

DOI: 10.13382/j.jemi.B2104248

基于 YOLOv4-tiny 和完整度排序的 标志物快速圆检测算法*

丁伟利^{1,2} 郝增号¹

(1. 燕山大学电气工程学院 秦皇岛 066004; 2. 智能控制系统与智能装备教育部工程中心 秦皇岛 066004)

摘要:针对现有圆检测算法在高分辨率标志物图像中检测效率低的问题,提出了一种新型标志物快速圆检测算法。首先通过 YOLOv4-tiny 算法快速定位图像中的标志物,滤除背景干扰,然后利用提出的弧段完整度排序方法从标志物中筛选出不同完整度的候选圆,最后通过距离和方向角约束筛选并拟合得到所有的标志物圆。基于高分辨率标志物图像数据集进行了实验,实验结果表明算法具有极快的检测速度,在检测精度基本不变时,算法的检测时间只有 EDcircle 算法的 11%,AAMED 算法的 5.0%,为需要快速定位标志物的视觉定位、标定等任务提供了一种高效的解决方案。

关键词: YOLO; 圆检测; 标志物; 完整度排序; 目标检测

中图分类号: TP391.4; TN98 **文献标识码:** A **国家标准学科分类代码:** 510.4050

Fast circle detection algorithm for markers based on YOLOv4-tiny and completeness ranking

Ding Weili^{1,2} Hao Zenghao¹

(1. Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China;

2. Engineering Research Center of Intelligent Control System and Intelligent
Equipment, Ministry of Education, Qinhuangdao 066004, China)

Abstract: Aiming at the low detection efficiency of existing circle detection algorithms in high-resolution marker images, this paper proposes a new type of rapid circle detection algorithm for markers. First, the YOLOv4-tiny algorithm is used to quickly locate the markers in the image to filter out the background interference, and then use the proposed arc completeness sorting method to screen out candidate circles with different completeness from the markers, and finally filter through the distance and direction angle constraints. And fit all the marker circles. This paper conducts experiments based on high-resolution marker images. The experimental results show that this algorithm has extremely fast detection speed. When the detection accuracy is basically unchanged, the detection time of this algorithm is only 11% of the EDcircle algorithm and 5.0% of the AAMED algorithm. It provides an efficient solution for tasks such as visual positioning and calibration that require rapid positioning of markers.

Keywords: YOLO; circle detection; markers; completeness ranking; object detection

0 引言

基于机器视觉的定位算法是智能移动机器人领域内的热点问题。特别是在工业机器人领域,基于标志物的视觉测量^[1]一直是机器人视觉引导最稳定可靠的手段之一。在众多标志物中,圆形组成的标志物易于辨别、鲁棒性高,因此圆形标志物在机器视觉领域极其常见。图像

中圆形标志物检测的速度和准确度将直接影响后续的视觉定位效果。检测圆形标志物常用方法是利用圆检测算法先提取出所有的圆形,然后利用这些圆形已知的属性从检测到的圆形中筛选出属于标志物中的圆。现有圆检测算法主要分为 3 类,为基于 Hough 变换的圆检测算法^[2-4]、基于弧段提取的圆检测算法^[5-7]和基于深度学习的圆形检测算法。

标准的 Hough 变换^[2]圆检测将图像点从图像空间变

收稿日期: 2021-05-04 Received Date: 2021-05-04

* 基金项目: 国家重点研发计划(2018YFB1308302, 2018YFB1308300)、国家自然科学基金(62073279)项目资助

换到参数空间,通过设置累加器对参数进行累加来得到圆的参数;随机 Hough 变换(RHT)^[3]与迭代 RHT 方法(IRHT)^[4]通过随机采样的方式提高了 Hough 变换方法的检测速度。基于弧段提取的椭圆检测算法是通过几何或者数学的方法检测边缘线段中属于同一个圆的集合并进行验证。Meng 等^[7]通过利用弧段邻接矩阵 AAM 来实现属于同一椭圆的圆弧的快速匹配,进而提高了椭圆拟合的速度。基于深度学习的方法也可以用于检测圆,如利用 Mask RCNN^[8]等分割网络直接从图像中直接分割出圆再拟合得到圆心坐标^[9],或者利用目标检测网络通过检测框^[10-14]的方式检测出目标圆的位置。

上述算法普遍针对的是通用场景,并没有很好的利用标志物的已有性质。因此当图像分辨率提高时,基于 Hough 变换和基于弧段提取的检测方法会产生存在速度慢与内存消耗大等问题;深度学习的方法可以实现较快的检测速度,但高分辨率图像输入网络中进行检测通常需要压缩图像尺寸,这会引入误差,不进行压缩尺寸的网络又会导致模型参数量急剧增大,这些问题导致基于深度学习的方法很难同时满足精度和速度的要求。

针对此问题,本文提出了一种新型快速圆检测算法。为了利用标志物图像的显著性特征,首先通过 YOLOv4-tiny^[15]算法提取出标志物的感兴趣区域(region of interest, ROI),其次利用 EDdrawing^[16-17]算法进行边缘检测,将检测到的边缘线段按曲率分开后使用文献[18]的方法得到平滑的弧段;然后提出了针对弧段的完整度验证法则,对所有的弧段先进行完整度验证,并对不同完整度的弧段采用不同的连接策略得到所有的候选圆,最后对椭圆集进行聚类与二次验证得到最终的检测结果。由于本算法采用了粗提取加精检测的策略,很好地利用了标志的已有性质,因此可以极大地提高检测速度。

1 基于 YOLOv4-tiny 和完整度排序的标志物快速圆检测算法

1.1 算法流程

本文提到的标志物为相机标定板或者视觉高精度定位过程中使用的由圆构成的显著性标志,标志物如图 1(a)所示,这些标志物的图像通过高分辨率相机拍摄获得,标志物通常自身的内部结构并不复杂,但周围背景存在较高的复杂性或不确定性。为优化此类场景下圆检测算法的效率,本文提出的圆检测算法分如下 5 步。

- 1) 使用 YOLOv4-tiny 检测标志物,提取出标志物的 ROI 区域;
- 2) 在 ROI 区域中检测边缘并提取出所有的弧段;
- 3) 计算每条弧段的完整度并将所有的弧段按照计算得到的完整度分为两部分,一部分为完整度较高的主弧,

另一部分为短弧;

- 4) 不同完整度的弧段分策略连接得到候选圆;
- 5) 对所有候选圆进行聚类和验证得到最终的输出。

1.2 YOLOv4-tiny 标志物 ROI 提取

在进行圆的检测时,通常会对全图进行边缘检测。但进行全图的边缘检测会产生大量的背景边缘,背景边缘的提取不仅会浪费大量的提取时间,而且会严重影响后续从边缘中获取圆的效率,这个问题在高分辨率或者背景复杂的图像中尤为突出。因此对于这类图像,本文先从图像中提取标志物的 ROI 区域^[19],然后只在 ROI 中进行圆检测,以降低误检率并提高检测速度。

为了提高 ROI 提取过程的效率,采用 YOLOv4-tiny 作为 ROI 提取的检测模型。基于 Yolo 的检测方法^[20]是目标检测领域效率最高的检测算法之一,其中 YOLOv4-tiny 是专为低端 GPU 而设计的一种架构,模型使用了 OSANet 作为整体结构,然后利用 CSPNet 对模型进行了优化得到了 CSPOSANet。YOLOv4-tiny 在优化模型计算量的同时也对模型对外围资源如内存访问带宽等进行了限制,其在 RTX 2080Ti 显卡上结合 TensorRT+FP16 等技术,可以达到惊人的 1774FPS@ batch = 4。由于 YOLOv4-tiny 这些特点使其能够在硬件条件并不是很高的情况下依旧获得很高的检测速度和较高的检测精度,可以满足本文高速提取 ROI 区域的要求,因此本文选用 YOLOv4-tiny 作为 ROI 提取模型。

模型训练时需要选取各种标志物的图片,并在不同背景、不同光照下对这些图片进行数据采集作为训练集。采集到图像后对图片进行手动标注,然后使用 K-means 算法聚类得到 6 种预选框,并将预选框尺寸写入 YOLOv4-tiny 的配置文件,使模型能够在此数据集上获得好的效果。

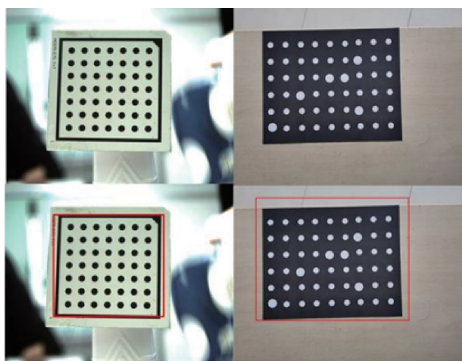
设待检测的图片为 I , 图像原始的宽高为 W 和 H , 将图片缩放到 416×416 大小然后输入到 YOLOv4-tiny 中。每张图片输出 n 个检测框,每个检测框的信息包括置信度得分 s_i 、检测框中心的横坐标位置 c_{xi} 、纵坐标位置 c_{yi} 、检测框的宽 w_i 、检测框的高 h_i 。

设模型检测框的置信度得分阈值为 t , 去掉 $s_i > t$ 的检测框,然后对剩余的检测框使用非极大值抑制,去掉重叠的检测框,此时最终检测框数量为 m 。由于模型输出的信息被归一化到了 0~1, 则第 i 个 ROI 区域的宽 W_i 和高 H_i 的计算方式如式(1)所示,左上角坐标 (x_{ai}, y_{ai}) 的计算如式(2)所示。

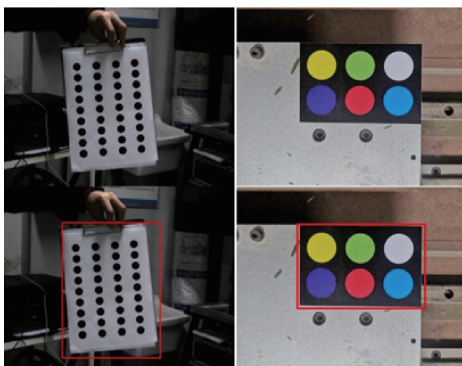
$$\begin{aligned} W_i &= w_i \times W \\ H_i &= h_i \times H \end{aligned} \quad (1)$$

$$\begin{aligned} x_{ai} &= c_{xi} \times W - W_i/2 \\ y_{ai} &= c_{yi} \times H - H_i/2 \end{aligned} \quad (2)$$

使用 YOLOv4-tiny 进行 ROI 区域提取的效果如图 1 所示。



(a) 标准标定板 ROI 提取结果
(a) Standard calibration board ROI extraction results



(b) 自制标志物 ROI 提取结果
(b) Manmade marker ROI extraction results

图 1 标志物 ROI 提取

Fig. 1 Marker ROI extraction

1.3 基于完整度排序的圆检测算法

在图像中获取到标志物的 ROI 后进行圆形检测的步骤如图 2 所示,图 2(a) 为一个标定板图像的 ROI,先在 ROI 区域中通过边缘检测与弧段提取得到所有的弧段,对得到的弧段进行完整度验证,将其分成如图 2(c) 所示完整度较高的主弧 A_L 和图 2(b) 所示短弧 A_S ;再将短弧中属于主弧 A_L 的部分连接到对应的主弧中,得到图 2(e) 所示的主弧候选圆。

以虚线框中的弧线为例,图 2(g) 中的弧线为短弧 A_S 中属于主弧 A_L 的部分,需要将其连接到图 2(c) 虚线框中对应的主弧上,连接后的结果如图 2(e) 中虚线框所示;

最后对图 2(f) 所示每个剩余的短弧进行搜索连接得到图 2(d) 所示剩余的候选圆。

1) 边缘检测与弧段提取

在获取到标志物的感兴趣区域 ROI 后,本文先采用 EDdrawing 算法^[16-17]对 ROI 区域提取二值化边缘。对于得到的边缘线条集合 E_{dge} ,由于任务的目标是检测圆,使用文献[18]中的算法,去除中边缘 E_{dge} 的直线,只保留弧

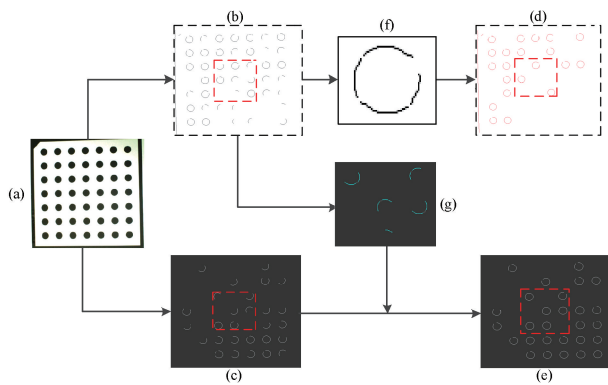


图 2 圆形检测示意图

Fig. 2 Schematic diagram of circular detection

线。具体方法为计算每个像素点 I_i 附近的局部曲率 s ,当像素点 I_i 处的曲率 s 过大时说明该像素点为角点,将边缘在角点处断开,得到平滑的边缘线条;然后使用文献[18]的方法对每条线条 $e \in E_{dge}$ 的边缘计算整体曲率,当其整体曲率较低时,认为该边缘线条为直线;去掉 E_{dge} 中所有直线,保留剩余的边缘线条,认为这些去掉直线剩余的边缘线条都是弧段,记为 A_{all} 。检测过程如图 3 所示,图 3(a) 为不同场景下的两张图像,图 3(b) 为边缘线条断开后的平滑边缘,图 3(c) 即为去掉直线后的弧段。

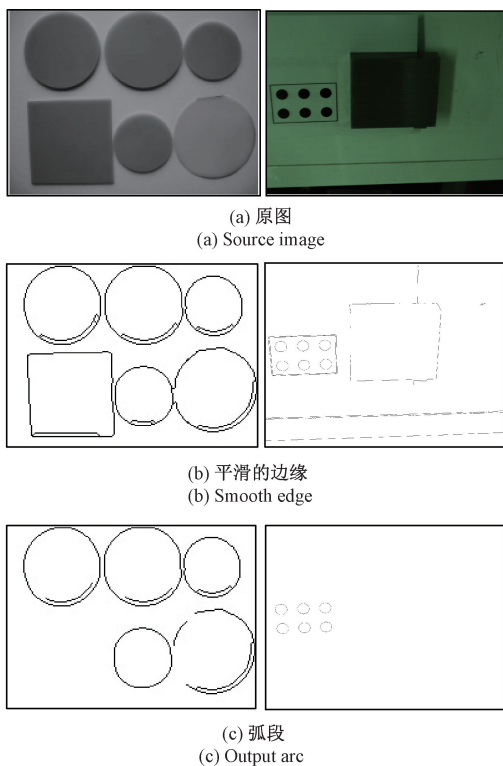


图 3 边缘检测与弧段提取

Fig. 3 Edge detection and arc extraction

2) 圆弧完整性计算获取主弧

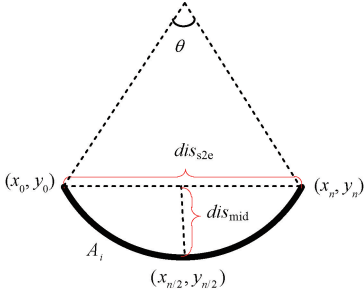
在高分辨率视觉定位场景下,通常大多数标志物中的圆轮廓清晰,可以直接获取,只有部分标志物中的圆由于遮挡或者噪声干扰导致弧段被断开成几段,需要通过弧段连接得到。因此本算法先获取到大部分较完整的弧段,再对其他弧段进行验证连接。

设圆的半径为 r , 圆心角为 θ , 则圆的参数方程如下:

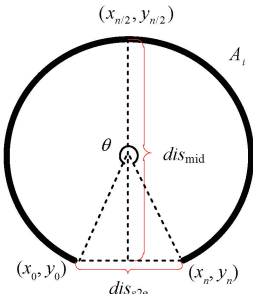
$$\begin{cases} x = r \cos \theta \\ y = r \sin \theta \end{cases} \quad (3)$$

如图4所示,对于一条长度为 n 的弧段 A_i , 其起点为 (x_0, y_0) , 终点为 (x_n, y_n) 。如式(4), 定义弧段 A_i 的中点 $(x_{n/2}, y_{n/2})$ 与起点终点连线的中点 $(\frac{x_0 - x_n}{2}, \frac{y_0 - y_n}{2})$ 的距离为弧中距 dis_{mid} 。

$$dis_{mid} = \sqrt{\left(x_{n/2} - \frac{x_0 + x_n}{2}\right)^2 + \left(y_{n/2} - \frac{y_0 + y_n}{2}\right)^2} \quad (4)$$



(a) 角度小于180°的弧
(a) Angle of arc that is less than 180°



(b) 角度大于180°的弧
(b) Angle of arc that is larger than 180°

图4 弧中距与弧边距

Fig. 4 Arc distance and arc margin

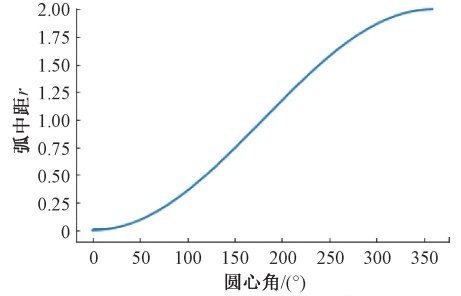
如图5(a)所示,若已知圆弧的半径 r , 当圆心角 θ 从 0° 变化到 180° 再到 360° 时, 弧中距 dis_{mid} 的值从0变化 r 再变化到 $2r$ 。且在 $30^\circ \sim 330^\circ$ 的区间内, 弧中距 dis_{mid} 值的变化接近于线性, 因此可以用弧中距 dis_{mid} 粗略的表示一条圆弧的完整性。但是对于一条未知的弧, 由于无法提前了解此圆弧的半径 r , 因此无法判断这条弧段的完整性。为了解决这个问题, 因此引入弧边距 dis_{s2e} 。如图4

所示, 设弧段的起点 (x_0, y_0) 到终点 (x_n, y_n) 直线距离为弧边距 dis_{s2e} , 其定义如式(5)。如图5(b)所示, 当 θ 从 0° 变化到 360° 时, 弧边距 dis_{s2e} 的值从0变化 $2r$ 再变化到0:

$$dis_{s2e} = \sqrt{(x_0 - x_n)^2 + (y_0 - y_n)^2} \quad (5)$$

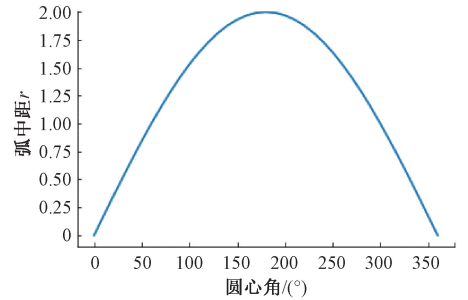
式(6)定义了一条弧的完整性 T_{hr} 为 dis_{mid} 与 dis_{s2e} 之比的对数。

$$T_{hr} = \ln(dis_{mid}/dis_{s2e}) = \ln\left(\frac{\sqrt{\left(x_{n/2} - \frac{x_0 + x_n}{2}\right)^2 + \left(y_{n/2} - \frac{y_0 + y_n}{2}\right)^2}}{\sqrt{(x_0 - x_n)^2 + (y_0 - y_n)^2}}\right) \quad (6)$$



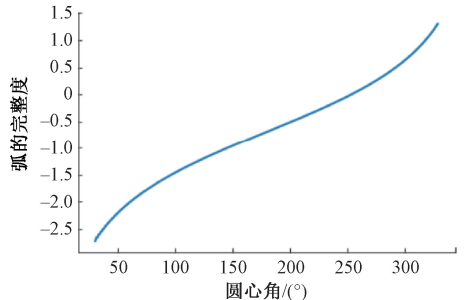
(a) 弧中距随圆心角变化

(a) The arc distance varies with the central angle



(b) 弧边距随圆心角变化

(b) The arc margin varies with the central angle



(c) 完整性随圆心角变化

(c) The integrity varies with the central angle

图5 圆弧完整度的各个参数变化示意图

Fig. 5 Schematic diagram of each parameter change of the completeness of the arc

当圆心角 θ 从 0° 变化到 360° 时, T_{hr} 的值为单调递增的, 并且如图5(c)所示, 在 $30^\circ \sim 330^\circ$ 的区间内, T_{hr} 值的变化接近于线性。由于这个值与弧的半径无关, 因此可

以用 T_{hr} 来代表一条弧段的完整度。

图6所示为不同 T_{hr} 的圆弧大小,可以看出随着 T_{hr} 变化,圆也呈现不同的完整度。当 $\theta = 180^\circ$, 即弧段的长度为整个椭圆的 $1/2$ 时, $T_{hr} = \ln(1/2) \approx -0.69$ 。设弧段完整度的阈值为 α , 当得到所有的弧段后, 计算每条弧段的完整度 T_{hr} , 当这条弧段的完整度 T_{hr} 大于 α 时, 认为这条弧段的完整度较高, 将其加入主弧集 A_L , 否则将其加入 A_s 中。经过实验, α 取 -1.06 附近时效果较佳。

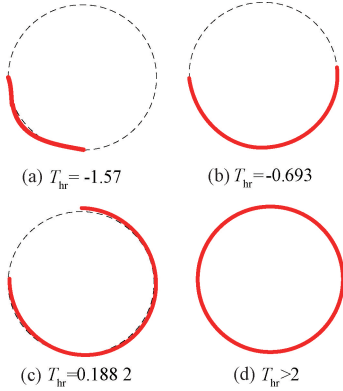


图6 不同完整度的圆弧

Fig. 6 Arcs of different completeness

3) 连接属于主弧的部分

获得弧段集合 A_L 与 A_s 之后, 主弧 A_L 中的弧段虽然完整度较高, 但拟合后只能得到对应圆的近似结果, 要计算得到该弧段精确的圆心需要获得尽可能多的边缘。因此需要将短弧 A_s 中可以连接到主弧 A_L 的部分连接到对应的主弧 A_L 上。如图7(a)所示, A_0 为一条 A_L 中完整度较高的主弧, A_1 为 A_s 中一条完整度较低的短弧, 需要将短弧 A_1 连接到主弧 A_0 上。

由于主弧的完整度大都超过圆的 $1/2$, 用最小二乘法拟合主弧 A_0 得到圆的参数, 其圆心为 $O(x_{l0}, y_{l0})$, 长轴为 a , 短轴为 b ; 设 $p_1(x_1, y_1)$ 到 $p_n(x_n, y_n)$ 为短弧 A_1 上的点, 若 dis_{avg} 为短弧 A_1 上的所有像素点到主弧 A_0 圆心 O 的平均距离, 则 dis_{avg} 的计算如式(7)所示。

$$dis_{avg} = \frac{\sum_{i=1}^n (dis(p_i, p_0))}{n} \quad (7)$$

设 P_{ct} 为 A_1 中到 A_0 的圆心 O 的距离在一定范围内的像素点占 A_1 像素点总数的比例, P_{ct} 定义如式(8)所示。

$$P_{ct} = \frac{\sum_{i=1}^n count(p_i, p_0)}{n} \quad (8)$$

其中,

$$count(p_i, p_0) = \begin{cases} 1, & \lambda_0 b < dis(p_i, p_0) < \lambda_1 a \\ 0, & \text{其他} \end{cases}$$

式中: λ_0 和 λ_1 为控制像素取值范围大小的超参数。

若 A_1 满足式(9), 即 A_1 上所有的像素点到 A_0 圆心 $O(x_{l0}, y_{l0})$ 的平均距离在一定范围内, 并且 A_1 中的像素满足在 A_0 半径附近的点的数量大于 85% 时, 认为可以将弧 A_1 连接到 A_0 上, 满足条件的弧如图7(a)所示。

$$\begin{cases} \beta_0 b < dis_{avg} < \beta_1 a \\ P_{ct} < 0.85 \end{cases} \quad (9)$$

如图7(b)所示, A_2 虽然满足 $\beta_0 b < dis_{avg} < \beta_1 a$, 但其 $P_{ct} < 0.85$, 因此不能将其连接到 A_0 上, 实验中, β_0 和 β_1 分别取 0.6 和 1.1 。

在计算 A_0 到 A_1 圆心 O 的平均距离 dis_{avg} 时, 为了减小计算量, 设置剪枝策略。当 A_0 的像素点中出现到圆心的距离小于 $0.5b$ 或者大于 $1.5a$ 时, 认为 A_0 不可以连接到 A_1 中, 不进行后续判断。

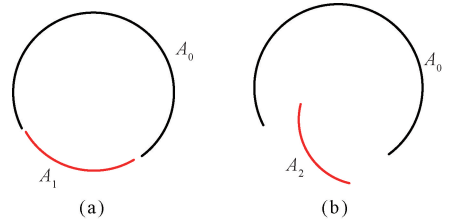


图7 主弧连接

Fig. 7 Main arc connection

4) 弧段搜索得到剩余的圆

已经提取到了完整度较高的圆弧构成的候选圆, 此时要对剩下的短弧 A_s 进行连接, 得到其余的候选圆。

由于在标志物中, 大部分需要连接的弧段相距较近, 且较平滑。因此认为, 当两条弧段满足有端点相距较近、较近的端点处局部方向相近、没有重叠且凹凸性一致时, 可以将这两个弧段在该端点连接到一起。

如图8所示, 设 A_3 和 A_4 为短弧 A_s 中的两条弧, A, B 为弧 A_3, A_4 的端点。设 d_i 为弧 A_3 端点 A 到弧 A_4 端点 B 的距离, $\vec{a}$ 为 A_3 端点处向内的切线向量, $\vec{b}$ 为 A_4 端点处向内的切线向量, $\vec{c}$ 为 A_3 端点 A 到 A_4 端点 B 的方向向量; θ_1 为 $\vec{a}$ 与 $\vec{c}$ 的夹角, 当 A_3 与 A_4 在端点处没有重叠时, θ_1 应较大保证两个端点可以连接; θ_2 为 $\vec{a}$ 与 $\vec{b}$ 末尾的局部方向的夹角, 设当 θ_2 较小时, 说明端点处方向不一致, 即弧段 A_3 的端点 A 无法通过延长连接到 A_4 端点 B ; s 定义如式(10), 当 $s < 0$ 时代表 $\vec{a}$ 到 $\vec{b}$ 与 $\vec{b}$ 到 $\vec{c}$ 的凹凸性不一致, 不属于同一个圆。

$$s = \text{sign}(-\vec{a}, \vec{c}) * \text{sign}(\vec{c}, \vec{b}) \quad (10)$$

式中: $\text{sign}(\vec{a}, \vec{b})$ 为 $\vec{a}$ 和 $\vec{b}$ 的向量积。

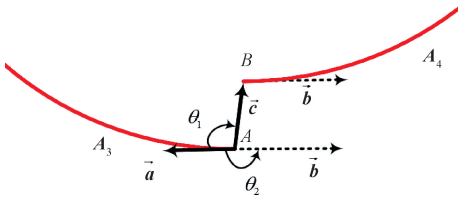


图8 弧段连接示意图

Fig. 8 Schematic diagram of arc connection

最终,当两条弧段存在两个端点满足式(11)时,将其在该端点处连接。递归搜索所有的弧段的两个端点,直到所有的弧都搜索一遍,然后对得到的所有连接后的弧段进行完整性验证,得到满足式(12)的弧段集 $A_{\text{connected}}$ 。

$$\begin{cases} d_i < d_0 \\ \theta_1 > \delta_1 \\ \theta_2 > \delta_2 \\ s > 0 \end{cases} \quad (11)$$

$$T_{\text{hr}} > \alpha \quad (12)$$

式中: d_0 为控制距离连接的阈值,增大 d_0 时可以连接的像素距离也相应提高; δ_1, δ_2 为控制连接角度的阈值,增大 δ_1 、减小 δ_2 可以降低误连接。在实验中, α 取-1.06, d_0 取 40, δ_1, δ_2 分别取 120 和 60,通常当输入图片分辨率提高时可以适当增大 d_0, δ_1 , 减小 δ_2 。

5) 候选圆拟合、筛选

得到了候选弧段 A_L 与 $A_{\text{connected}}$, 将其合并得到所有的候选弧段 $A_{\text{candidate}}$ 。此时,需要对这些候选圆弧进行验证,剔除重叠的和不完全的圆弧,得到最终输出 O_{output} , 筛选准则如下。

(1) 圆度检查

圆度 C 定义如式(13), 当候选圆的圆度 C 接近 1 时, 代表其更接近一个圆, 设 C_0 为圆度阈值, 对候选圆进行筛选, 保留 $C > C_0$ 的候选圆。

$$C = \frac{4\pi S}{P^2} \quad (13)$$

式中: S 为轮廓的面积, P 为轮廓的周长。

(2) 最小二乘拟合为椭圆, 进行长短轴限制

使用最小二乘法将所有候选弧段拟合为椭圆, 得到每个候选椭圆的圆心 (x_i, y_i) 跟长短轴 a_i, b_i 。计算每个候选圆的长宽比 $f_i = a_i/b_i$, 只保留近圆椭圆, 剔除长宽比过大与过小, 即 $f_i < 1/5$ 或者 $f_i > 5$ 的候选圆。

(3) 同圆剔除

以横坐标 x_i 为索引对所有候选圆进行排序, 然后按顺序遍历候选圆, 将每个候选圆与其后 3 个相邻的候选圆进行重叠判断。重叠判断的标准如式(14), 当两个圆的圆心距离 d 小于阈值 d_1 而且长宽接近时, 认为两个圆

为同一个圆, 去除其中完整性较低的那个。

$$\begin{cases} d < d_1 \\ \lambda_0 < (a_i b_{i+1}) / (a_{i+1} b_i) < \lambda_1 \end{cases} \quad (14)$$

式中: d_1 为距离阈值, $(a_i b_{i+1}) / (a_{i+1} b_i)$ 代表两个圆的大小比。在本实验中, d_1 取 40, λ_0 和 λ_1 分别取 0.9 和 1.1。候选圆经过以上筛选后得到了最终的结果 O_{output} 。

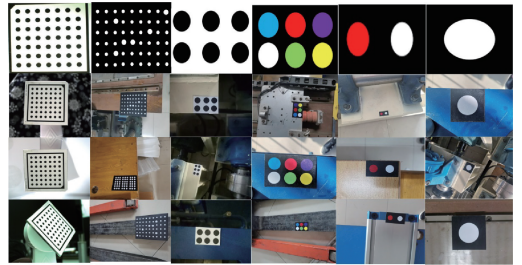
2 实验与结果

为了检验本文算法的有效性, 使用 C++ 语言在 VS2019 平台下编写了实验程序。操作系统为 Windows 10, PC 的 CPU 为 i7-8700 3.2 GHz, 16 GB 内存, 显卡为 NVIDIA GeForce GTX1660。

2.1 数据集与评估标准

1) 数据集采集与分类

本文先制作了多个圆形数量不同的标志物, 然后使用相机分别拍摄了不同背景、不同光照、不同角度标志物的图像共 682 张, 数据集图像的平均分辨率为 $3\,000 \times 4\,000$ 。将这个数据集称为 D_1 , 数据集的图像如图 9 所示。

图9 数据集 D_1 Fig. 9 Dataset D_1

为了与其他圆检测算法进行对比, 本文从数据集 D_1 中选取了 170 幅标志物图像如图 10 所示, 这些图像中只有标志物所在的区域有圆, 而在图像的其他区域中都不存在可以被认为是圆的物体, 将这个数据集称为 D_2 , 对 D_2 数据集中的图像进行了标注, 标注信息为检测框的大小和检测框内圆的数量。

图10 数据集 D_2 Fig. 10 Dataset D_2

2) 评估标准

在进行 ROI 提取时,本文使用交并比 (IOU) 来判断正负样本,如式 (15),交并比 IOU 可以衡量目标检测中检测框与真实检测框的检测效果,当检测到的矩形框与人工标注的矩形框的 IOU 值大于阈值时为正样本,否则负样本。

$$IOU = \frac{|A \cap B|}{|A \cup B|} \quad (15)$$

各对比算法在数据集 D_2 中进行圆检测时,采用了简单的标注方式,即只标注了图片中的正确圆的数量,并没有对每个圆的圆心以及半径做出标注。此时正负样本计算方式为:若在一张图片中检测到圆的数量小于等于标志物中的标注数量时,认为正样本为检测数量,漏检数等于标注数量减去检测数量;当检测数量大于标注数量时,记正样本数为标注数量,多出来的部分为误检数量。可以这样计算是由于标志物中的目标圆为标志物中的显著目标,当圆检测算法检测到圆时,各方法都会优先检测到标志物中的显著圆。

为了准确评估模型检测效果,计算被检目标的精准率 P 和召回率 R ,其定义如式 (16)、(17),其中, TP 代表模型正确检测出来的目标数量, FP 代表模型误检的目标数量, FN 代表模型漏检的目标数量。 F -measure 综合了精确率和召回率的指标,表示检验结果的有效性, F -measure 值越高说明试验方法较为理想,其计算方式如式 (18) 所示。

$$P = \frac{TP}{TP + FP} \quad (16)$$

$$R = \frac{TP}{TP + FN} \quad (17)$$

$$F\text{-measure} = \frac{2PR}{P + R} \quad (18)$$

2.2 实验

1) 基于完整度排序的圆提取算法实验与结果

图 11 所示为本文所述圆提取算法的检测过程,为了区分不同的弧段,对获取到的每条弧段进行了随机颜色填充处理。其中图 11(a) 为一张图像的 ROI 部分,其大小为 611×643 ,在图 11(b) 中,由于这些边缘在角点处断开,因此得到了大小不一的多个圆弧;经过完整度验证后,这些圆弧被分为两个部分,一部分为完整度较高的主弧,如图 11(c) 所示,另一部分为完整度较低的短弧,如图 11(d) 所示;将属于图 11(c) 中主弧的部分连接到对应的主弧得到如图 11(e) 所示完整弧段,然后将剩余的部分搜索连接得到如图 11(f) 所示的完整弧段;最终图 11(e) 与 (f) 中的弧段合并得到了如图 11(g) 所示的候选圆。

在 ROI 区域内使用本文算法进行圆检测的结果如图

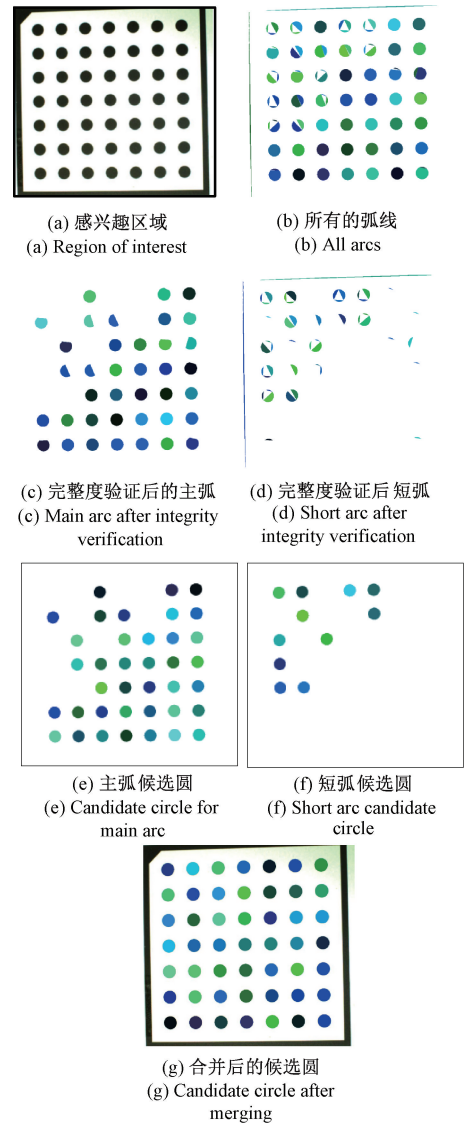


图 11 ROI 区圆检测算法过程

Fig. 11 ROI area circle detection algorithm process

12 所示,可以看到在包含不同形状、不同数量、同心圆、以及添加了椒盐噪声和高斯噪声的图像中本算法都可以准确的检测到目标圆。

2) YOLOv4-tiny ROI 提取实验与结果

训练时,先对数据集先使用 k 均值聚类算法聚类,得到了大小为 (39, 51)、(217, 212)、(124, 147)、(246, 366)、(182, 293)、(71, 111) 的锚框,修改对应的配置文件后进行 YOLOv4-tiny 进行的模型训练。为了测试 ROI 提取步骤对检测速度的影响,使用了图 13(a) 所示的图像进行了实验测试。

图 13 左右两列分别为在进行 ROI 弧段提取和不进行 ROI 区域提取时的圆检测效果,图 13(b) 为检测过程中产生的边缘,图 13(c) 为提取到的弧段。由于场景较复杂,图 13(b) 左图中产生了许多边缘,但如果只在右图

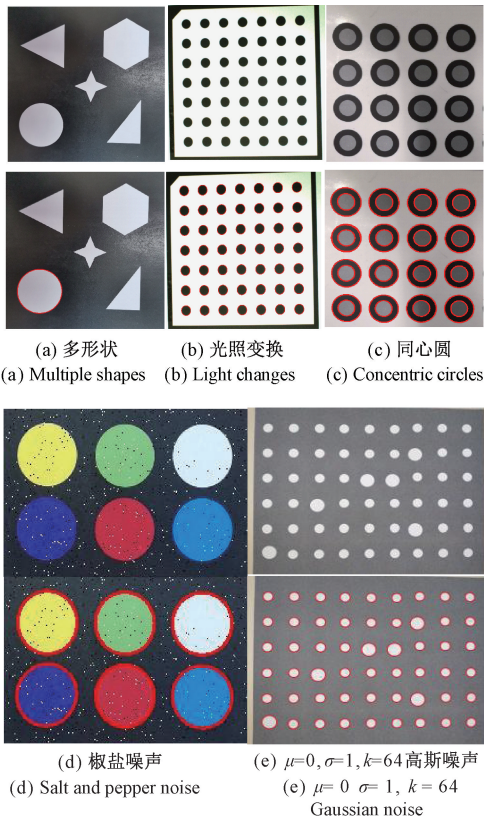


图 12 感兴趣区域中的检测结果
Fig. 12 Detection result in ROI

的 ROI 中进行检测,则只剩下很少的边缘;同样图 13(c)左图中显示了从原图像中获取到的众多弧段,而如图 13(c)右图在 ROI 中只检测到的很少的弧段。图 13(d)为两者最终输出的候选圆。

表 1 所示为两者的检测时间对比,可以看到虽然使用 YOLOv4-tiny 进行 ROI 提取占用了 31 ms 的检测时间,但是在 ROI 区域提取后,由于 ROI 区域中后续每个步骤的检测时间都大幅降低,总的时间性能反而有了很大的提高。

表 1 检测时间对比		
Table 1 Detection time		
算法过程	仅使用圆检测	ROI 提取+圆检测
ROI 提取	0.00	31.6
边缘检测	70.98	1.14
弧段提取	38.34	0.48
完整性验证	6.13	0.56
主弧提取	0.81	0.02
短弧连接	13.25	0.01
聚类拟合	11.64	1.08
总时间	141.15	34.89

2.3 对比实验

在数据集 D_1 中的检测结果如图 14 所示,数据集 D_2

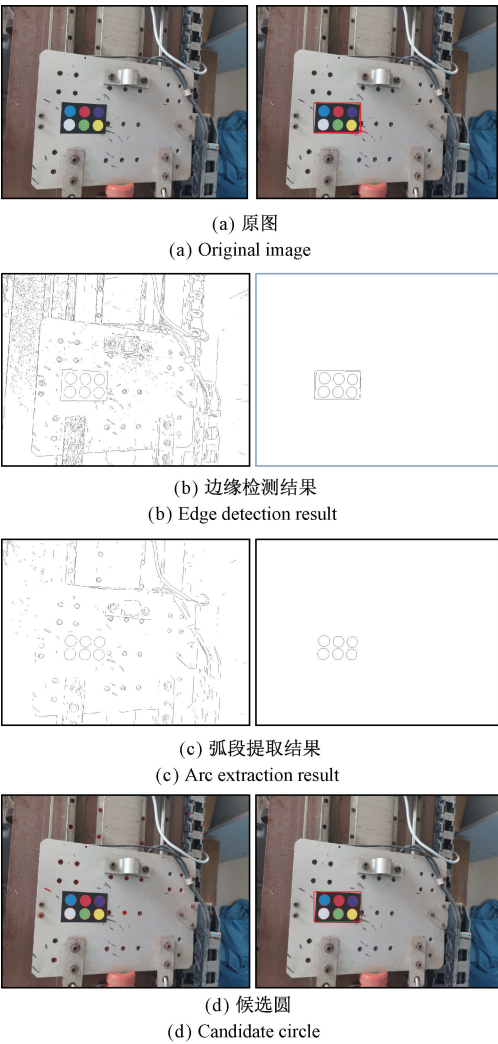


图 13 ROI 提取前后的检测结果

Fig. 13 Detection results before and after ROI extraction

中进行检测的结果如图 15 所示。可以看到无论简单或者复杂背景下的图像,本文中的算法都能很好的完成标志物中圆提取的任务。但由于在复杂图像中进行圆检测无法定义螺母等轮廓是否为圆,导致无法统计正确率,因此本文只在数据集 D_2 中进行了对比实验。

数据集的拍摄分辨率 $3\,000\times4\,000$,为了测试本文算法在不同分辨率下的检测效果。将图像以原图像的 0.1 倍大小为基准进行缩放后测试了算法的运行时间和检测精度。各个算法的检测时间如图 16(a) 所示,算法测试时以图像的 0.1 倍为输入基准,横坐标为 10 倍时代表输入算法的图像尺寸为原图像的大小。从图 16 可以看到,在本文所用算法在分辨率提高时仍然保持较低检测时间,而 AAMED 与 EDCircle 算法的检测时间都会随分辨率的提高大幅度的增加。算法检测精度用 F -measure 表示,检测结果如图 17 所示,可以看到本文所提算法的检测精度在分辨率较低时,由于本文的参数并非自适应设

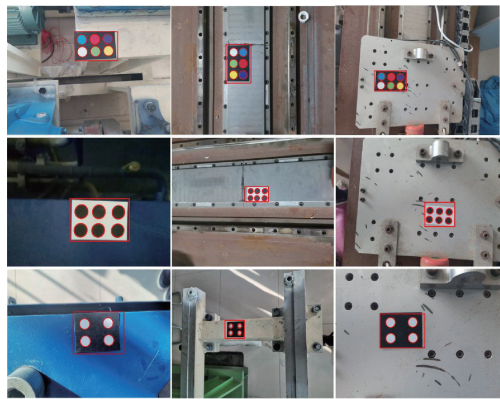


图 14 数据集 D_1 中的结果
Fig. 14 Results in dataset D_1



图 15 数据集 D_2 中的检测结果
Fig. 15 Results in dataset D_2

置,在分辨率较低时会有较多的弧线被忽略,因此检测效果略低于 EDCircle 算法。但当分辨率提高到原图的 0.4 倍以及更高的分辨率时,本算法可以达到与其他的检测算法相当的检测精度。

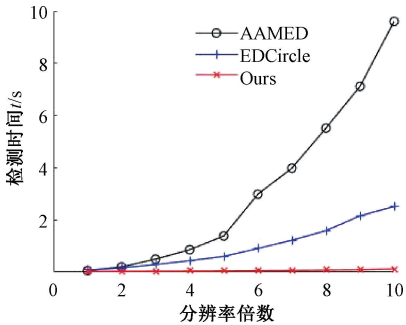


图 16 不同分辨率下的检测时间
Fig. 16 Detection time under different resolutions

从图 16 可以看出,由于本算法通过固定时间的 ROI 提取过程先去除了大部分的边缘,同时算法具有较好的时间复杂度,因此算法在分辨率逐渐提高时仍然保持较低的检测时间,而 AAMED 与 EDCircle 算法在分辨率的提高时,由于边缘像素数量的增加,其检测时间大幅度的

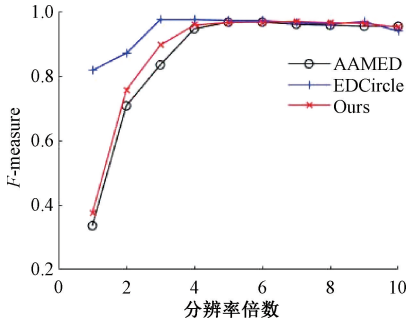


图 17 与 F -measure 分数
Fig. 17 F -measure score under different resolutions

增加。
当本算法在原图 0.5 倍分辨率下即分辨率为 $1\,500 \times 2\,000$ 时的检测结果如表 2 所示,由表 2 可见,此时本文算法的检测精度指标 F -measure 与其他两种算法相近,但检测时间只有 EDCircle 算法的 0.11, AAMED 算法 0.05, 平均每张图片的检测时间只有 68.8 ms。

表 2 0.5 倍分辨率下算法结果
Table 2 Algorithm results at 0.5× resolution

算法	$P/\%$	$R/\%$	$F/\%$	t/ms
EDCircle	99.3	95.1	97.2	602
AAMED	98.2	95.1	96.7	1384
Ours	98.2	94.9	96.5	68.8

3 结 论

针对在高分辨率图像中的传统标志物圆检检测算法检测速度慢、误检率高的问题,提出了一种基于完整度排序的快速圆提取算法,所设计算法在高分辨率图像中取得了比已有算法更快速的检测效果。本算法先使用 YOLOv4-tiny 这一目标检测模型在图像中检测标志物 ROI 区域。然后针对在 ROI 中的圆检测问题,本文提出了一种分策略的圆检测算法。算法先使用 EDdrawing 算法进行边缘检测,针对从圆中提取出的弧段,设计了弧段的完整度验证法则,对不同完整度的弧段采用不同的策略获取候选圆,最后合并得到模型的输出。本算法适用于背景复杂但标志物内结构简单的视觉定位或者相机标定场景,在此类场景下的应用本章算法,尤其在高分辨图像中进行检测时可以在保持检测精度的同时取得极高的检测速度,算法在分辨率为 $2\,000 \times 1\,500$ 的数据集中能够达到 68.8 ms 一帧的检测速度。

参考文献

[1] 黄志清, 苏毅, 王庆文, 等. 二维激光雷达与可见光相机外参标定方法研究[J]. 仪器仪表学报, 2020,

- 41(9): 121-129.
- HUANG ZH Q, SU Y, WANG Q W, et al. Research on extrinsic parameter calibration method of 2D laser rangefinder (LRF) and visible light cameras [J]. Chinese Journal of Scientific Instrument, 2020, 41(9): 121-129.
- [2] 吕侃徽, 张大兴. 基于改进 Hough 变换耦合密度空间聚类的车道线检测算法[J]. 电子测量与仪器学报, 2020, 34(12): 172-180.
- LYU K H, ZHANG D X. Lane detection algorithm based on improved hough transform coupled density space clustering[J]. Journal of Electronic Measurement and Instrumentation, 2020, 34(12): 172-180.
- [3] 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. Improved fast circle detection algorithm based on random Hough transform[J]. Computer Engineering and Design, 2018, 39(7): 1978-1983.
- [4] 毛庆洲, 潘志敏, 高文武. 利用迭代霍夫圆变换实现成捆棒材可靠计数[J]. 武汉大学学报(信息科学版), 2014, 39(3): 373-378.
- MAO Q ZH, PAN ZH M, GAO W W. Using iterative Hough round transform and connected area to count steel bars reliably[J]. Geomatics and Information Science of Wuhan University, 2014, 39(3): 373-378.
- [5] PĂTRĂUCEAN V, GURDJOS P, VON GIOI R G, et al. A parameterless line segment and elliptical arc detector with enhanced ellipse fitting[C]. European Conference on Computer Vision, 2012(7573): 572-585.
- [6] AKINLAR C, TOPAL C. EDCircles: A real-time circle detector with a false detection control [J]. Pattern Recognition, 2013, 46(3): 725-740.
- [7] MENG C, LI Z, BAI X, et al. Arc adjacency matrix-based fast ellipse detection[J]. IEEE Transactions on Image Processing, 2020, 29: 4406-4420.
- [8] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017(42): 386-397.
- [9] 王陶然, 王明泉, 张俊生, 等. 基于 Mask R-CNN 的轮毂缺陷分割技术[J]. 国外电子测量技术, 2021, 40(2): 1-5.
- WANG T R, WANG M Q, ZHANG J S, et al. Automatic instance segmentation technology of automobile wheel hub defects based on Mask R-CNN[J]. Foreign Electronic Measurement Technology, 2021, 40(2): 1-5.
- [10] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector [C]. European Conference on Computer Vision, 2016.
- [11] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.
- [12] TAN M, PANG R, LE Q V. EfficientDet: Scalable and efficient object detection [C]. Conference on Computer Vision and Pattern Recognition (CVPR), 2020: 10778-10787.
- [13] LAW H, DENG J. CornerNet: Detecting objects as paired keypoints[C]. European Conference on Computer Vision, 2018: 642-656.
- [14] TIAN Z, SHEN C, CHEN H, et al. FCOS: Fully convolutional one-stage object detection[C]. IEEE/CVF International Conference on Computer Vision (ICCV), 2019: 9626-9635.
- [15] WANG C Y, BOCHKOVSKIY A, LIAO H. Scaled-YOLOv4: Scaling cross stage partial network [J]. Computer Science, 2020, arXiv:2011.08036.
- [16] TOPAL C. Edge drawing: A Heuristic approach to robust real-time edge detection [C]. 20th International Conference on Pattern Recognition, 2010: 2424-2427.
- [17] TOPAL C, OZSEN O, AKINLAR C. Real-time edge segment detection with edge drawing algorithm[C]. 7th International Symposium on Image and Signal Processing and Analysis (ISPA), 2011: 313-318.
- [18] LEE Y S, KOO H S, JEONG C S. A straight line detection using principal component analysis[J]. Pattern Recognition Letters, 2006, 27(14): 1744-1754.
- [19] 李杰, 史艺丹, 聂泽东, 等. 手指静脉图像感兴趣区域快速提取方法研究[J]. 电子测量与仪器学报, 2020, 34(4): 42-49.
- LI J, SHI Y D, NIE Z D, et al. Research on fast extraction method of region of interest in finger vein images [J]. Journal of Electronic Measurement and Instrumentation, 2020, 34(4): 42-49.
- [20] 化嫣然, 张卓, 龙赛, 等. 基于改进 YOLO 算法的遥感图像目标检测[J]. 电子测量技术, 2020, 43(24): 87-92.
- HUA Y R, ZHANG ZH, LONG S, et al. Remote sensing image target detection based on improved YOLO algorithm [J]. Electronic Measurement Technology, 2020, 43(24): 87-92.

作者简介



丁伟利, 2003 于辽宁工学院获学士学位, 2004 于中国科学技术大学获得硕士学位, 2008 年于中国科学院沈阳自动化研究所获得博士学位, 现为燕山大学教授, 博士生导师, 主要研究方向为计算计算机视觉(视觉测量、三维重建)、虚拟现实、人机交互、医学图像处理。

E-mail: weiy51@ysu.edu.cn

Ding Weili received her B. Sc. degree from Liaoning University of Technology in 2003, M. Sc. degree from University of Science and Technology of China in 2004 and Ph. D. degree from Shenyang Institute of Automation Chinese Academy of Sciences in 2008, respectively. Now she is a Ph. D. supervisor at

Yanshan University. Her main research interests include computer human-computer interaction, and medical image processing.



郝增号, 2016 年于山东科技大学获得学士学位, 2021 年获得燕山大学硕士学位, 现为推想医疗科技股份有限公司算法工程师, 主要从事医学影像处理、三维视觉分割与检测、虚拟仿真方面的研究。

E-mail: 1293194431@qq.com

Hao Zenghao received his B. Sc. degree from Shandong University of Science and Technology in 2016 and M. Sc. degree from Yanshan University in 2021. He is currently an algorithm researcher at Beijing Infervision Technology Co., Ltd. His main research interests include medical image processing, 3D visual segmentation and detection, and virtual simulation.