

基于机器视觉的最大内接矩形快速检测算法

邹哲康, 朱铮涛, 陈映谦, 孟令龙, 王瑞丰

(广东工业大学 机电工程学院, 广州 510006)

摘要: 在进行皮革裁切、板材边角料再利用、PCB板布线等过程时都会遇到求解不规则图型最大内接矩形的问题; 当前工厂的裁定手段普遍由人工估测来完成, 这种作业方式不仅主观性强, 而且效率低下; 在研究了基于数字图像处理的遍历法、中心扩散法、遍历中心扩散法后, 结合以上方法提出一种高效实用的最大内接矩形快速检测算法: 边界排序生长法; 实验结果证实: 遍历法无法正确适用于凹多边形; 中心扩散法虽然可适用于任意不规则图型, 但其检测效果不够理想; 遍历中心扩散法的检测结果较为理想, 但该方法耗时巨大; 而边界排序生长法在保证高效率的同时依旧可以拟合出理想的内接矩形。

关键词: 图像处理; 最大内接矩形; 遍历法; 中心扩散法; 边界排序生长法

A Fast Detection Algorithm for Maximum Inlined Rectangle Based on Machine Vision

Zou Zhekang, Zhu Zhengtao, Chen Yingqian, Meng Linglong, Wang Ruifeng

(College of Electromechanical Engineering, Guangzhou University of Technology, Guangzhou 510006, China)

Abstract: The detection of irregular polygons with maximum enclosed rectangle is of great significance to the industrial scenes such as leather cutting, the reuse of board scraps and PCB board routing. At present, the method of adjudication is generally completed by manual estimation, this method has some problems such as strong subjectivity and low efficiency. After studying the traversing method, center diffusion method and traversal center diffusion method, an efficient and practical fast detection algorithm for maximum enclosed rectangle, boundary-sorting growth method, is proposed based on the above ideas. Experimental results show that the traversal method is not suitable for concave polygons. Although the central diffusion method can be applied to any graph, its detection effect is not ideal. The traversal center diffusion method has an ideal detection result, but it takes a lot of time. The boundary sorting growth method can ensure high efficiency and still obtain the ideal maximum enclosed rectangle.

Keywords: image processing; maximum enclosed rectangle; traversing method; center diffusion method; boundary-sorting growth method

0 引言

不规则多边形的最大内接矩形的检测对于皮革裁切、板材边角料再利用等工业场景有着重大的影响。目前, 国内许多工厂仍是以纯人工的方式进行裁定的, 人工裁定通常是根据工人的经验来确定矩形, 这种方式存在很强的主观性, 不仅容易造成资源浪费, 而且效率也很低。使用机器视觉技术是生产制造业迈向高效化、智能化、经济化的一个关键途径, 实现自动化生产能在大大节约人力成本、材料资源的同时还能显著提高生产的效率, 而其中存在的最大难题便是如何快速准确地检测出目标物体的最大内接矩形。

随着计算机视觉的振翻和工业智能化的发展, 图像处理技术已经广泛应用于工厂的自动化生产当中^[1]。赵凤^[2]等人研究了基于灰度和非局部空间灰度特征的二维 Otsu 曲线阈值分割法。本文团队^[3]于 2019 年研究了大理石板的图像分割方法, 此方法速度快且精度达到预期要求。吴晓光^[4]等人研究了获取图像区域最小外接矩形的算法。这些都为

检测不规则物体最大内接矩形提供了思路。对于求解最大内接矩形, 谢新华^[5]等的研究表明目标物体的摆放角度会影响最大内接矩形检测, 其提出的遍历中心扩散法虽有较高的准确率, 但其算法时间复杂度高, 耗时巨大, 不适合运行在高速运转的自动化流水作业当中。袁哲^[6]等人提出了基于改进遗传算法的任意图形最大内接矩形求解法, 该方法的适应变化规律显示其需要遗传 60 代以上才能获得较为理想的最大内接矩形, 显然也不符合快速检测的原则。

针对以上问题, 本文在谢新华^[5]等人的研究基础上, 提出一种快速高效的最大内接矩形检测算法: 边界排序生长法。该方法能在极短的时间内有效地检测出理想的内接矩形, 采用该方法结合标定的摄像头就能实现目标物体最大内接矩形的高速检测。

1 系统算法实现

目标物体最大内接矩形检测过程分为图像预处理和矩形检测两个部分。首先, 传入工厂提供的实验数据图像,

收稿日期: 2020-11-21; 修回日期: 2020-12-07。

作者简介: 邹哲康(1996-), 男, 江西万年人, 硕士, 主要从事计算机视觉方向的研究。

朱铮涛(1967-), 男, 江苏启东人, 博士, 副教授, 硕导, 主要从事视觉检测技术应用、嵌入式系统设计等方向的研究。

引用格式: 邹哲康, 朱铮涛, 陈映谦, 等. 基于机器视觉的最大内接矩形快速检测算法[J]. 计算机测量与控制, 2021, 29(6): 194-198.

对这些数据图像进行处理, 得到适合检测的图像。然后运用不同的视觉检测算法找到其最大内接矩形并记录算法运行时间及内接矩形的面积。像处理的总体流程包括图像采集、图像语义分割、图像几何变换、边缘检测、最大内接矩形检测, 算法总体流程如图 1 所示。

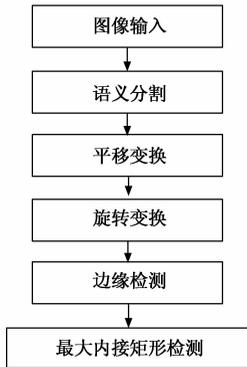


图 1 算法流程图

具体算法步骤如下:

- 1) 首先将工厂提供的大理石板材毛胚图像进行语义分割, 保留图像中需要的部分;
- 2) 选取分割后的图像的最大连通域, 将其转换为二值图像并进行平移变换, 使得目标物体坐落于图片的中央位置;
- 3) 对平移后的图像进行旋转变换, 使得目标物体与水平方向平行;
- 4) 检测物体的边缘, 并记录外边缘的坐标信息;
- 5) 通过视觉检测算法, 找到目标物体的最大内接矩形。

2 图像预处理

2.1 提取 ROI

提取 ROI (region of interest) 指的是将图像中需要的物体从背景中分离出来。相比直接将整个图像作为算法输入, 提取 ROI 能排除背景产生的干扰, 以供后续算法检测出最优的最大内接矩形。由于大理石板材纹路的复杂性, 利用传统图像处理的方法很难将其准确地分割, 因此, 本文采用深度学习中的语义分割网络来提取采集好的大理石板图像的 ROI。

本实验使用孟令龙等^[3]于 2019 年发布的针对大理石板材分割的高效语义分割网络, 模型结构如图 2 所示。该模型在 U-Net 的基础上修改卷积层的通道数以减少运算量, 使得模型的推理速度减少到 5 ms 左右。同时加入含有空洞卷积的 RASPP 模块以获得更大的感受野^[7]。最后, 辅以 Focal Loss 来拟合出更准确的结果。Focal Loss 定义如式 (1), 其中 α 为平衡因子, 用来平衡本身比例不均的正负样本, γ 是调节损失函数的一个超参数, 其值通常设置为 2, p 为模型的预测, 是一个概率, 其值介于 0~1 之间。大理石板材图片的语义分割效果如图 3 (b) 所示。

$$FL = \begin{cases} -\alpha(1-p)^\gamma \log(p), & y = 1 \\ -(1-\alpha)p^\gamma \log(1-p), & y = 0 \end{cases} \quad (1)$$

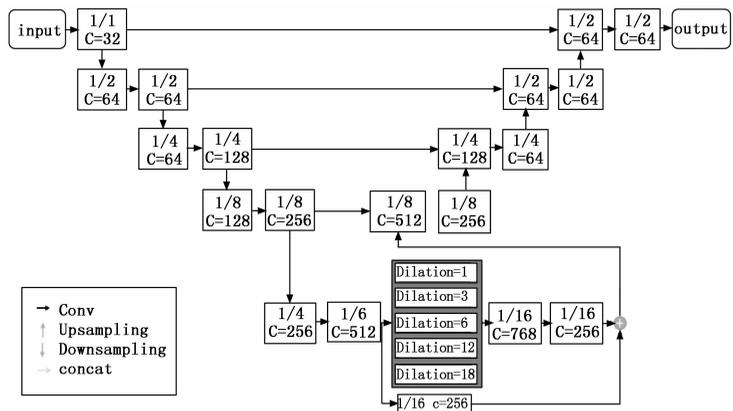


图 2 语义分割模型

2.2 旋转摆正

目标物体的摆放方向切中了算法找到最大内接矩形面积的青紫。为了得到面积最大的内接矩形, 需要将目标物体旋转到与水平方向平行的位置, 即使得目标物体水平方向上的最小外接矩形最小。

在旋转前需要将目标图像移动到图片中央位置以最大程度避免目标物体旋转后超出图像边界。首先求取图像的中心以及目标物体的中心, 图像中心如式 (2) 所示, 中心点横坐标记为 $center_x$, 纵坐标记为 $center_y$, 其中 img_w 代表图片的宽度, img_h 代表图片的高度。

$$\begin{cases} center_x = mg_w/2 \\ center_y = mg_h/2 \end{cases} \quad (2)$$

使用 OpenCV 中的 “cv2.moments” 函数求取目标物体的各阶矩, 并根据函数返回的矩值得到目标物体的重心^[8], 即该物体的中心, 将中心点的横坐标记为 C_x , 纵坐标记为 C_y , 可由公式 (3) 计算得出:

$$C_x = \frac{M_{10}}{M_{00}}, C_y = \frac{M_{01}}{M_{00}} \quad (3)$$

其中: M_{00} 为目标物体的零阶矩; M_{10} 和 M_{01} 均为目标物体的一阶矩。

求得图像中心和目标物体中心后可根据公式 (4) 计算出横坐标需要移动的距离 d_x 与纵坐标需要移动的距离 d_y , 即图像中心与目标物体中心横坐标与纵坐标的差值。

$$\begin{cases} d_x = center_x - C_x \\ d_y = center_y - C_y \end{cases} \quad (4)$$

根据所求得的 d_x 和 d_y 将目标图像平移变换到图片中央位置, 用空间变换矩阵表示如式 (5), 式中 \mathbf{M} 为变换矩阵。平移变换后的结果如图 3 (c) 所示:

$$[x', y', 1] = [[x, y, 1]\mathbf{M}; \mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ d_x & d_y & 1 \end{bmatrix} \quad (5)$$

选取平移变换后的图像的最大连通域, 根据其标定区域边界的像素点集找到目标物体的最小外接矩形^[4], 如图 3 (d) 所示。选取最小外接矩形中横坐标最小的顶点和纵坐标最小的顶点, 分别记为 ($left_x$, $left_y$) 和 ($top_$

x, top_y), 根据如下公式求出最小外接矩形和水平方向的夹角 θ 。

$$\theta = \arctan\left[\frac{|left_y - top_y|}{|left_x - top_x|}\right] \quad (6)$$

利用公式 (6) 求得的角度 θ , 对图像进行旋转变换。用空间变换矩阵表示为式 (7)。

$$[x', y', 1] = [[x, y, 1]M \quad (7)$$

式 (7) 中的 M 为变换矩阵, 有顺时针和逆时针两种形式:

$$M = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{顺时针})$$

$$M = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{逆时针})$$

旋转后的图像如图 3 (e) 所示。目标图片旋转后的最小外接矩形如图 3 (f) 所示, 证明此时的目标物体已经摆放水平。

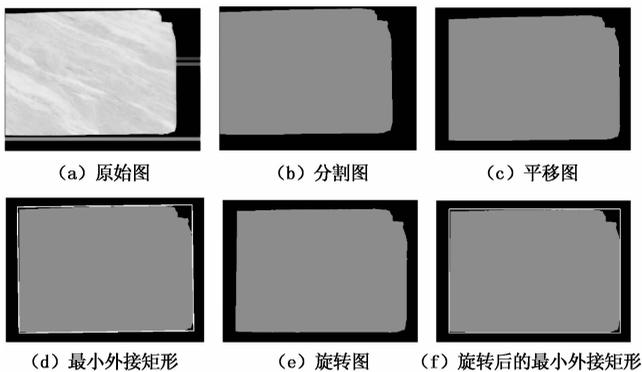


图 3 大理石板材图像预处理

2.3 边缘检测

边缘指的是图像中不连续性的特性, 如灰度、纹理结构以及颜色的突变等, 是物体与背景之间的重要信息。边缘检测^[9]是指提取目标图像的边缘轮廓信息并丢弃非相关的其他信息。对数字二值图像进行拓扑分析^[10]来确定图像的外边界, 以非压缩的形式记录所有外边界的轮廓点并保存于数组当中, 即保存的相邻的两个点的像素位置之差不超过 1。

3 最大内接矩形检测

3.1 遍历法

遍历法的思想是遍历所有边界点, 在其中找面积最大的矩形作为结果, 其主要原理如下:

1) 以边缘检测所得数组中的第一个边界点作为初始点, 记为 $A1$, 从 $A1$ 起向右移动直到到达图像的边界, 即到达像素值为 0 的位置, 记为 $A2$, 再从 $A2$ 向下移动直到再次到达图像的边界, 记为 $A3$, 再从 $A3$ 向左移动直到到达图像边界, 记为 $A4$ 。若在移动点到达边界前的 x 坐标等于 $A1$ 的 x 坐标则需要提前停止移动, 并在此记为 $A4$ 。由 $A2$ 、 $A3$ 、 $A4$ 构成矩形 1。

2) 将过程 1 选取的初始点 $A1$ 记为 $B1$, 从 $B1$ 起向下移动直到到达图像的边界, 即到达像素值为 0 的位置, 记为 $B2$, 再从 $B2$ 向右移动直到再次到达图像的边界, 记为 $B3$, 再从 $B3$ 向上移动直到到达图像边界, 记为 $B4$ 。相应的, 在移动点到达边界前的 y 坐标等于 $B1$ 的 y 坐标时也需要提前停止移动, 并在此记为 $B4$ 。由 $B2$ 、 $B3$ 、 $B4$ 构成矩形 2。

选取边缘检测所得数组中的下一个边界点作为初始点, 往复过程 1 和 2, 直到遍历完目标物体所有的边界点, 选择其中面积最大的矩形为检测结果, 算法原理如图 4 所示, 该方法检测效果如图 5 (a) 所示。

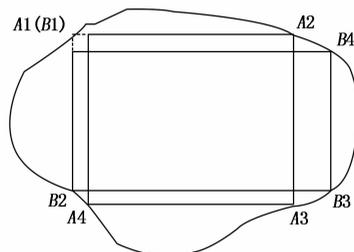


图 4 遍历法检测原理

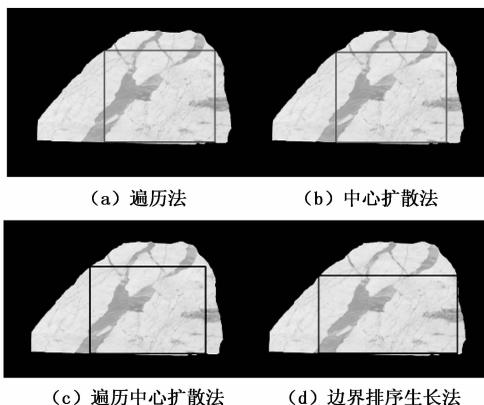


图 5 不同方法检测效果对比

3.2 中心扩散法

3.2.1 确定扩散中心

经过预处理中的平移变换后, 整幅图像的中心就是目标物体的中心点, 因此, 可由公式 (2) 确定目标物体的中心点, 中心点横坐标记为 $center_x$, 纵坐标记为 $center_y$ 。

定义 $left$ 、 $right$ 、 top 、 $bottom$ 四个初始变量, 变量的初始值与中心点的对应关系分别由公式 (8) 所示的关系确定:

$$\begin{cases} top = center_y - 1 \\ bottom = center_y + 1 \\ left = center_x - 1 \\ right = center_x + 1 \end{cases} \quad (8)$$

3.2.2 向外扩散

1) 以 $(left, top)$ 和 $(right, top)$ 两点确定一条直线, 判断该直线是否在目标图像区域内, 即该线段内所有的点的像素值都不为 0, 若该线段在目标图像区域内, 则向上扩展, 即变量 top 减 1, 若该直线已经到达了目标物体的边界, 则停止向上扩展。

2) 以 $(left, top)$ 和 $(left, bottom)$ 两点确定一条直线, 判断该直线是否在目标图像区域内, 即该线段内所有的点的像素值都不为 0, 若该线段在目标图像区域内, 则向左扩展, 即变量 $left$ 减去 1, 若该直线已经到达了目标物体的边界, 则停止向左扩展。

3) 以 $(left, bottom)$ 和 $(right, bottom)$ 两点确定一条直线, 判断该直线是否在目标图像区域内, 即该线段内所有的点的像素值都不为 0, 若该线段在目标图像区域内, 则向上扩展, 即变量 $bottom$ 加 1, 若该直线已经到达了目标物体的边界, 则停止向下扩展。

4) 以 $(right, bottom)$ 和 $(right, top)$ 两点确定一条直线, 判断该直线是否在目标图像区域内, 即该线段内所有的点的像素值都不为 0, 若该线段在目标图像区域内, 则向上扩展, 即变量 $right$ 加 1, 若该直线已经到达了目标物体的边界, 则停止向右扩展。

依次往复 1~4 过程, 直到四条直线都到达物体的边界, 此时这四条直线构成的矩形便是中心扩散法的检测结果。其检测结果如图 5 (b) 所示。

3.3 遍历中心扩散法

该方法的作者认为其同时具有遍历法的准确性和中心扩散法的适用性。原理是首先计算由 (2) 式确定的目标图像中心点的位置, 然后对该点向上、下、左、右移动一些位置来改变中心点的坐标, 利用中心扩散法以这些点为中心点检测内接矩形并记录, 直到遍历完所有的“中心点”, 选取所找到的内接矩形中面积最大的一个作为该方法的检测结果。以向上、下、左、右增加或减少 1 至 3 个像素为例, 共使用了 49 个点, 其检测结果如图 5 (c) 所示。

3.4 边界排序生长法

边界排序生长法的思想是以最快的速度选定一个最有可能成为最大内接矩形的初始矩形, 对其超出的部分进行收缩, 不足的部分进行扩张, 从而达到又快又准的效果。主要分为选择初始矩形、逆生长和生长三个步骤。

3.4.1 选择初始矩形

搜寻目标物体边界点中横坐标与纵坐标之和最大的点和最小的点, 以及横坐标与纵坐标之差最大的点与最小的点。将以上四个点的横坐标与纵坐标分别按从小到大排序, 取其中排第二的横坐标和纵坐标记为 (x_1, y_1) , 排第三的横坐标和纵坐标记为 (x_2, y_2) , 以 (x_1, y_1) 为左上角 (x_2, y_2) 为右下角确定初始矩形。

3.4.2 逆生长

对于不同形状的目标物体, 其选择出的初始矩形不一定会完全落在目标图像区域内。因此, 对于初始矩形超出的部分要进行一系列收缩操作, 该过程类似于中心扩散法的逆过程。

判定 (x_1, y_1) 和 (x_2, y_1) 两点确定的直线是否在目标图像区域内, 即该线段内所有的点的像素值都不为 0, 若已经超出目标图像区域, 则 x_1 加上 1; 判定 (x_1, y_1) 和 (x_1, y_2) 两点确定的直线是否在目标图像区域内, 若

已经超出目标图像区域, 则 y_1 加上 1; 判定 (x_1, y_2) 和 (x_2, y_2) 两点确定的直线是否在目标图像区域内, 若已经超出目标图像区域, 则 x_2 减去 1; 判定 (x_2, y_1) 和 (x_2, y_2) 两点确定的直线是否在目标图像区域内, 若已经超出目标图像区域, 则 y_2 减去 1, 依次重复以上 4 个步骤, 直到该矩形完全进入到目标图像内, 即完成了“逆生长”。

3.4.3 生长

同样地, “逆生长”后的矩形有时也含有一定的生长空间, 既该矩形四条边不一定都能达到目标图像的边界, 因此, 要对该矩形进行相应的扩散操作, 以获得最大的内接矩形。

首先, 将“逆生长”后矩形的左上角的坐标 (x_1, y_1) 与右下角的坐标 (x_2, y_2) 分别对应中心扩散法的四个初始数值 $left$ 、 top 、 $right$ 、 $bottom$, 然后使用中心扩散法对逆生长后的矩形继续扩散, 直到该矩形的四个边都抵达目标物体的边界, 即进行所谓的“生长”。将扩散后的矩形作为边界排序生长法的检测结果。

边界排序生长法的检测过程可由图 6 表示, 图中的不规则五边形代表待检测的目标物体, 左侧矩形区域代表选定的初始矩形, 箭头代表该矩形需要“逆生长”的方向, 中间图片的矩形区域表示初始矩形“逆生长”后的矩形, 周边的箭头表示该矩形需要“生长”的方向, 右侧矩形为左侧矩形沿箭头方向“生长”后的矩形, 即边界排序生长法的检测结果。边界排序生长法的检测结果如图 5 (d) 所示。

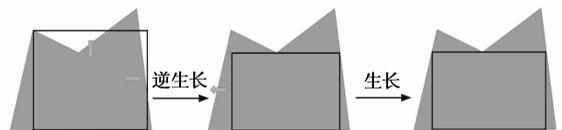


图 6 生长示意图

4 实验结果与分析

本文实验设备为 AMD R31200@3.4 GHz CPU, 32 GB 2 933 MHz 内存, NVIDIA GTX 1660 6 GB 显卡, 64 位 Windows 10 操作系统。实验使用 python3.6 语言在 jupyter notebook 编译环境下结合 OpenCV、time 及 Numpy 模块编程实现。其中使用“cv2.threshold”函数将灰度图转换为二值图, 使用“cv2.findContours”函数获取目标图像的轮廓, 使用“cv2.minAreaRect”函数确定目标图像的最小外接矩形, 使用“cv2.moments”函数计算目标图像的各阶矩并由此推导出目标图像的中心, 使用“cv2.getRotationMatrix2D”函数和“cv2.warpAffine”函数对图像进行平移、旋转变换等^[1]操作。

为了研究遍历中心扩散法不同遍历量对检测面积及速度的影响, 本文采用了以目标物体中心点为中心的 3×3 、 7×7 、 11×11 、 15×15 方格内的所有点作为“中心点”的遍历中心扩散法对 4 种不同形状的板材进行验证, 以矩形包含的像素点数表示检测面积的大小, 检测面积与耗时如表 1 所示。为了研究 4 种方法的检测效果及运行速率, 本文使用 4 种方法分别对 8 种不同形状的板材进行验证, 其中遍

历中心扩散法的遍历量为 225 个点，以矩形包含的像素点数表示检测面积的大小，检测面积与耗时如表 2 所示。

表 1 不同遍历量的遍历中心扩散法检测结果对比

遍历量	检测面积/PX				检测耗时/s			
	板材 1	板材 2	板材 3	板材 4	板材 1	板材 2	板材 3	板材 4
9	199 576	173 055	188 734	376 282	0.034	0.027	0.029	0.039
49	199 794	173 888	189 550	458 658	0.301	0.236	0.252	0.371
121	200 099	174 723	189 952	458 658	0.831	0.649	0.705	1.056
225	200 546	175 560	190 350	458 658	1.623	1.286	1.407	2.089

表 2 4 种方法的检测结果对比

	检测面积/PX			
	遍历法	中心扩散法	遍历中心扩散法	边界排序生长法
板材 1	200 490	199 485	200 546	210 540
板材 2	206 540	172 224	175 560	181 882
板材 3	201 310	188 718	190 350	210 098
板材 4	错误	375 769	458 658	422 624
板材 5	错误	209 763	211 140	495 733
板材 6	328 848	407 652	417 962	644 480
板材 7	错误	209 763	211 140	495 733
板材 8	394 875	223 256	227 052	357 808

	检测耗时/s			
	遍历法	中心扩散法	遍历中心扩散法	边界排序生长法
板材 1	1 343.22	10.64	1 623.21	3.15
板材 2	1 739.31	7.83	1 286.38	1.01
板材 3	975.66	7.02	1 407.96	2.05
板材 4	2 213.91	10.14	2 089.38	3.54
板材 5	799.17	7.61	1 419.11	3.14
板材 6	889.96	10.54	1 953.66	0.99
板材 7	790.03	8.30	1 411.98	3.31
板材 8	3 111.27	8.54	1 447.68	1.22

注:错误代表该矩形超出目标物体边界。

本文通过控制变量法，研究同一石板在不同算法下的检测面积及计算耗时。表 1 结果显示遍历中心扩散法的检测结果与选取的“中心点”数呈正线性相关，随着选取的“中心点”数的增多，该方法的检测结果越接近完美，但其相应的检测耗时也会成倍增长。由图 7 第三行子图和表 2 可知，遍历法有很好的鲁棒性，但在应用于凹多边形时有一定的机率出现错误，分析其原理不难得出遍历法在运用于凹多边形时，矩形的最后一条边有可能因超出目标图像边界而导致出错。并且该算法时间复杂度高，不适合应用与对速度有高需求的场景中。由图 7 第三行结果可以看出，中心扩散法在每个方向上的扩散速度都是相同的，因此在处理长条状的多边形时效果极为不理想。由表 2 和图 7 可以得出，边界排序生长法不仅检测效果好，适用性广，而且检测速度大大领先其他三种方法，能在 5 ms 的时间内完成检测，因此更适合应用于对速度有高要求的不规则多边形最大内接矩形检测场景。

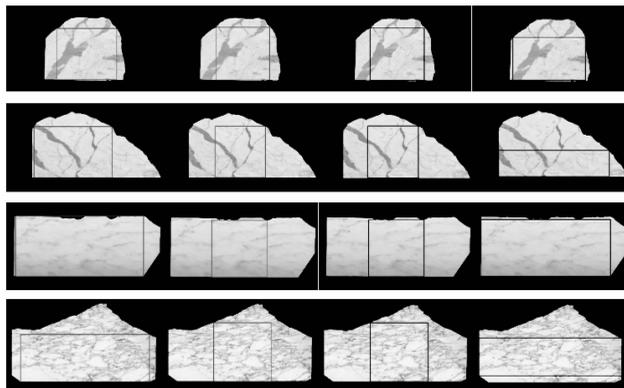


图 7 不同形状石板四种方法检测结果

5 结束语

本文通过分析其它方法发不足之处，思考并设计出了边界排序生长法。该方法不仅检测效果好，而且在计算速度上大大领先文中另外三种方法，值得在高速运行的流水线中推广和应用。此外本文也存在着一定的问题。经分析，本文的不足之处主要体现在边界排序生长法的检测结果仍有一部分提升空间，以及对于遍历中心扩散法是否存在某种中心点的选取规则能使得该算法的检测效果又快又好，将有待未来进行深入的研究。

参考文献:

- [1] 何新宇, 赵时璐, 张震, 等. 机器视觉的研究及应用发展趋势 [J]. 机械设计与制造, 2020 (10): 281-283.
- [2] 赵凤, 范九伦, 潘晓英, 等. 基于灰度和非局部空间灰度特征的二维 Otsu 曲线阈值分割法 [J]. 计算机应用研究, 2012, 29 (5): 1988-2000.
- [3] 孟令龙, 朱铮涛, 李伟业, 等. 大理石板图像分割方法、装置设备及存储介质 [P]. 中国: CN202010250174.9, 2020.
- [4] 吴晓光, 王涤琼, 盛慧. 一种获取图像区域最小外接矩形的算法及实现 [J]. 计算机工程, 2003, 30 (12): 124-125.
- [5] 谢新华, 梁栋, 张香倩, 等. 基于图像处理的目标物体最大内接矩形面积的检测 [J]. 科学技术与工程, 2015, 15 (17): 193-197.
- [6] 袁哲, 王永振, 石怀涛, 等. 基于改进遗传算法的任意图形最大内接矩形求解 [J]. 控制工程, 2016 (3): 400-404.
- [7] Liu Y G, Yu J Z, Han Y H. Understanding the effective receptive field in semantic image segmentation [J]. Springer Science + Business Media, 2018, 77 (17): 22159-22171.
- [8] 单德强, 胡超, 刘飞飞. 关于光斑图像阈值分析及其中心点的提取 [J]. 计算机与数字工程, 2020 (7): 1753-1756.
- [9] 曾建华, 黄时杰. 典型图像边缘检测算子的比较与分析 [N]. 河北师范大学学报 (自然科学版), 2020, 44 (4): 295-301.
- [10] Suzuki S, Abe K. Topological structural analysis of digitized binary images by border following [J]. Computer Vision, Graphics, and Image Processing, 1985, 30 (1): 32-46.
- [11] Joshi P, et al. OpenCV 实例精解 [M]. 北京: 机械工业出版社, 2017.