

DOI:10.19651/j.cnki.emt.2417215

# 基于霍夫梯度的同步并行圆检测方法<sup>\*</sup>

苗少杰<sup>1</sup> 李宪<sup>1,2</sup> 赵东杰<sup>1,2</sup> 孙宇<sup>1</sup>

(1. 青岛大学自动化学院 青岛 266000; 2. 青岛大学未来研究院 青岛 266000)

**摘要:** 机器人视觉伺服控制主要依靠其视觉系统对工件角点、边线和圆等的检测为其后续决策与控制提供依据。随着系统作业密度的骤增,边缘模组已很难满足其检测负荷,尤其面对复杂场景的连续批量检测,终端视觉的实时性和准确性均受到巨大挑战,降低了系统作业效率。针对上述瓶颈,尤其是难度更高的圆检测,创新性的提出了一种基于霍夫梯度的同步圆检测方法:通过对边缘图像进行边缘筛选去除图像中的干扰信息;然后通过八点法同步确定圆心和半径,并通过半径再搜索减少半径误差,通过圆心位置约束减少无效计算;最后通过候选圆再搜索和最优圆获取实现图像中圆目标的准确检测。为进一步提高圆检测速度和效率,上述算法与 CUDA 并行技术进一步融合,提出了一种基于霍夫梯度的同步并行圆检测方法,能够充分利用并行计算的优势加速圆检测过程。实验结果表明,与 GHT、CACD、RCD 和 Zhao 等相比,该方法显著提升了圆检测的精度和效率,具备更强的抗噪声和抗扰动能力。其精确率、召回率、F 值分别为 99.1%、90.7% 和 94.7%;单张图像的平均检测时间为 0.09 s,检测效率最高提升 26 倍,使其在工业领域的批量图像处理任务中具有较好的实用价值。

**关键词:** 霍夫变换;圆检测;同步;并行计算;八点法

中图分类号: TP391.41; TN06 文献标识码: A 国家标准学科分类代码: 510.4050

## Synchronous parallel circle detection method based on Hough gradient

Miao Shaojie<sup>1</sup> Li Xian<sup>1,2</sup> Zhao Dongjie<sup>1,2</sup> Sun Yu<sup>1</sup>

(1. Department of Automation, Qingdao University, Qingdao 266000, China; 2. Institute for Future, Qingdao University, Qingdao 266000, China)

**Abstract:** Visual servo control robots mainly rely on their vision systems to detect corner points, edges, and circles of workpieces, providing the basis for subsequent decision-making and control. With the rapid increase of its workload density, the edge modules struggle to meet these detection loads, especially for continuous batch detection in complex scenarios, where the real-time and accuracy of visual system are greatly challenged, reducing the system efficiency. To deal with the above bottlenecks in circle detection, particularly the more challenging circle detection, we proposed a novel synchronous circle detection method based on Hough gradient: the interference information in the image is removed by edge screening of the edge image; the center and radius are synchronously determined by the eight-point method; the radius error is reduced by radius re-search, and invalid calculations are reduced by center position constraints; the accurate detection of circular targets in the image is achieved by candidate circle re-search and optimal circle acquisition. To further enhance the speed and efficiency of circle detection, the above method is integrated with CUDA parallel technology for a synchronous parallel circle detection method based on Hough gradient, which fully utilizes its parallel computing to accelerate the circle detection. In comparison with GHT, CACD, RCD, and Zhao, the proposed method significantly improves the circle detection accuracy and efficiency with stronger anti-noise and anti-disturbance capabilities, where its precision, recall rate and F value are 99.1%, 90.7% and 94.7% respectively; the average detection time is 0.09 s per image with an efficiency increase by up to 26 times, making it suitable for batch image processing in industry.

**Keywords:** Hough transform; circle detection; synchronization; parallel computing; eight-point method

收稿日期:2024-10-26

\*基金项目:山东省自然科学基金(ZR2021QF038)项目资助

## 0 引言

智能系统中,针对作业目标的形状检测是终端机器人环境感知的关键,为其后续决策与控制提供了关键依据<sup>[1]</sup>。尤其在工业场景,作业机器人主要依靠其视觉系统对工件角点、边线和圆等目标的精准检测,通过各类坐标系转换,最终形成机器人的路径规划和运动控制参数。在此类场景中,如交通标志<sup>[2]</sup>和工业零件<sup>[3]</sup>等,圆检测难度大、计算负担重,现有算法始终无法兼顾其检测实时性和准确率。因此,如何快速准确地从数字图像中检测圆形目标已成为一个重要的研究方向。

作为圆检测的经典方法,文献[4]提出的标准霍夫变换圆检测存在占用内存空间大,对背景噪声敏感等缺点<sup>[5]</sup>。为了克服这些局限,研究者们提出了多种优化思路。文献[6]提出基于霍夫梯度的圆检测方法(gradient Hough transform, GHT),利用圆边缘的梯度有效减少了投票的参数空间,在一定程度减少了空间的占用,但该方法存在计算效率低和无效采样次数多的挑战。文献[7-8]提出随机霍夫变换圆检测算法(randomized Hough transform, RHT),通过随机采样边缘点求取圆参数并进行累积,最终和设定的阈值进行比较,在一定程度上降低了计算负荷,但当目标图像复杂度提升时,无效样本的增加将导致计算负荷急剧增加和内存需求显著上升<sup>[9]</sup>。文献[10]提出的随机圆检测算法(randomized circle detection, RCD),通过随机选择图像中4个边缘像素,并定义一个标准来确定图像中的候选圆。RCD算法不需要累积器保存参数空间信息,在内存需求和计算效率方面均有显著优势,但需大量迭代尝试才能找到潜在候选圆。文献[11]提出随机抽样一致性圆检测算法(random sampling consistency, RANSAC),仅对长边缘进行抽样,当首次检测到候选圆时,利用最小二乘法对候选圆参数进行优化,提高了检测的准确性,但大量圆外点会带来大量无效抽样,使算法效率难以提升。文献[12]提出了曲率辅助 Hough 变换(curvature aided Hough transform, CACD),利用边缘点的曲率信息辅助 Hough 变换,实现了圆检测的快速性和准确性,但该算法对边缘检测的准确性依赖较高,在复杂场景中,候选圆的生成仍然需要多次尝试<sup>[13]</sup>。Zhao 等<sup>[14]</sup>提出了基于圆内切三角形的检测算法,根据位置约束和内切三角形约束组合属于同一圆的圆弧,基于 Theil-Sen 估计器和线性误差修正,利用内切三角形估计圆参数,这提高了算法的抗遮挡能力,但增加了算法的计算复杂度。

随着图形处理单元(graphics processing unit, GPU)的飞速发展,其已经广泛应用于数据处理和分析领域,NVIDIA 的通用并行计算架构(compute unified device architecture,CUDA)也为并行计算提供了强大的加速能力。文献[15]提出了基于 GPU 的并行车道检测算法,通过合理地将计算资源和任务分配给主机(CPU)和设备

(GPU),显著减少了处理数据的时间成本。文献[16]提出了基于 CUDA 的航空 γ 能谱数据并行算法,对航空 γ 能谱数据小波降噪处理过程进行加速优化,提升了计算效率。这些研究充分展示了 GPU 并行计算在处理高密度计算任务时的巨大潜力。

基于 GPU 在通用计算领域处理数据的特点和优势,特别是其强大的并行处理能力,结合梯度计算和 Hough 变换这两种本身适宜于并行处理的算法特点,创新性的提出了一种基于霍夫梯度的同步并行圆检测方法(synchronous parallel gradient Hough transform, SPGHT),通过滤波和边缘段筛选去除了多余的像素点和边缘,并利用梯度求取圆的参数,减少了无效圆的获取,通过候选圆再搜索有效减少圆的漏检,最后保留最优圆,并结合 CUDA 并行运算,实现了算法实时性的提升。通过对不同场景下的圆检测、本文方法和对比方法的对比分析、本文方法并行处理与否的对比分析,验证本文所提方法的有效性。

## 1 传统霍夫梯度圆检测方法及其改进

### 1.1 基于霍夫梯度的圆检测原理

霍夫梯度圆检测算法利用图像边缘梯度,将在三维空间的投票简化为二维空间的投票,具体可分为图像预处理、圆心投票、半径查找和圆参数的确定。

#### 1) 图像预处理

通过图像滤波减少噪声和细节的干扰,利用边缘检测算法生成边缘图  $E$ ,并计算其边缘点在 X、Y 方向的梯度以获得梯度图  $G_x, G_y$ 。

$$G_x = F_x * E \quad (1)$$

$$G_y = F_y * E \quad (2)$$

$$\theta = \arctan\left(\frac{G_x}{G_y}\right) \quad (3)$$

式中:  $F_x$  和  $F_y$  分别为 Prewitt 算子在 X 方向和 Y 方向梯度卷积核,  $\theta$  是像素梯度方向。

#### 2) 圆心投票

初始化圆心投票矩阵。对于图像中的每个边缘点,获取其梯度值  $k$  和像素位置  $(x_i, y_i)$ 。基于此,确定每个边缘点的梯度线方程。随后,对每条梯度线穿过的像素位置投票,最终得到圆心投票矩阵。

#### 3) 半径查找

对圆心投票矩阵进行阈值处理或局部最大值检测,以识别出潜在圆心。针对每一圆心,遍历边缘点并统计各边缘点到圆心的距离,得到每个圆心的半径直方图。

#### 4) 圆参数的确定

对每个半径直方图进行筛选,选择统计峰值作为真实圆的半径,也可求取多个统计峰值的均值作为半径。

相较于霍夫变换圆检测,加入了梯度的圆检测算法具有更强的鲁棒性,同时梯度能够减少参数空间的投票范围,在一定程度上减少了计算复杂度,使其在准确性和效率上

均有显著的提升。

## 1.2 基于霍夫梯度的圆检测改进方法

在圆检测领域,霍夫梯度法得到了广泛的应用和深入的研究。针对传统的霍夫变换的局限性,如计算复杂度高,对噪声敏感等问题,研究者们提出了多种改进策略。

文献[17]提出了基于梯度方向约束的随机圆检测算法,该算法首先随机采样 3 个边缘点,然后通过计算圆上两条弦的中垂线交点与这 3 个点的梯度线的距离,作为评估候选圆有效性的依据。这在一定程度上提升了计算效率,节省了所需的内存空间,但这类基于随机采样的方法不具备可重复性。文献[18]提出了融合连通域和霍夫梯度技术的纱笼坐标检测方法,利用霍夫梯度法确定圆心位置,判断圆心是否位于求取的连通域内,进而评估纱杆损害情况并确定其坐标位置。该方法提高了对背景噪声的抗干扰能力,但完成一次圆检测功能长达 6 min,检测速度成为应用的一个问题。文献[19]提出了一种基于改进 Hough 变换的圆形防松垫片安装检测技术,该方法利用圆的梯度特性和垫片半径先验知识,有效减少了计算量,但检测效果依赖于半径参数的设定,且计算效率不高。

针对上述方法局限性,以及梯度在圆检测过程中的关

键作用,本文对基于霍夫梯度的传统圆检测方法进行了改进。本文仍使用梯度获取圆,但不同于传统的投票机制,提出了八点判断法,同步检测圆心和半径。针对圆心半径检测的高度并行化特性,利用 CUDA 技术对其进行了并行化设计,克服多线程中实现负载均衡和数据同步问题,将计算迁移到 GPU,同时协调 CPU 和 GPU 之间的数据传输,实现圆检测的并行化,提高检测速度。在圆检测过程中,如何提升候选圆的准确性以及筛选最准确的圆,也是影响实验结果的关键性问题。为此提出了三维连通域的候选圆的再搜索策略,并对最优圆的选取算法进行了改进。

## 2 基于霍夫梯度的同步并行圆检测

### 2.1 边缘筛选

图 1 是本文提出的基于霍夫梯度的同步并行圆检测算法流程图。图像噪声不仅增加了图像处理的复杂性,同时还降低了圆检测的准确性<sup>[20]</sup>。因此,在边缘检测前引入高斯滤波,以保留图像边缘细节的同时抑制其噪声。之后,通过 Canny 算法检测边缘,通过连通域搜索算法得到边缘连通域,并通过下述过程去除对圆检测贡献小的边缘连通域。

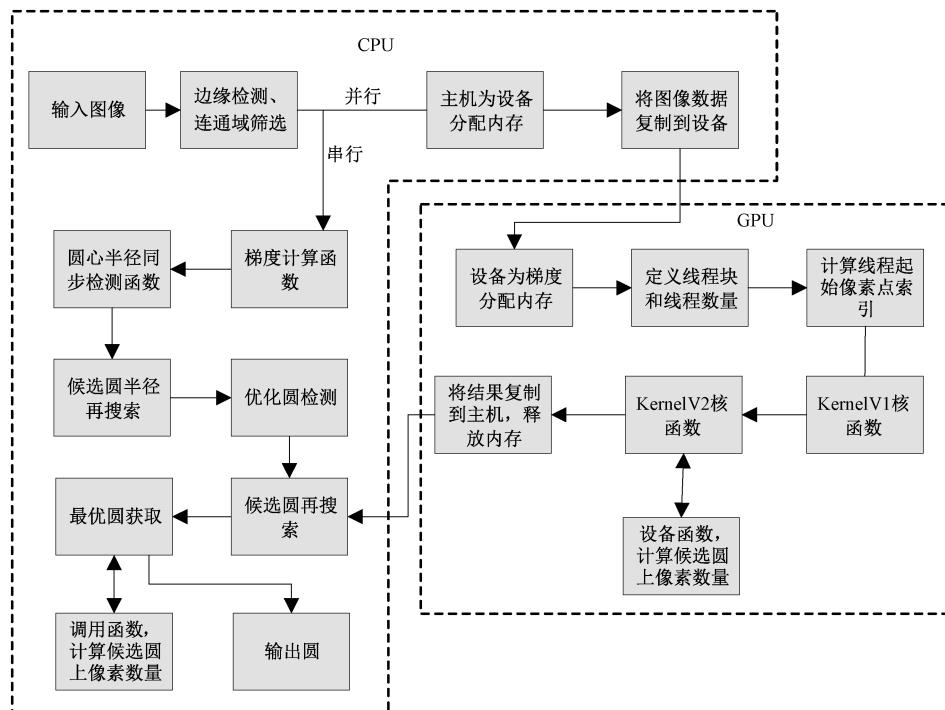


图 1 SPGHT 算法流程  
Fig. 1 SPGHT algorithm flowchart

1) 连通域筛选:去除图像中像素数量少的边缘连通域。基于文献[21]与本文的研究,针对尺寸为 512 的 PCB 图像,圆的最大半径为 256,本文巧妙的选取 256 的平方根,即 16,作为区分显著和不显著连通域的像素数量阈值,通过去除不显著连通域,加快后续的圆检测速度。

2) 边缘线删减:删除边缘图像中直线段、曲率小的曲线。通过计算其端点  $s_1, s_2$  和中点构成的三角形  $\triangle s_1 s_2 s_m$  面积与边缘线长度  $L$  的比值  $\triangle s_1 s_2 s_m / L$ ,该比值结果依赖于圆心角  $\theta$  和半径  $r$ ,对于较大的比值结果则表示着更弯曲的边缘段,删除比值小于预设阈值  $T_a = 1.8$  的边缘段。

## 2.2 改进的候选圆检测

本文提出圆心半径同步检测方法检测候选圆,同时加入了圆心位置约束的策略。详细步骤如下:

### 1) 边缘梯度计算

根据边缘图像  $E$ ,通过上述式(1)~(3)计算得到边缘图像梯度和梯度方向  $\theta$ 。

### 2) 圆心半径同步检测

遍历图像边缘像素点,并针对每个像素点,沿着其梯度方向逐一检查梯度线上的像素否是潜在圆心。如图 2(a)所示,对于边缘点  $M$ ,当其梯度线经过点  $N(x_0, y_0)$  时,通过点  $N$  和边缘点  $M$  得到潜在候选圆的圆心  $N(x_0, y_0)$  和半径  $r$ ,并由八点法判定其是否为潜在候选圆。具体如下:把八点分为两组:如图 2(b)所示,第 1 组点为潜在候选圆的上、下、左、右像素点,分别记为  $P_U, P_D, P_L, P_R$ ;第 2 组为潜在候选圆的 4 个斜对角像素点,分别记为  $P_{LU}, P_{LD}, P_{RU}, P_{RD}$ ,像素坐标计算均以图像左上角为坐标系原点。规则如下:

$$C_1(P_U, P_D, P_L, P_R) \geqslant 1 \quad (4)$$

$$C_2(P_{LU}, P_{LD}, P_{RU}, P_{RD}) \geqslant 1 \quad (5)$$

$$C_1(P_U, P_D, P_L, P_R) + C_2(P_{LU}, P_{LD}, P_{RU}, P_{RD}) \geqslant 3 \quad (6)$$

式中:  $C_*(\cdot, \cdot, \cdot)$  为该组中满足梯度指向圆心的个数。统计边缘点在该圆中的数量  $nr(x, y, r)$ ,计算边缘点密度  $n_r(x, y, r)/(r \times (C_1 + C_2))$ ,将结果大于阈值  $T_c = 1.85$  的圆作为潜在候选圆。

### 3) 候选圆半径再搜索

在检测潜在候选圆的基础上,进一步搜索候选圆的同心圆,半径分别为  $r - 1$  和  $r + 1$ ,并统计边缘点数量  $n_{r-1}(x_0, y_0, r-1), n_r(x_0, y_0, r), n_{r+1}(x_0, y_0, r+1)$ ,将边缘点数量最多的潜在候选圆添加到候选圆队列。

### 4) 圆心位置约束

通过下述策略避免在不可能形成完整圆的区域进行无效检测:

(1) 边界检查:采用下述方法对潜在圆心判定前执行边界检查。

$$r = \min(x_0, w - x_0, y_0, h - y_0) \quad (7)$$

式中:  $x_0, y_0$  为潜在圆心  $x$  和  $y$  坐标,  $h$  和  $w$  为图像的高和宽,  $r$  该潜在候选圆的半径。把满足式(7)的潜在圆心点记作临界点。如图 2(c)所示,以  $N_1$  为圆心的潜在候选圆与图像边界重合,记  $N_1$  为临界点。

(2) 梯度线截断:终止梯度线临界点以下的潜在圆心的搜索。如图 2(c)所示,梯度线上  $N_1$  临界点后的虚线部分不进行圆心检测。

## 2.3 改进的候选圆再搜索

由于像素的离散化和图像边缘呈现的模糊特性,检测圆心可能会偏离真实圆心位置。采用基于三维坐标空间的方法降低真实圆心的漏检,具体如下:将检测到的候选

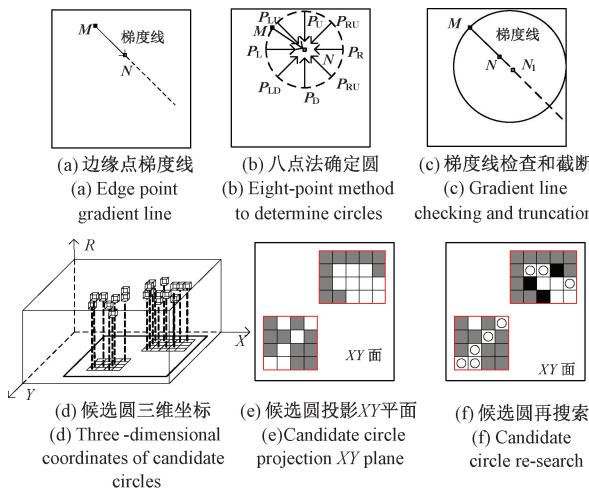


图 2 候选圆检测过程

Fig. 2 Candidate circle detection process

圆  $(x, y, r)$  映射到三维坐标系中,并把这些坐标点的值设为 1(表示在该点处存在候选圆)。通过搜索三维连通域,将圆心位置和半径大小相近的候选圆划分到同一个三维连通域,如图 2(d)所示,形成两个连通域。

针对每个连通域,将三维坐标系中的点投影到 XY 平面,如图 2(e)所示,其中灰块标记点代表每个候选圆的圆心坐标位置。在上述候选圆的周围进一步搜索,规则如下:首先确定每个连通域的最小外接矩形,图 2(e)中红色矩形即为两个连通域对应的外接矩形;然后在连通域中,遍历每个候选圆心的 4-邻域像素并把 4-邻域像素位于最小外接矩形内的像素位置作为新圆心,把该连通域内候选圆的半径集作为新半径,生成新的候选圆,如图 2(f)所示,黑块标记和“○”标记点是新的候选圆;最后计算每个新的候选圆边缘点的数量与其半径比值  $nr(x, y, r)/r$ ,保留比值大于阈值  $T_d$  的新的候选圆,舍弃小于阈值的新的候选圆。“○”标记为新的候选圆,黑块标记为舍弃圆。

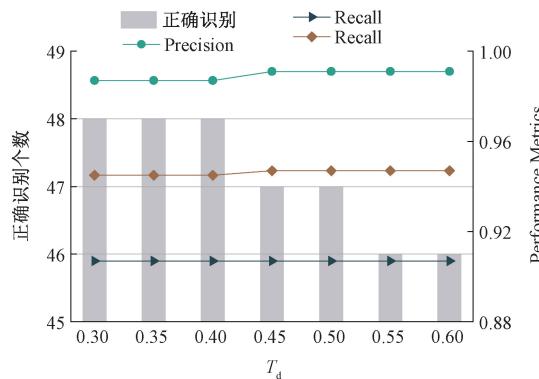
## 2.4 改进的最优圆获取

在检测真实圆的过程中得到的多个检测结果。采用下述方法保留最优圆:对于圆  $a$  和圆  $b$ ,圆心分别为  $Oa$  和  $Ob$ 。

$$D = D_1 + D_2 \quad (8)$$

式中:  $D_1$  为圆  $a$  和圆  $b$  圆心距离的平方,  $D_2$  是两圆半径差的平方。

当  $D < T_d$  时,则认为两圆表示同一个真实圆,将统计圆上边缘点数量,并把边缘点数量较多的圆视为最优圆。图 3 展示了阈值  $T_d$  对两组实验检测结果的影响,分别为 MINI 图像的正确识别数量和 PCB 图像的精确率、召回率和  $F$  值。当  $T_d = 40$  时,正确识别数量为 48,当  $T_d = 45$  时,正确识别数量为 47,其他指标出现上升,当  $T_d = 55$  时,正确识别数量为 46,其他指标保持不变。根据分析得知,该阈值大小对半径相近的同心小圆产生更大影响,导致两圆被认为同一个圆。因此,结合分析数据,令  $T_d = 45$ 。

图 3 阈值  $T_d$  的影响Fig. 3 The influence of threshold  $T_d$ 

## 2.5 基于 CUDA 的候选圆并行检测

为加速算法在检测大尺寸、多像素图像时的检测速度,通过引入 CPU 和 GPU 的异步架构处理优化。具体来说,主机负责图像的预处理和最终处理工作,设备负责多线程并行化候选圆检测,通过 CPU-GPU 协同工作,能够显著提升圆的检测速度。

在 CUDA 计算模型中,通过分配线程网格和线程块,把圆检测任务拆解成多个小任务,每个线程负责计算对应的像素块。线程网格由多个线程块组成,而线程块内包含多个线程。在设备端,通过定义 KernelV1 和 KernelV2 核函数实现候选圆的检测功能,其中 KernelV1 负责并行化图像梯度计算,KernelV2 负责并行化圆心半径检测,并通过 CUDA 流控制核函数顺序执行。在 KernelV1 核函数中,每个线程遍历像素块,计算出像素点的梯度,并将梯度保存到全局内存。在等待 KernelV1 的所有线程执行完毕后,运行 KernelV2 核函数。在 KernelV2 中,每个线程从全局区获取不为 0 的像素点的梯度,并基于这些梯度执行候选圆的检测,待每个线程检测后,将检测到的候选圆拷贝回主机端,主机端随后执行下一步的判断和验证步骤。图 4 展示了算法的并行化设计,可将这种并行化设计封装为易于调用的包,引入其他设备使其具备并行计算提高图像处理效率的能力。

## 3 实验结果与讨论

为验证算法的有效性,本文选取 MINI 图像<sup>[14]</sup>、PCB 图像<sup>[14]</sup>和新能源汽车快充接口图像进行实验。由于其他潜在的基准方法没有提供源码或者无法直接应用于本文检测中,故选取当前圆检测领域中 4 种具有代表性的最新算法 GHT、CACD、RCD 和 Zhao 等<sup>[14]</sup>来进行对比分析<sup>[22]</sup>。具体为文献[6]提出的 GHT 算法,其作为经典的圆检测方法,通过梯度有效减少了参数空间搜索的复杂度;文献[12]提出的 CACD 算法,CACD 算法是 GHT 的改进版本;文献[10]提出的 RCD 算法,不同于投票机制,RCD 通过随机采样策略并验证的方式获得候选圆;Zhao 等<sup>[14]</sup>

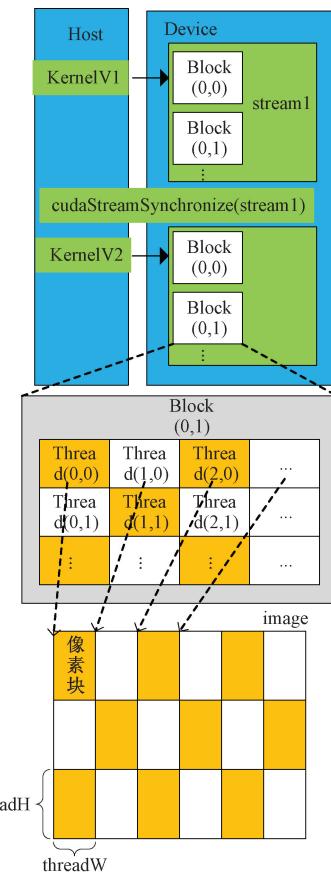


图 4 CUDA 并行运算结构

Fig. 4 CUDA parallel computing architecture

提出的基于圆内切三角形的圆检测算法。计算机配置为 Intel ® Core™ i5-8265U CPU @ 1.60 GHz × 8 处理器和 NVIDIA Geforce MX150 图形处理器。

### 3.1 评价指标

对于检测结果,采用常用的圆检测结果计算方法,分别为 Precision、Recall、F-score 和单张图像的平均处理时间(t/s),具体如下:

$$\text{Precision} = \frac{|TP|}{|TP + FP|} \quad (9)$$

$$\text{Recall} = \frac{|TP|}{|TP + FN|} \quad (10)$$

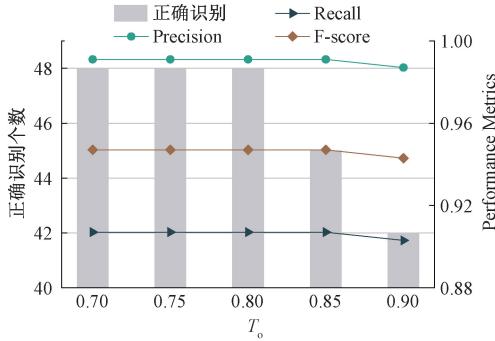
$$F\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

为衡量候选圆的准确性,本文采用文献[21-22]提出的候选圆和真实圆的重叠率作为其评价指标。

$$\text{OverlapRatio} = \frac{s(c1) \cap s(c2)}{s(c1) \cup s(c2)} \quad (12)$$

式中: $s(c*)$  代表的圆  $c*$  的面积,  $c1$  为检测圆,  $c2$  为标定圆。当  $\text{OverlapRatio} \geq T$  时,此情况被定义为正阳性实例。在此基础上,进一步检测结果的几何匹配度,计算正阳性圆与标定圆之间的多个关键指标,包括重叠率(OvlR)、圆心距( $\Delta O/\text{pixel}$ )和半径差其他( $\Delta R/\text{pixel}$ )。

图5展示了阈值  $T_o$  对检测结果的影响,当  $T_o$  越大时,要求检测结果越严格,圆的正确识别数和评价指标均在降低,当  $T_o = 0.9$  时,各指标出现了明显的下降,在  $T_o < 0.85$  时,各指标保持稳定。因此,结合上述阈值  $T_d$  取值,令  $T_o = 0.8$ 。

图5 阈值  $T_o$  的影响Fig. 5 The influence of threshold  $T_o$ .

### 3.2 MINI 图像对比实验

如图6所示,MINI图像包含10张涵盖多种复杂的圆检测的图像,以验证圆检测算法的有效性,图7是各种算法在MINI图像上的圆检测结果。从表1中看出本文方法正确识别数量仅次于Zhao等<sup>[14]</sup>的方法,检测时间相较于RCD和CACD短,但相较于其他算法长,圆的重叠率和半径差表现优异,采用CUDA加速后获得高达16倍的加速比,这验证了本文算法在多种复杂图像中的有效性以及较高的检测准确率。

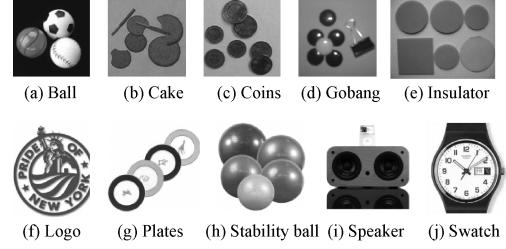


图6 MINI 图像

Fig. 6 MINI images

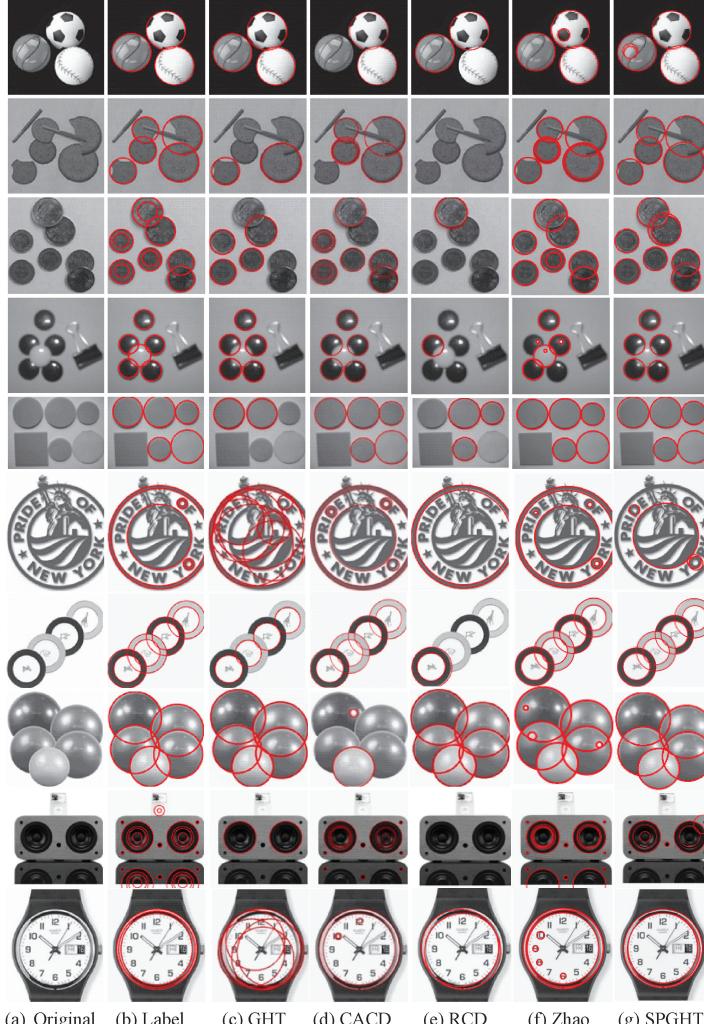


图7 不同算法在 MINI 图像上的检测结果

Fig. 7 Detection results of different algorithms on MINI images

**表 1 不同算法在 MINI 图像上的检测结果**  
**Table 1 Detection results of different algorithms**  
**on MINI images**

| 算法    | 正确识别 | OvlR  | $\Delta O$ | $\Delta R$ | Time  |
|-------|------|-------|------------|------------|-------|
| GHT   | 29   | 0.927 | 1.862      | 1.241      | 0.006 |
| CACD  | 46   | 0.933 | 0.582      | 0.647      | 0.348 |
| RCD   | 18   | 0.939 | 2.105      | 1.722      | 2.566 |
| Zhao  | 52   | 0.941 | 1.253      | 1.570      | 0.087 |
| SGHT  | 47   | 0.940 | 1.589      | 0.826      | 4.162 |
| SPGHT | 47   | 0.940 | 1.589      | 0.826      | 0.255 |

### 3.3 PCB 图像对比实验

PCB 图像包含 100 张尺寸各异的图像, 其中, 97 张尺寸为  $512 \times 456$ , 其余 3 张尺寸各不相同, 每张图像包含 1~4 个圆。为进一步测试算法效率及并行处理能力, 本文对 PCB 图像进行预处理, 将所有图像尺寸裁切至  $512 \times 456$ , 图 8 是各种算法在 PCB 图像上的圆检测结果。从表 2 中可以看出, 本文提出的方法展现了优秀的性能表现。具体而言, 其精确率、召回率和 F 值分别为 99.1%、90.7% 和 94.7%, 验证了候选圆再搜索和圆心位置约束在圆检测过

程中起到的关键作用。此外, 通过候选圆半径再搜索, 使得半径误差相较于其他算法小。更值得一提的是, 通过将多张相同尺寸的图像进行并行检测, 算法执行效率实现高达 26 倍的显著提升。与 MINI 图像的处理相比, 这一加速效果更为显著, 从而进一步提升了算法的整体性能和效率。尽管在圆心距的表现上, 本文算法表现劣于其他检测算法, 但本文算法依然展现了低误检率和高准确率的优异特性。检测时间的缩短和加速比的提升也验证了并行检测对提升圆检测效率的关键作用。

**表 2 不同算法在 PCB 图像上的检测结果**  
**Table 2 Detection results of different algorithms**  
**on PCB images**

| 算法    | P     | R     | F     | OvlR  | $\Delta O$ | $\Delta R$ |
|-------|-------|-------|-------|-------|------------|------------|
| GHT   | 0.904 | 0.482 | 0.629 | 0.967 | 1.639      | 0.982      |
| CACD  | 0.622 | 0.822 | 0.708 | 0.955 | 0.782      | 0.674      |
| RCD   | 0.717 | 0.814 | 0.763 | 0.969 | 1.420      | 0.801      |
| Zhao  | 0.841 | 0.968 | 0.900 | 0.959 | 1.473      | 1.461      |
| SGHT  | 0.991 | 0.907 | 0.947 | 0.964 | 2.026      | 0.404      |
| SPGHT | 0.991 | 0.907 | 0.947 | 0.964 | 2.026      | 0.404      |

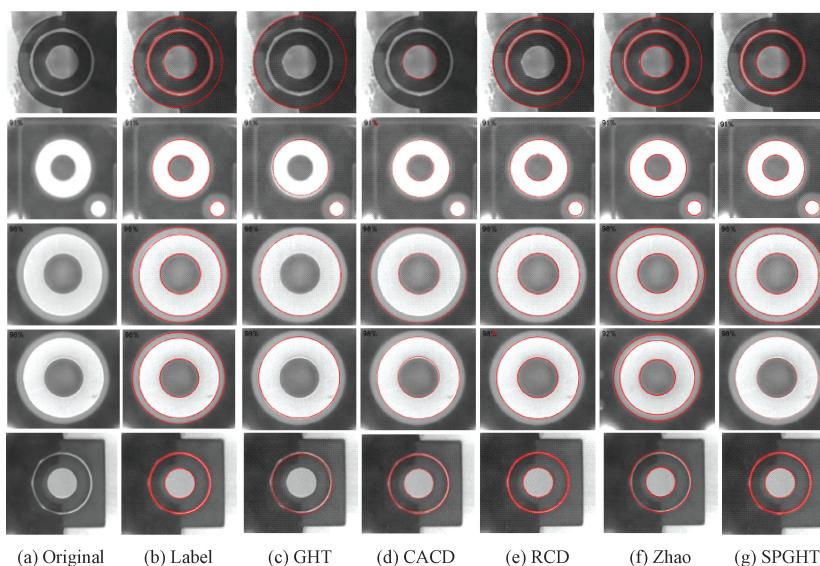


图 8 不同算法在 PCB 图像上的检测结果

Fig. 8 Detection results of different algorithms on PCB images

### 3.4 汽车快充接口图像对比实验

图 9 是汽车快充接口图像, 并对该充电口从不同视角拍摄的图像进行圆检测, 图像尺寸为  $640 \times 480$ 。图 10 展示了各种算法在快充接口上的圆检测结果, 结果均截取于检测完成后图像的充电口部分。从图 10 得知, 在处理复杂背景的图像时, GHT 算法和 RCD 算法丢失了许多圆形, 其中 RCD 由于采用随机采样策略, 检测到更多的干扰圆, 且检测时间长, 平均为 4.79 s/张, GHT 则为 0.014 s/张;

CACD 算法在检测弱边缘圆形时表现出更好的检测效果, 但该方法检测速度较慢, 平均为 3.51 s/张, 由于 CACD 对边缘检测的准确性依赖高, 对模糊边缘的圆检测不如 Zhao 等<sup>[14]</sup>算法; 相比之下, Zhao 等<sup>[14]</sup>算法对小圆的检测能力更强, 且检测速度快, 达到 0.213 s/张, 但对非小圆会产生误检。本文算法的平均检测速度为 0.464 s/张, 具有稳定的圆检测性能, 能够准确提取充电口中的关键圆形特征, 从而实现对充电口位置的精确定位, 有效应对复杂

背景下的圆检测任务。



图 9 汽车快充接口图像

Fig. 9 Car fast charging port image



图 10 不同算法在汽车快充接口图像上的检测结果

Fig. 10 Detection results of different algorithms on the car fast charging port images

### 3.5 干扰性测试实验

在 Plates 和 stability ball 两张图像上引入不同程度的

高斯噪声和椒盐噪声验证算法的抗干扰性能。高斯噪声的均值被设定为 0, 其方差则分为 10%、20%、30% 和 40% 多个级别; 椒盐噪声按照每张图像的总像素随机生成 2%、5%、8%、和 10% 数量的噪声。在这两幅图像中, 共包含 13 个圆形目标。图 11 和图 12 展示了本文算法在两种噪声下的检测结果, 从表 3 中看出, 高斯噪声为 10% 时, 该算法检测到全部的圆, 在椒盐噪声为 2% 时, 算法检测到 12 个圆; 在高斯噪声为 30% 时, 检测到 6 个圆, 在椒盐噪声为 8% 时, 检测到 4 个圆, 验证了算法在噪声下的鲁棒性和抗干扰能力。

综上所述, 实验验证了本文算法的有效性以及并行检测的高效性, 为圆检测领域提供了一种既高效又可靠的解决方案, 具有广阔的应用前景。

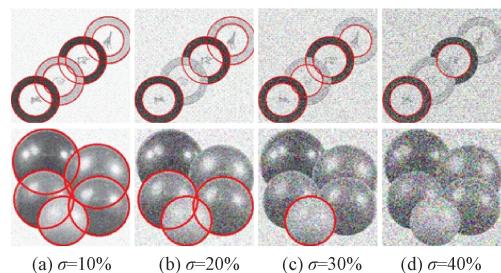


图 11 SPGHT 算法在不同高斯噪声级别的检测结果

Fig. 11 Detection results of SPGHT algorithm under different Gaussian noise levels

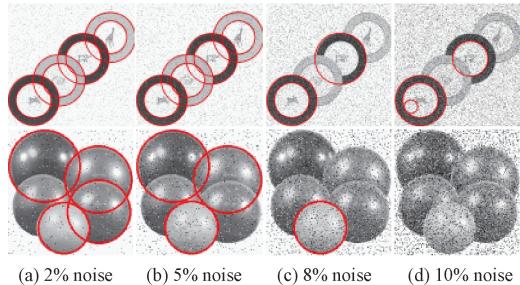


图 12 SPGHT 算法在不同椒盐噪声级别的检测结果

Fig. 12 Detection results of SPGHT algorithm under different salt and pepper noise levels

表 3 SPGHT 算法在噪声下的检测结果

Table 3 Detection results of SPGHT algorithm under noise

| 检测结果  | 高斯噪声 |     |     |     | 椒盐噪声 |    |    |     |
|-------|------|-----|-----|-----|------|----|----|-----|
|       | 10%  | 20% | 30% | 40% | 2%   | 5% | 8% | 10% |
| 正识别个数 | 13   | 7   | 6   | 1   | 12   | 11 | 4  | 3   |

## 4 结论

本文提出了一种基于霍夫梯度的同步并行圆检测方法。相较于传统霍夫梯度圆检测方法, 该方法能够去除边缘图像中的干扰信息, 实现了圆心半径的同步检测。通过

引入半径再搜索和圆心位置约束, 进一步提升了圆检测的精度和效率; 通过候选圆再搜索和最优圆获取有效减少候选圆的漏检概率和去除了干扰圆。实验测试表明, 该方法在终端机器人视觉图像上实现了高精度的圆检测性能, 其精确率、召回率和 F 值为 99.1%、90.7% 和 94.7%, 单张图

像的平均检测耗时为 0.09 s,使其在复杂图像中的圆检测性能显著提升且具备更好的抗噪能力。此外,该算法利用 CUDA 并行处理技术优化了圆心半径同步检测,实现了最高 26 倍的加速比,显著提升了圆检测效率。该功能可封装为易于调用的包,便于类似算法调用和移植,以加速其图像处理效率,为复杂场景中的圆检测和工业图像的批量处理提供有效的技术支持。

## 参考文献

- [1] MUKHOPADHYAY P, CHAUDHURI B B. A survey of Hough transform[J]. Pattern Recognition, 2015, 48(3): 993-1010.
- [2] BERKAYA S K, GUNDUZ H, OZSEN O, et al. On circular traffic sign detection and recognition [J]. Expert Systems with Applications, 2016, 48: 67-75.
- [3] 陈励威,刘苏苏,袁辉,等.融合装配特征和回归分析的锯链图像分割算法[J].电子测量技术,2022,45(23):139-146.  
CHEN L W, LIU S S, YUAN H, et al. Saw chain image segmentation algorithm fusion assembly features and regression analysis [J]. Electronic Measurement Technology, 2022,45(23):139-146.
- [4] YE H SH, SHANG G C, WANG L N, et al. A new method based on Hough transform for quick line and circle detection[C]. 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI), IEEE, 2015: 52-56.
- [5] LIN EN Y, TU C T, LIEN J J J. Nut geometry inspection using improved Hough line and circle methods[J]. Sensors, 2023, 23(8): 3961.
- [6] 顾忠庆,苏卫华,陈旭义.基于 Hough 变换圆检测的颅脑穿刺靶标信息提取算法对比研究[J].军事医学,2023,47(12):934-941.  
GU ZH Q, SU W H, CHEN X Y. Comparative study of algorithms for information retrieval of brain puncture targets based on Hough transform circle detection[J]. Military Medical Sciences, 2023, 47 (12): 934-941.
- [7] 李艳荻,徐熙平,钟岩.特征弦约束随机 Hough 变换在椭圆检测中的应用[J].仪器仪表学报,2017,38(1):50-56.  
LI Y D, XU X P, ZHONG Y. Application of RHT based on character string constraint in ellipse detection [J]. Chinese Journal of Scientific Instrument, 2017,38 (1):50-56.
- [8] 章飘艳,陈仁文,朱霞,等.改进的 RHT 算法在弹簧卡箍检测中的应用[J].计算机应用与软件,2016,33(2):280-283.  
ZHANG P Y, CHEN R W, ZHU X, et al. Applying improved RHT algorithm to spring clamps detection [J]. Computer Applications and Software, 2016, 33 (2):280-283.
- [9] JIANG L Y. A fast and accurate circle detection algorithm based on random sampling [J]. Future Generation Computer Systems, 2021, 123: 245-256.
- [10] CHEN T C, CHUNG K L. An efficient randomized algorithm for detecting circles[J]. Computer Vision and Image Understanding, 2001, 83(2): 172-191.
- [11] 邓仕超,高阳,韩海媚.改进的 RANSAC 圆检测算法[J].郑州大学学报(理学版),2018, 50(1): 77-81.  
DENG SH CH, GAO Y, HAN H M. Improved RANSAC algorithm for circle detection[J]. Journal of Zhengzhou University ( Natural Science Edition ), 2018, 50(1): 77-81.
- [12] YAO ZH J, YI W D. Curvature aided Hough transform for circle detection[J]. Expert Systems with Applications, 2016, 51: 26-33.
- [13] LIU Y, DENG H G, ZHANG Z Y, et al. A fast circle detector with efficient arc extraction [J]. Symmetry, 2022, 14(4): 734.
- [14] ZHAO M Y, JIA X H, YAN D M. An occlusion-resistant circle detector using inscribed triangles[J]. Pattern Recognition, 2021, 109: 107588.
- [15] XU Y, FANG B, WU X G, et al. Research and implementation of parallel lane detection algorithm based on GPU[C]. 2017 International Conference on Security, Pattern Analysis, and Cybernetics(SPAC), IEEE, 2017: 351-355.
- [16] 熊超,王欣,王鑫杰,等.基于 CUDA 的航空  $\gamma$  能谱数据小波降噪并行加速算法[J].核技术,2024,47(4):23-33.  
XIONG CH, WANG X, WANG X J, et al. CUDA-based parallel acceleration algorithm for wavelet denoising of airborne  $\gamma$ -ray spectrometry data [J]. Nuclear Techniques, 2024,47(4):23-33.
- [17] 宁小鸽,牟莉.基于梯度方向约束的瞳孔定位算法研究[J].国外电子测量技术,2021,40(7):115-121.  
NING X G, MU L. Research on pupil localization algorithm based on gradient direction constraint[J]. Foreign Electronic Measurement Technology, 2021,40(7):115-121.
- [18] 冉宇辰,黄民,王文胜,等.一种改进霍夫梯度变换的纱笼坐标检测方法研究[J].机械设计与制造,2022(10):94-99.  
RAN Y CH, HUANG M, WANG W SH, et al. Research on an improved Hough gradient transform method for sarong coordinate detection[J]. Machinery Design & Manufacture, 2022(10):94-99.

- [19] 崔志勋,张仁杰. 基于改进 Hough 变换的圆形防松垫片安装检测[J]. 电子测量技术,2018,41(22):95-98.  
CUI ZH X, ZHANG R J. Detection of round lock washer mounting based on improved Hough transform [J]. Electronic Measurement Technology, 2018, 41 (22):95-98.
- [20] 朱正伟,宋文浩,焦竹青,等. 基于随机 Hough 变换改进的快速圆检测算法[J]. 计算机工程与设计,2018,39(7): 1978-1983.  
ZHU ZH W, SONG W H, JIAO ZH Q, et al. Fast circle detection algorithm based on improved randomized Hough transform [J]. Computer Engineering and Design, 2018, 39(7): 1978-1983.
- [21] JIA Q, FAN X, LUO ZH X, et al. A fast ellipse detector using projective invariant pruning[J]. IEEE Transactions on Image Processing, 2017, 26 (8): 3665-3679.
- [22] XU X G, YANG R G, WANG N G. A robust circle detector with regionalized radius aid [J]. Pattern Recognition, 2024, 149: 110256.

### 作者简介

苗少杰,硕士研究生,主要研究方向为智能控制。  
E-mail:17633539220@163.com

李宪(通信作者),助理教授,博士,主要研究方向为复杂系统建模、优化控制、无人系统和数据挖掘。  
E-mail:lixian@qdu.edu.cn