

A Circle Positioning Algorithm for Electronic Element Detection

Zhihan Yang, Xiuli Shao, Zhaozheng Yuan, Huadong Zhang

School of Computer and Control Engineering, Nankai University, Tianjin
Email: shaoxl@nankai.edu.cn

Received: Oct. 6th, 2016; accepted: Oct. 24th, 2016; published: Oct. 28th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

At present, the image processing of the PCB circuit detection system based on machine vision is mainly to achieve the accurate positioning of components. According to machine vision positioning of the circular electronic components on the PCB board, we put forward the method: 1) to use Canny edge detection combined with Sobel transform coarse positioning detection method, 2) according to the detection target circle on the edge, and the gray gradient, to get the line set across edge points, 3) and then to use a sliding window to implement positioning of the center of the circle, 4) then to use the method of concentric rings, find the rounded edge radius, which can accurately locate the position of the circular electronic components. In practical application, the algorithm meets the requirements of real-time and accuracy in the industrial production of electronic enterprises.

Keywords

Circle Location, PCB Optical Detection, Edge Detection, Gray Gradient

一种用于电子元件检测的圆定位算法

杨志涵, 邵秀丽, 袁兆争, 张华东

南开大学计算机与控制工程学院, 天津
Email: shaoxl@nankai.edu.cn

收稿日期: 2016年10月6日; 录用日期: 2016年10月24日; 发布日期: 2016年10月28日

摘要

目前基于机器视觉、图像处理的PCB电路检测系统首要实现的就是对元件的准确定位。本文针对PCB板上的圆形电子元件进行机器视觉定位,提出了先使用Canny边缘检测结合Sobel变换的粗定位检测方法,再根据检测目标圆的边缘上的点,以及各点的灰度梯度,得到过边缘点的直线集合,继而使用正方形滑动窗口,实现圆心的定位,最后再使用同心圆环的方法,找到圆形边缘的半径,从而精确定位圆形电子元件的位置。经实际使用,本文算法满足电子企业工业生产中实时性和精确性的要求。

关键词

圆定位, PCB光学检测, 边沿检测, 灰度梯度

1. 前言

在常见的PCB板的电子元件中,大多数的电子元件外形无非是矩形、圆形两种。如果能在检测中,准确快速的对这两种外形的元件的外轮廓进行定位,将极大程度的提高企业生产线上对其识别的准确性和稳定性。在PCB定位中,如果预知每个圆形元件的坐标区域,在实际匹配计算时,只需适当扩大搜索区域,在该区域内即可找到一个最大的最佳匹配圆即可。

在图像处理中,Hough圆定位算法是Paul Hough在1962年提出的一种图像边缘检测技术,它可以检测图像空间中的任意解析曲线[1]。目前很多已有的圆定位算法,大都是基于Hough算法的改进。目前Hough圆定位算法会在一张图中会计算得到多个符合条件的圆,而非在区域内优选出单一的圆,因此,无法满足PCB定位中圆定位算法的目的,在一个区域内找到一个最优圆的需求。

基于模板匹配的圆定位算法在实际中也有使用,该方法可以在一个区域内找到和模板匹配最佳的圆,但是该方法,无法用同一个圆模板,满足不同半径的圆的定位。PCB定位中圆的直径,可能会由于拍摄相机的位置变化而有改变,这是模板匹配的算法所无法满足的。

目前PCB电路圆的定位,有基于Hough变换[2][3][4]、基于势函数[5]等方法,但是都没有很好的解决在设定区域内快速定位最大半径圆的问题。

本文设计了一套基于边沿梯度方向统计的圆定位算法,在确定的图像内找到最大的匹配圆,首先使用Canny边缘检测结合Sobel变换先计算得到圆心区域的大概位置,然后统计对圆心区域有贡献的点,得到目标圆的圆周上的点,然后再以得到的圆周,对圆心进行二次定位,从而得到最终的圆心和半径。实验表明对于边沿清晰的目标圆,该算法速度较快,长宽200像素内的图像,运算速度在50ms以内。对于圆形的电子元件允许一定程度的圆周不完整(大于70%周长),允许一定程度的圆周变形(椭圆),对圆的半径变化也有一定是适应性。该算法于灰度图像进行计算,所以对于彩色、灰度CCD相机都可满足,对于边沿清晰的圆形物体识别准确度高。

2. PCB板的圆定位算法

图1给出了圆定位算法,其主要包含四个主要处理过程:预处理、圆心粗定位、半径计算、圆心再定位。

图1所示的圆定位算法流程图中虚框框起来的是四个主要处理工作,接下来以最常见的电容元件为例,介绍论文提出的定位算法的这四个具体处理工作。介绍每一处理工作的内容、处理的结果,以及得到的结果。

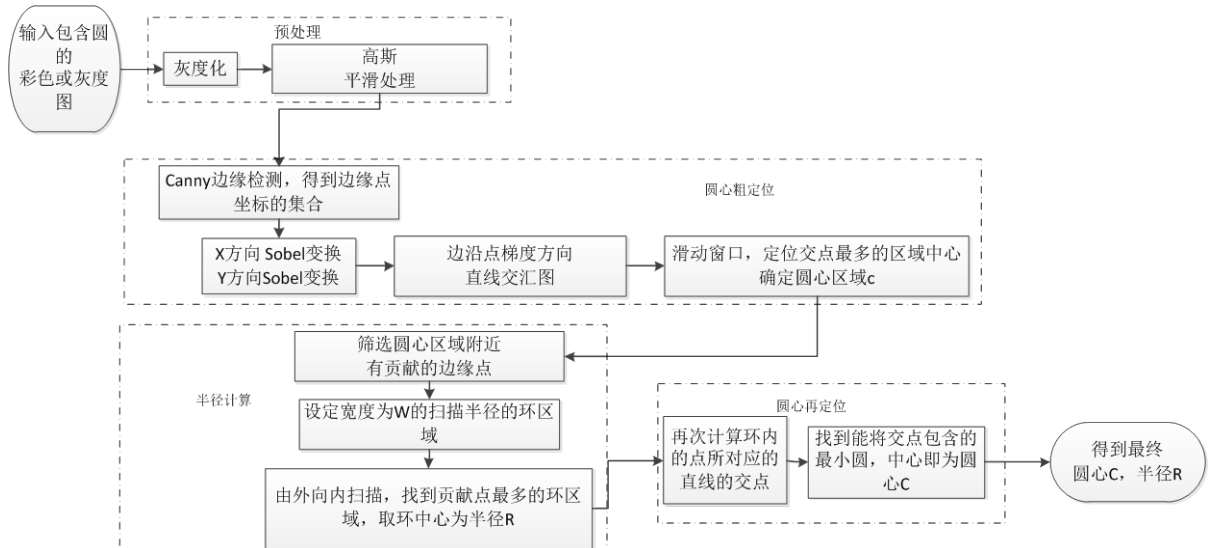


Figure 1. Circle location algorithm

图 1. 圆定位算法流程

2.1. 预处理

2.1.1. 灰度化

本步工作主要完成对所拍摄的彩图进行灰度化转换。方法是基于计算公式 1 把输入的彩色图像(图 2 左所示)的每一坐标值对应计算,从而使其转换为单通道的灰度图像(如图 2 中所示),即将 RGB 三通道彩色图像通过变换,转化为灰度图像的计算。

$$Gray(x, y) = 0.299 \times R(x, y) + 0.587 \times G(x, y) + 0.144 \times B(x, y) \quad (\text{公式 1})$$

其中 x, y 为图中每个点的坐标值。

2.1.2. 高斯平滑处理

对于图像非圆周的细节信息,并不会对圆定位有所贡献,反而会对定位工作带来干扰,所以要减弱这种干扰。

本文采用高斯平滑算法[6] [7],用它来减少图像噪音以及降低细节层次,平滑窗口取 5×5 像素。图 2 右显示出图像的细节已被模糊处理。

2.2. 粗定位圆心区域

粗定位圆心区域的工作可分为:计算边沿、计算边沿点 Sobel 切线、生成交点图和计算滑动窗口圆心四个步骤,从而得到一个圆心的圆区域。

2.2.1. 边沿的计算

边沿(边缘): 图像的边缘是指图像局部区域亮度变化显著的部分,该区域的灰度梯度变化一般可以看作是一个阶跃,既从一个灰度值在很小的缓冲区域内急剧变化到另一个灰度相差较大的灰度值。通常图像的边沿集中了图像的很多信息,是图像场景识别和理解的基础,是图像分割、定位、测量、识别的重要手段。图 3 给出了边沿方向和梯度方向,图中高度表示灰度值。

圆心在圆定位的过程中是通过圆周的定位来确定的,圆周在图像中即为的圆的边沿。本文对此采用最常用的 Canny 算法来计算圆的边沿图,得到图 4 所示结果。Canny 边缘检测算子是 John F. Canny [8]

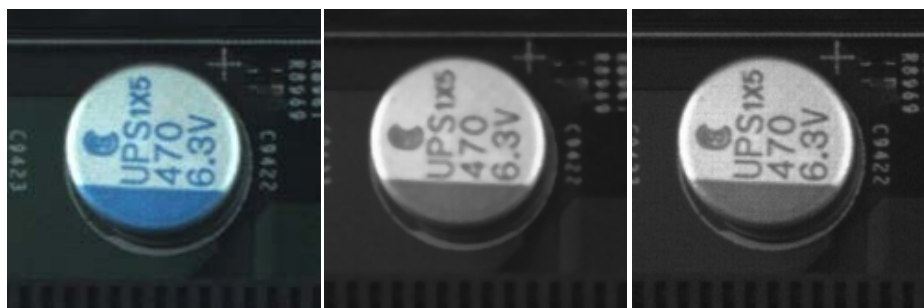


Figure 2. Gray and smooth processing of color image
图 2. 彩色图的灰度化和平滑处理

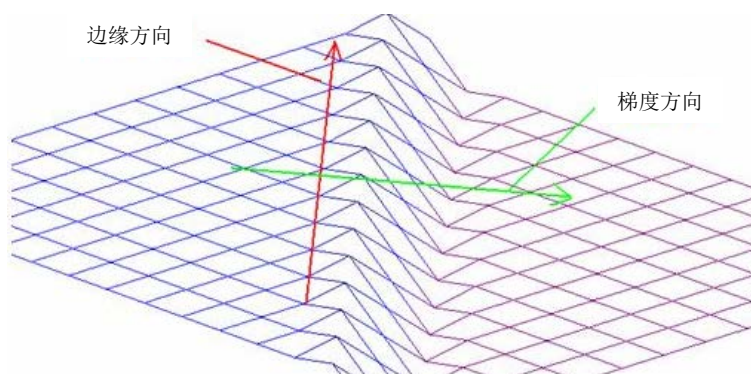


Figure 3. Edge and gradient direction
图 3. 边沿和梯度方向



Figure 4. Based on the edge of Canny computing
图 4. 基于 Canny 计算边沿

于 1986 年开发出来的一个多级边缘检测算法，主要包括五个处理步骤：

- 1) 应用高斯滤波来平滑图像去除噪声；
- 2) 找寻图像的强度梯度(intensity gradients)，用一阶偏导(Sobel 算子)计算梯度的方向和幅度值；
- 3) 应用非最大抑制(non-maximum suppression)技术来消除边误检；
- 4) 应用双阈值的方法来决定可能的边界；
- 5) 利用滞后技术来跟踪边界。

2.2.2. 边沿点 Sobel 切线方向的计算

Sobel [9] [10] [11]算子常用于边缘检测中计算图像亮度函数的一阶梯度之近似值。在图像的任何一点使用此算子，将会产生该点对应的梯度矢量或是其法矢量。本文使用的 Sobel 算子如下：

Sobel X: X 方向求导，将图像(I)与一个奇数大小的内核进行卷积。

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \times I$$

Sobel Y: Y 方向求导，将图像(I)与一个奇数大小的内核进行卷积。

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \times I$$

对灰度图的每一个点分别计算 Sobel X Y 卷积得到 X、Y 两个方向的 Sobel 图得到图 5 所示结果。

从图 5 左和中图中可看出 X, Y 方向的 Sobel 计算结果即为 X、Y 方向的灰度变换梯度，将其视为该点的正弦、余弦，那么过该点的正切即为 $\frac{\text{Sobel Y}}{\text{Sobel X}}$ 。

由于本文只对图像的边沿感兴趣，所以只需要计算 Canny 边沿图上的每一个边沿点的正切线即可。

以边沿图中边沿点(56,56)为例，Sobel X 的(56,56)值为 $s_x = 573$ ，Sobel Y 的(56,56)值为 $s_y = 653$ ，(56,56)点的正切值为，(i,j)为计算点的坐标，计算得到 $\frac{653}{573} = 1.139616$ ，计算反正切，得到角度为 $\theta = 48.73^\circ$ ，从图 6 右中看到计算和实际方向一致。

2.2.3. 绘制交点图

对 Canny 的边沿图(图 4)中白色的边沿上的每一个点，计算正切值，并将计算后得到的切线绘制到统计图中，最终得到对应的图 6 所示的切线交点图，处理步骤如下：

- 1) 创建空白的切线交点图 Cross;
- 2) Canny 图中按坐标顺序，取一点边沿点 C;
- 3) 根据 SobelX, Sobel 计算 C 点的正切;
- 4) 在 Corss 图中，绘制经过 C 的切线;
- 5) 如果切线经过点 P, P 当前的值位 D;
- 6) 修改 P 点的值为 $D + 1$;
- 7) 依次计算切线上每一个点的新值;
- 8) 计算 Canny 图中下一个边沿点，直到遍历完毕。

2.2.4. 滑动窗口圆心区域粗定位

在图 6 右中可看出，亮度最高的区域就是边沿切线交汇最多的区域，即圆心所在的区域。粗定位步骤如下：

- 1) 图的宽 W ，高 H ，设定滑动窗口的宽为 $s_w = \frac{W}{10}$ ，高为 $s_h = \frac{H}{10}$;
- 2) 设定搜索步长 X 方向为 $p_w = \frac{s_w}{5}$ ，Y 方向为 $p_h = \frac{s_h}{5}$;
- 3) 遍历全图(图 6 右)，找到滑动窗口像素和最大的区域，得到粗定位的圆心区域;

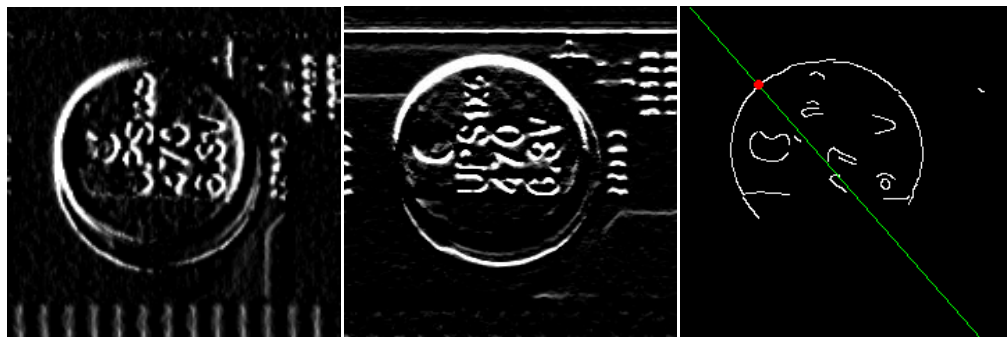


Figure 5. Calculation of X Sobel and Y Sobel and its tangent direction diagram
图 5. 计算 Sobel X 和 Sobel Y 及其切线方向示意图

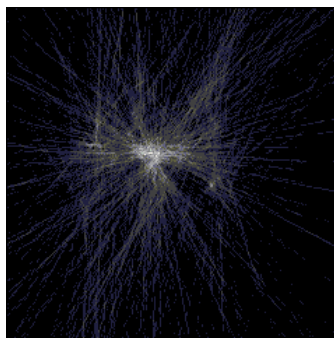


Figure 6. Tangent intersection graph
图 6. 切线交点图

- 4) 初始化滑动窗口像素和最大值 $sum_max = 0$, 最大值坐标位置 $(X_{max}, Y_{max}) = (-1, -1)$;
- 5) 滑动窗口左上角坐标 (X, Y) 从图 7 左上角 $(0, 0)$ 开始, 计算 $(X, Y, W, H) = (0, 0, s_w, s_h)$, 区域的像素值总数 sum , 如果 $sum > sum_max$:
 - a) $sum_max = sum$, $(X_{max}, Y_{max}) = (X, Y)$;
 - b) 更新 (X, Y) 坐标为 (X_1, Y_1) , 计算新坐标 (X_1, Y_1) 的 sum , 直到遍历完毕;
 - c) $X' = X + p_w$, $Y' = Y + p_h$;
 - d) 如果 $X' < (W - s_w)$, 新坐标为 $(X_1, Y_1) = (X', Y)$;
 - e) 如果 $X' > (W - s_w)$ 且 $Y' < (H - s_h)$, 新坐标为 $(X_1, Y_1) = (0, Y')$;
 - f) 如果 $Y' \geq (H - s_h)$, 遍历完毕, 结束运算。

通过上面的步骤, 得到图 7 中绿框所示的圆心区域, 取绿色矩形的中心点为粗定位圆心 Center。

2.3. 半径计算

计算得到粗定位圆心后, 再对边沿点集中的点进行筛选, 步骤如下:

- 1) 计算 Canny 图(图 4)的边沿点中, 对圆心区域(图 8)有贡献的边沿点。
 - a) 对 Canny 中每一个白点;
 - b) 取 Canny 中白点 $P(X, Y)$;
 - c) P 点切线穿过圆心粗定位区域(图 8 的绿色区域);
 - d) P 点到矩形中心的连线, 和 P 点切线的角度小于设定值 15° ;
 - e) P 点满足条件则保留, 否则删除, 最终得到有贡献的边沿图, 如图 8 所示。

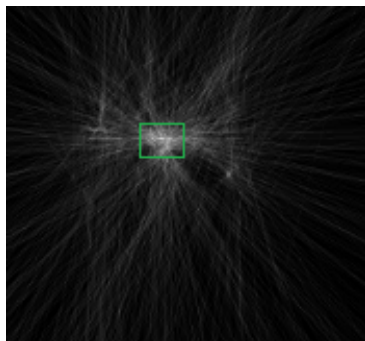


Figure 7. The center location
图 7. 圆心粗定位



Figure 8. Center edge contribution
图 8. 圆心有贡献边沿

2) 以粗定位圆心 $Center(X,Y)$ 为中心, 从外向内确定半径 R 。

a) 图像宽高 (W,H) 。 X , $(W-X)$, Y , $(H-Y)$ 四个值中最小值, 作为搜索半径的最大值 R_{max} , 半径的搜

索区域为 $[R_{max}, R_{min}]$, $R_{min} = \frac{R_{max}}{3}$, 搜索宽度为 $s_w = \frac{R_{max}}{20}$;

b) 初始 $R_x = R_{max} - \frac{s_w}{2}$, 计算 $R_1 = R_{max}$, $R_2 = R_{max} - s_w$, 环形区域内的所有白点数目 Sum ;

c) 如果 $Sum > 2 \times \pi \times R_x \times 0.7$, 即环形区域达到 70% 的圆半径, 则得到最终的半径 R ;

d) 否则 $R_x = R_x - 1$, 继续搜索, 直到找到。如果 $R_x = R_{min}$, 还未找到, 则定位半径失败。

最终计算得到的半径 R 如图 9 所示。

2.4. 圆心再定位

粗定位的圆心并不准确, 在计算得到半径后, 我们还需要对圆心再一次定位, 步骤如下。

1) 将图 9 中, 半径 $\left[R - \frac{s_w}{2}, R + \frac{s_w}{2} \right]$, 圆环区域内的边沿点保留, 去掉其他的边沿点。

2) 如图 10 中白色边沿所示。

3) 根据 SobelX、SobelY, 对图 10 中每个边沿点, 按照圆心粗定位的算法, 得到切线交汇图(图 11)。

4) 在切线交汇图中, 使用圆心滑动窗口(半径为 $\frac{R}{5}$), 找到切线交汇最多的区域, 小圆圆心即为最终

的圆心, 如图 10 中心点所示。

最终, 获得圆定位结果如图 12 所示。

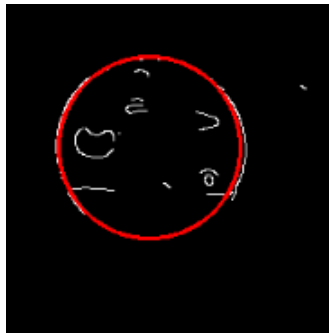


Figure 9. Radius localization
图 9. 半径定位

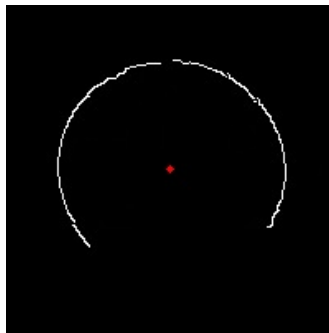


Figure 10. Ring edge
图 10. 圆环边沿

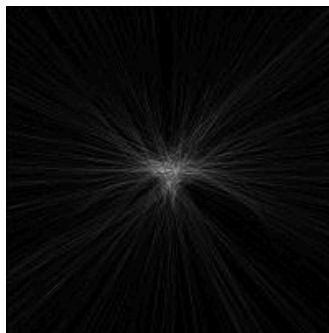


Figure 11. Tangent intersection graph
图 11. 切线交汇图

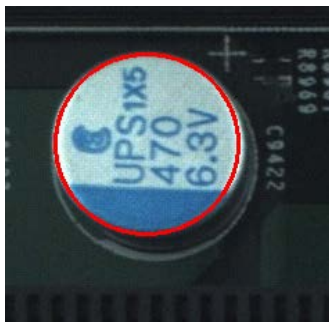


Figure 12. Location result
图 12. 定位结果

2.5. 算法验证



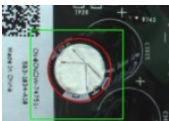


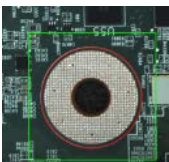
本文算法在电子电路检测中，主要针对的圆形检测对象可以是：电容、蜂鸣器、锂电池等圆形手插件、板卡上的各种孔、洞、定位点、螺丝等各种紧固件，以及其他圆形边缘元件。为验证本文算法的准确性和实际性能，首先要对相应的电路板进行拍照，应用本文提出的圆定位算法对其上面的圆形元件或板卡上的孔洞进行检测实验。

为此，本文搭建了包括拍摄平台、算法平台、识别程序三部分组成的测试环境。其中：拍摄实验平台是由千万像素 CCD 工业相机、镜头、LED 光源、封闭机柜、工控机组成；算法测试平台是在高性能 PC (intel i5 CPU、4G DDR3 1600 内存、运行 Win7 操作系统)运行；识别程序是使用 VS2005 的 MFC 编写，图像库使用 OpenCV2.0。选择了几种典型的圆形元件，如：电容，螺丝，圆形打孔，塑料、金属支撑立柱等常见的圆形元件。实验结果如表 1 所示。

采用圆定位算法得到的圆心平均偏移误差在 2 个像素以内，半径误差在 4 个像素之内，运算时间和图像的像素数目有关，对于 200×200 像素的图像，运算时间在 30 ms 左右，可以满足大多数圆形元件的准实时定位需求。与传统的基于 Hough 变换的圆定位算法相比，该算法数学模型清晰、易于实现、更适用于 PCB 光学检测系统的定位需求。

Table 1. Experimental result

表 1. 实验结果

元件	搜索区域 (像素)	平均耗时 (ms)	平均误差 (x,y,r)
	200 × 200	37	(1,1,1)
	200 × 200	32	(1,1,1)
	250 × 250	47	(1,1,2)
	200 × 200	30	(1,1,1)
	280 × 280	58	(2,3,2)
	480 × 480	67	(2,2,4)

2.6. 结论

本文提出的圆形电池定位算法具有一定的通用性, 实际应用时, 可根据实际需要调节模、canny 变换、滑动窗口、圆周比例等参数。如果效果不理想, 还可以对模糊、圆心定位、半径搜索的算法加以改进, 即可得到满意的效果。

基金项目

天津市支撑项目(15ZXHLGX00380); 天津市支撑项目(15ZXHLGX00360)。

参考文献 (References)

- [1] Uda, R.O. and Hart, P.E. (1972) Use of the Hough Transform to Detect Lines and Curves in Pictures. *Communications of the ACM*, **15**, 11-15. <http://dx.doi.org/10.1145/361237.361242>
- [2] 夏磊, 蔡超, 周成平, 丁明跃. 一种用 Hough 变换检测圆的快速算法[J]. 计算机应用研究, 2007, 10(24): 197-210.
- [3] Pei, S.-C. and Horng, J.-H. (1995) Circular Arc Detection Based on Hough Transform. *Pattern Recognition Letters*, **16**, 615-625. [http://dx.doi.org/10.1016/0167-8655\(95\)80007-G](http://dx.doi.org/10.1016/0167-8655(95)80007-G)
- [4] 叶峰, 陈灿杰, 赖乙宗, 等. 基于有序 Hough 变换的快速圆检测算法[J]. 光学精密工程, 2014, 22(4): 1105-1111.
- [5] 郭峰林, 管庶安, 何健. 基于势函数的印刷电路板定位孔快速定位算法[J]. 科技导报, 2010, 28(16): 71-73.
- [6] 王耀贵. 图像高斯平滑滤波分析[J]. 计算机与信息技术, 2008(8).
- [7] Jain, R., Kasturi, R. and Schunck, B.G. (1995) *Machine Vision*. McGraw-Hill, Inc., Boston, 112-139.
- [8] Canny, J. (1986) A Computational Approach to Edge Detection. *IEEE Trans Pattern Analysis and Machine Intelligence*, **8**, 679-698. <http://dx.doi.org/10.1109/TPAMI.1986.4767851>
- [9] 何春华, 张雪飞, 胡迎春. 基于改进 Sobel 算子的边缘检测算法的研究[J]. 光学技术, 2012(3): 323-327.
- [10] Vincent, O.R. and Folorunso, O. (2009) A Descriptive Algorithm for Sobel Image Edge Detection. *Proceedings of Informing Science & IT Education Conference (InSITE)*, 97-107.
- [11] 沈德海, 侯建, 鄂旭. 基于改进的 Sobel 算子边缘检测算法[J]. 计算机技术与发展, 2013(11): 22-25.

期刊投稿者将享受如下服务:

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org