

一种基于OpenCV的二值图像线状要素坐标识别方法

尹舒祚, 张乃丹, 孙志民, 梁 博

长江大学地球科学学院, 湖北 武汉
Email: 824247383@qq.com

收稿日期: 2020年10月29日; 录用日期: 2020年11月13日; 发布日期: 2020年11月20日

摘 要

针对传统八邻域追踪算法在提取图像像素坐标的问题, 对于规模较大、线状要素数量过多、线状要素空间关系复杂的裂缝图像, 易产生大量计算冗余, 加大时间消耗, 本文将OpenCV轮廓定位融入八邻域追踪算法, 提出了一种基于OpenCV的二值图像线状要素坐标识别方法。通过对裂缝图像实验测试, 在保证提取图像效果基础上, 降低了计算冗余和时间消耗, 极大地提高了运行效率。

关键词

OpenCV, 八邻域追踪, 图像处理, 线状要素

A Coordinate Recognition Method of Linear Elements in Binary Image Based on OpenCV

Shuzuo Yin, Naidan Zhang, Zhimin Sun, Bo Liang

School of Geosciences, Yangtze University, Wuhan Hubei
Email: 824247383@qq.com

Received: Oct. 29th, 2020; accepted: Nov. 13th, 2020; published: Nov. 20th, 2020

Abstract

Aiming at the problem of traditional eight-neighborhood tracking algorithm in extracting image pixel coordinates, for crack images with large scale, too many linear elements and complex spatial relationship of linear elements, it is easy to produce a large amount of computational redundancy and increase time consumption. In this paper, OpenCV contour location is integrated into the

eight-neighborhood tracking algorithm, and a coordinate recognition method of linear elements in binary images based on OpenCV is proposed. Through the experimental test of the crack image, on the basis of ensuring the effect of extracting the image, the computational redundancy and time consumption are reduced, and the running efficiency is greatly improved.

Keywords

OpenCV, Eight-Neighborhood Tracking, Image Processing, Linear Elements

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

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



Open Access

1. 引言

随着计算机技术的发展,线状要素在地理信息科学、计算机视觉以及地质学等众多领域得到了广泛的应用[1][2],将有意义的线状要素获取后,对人们将目标线状要素特征进行进一步的处理和分析具有重要的意义。现阶段基于图像的裂缝识别算法已经相当成熟,李晋惠[3]等人根据路面图像中裂缝像素分布原理,利用“最大距离法”的基本原理提出了基于形态学的裂缝边缘检测算法;李刚[4]等人使用多方位结构的边缘检测算法来对裂缝进行边缘提取;刘玉臣[5]使用数字图像处理技术来对公路的表面的裂缝进行识别,既提高了裂缝识别的效率又减少人工识别的出错率。在调研相关裂缝识别方法后,本文前期工作中利用图像处理边缘检测算法已经识别出某岩石表面裂缝栅格数据,在实际生产应用中,如何精确、高效地将裂缝栅格数据提取成像素坐标,这是本文研究工作的重点。

现阶段,利用一种改进的八邻域追踪算法可以实现线状要素矢量化[6],但是该算法适用范围较窄,无法高效地应用于裂缝图像规模较大、裂缝数量较多、裂缝空间关系较为复杂(存在相交,相切)的情况。为了减少像素形式的背景图像所带来的计算冗余,周晓波[7],杨广林[8]等人将图像分块算法融入其相关实验,提高运行效率,但是图像分块算法存在局限性,在岩石表面裂缝栅格数据提取为像素坐标的过程中,会造成裂缝的断裂,无法兼顾算法的运行效率和提取效果。

计算机视觉库(OpenCV)是一种开源、跨平台的视觉库,可用来实现常用的图像处理及计算机视觉算法,具有独立性强、开放性高、高效率、开发平台自由等优点[9]。该视觉库由一系列C函数和C++类组成,并且提供了Python、MATLAB等多个语言的接口。

针对以上问题,本文结合OpenCV的优势,提出了一种基于OpenCV的二值图像线状要素坐标识别方法,将OpenCV的目标轮廓定位融入八邻域追踪算法,在保证线段连续性的条件下,极大程度地提高了线状要素像素坐标的提取效率。

2. 方法设计

2.1. 总体思路

前期准备工作中,拍摄获取岩石表面RGB图像,利用数字图像处理技术对该图像进行增强操作和分割处理,以消除图像噪声和背景光照不均匀,并突出岩石表面裂缝的主要特征。建立BP神经网络模型,通过训练样本对网络进行训练,用实际岩石裂缝图像验证网络的有效性,得到最终的岩石裂缝RGB图像,该图像像素规模较大、裂缝数量较多、裂缝空间关系复杂。

关键在于如何精确、高效地提取岩石裂缝线状要素像素坐标，本文针对八邻域追踪算法存在大量计算冗余问题，提出并使用基于 OpenCV 区域定位融合八邻域追踪的方法，总体实验设计如图 1。

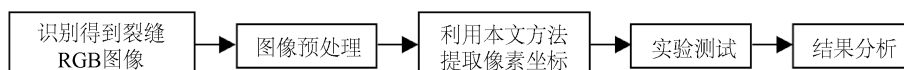


Figure 1. Overall experimental design

图 1. 总体实验设计

使用获取到的岩石裂缝图像，对其进行图像预处理，将裂缝图像最终转为易于计算的二值图像。利用本文提出方法进行像素坐标的提取，并保存输出提取坐标。使用原始八邻域追踪算法、图像分块法和本文方法测试不同像素大小和不同裂缝数量的裂缝图片，通过测试识别速度和识别效果，来验证提出方法的实用性和可行性。

2.2. 二值图像线状要素坐标识别方法

2.2.1. 图像预处理

图像预处理环节主要包含三个部分：图像灰度化、图像二值化和图像细化。

图像灰度化处理是将彩色图像转化为灰度图像。处理三个通道的数据比较复杂，且计算量大，灰度化将裂缝 RGB 图像变为单通道图像，简化矩阵运算。二值化对灰度图像进行处理，使用最大类间方差 (OTSU)，自适应确定阈值，将图像中白色部分(线状目标)像素值变为 255，黑色部分(图像背景)像素值变为 0，便于八邻域追踪查找像素坐标，并将查询到的白色线状要素像素值置为 0，避免重复查找。细化又称为提取骨架，通过对裂缝图像进行细化，得到较为平滑的单像素宽度裂缝线条。

2.2.2. 基于 OpenCV 的线状要素轮廓定位

轮廓定位是一种轮廓扫描技术，可以得到线状目标的具体位置，便于接下来对定位目标提取、分析。OpenCV 是具有强大图像和矩阵运算能力的开源计算机视觉库，可实现图像处理和计算机视觉方面的很多通用算法，是二次开发的理想工具[10]。本文采用 OpenCV 中 findContour 方法来检测裂缝图像中线状要素的轮廓。

findContour 方法中主要包含以下参数。

参数 1: image, 输入图像类型，即预处理中的二值裂缝图像。

参数 2: contours, 保存检测的轮廓对象，轮廓对象为一个点向量，存储到 Point 类型的 vector 中。

参数 3: mode, 轮廓的检索模式。本文只需要检测线状要素的外部轮廓，故采用 CV_RETR_EXTERNAL 方法。

参数 4: method, 轮廓近似方法。用八邻域提取像素坐标时，避免遗漏线状要素与检测轮廓重合的像素坐标点，影响识别裂缝的完整性，故使用 CV_CHAIN_APPROX_NONE 方法，存储所有的轮廓坐标信息。

2.2.3. 八邻域追踪算法

八邻域追踪算法的基本步骤包含：(1) 在单像素宽度的二值图像基础上，遍历图像像素(从上到下，从左到右)，寻找起始点 P0 (白色线条点)；(2) 在起始点的八邻域范围内查找下一个点 P1 (起始点不一定是线条端点，所以分别向两边进行查找)；(3) 将找到的点像素值设为 0 (即将找到线条颜色置为背景色)，并保存该点坐标；(4) 重复步骤(2)~(3)，直到八邻域内没有找到下一个白色点，表示一端搜索结束；(5) 重复上述步骤直到图像内所有线条遍历完毕。

八邻域追踪算法在查找下一个点时会更新记录当前点搜索方向 `currFlag`, 在下次使用八邻域遍历搜索时, 会避免与上一次遍历重复, 减少无用的计算。如图 2。

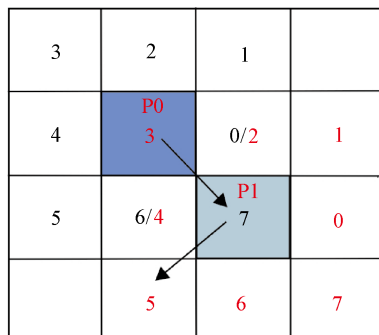


Figure 2. Counterclockwise search in eight neighborhoods

图 2. 八邻域逆时针搜索

当前起始点 P_0 为红色 3, 逆时针遍历一次后找到 P_1 (黑 7)。上图八邻域搜索为 0~7 逆时针方向, 第一次找到 P_1 , 意味着黑色 0~6 不存在像素点, 第二次遍历时, 因为 P_0 、黑色 0 和 6 位置与红色 2、3、4 重合, 不需再次遍历, 故第二次遍历从 5 方向开始, 减少了搜索次数(顺时针方向搜索同理)。

3. 实验过程及结果

3.1. 算法实现

针对传统八邻域追踪过程中运算量大, 易产生计算冗余, 在处理复杂图像的情况下, 时间消耗较大, 提出了一种基于 OpenCV 的二值图像线状要素坐标识别方法, 算法流程图如图 3。传统八邻域算法在提取像素坐标, 搜索顺序是从左到右, 从上到下, 每搜索结束一条线状要素, 会再次从整幅图像首坐标开始搜索。若一幅二值图像有 N 条线状要素, 则全部搜索结束, 会从图像坐标起始点开始搜索 N 次, 直到全图线状要素搜索完毕。本文方法在搜索之前, 在八邻域算法中融入 OpenCV 的轮廓定位, 首先确定每条线状要素的坐标范围, 在坐标范围内开始搜索, 一幅图像搜索结束只需搜索一次, 且裂缝数量越多, 搜索量相对会越小。

使用前期工作中的岩石裂缝 RGB 图像, 具体的工作流程如下:

① 使用 `opencv` 对图像进行预处理, 调用 OpenCV 中的 `cvtColor` 函数对读取图像进行灰度化, 调用 OpenCV 中的 `threshold` 函数对灰度化图像进行二值化处理, 具体函数细节如下:

```
cvtColor(srcImg, srcImg, COLOR_BGR2GRAY); //灰度化
```

```
threshold(srcImg, srcImg, 100, 255, THRESH_BINARY); //二值化
```

② 获取每个轮廓对角坐标信息, 调用 `boundingRect` 函数, 该函数有四个返回值 x , y , w , h , 分别对应轮廓左上角 x , y 坐标和轮廓的宽高, $(x+w, y+h)$ 即对应右下角坐标值, 通过这些返回值可以计算出轮廓的坐标范围。

③ 轮廓内八邻域搜索, 首先判断像素坐标是否在轮廓范围内, 如果在搜索范围内, 则进行八邻域追踪像素坐标。判断代码(伪代码)如下:

```
if (当前坐标在轮廓范围内)
```

```
for (当前线条外接矩形行数>i)
```

```
指针指向第 i 行
```

```

for (当前线条外接矩形列数>j)
  获取第 i 行第 j 列像素坐标
  if (当前像素坐标等于 255)
    获取该点坐标, 并将像素值置为 0
  end
end
end
④ 保存提取的像素坐标
⑤ 输出识别的 RGB 裂缝图像文件

```

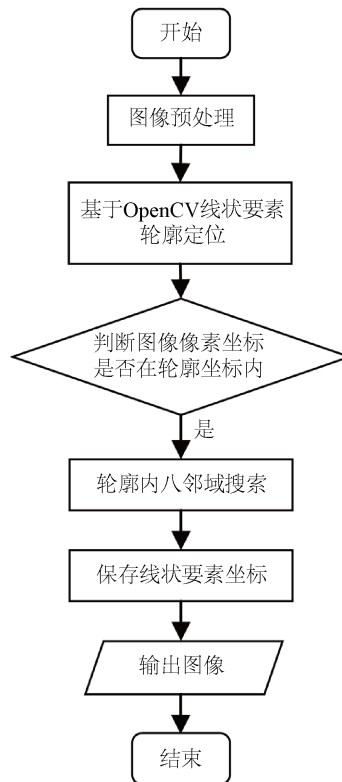


Figure 3. Algorithm flow chart
图 3. 算法流程图

3.2. 结果分析

在操作平台: AMD A10-7400P Radeon R6, CPU 四核, 2.5 GHz, Visual Studio 2019 上分别使用原八邻域算法、图像分块法、本文提出方法进行裂缝二值图像像素坐标提取, 对比不同方法的识别效果和时间消耗。

三种方法图像识别效果测试结果(部分图像)如图 4。

从图 4 中可以看出, 在最终提取效果上, 图像分块法和本文方法整体上相似。但是图像分块法在分块处, 会造成连续线状要素像素点断裂, 影响最终效果(断裂放大图如图 5); 本文方法首先定位要素坐标范围, 在线状要素框内识别, 未对图像进行分割, 保留了原图像的要素形状完整性。

测试不同裂缝规模图像的时间消耗, 每个图像取三次计算时间的平均值。三个方法时间消耗测试结果如图 6。

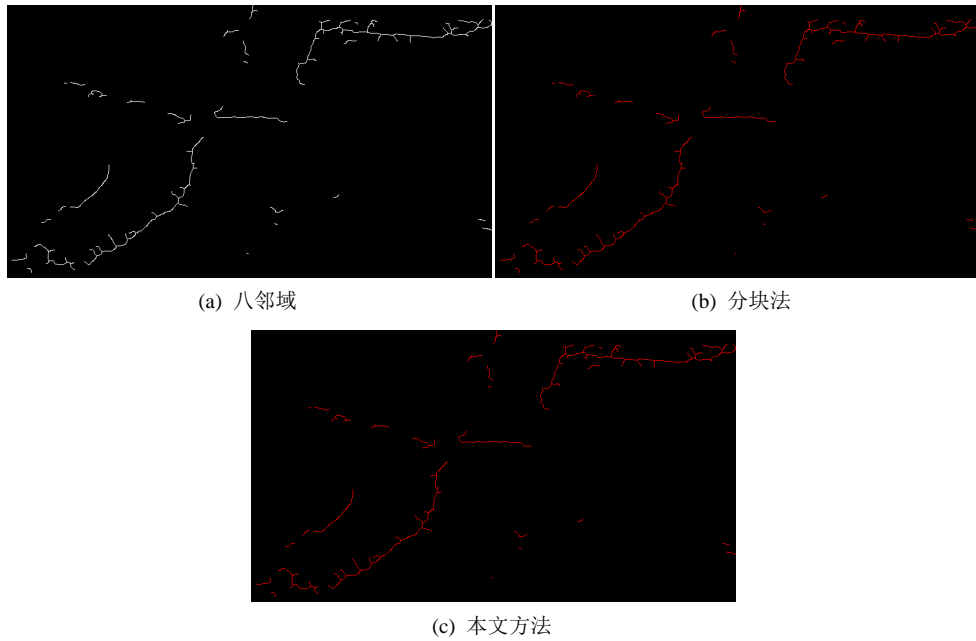


Figure 4. Test result graph
图 4. 测试结果图



Figure 5. Schematic diagram of continuous linear element fracture
图 5. 连续线状要素断裂示意图

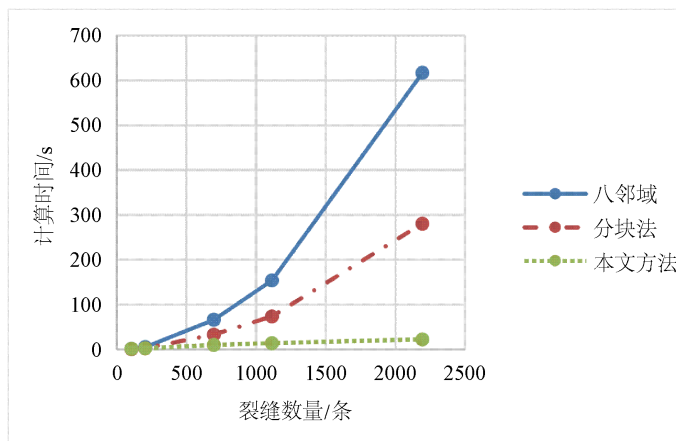


Figure 6. Time comparison diagram of three methods
图 6. 三种方法时间对比图

在同等硬件条件上, 计算用时和原方法相比较, 图像分块(CPU 为四核, 图像分为四块)运算速度降低了一半左右。本文提出方法在测试过程中, 结果如图 6, 第三组提取时间节省了 85.26%, 第四组节省了 91.17%, 第五组甚至达到了 96.39%。可以看出, 随着图像分辨率越大, 图像中需要处理的数据越多, 时间上消耗差别就会越大, 越体现出本文方法的优越性。

4. 结论

本文提出的基于 OpenCV 的二值图像线状要素坐标识别方法, 采用 Visual C++ 语言将轮廓定位融入八邻域追踪算法, 降低了原算法的计算冗余, 对于图像规模较大的裂缝图像, 可以快速获取其像素坐标。

利用 OpenCV 强大的矩阵运算能力, 使用轮廓定位, 而不将图像进行分块, 保证像素坐标提取完整性前提下, 极大地提高了运行速度。

参考文献

- [1] 胡瑞, 樊养余, 郝重阳, 黄志理. 基于线状要素提取的等高线图像矢量化方法研究[J]. 计算机工程与科学, 2006(4): 50-52.
- [2] 刘彦婷. 岩体裂缝动态识别算法研究[D]: [硕士学位论文]. 赣州: 江西理工大学, 2012.
- [3] 李晋惠, 楼伟, 姜寿山. 基于 CCD 的公路路面病害检测技术研究[J]. 西安工业学院学报, 2002(2): 95-99.
- [4] 李刚, 贺昱曜. 多方位结构元素路面裂缝图像边缘检测算法[J]. 计算机工程与应用, 2010, 46(1): 224-226.
- [5] 刘玉臣. 公路路面裂缝图像自动识别技术研究[D]: [硕士学位论文]. 长春: 吉林大学, 2005.
- [6] 梁博, 曾齐红, 刘远刚, 邓帆, 邵燕林. 基于 C++ 与 MATLAB 混合编程的露头表面裂缝识别及矢量化方法[J]. 计算机时代, 2020(7): 72-75+79.
- [7] 周晓波, 程海龙, 贾琦. 图像分块融合算法速度优化处理研究[J]. 北京交通大学学报, 2014, 38(5): 33-36.
- [8] 杨广林, 孔令富. 基于图像分块的背景模型构建方法[J]. 机器人, 2007(1): 29-34.
- [9] 张宇飞, 张义顺. 基于 OpenCV 的焊缝图像识别算法研究[J]. 焊接技术, 2020, 49(5): 12-14.
- [10] 梅迎春, 王朝立, 廖勤武. 基于 OpenCV 的摄像机标定的应用研究[J]. 计算机工程与设计, 2009, 30(16): 3856-3858+3862.