点将被跳过；

【参考距离】距离圆小于此距离的边缘点也将被跳过；

## 3.3属性表示方式

Mat  src           ：输入图像  
scanArea detectArea       :检测区域   
scanArea templateArea     :模板区域   
int selectPointCount    :选点数量  
int blurWidth         ：滤波宽度  
int edgeThreshold      ：边缘阈值  
int selectionMethod   ：选点方式，0为峰值点，1为最先找到点  
int colorChange        ：颜色变换，0为黑到白，1为白到黑  
int blurLength        ：差分滤波器长度  
int depthThreshold     : 深度阈值  
float score          : 找圆得分  
float step          : 点距  
typedef struct scanArea{  
    Point center;//检测区域圆心坐标      
    int exRadius;//外半径长度  
    int inRadius;//内半径长度  
}scanArea;  
typedef struct node{  
    Point point;//位置  
    float value=0;//求平均值后的结果  
    float result=0;//滤波后的结果  
}node;    
     
typedef struct Circle{ 自定义检出圆的格式  
    Point2f center;  
    float radius;  
}Circle;

## 3.4主要方法

## 3.5解决思路

1. 根据选点数量确定扫描线间隔角度；
2. 根据检测区域圆环的内外半径和角度确定扫描线的两端点，进而确定扫描线；
3. 根据扫描线扫描得到满足条件的边缘点；
4. 对边缘点进行圆拟合；

# 4.边检出

## 4.1功能说明

【边检出】模块在【有向矩形ROI】中寻找直线段（直线段简称为“边”）的，以确认是否有满足参数范围的边的存在。

在【视觉模块列表】中双击【边定位】后，如图4-4-41所示，【图像及数据显示区域】中会出现一个红色【有向矩形ROI】，表示【边定位】的ROI区域和【边定位】的扫描方向。拖动红色ROI到达预定位置，通过拖动ROI上红色小圆点调整扫描方向角度到预定方向后，点击【添加】后将此模块添加到【处理流程】中，如图4-4-42所示。

如图4-4-42所示，检出的边用绿色直线显示出来。

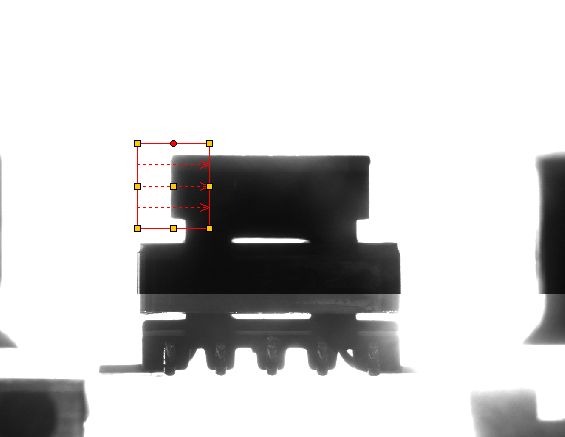


图4-4-41添加【边定位】及ROI设定

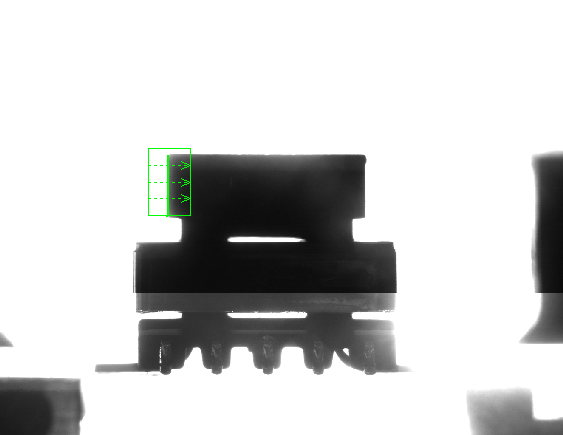


图4-4-42添加后的【边检出】模块



图4-4-43【边检出】的参数

## 4.2参数说明

边检出算法流程如下：

【颜色变化】沿着ROI设定的扫描方向进行扫描，寻找从【黑到白】或者【白到黑】的边缘点，二选一。

【选点方式】沿着ROI设定的扫描方向进行扫描，寻找【峰值点】或者【最先找到点】作为边缘点。【峰值点】指扫描方向上边缘最强且超过【检出阈值】的点，【最先找到点】指扫描方向上最先找到的边缘强度超过【检出阈值】的点。

【选点数量】沿着ROI设定的扫描方向等间距设置扫描线进行扫描，扫描线的数量。数量为0时表示自动选择，一般为逐像素设置扫描线；数量大于0时表示实际的扫描线数量。

【边缘阈值】，图像上检出边缘点时的阈值，参数范围为0到255。默认值20。【检出阈值】也可以通过【直方图显示窗口】中调整蓝色竖直线来调节。

【线宽度】是边缘检出时的扫描线宽度；

【滤波长度】边缘检出差分滤波器的半长度；

【深度阈值】选点方式为【最先找到点】时边缘点的深度阈值。

【长度范围】找到的边缘点拟合得到直线段之后，此直线段长度范围，最大值为0时表示不限制。

【角度差异】找到的边缘点拟合得到直线段之后，此直线段与【边定位】ROI的角度差异最大值，参数范围为0-180度，默认值为10度。

点击【高级】弹出对话框可设定高级参数。

【线宽度】为了抑制噪声，在扫描方向上扫面边缘点时，可以选择一定的线宽，综合扫描方向上多个像素的信息寻找边缘，边缘位置会更稳定一点。参数范围为非负整数，默认值为3。

【滤波宽度】差分滤波器的半长度。

## 4.3属性表示方式

Mat image;         输入图像  
Mat ROI;             ROI区域  
int selectPointCount;选点数量  
int blurWidth;       滤波宽度  
int edgeThreshold;  边缘阈值  
int selectionMethod; 选点方式，0为峰值点，1为最先找到点  
int colorChange;     颜色变换，0为黑到白，1为白到黑  
int blurLength;      差分滤波器长  
int edgeDepth;       边缘深度  
int minLength;       最小边缘长度  
int maxLength;       最大边缘长度  
int angleDiffer;     测出边缘与ROI的最大角度差异\*/  
typedef struct scanArea{             //检测区域  
    Point p1;  
    Point p2;  
    Point p3;  
    Point p4;  
}scanArea;  
typedef struct Node{  
    Point point;//位置  
    double value;//求平均值后的结果  
    double result=0;//滤波后的结果  
}node;  
typedef struct Line{  
    Point p1;  
    Point p2;  
}Line;

## 4.4主要方法

## 4.5解决思路

1.根据参数要求在ROI区域中生成一组平行的扫描线；

2.每条扫描线上计算边缘点，返回最多一个边缘点；

3.利用所有扫描线上的边缘点进行直线拟合；