



(12) 发明专利

(10) 授权公告号 CN 112419250 B

(45) 授权公告日 2023.06.16

(21) 申请号 202011265910.4

G06T 5/00 (2006.01)

(22) 申请日 2020.11.13

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 107230202 A, 2017.10.03

申请公布号 CN 112419250 A

CN 110264459 A, 2019.09.20

CN 111145161 A, 2020.05.12

(43) 申请公布日 2021.02.26

US 2013155061 A1, 2013.06.20

(73) 专利权人 湖北工业大学

CN 109919942 A, 2019.06.21

地址 430068 湖北省武汉市洪山区南李路
28号

CN 111709944 A, 2020.09.25

谢芳.《大学计算机基础》.北京邮电大学出版社,2010,第218页.

(72) 发明人 朱清波 董李 李婉 秦筠婷

Ciprian Sorandaru etc.Detecting Road Irregularities by Image Enhancements and Thresholding.《IEEE EUROCON 2019 -18th International Conference on Smart Technologies》.2019,全文.

李磊 段志飞 刘洋 陈前

(74) 专利代理机构 武汉科皓知识产权代理事务所(特殊普通合伙) 42222

专利代理师 肖明洲

严盛龙;徐飞鸿;.基于亚像素算法的混凝土结构裂缝测量.交通科学与工程.2017,(第04期),全文.

(51) Int.Cl.

G06T 7/00 (2017.01)

G06T 7/10 (2017.01)

G06T 7/60 (2017.01)

G06T 7/62 (2017.01)

审查员 刘江平

权利要求书5页 说明书15页 附图11页

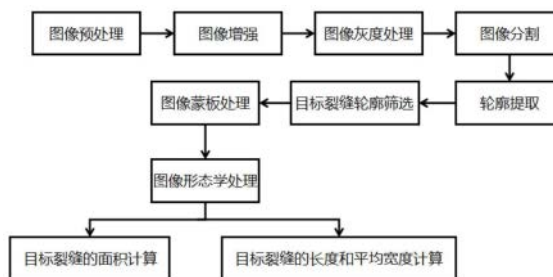
(54) 发明名称

路面裂缝数字图像提取、裂缝修补及裂缝参数计算方法

准度。

(57) 摘要

本发明公开了一种路面裂缝数字图像提取、裂缝修补及裂缝参数计算方法,首先路面裂缝图像的预处理;基于伽马变换的图像增强;图像灰度处理;基于自适应阈值法的图像阈值分割;图像的轮廓提取;目标裂缝轮廓筛选;图像蒙板处理;图像形态学处理;最后目标裂缝的面积和长度计算。本发明计算出的裂缝长度为像素长度,若要获得图像中裂缝的真实长度,则应知道一幅图像所拍摄区域的真实尺寸和数字图像的分辨率。本发明可以在具有噪声干扰的图像中提取出所需目标裂缝,大大减少了在抑制噪声过程中对图像细节的影响,较好地保存了裂缝的轮廓细节,提高了裂缝面积、长度和平均宽度计算的精



1. 一种路面裂缝数字图像提取方法,其特征在于,包括以下步骤:

步骤1:路面裂缝图像预处理;

采集并读取路面裂缝图像,并将路面裂缝图像进行缩放处理;

步骤2:图像增强;

对路面裂缝图片进行处理,将相机过曝或者曝光不足的图片进行修正,并且对裂缝图像进行增强突出,便于图像分割时的阈值选择;

步骤3:图像灰度处理;

根据RGB和YUV颜色空间的变化关系来建立亮度Y与R、G、B三个颜色分量的对应关系,以亮度值表达图像的灰度值,得到灰度图像;

步骤4:图像分割;

对裂缝灰度图像进行降噪处理,然后将图像进行二值化处理;

步骤5:轮廓提取;

将图像中的裂缝图像以及噪声全部进行轮廓提取并标上序号,并获得提取出的所有轮廓的像素坐标;计算每个轮廓所围成的像素面积;

所述轮廓提取,是使用光栅扫描法对整个图像的像素点进行扫描,即从左往右、由上往下,先扫描完一行,再移至下一行起始位置继续扫描;

从边界开始点 (i, j) 以边界跟踪算法得到一条边界,为每条新找到的边界B赋予一个新的唯一的编号,NBD表示当前跟踪的边界的编号;在光栅扫描的过程中,保存最近遇到的边界B'的编号,记为LNBD;每当扫描到图片的新行的起始位置时,将LNBD重置为1;

假设输入图像为 $F = \{f_{ij}\}$,将初始的NBD设为1,即把图像F的图像框架看成第一个边界;使用光栅扫描法扫描图像F,当扫描到某个像素点 (i, j) 的灰度值 f_{ij} 不为0时执行以下步骤;

(1) 将 f_{ij} 分三种情况讨论:

(a) 如果 $f_{ij} = 1$ 并且 $f_{i, j-1} = 0$,则 (i, j) 是外边界开始点,NBD加1,且将像素坐标 $(i, j-1)$ 赋给坐标 (i_2, j_2) ;

(b) 如果 $f_{ij} \geq 1$ 并且 $f_{i, j+1} = 0$,则 (i, j) 是孔边界开始点,NBD加1,且令 $i_2 = i, j_2 = j+1$;如果 $f_{ij} > 1$,则将 f_{ij} 的值赋给LNBD;

(c) 其他情况,则执行步骤(4);

(2) 根据上一个边界B'和当前新遇到边界B的类型,从边界类型表得到当前边界B的父边界;

(3) 从边界开始点 (i, j) 开始按步骤3.1到步骤3.5进行边界跟踪;

(3.1) 以 (i, j) 中心, (i_2, j_2) 为起始点,按顺时针方向查找 (i, j) 的4或8邻域是否存在非0像素点;若找到非0像素点,则令 (i_1, j_1) 是顺时针方向的第一个非0像素点;否则令 $f_{ij} = -NBD$,转到步骤(4);

(3.2) 将像素点的坐标 (i_1, j_1) 赋给坐标 (i_2, j_2) ,坐标 (i, j) 赋给坐标 (i_3, j_3) ;

(3.3) 以 (i_3, j_3) 为中心,按逆时针方向, (i_2, j_2) 的下一个点为起始点查找 (i_3, j_3) 的4或8邻域是否存在非0像素点;

(3.4) 分三种类型进行讨论:

(a) 如果 (i_3, j_3+1) 是步骤3.3中已经检查过的像素点且是0像素点,则将 $-NBD$ 的值赋给像素点 (i_3, j_3) 的灰度值 f_{i_3, j_3} ;

(b) 如果 (i_3, j_3+1) 不是步骤3.3中已经检查过的0像素点, 并且 $f_{i_3, j_3} = 1$, 则将 -NBD 的值赋给像素点 (i_3, j_3) 的灰度值 f_{i_3, j_3} ;

(c) 其他情况, 不改变 f_{i_3, j_3} ;

(3.5) 如果 $(i_4, j_4) = (i, j)$ 且 $(i_3, j_3) = (i_1, j_1)$, 即回到了边界开始点, 则转到步骤4; 否则令将像素点的坐标 (i_3, j_3) 赋给坐标 (i_2, j_2) , 坐标 (i_4, j_4) 赋给坐标 (i_3, j_3) , 转到步骤3.3;

(4) 如果 $f_{i, j} \neq 1$, 则将 $|f_{i, j}|$ 的值赋给LNBD, 从点 $(i, j+1)$ 继续光栅扫描; 当扫描到图片的右下角顶点时结束;

所述计算每个轮廓所围成的像素面积 S_D , 采用格林公式法, 其面积公式为:

$$S_D \approx \frac{1}{2} \sum_{k=1}^n \int_{P_{k-1}P_k} -ydx + xdy = \frac{1}{2} \sum_{k=1}^n \begin{vmatrix} x_{k-1} & x_k \\ y_{k-1} & y_k \end{vmatrix};$$

式中, 设需测面积在xoy平面上, D是xoy平面上的多连通闭区域, P_k 是该平面上的第k个坐标点, 其中k为整数, 取值范围为 $1 \leq k \leq n$, 该坐标点的坐标为 (x_k, y_k) ; 按照坐标点进行连线, 将曲线近似用直线段表示, $\overline{P_{k-1}P_k}$ 表示相邻顺序两点连接所得的直线段, n是坐标点的总数; 步骤6: 目标裂缝轮廓筛选;

对所有裂缝的面积进行排序, 通过面积数据筛选出目标裂缝的轮廓的序号, 然后, 获取目标裂缝轮廓的像素坐标和所围成的像素面积;

步骤7: 图像蒙板处理;

取一张纯黑色的图像, 将裂缝轮廓的像素坐标赋值到纯色图像上, 并对轮廓内部进行颜色填充, 填充颜色与背景颜色相反; 然后, 获取图像中裂缝所占的矩形区域, 将其他多余的区域裁剪掉, 获得路面裂缝数字图像。

2. 根据权利要求1所述的路面裂缝数字图像提取方法, 其特征在于: 步骤1中, 采用光电摄像头对沥青或者水泥路面裂缝进行现场采集, 摄像头安装在车的底部, 拍摄角度与路面垂直; 采用车载计算机按照一定频率对路面图像进行自动拍摄, 采集路面裂缝、车辙、路面结构信息; 采用JPG格式和PNG格式作为路面图像读取的格式;

对图像的长和宽进行调整, 以适应屏幕, 其缩放表达式为:

$$X_1 = fx * src.cols;$$

$$Y_1 = fy * src.rows;$$

式中, $src.cols$ 、 $src.rows$ 分别表示原始图像的宽和长, fx 表示水平轴上的比例因子, fy 表示垂直轴上的比例因子, X_1 、 Y_1 分别表示处理后图像的长和宽, fx 与 fy 相等时可实现对图像的等比例缩放。

3. 根据权利要求1所述的路面裂缝数字图像提取方法, 其特征在于: 步骤2中, 采用伽马变换算法对路面裂缝图片进行处理;

所述伽马变换算法具体为: 对整个图像的所有像素点的值使用伽马变换公式进行处理, 伽马变换的彩色图像公式为:

$$f(R, G, B) = C * h(R, G, B)^\gamma;$$

式中, C 为常数, $h(R, G, B)$ 为原始图像像素点的R、G、B通道的亮度值, $f(R, G, B)$ 为处理后的像素点的R、G、B通道的亮度值, γ 为伽马系数。

4. 根据权利要求1所述的路面裂缝数字图像提取方法, 其特征在于: 步骤6中, 采用冒泡

排序法将轮廓的面积进行排序,由大到小排序的冒泡排序法的运行过程如下:

(1) 比较相邻的数字,如果第一个比第二个小,则交换这两个数字;

(2) 对每一对相邻数字做同样的处理,从开始第一对到结尾的最后一对,处理完一遍后,最后一个数字会是最小数,这个数的排序已经完成;

(3) 针对所有的数字重复以上步骤,除了最后已经排序好的数字;

(4) 持续每次对越来越少的数字重复上面步骤,直到没有任何一对数字需要比较,则序列最终有序。

5. 根据权利要求1-4任意一项所述的路面裂缝数字图像提取方法,其特征在于:步骤7中,所述图像蒙板处理,是通过目标裂缝的轮廓坐标获取,在新图像中按坐标信息对相应像素点重新赋值、裁剪图像的方式,获得一张只含有完整的目标裂缝图像;具体步骤如下:

(1) 选取所需要的目标裂缝轮廓,获取目标裂缝轮廓的像素坐标;

(2) 导入一张黑色背景的纯色图,该图的长和宽都要比正在处理的路面裂缝图片大,将轮廓像素坐标按1:1的比例映射到纯色图上,并将这些点的像素值改为255,如果目标裂缝覆盖的面积中有完好的路面,则需要对轮廓目标进行分别提取,将所有裂缝轮廓目标整合到一张图片中;

(3) 对纯色图上的轮廓内部进行颜色填充,填充颜色为白色,像素值为255;

(4) 以步骤(3)选择的这个点为起点,将它压入栈中,所要填充的颜色为白色,因此将该点设置为白色,判断这个点的四邻域像素;然后,设置一个颜色阈值T,假设当前像素灰度值为 $P(x,y)$,四邻域像素为 $M(n)$, $n=1,2,3,4$,以该阈值T判断当前像素与四邻域像素的灰度差值 $D=|P-M|$,如果该差值D小于阈值T,则把这个像素点M作为下一个种子点,压入栈中,否则继续判断;当栈为空时,则种子填充结束,否则重复步骤(4);

(5) 采用最小矩形边框选择法,从图像左上角的像素点开始,依照高和宽绘制轮廓的垂直边界最小矩形,矩形是与图像的上下边界平行的,直到目标像素点不处在最小矩形外为止;

(6) 使用ROI选择和图像裁剪,在最小矩形框选处理后,得到最小矩形的四个边界的坐标点,这些坐标点将作为ROI区域的坐标,根据ROI的坐标直接将此矩形区域围成的图像从原图中扣出,从而获得处理数据最小的含有目标裂缝的图像。

6. 一种路面裂缝数字图像中裂缝修补方法,基于权利要求1-5任意一项所述的方法获得的路面裂缝数字图像;其特征在于:对传统的闭运算进行改进,通过膨胀、滤波和腐蚀的方法,对路面裂缝数字图像中断裂的裂缝进行修补;

改进后的闭运算基本步骤为:

(1) 对路面裂缝的二值化图像进行膨胀处理,邻域大小选为 $(M1,M1)$,处理次数为 $N1$ 次, $M1$ 和 $N1$ 的取值由所需连接裂缝的间距决定;

(2) 使用中值滤波器对膨胀后的图像进行滤波,中值滤波所选的模板大小由膨胀处理后的裂缝最窄宽度决定,要求滤波后的裂缝图像没有出现断裂带;

(3) 针对步骤2中所述中值滤波的特性,对图像进行腐蚀处理,邻域大小选为 $(M2,M2)$,处理次数为 $N2$,其中 $M2$ 与 $M1$ 要相等, $N2$ 的取值需要比 $N1$ 小。

7. 根据权利要求6所述的路面裂缝数字图像中裂缝修补方法,其特征在于:通过图像细化算法提取出裂缝的骨架,首先对对二值化图像进行颜色反转,然后采用索引表图像细化

算法;

细化过程需要满足两个条件,一是在细化过程中,图像应该有规律地缩小;二是在图像逐步缩小的过程中,应当使图像的连通性质保持不变;在图像细化中,要判断一个点能否去掉,是根据它的八个相邻点的情况来判断的,八个点组成六种不同的组成情况,得出以下四个判据:

- (1) 内部点不能删除;
- (2) 孤立点不能删除;
- (3) 直线端点不能删除;
- (4) 如果P是边界点,去掉P后,如果连通分量不增加,则P删除;

所述索引表图像细化算法是指做出一张索引表,这个表含有需要细化的点的八个邻域的情况,该算法需要不断地遍历边缘,按照索引表对邻域点进行是否删除判断,直到边缘的每个点都不能再被删除为止;

索引表的查表方法是:设白点为1,黑点为0,边缘点为P,则P的八连通区域为:

$$\begin{bmatrix} P0 & P1 & P2 \\ P3 & P & P4 \\ P5 & P6 & P7 \end{bmatrix};$$

对于这些点,需要对其赋予不同的价值,若P点八邻域的某个点为黑色,则认为其价值为0,如果为白色,则按照权值表赋予对应的价值,权值表为:

$$\begin{bmatrix} P0=1 & P1=2 & P2=4 \\ P3=8 & P & P4=16 \\ P5=32 & P6=64 & P7=128 \end{bmatrix};$$

该表经过计算后得到如下表达式:

$$M=P0*1+P1*2+P2*4+P3*8+P4*16+P5*32+P6*64+P7*128;$$

式中,M表示计算结果,通过该计算结果查找索引表中对应的数值,即可知道该点是否需要删除,用这个方法能把所有点的情况映射到0~255的索引表中。

8. 一种路面裂缝参数计算方法,基于权利要求6-7任意一项所述的方法获得的路面裂缝数字图像;其特征在于:所述路面裂缝参数包括目标裂缝的面积、裂缝长度和裂缝平均宽度计算;

读取轮廓提取时轮廓围成的像素面积即可得到目标裂缝的面积;通过图像细化算法,提取出图像骨架,展现出裂缝的形态信息,经过细化处理后的二值化图像,其裂缝像素是黑色的,灰度值为0;背景是白色的,灰度值为255;然后,对图像进行遍历运算,按照骨架像素点的间距以及是否为背景像素点,将所有像素点进行分类,按下列公式进行裂缝长度计算:

$$\text{Length}=1 \bullet X + \sqrt{2} \bullet Y + 0 \bullet Z;$$

式中,Length表示裂缝的长度,X表示与相邻骨架像素点距离为1的像素点总数,Y表示与相邻骨架像素点的距离为 $\sqrt{2}$ 的像素点总数,Z表示背景像素点的总数;

裂缝平均宽度计算公式为:

$$\text{Width} = \frac{\text{Area}}{\text{Length}};$$

式中,Width表示裂缝的平均宽度,Area表示图像中裂缝的面积,Length表示裂缝的长度。

9. 根据权利要求8所述的路面裂缝参数计算方法,其特征在于:所述对图像进行遍历运算,是采用指针遍历算法,对图像中的像素点按照既定规则进行分类,将中心点为黑色的两种不同分类的像素点按照所对应间距进行累加求和,将会得到更为精确的裂缝长度;

指针遍历法的算法步骤为:

(1) 令图像中的任一像素点的坐标为 $R[i, j]$,其中 i 表示横坐标值, j 表示竖坐标值,图像从左上角开始进行遍历运算;则它的八邻域的像素点的坐标按照八邻域关系图表示;

(2) 根据八邻域的坐标关系,取每个像素的中心点作为衡量像素距离的因素,将像素点分为三类:

中心点和邻域点的距离为1的点有: $R[i, j-1]$ 、 $R[i-1, j]$ 、 $R[i+1, j]$ 、 $R[i, j+1]$;

中心点和邻域点的距离为 $\sqrt{2}$ 的点有: $R[i-1, j-1]$ 、 $R[i+1, j-1]$ 、 $R[i-1, j+1]$ 、 $R[i+1, j+1]$;

中心点的八邻域区域没有黑色像素点,这种情况则将该点视为背景点;

(3) 对以上三种情况进行分类统计,并进行裂缝长度计算;

$$\text{Len} = N_{str} * 1 + N_{sla} * \sqrt{2};$$

式中, N_{str} 为中心点和邻域点距离为1的点数, N_{sla} 为中心点和邻域点距离为 $\sqrt{2}$ 的点数,Len为裂缝的总长度。

路面裂缝数字图像提取、裂缝修补及裂缝参数计算方法

技术领域

[0001] 本发明属于Python-OpenCV图像处理技术领域,涉及一种道路裂缝检测方法,具体涉及一种路面裂缝数字图像提取、能够保留裂缝边缘细节且具有修补裂缝断裂区域及裂缝参数计算的方法。

背景技术

[0002] 随着我国经济的成长和交通运输行业的不断发展,道路运营交通量和交通荷载等级都发生了变化,水泥路面和沥青路面是目前两种最为常用的路面结构层,其中沥青路面由于具有表面平整无缝、产生噪音低、施工期短、养护维修简便等优势,在较高等级的道路上广泛应用。但沥青路面容易受到气候环境的作用,也会受到车辆荷载的影响,逐渐产生各种路面病害,裂缝便是常见的路面病害之一。裂缝的种类可以大致分为横向裂缝、纵向裂缝、分叉裂缝、网状裂缝和不规则裂缝等。各种裂缝的产生,会降低道路的使用寿命,如不及时检测并修理,则会影响到行车安全,因此路面裂缝的识别和检测显得非常重要。

[0003] 目前,国内的公路路面病害检测主要是人工检测,即使用人的肉眼以及人工仪器检测,这种方式存在检测效率低、检测结果错误率高、劳动强度大等缺点。随着图像处理技术的发展,具有高效、准确、实时的路面病害检测技术必将获得广阔市场前景,但目前路面图像光照不均、裂缝形状多样、背景噪声大是路面实景图像检测的严重问题,现有的部分图像处理算法虽然可以解决上面部分问题,但却大多会对目标裂缝的形态、边缘、面积数据等产生较大影响,其计算精度、裂缝细节保留、裂缝断裂带处理相对较差。

发明内容

[0004] 为了解决上述技术问题,本发明通过一种具有面积筛选和边缘检测的目标裂缝提取方法,使得提取出的目标裂缝具有更加完整的细节,从而提高裂缝面积等数据计算的精度。

[0005] 本发明首先提供了一种路面裂缝数字图像提取方法,其特征在于,包括以下步骤:

[0006] 步骤1:路面裂缝图像预处理;

[0007] 采集并读取路面裂缝图像,并将路面裂缝图像进行缩放处理;

[0008] 步骤2:图像增强;

[0009] 对路面裂缝图片进行处理,将相机过曝或者曝光不足的图片进行修正,并且对裂缝图像进行增强突出,便于图像分割时的阈值选择;

[0010] 步骤3:图像灰度处理;

[0011] 根据RGB和YUV颜色空间的变化关系来建立亮度Y与R、G、B三个颜色分量的对应关系,以亮度值表达图像的灰度值,得到灰度图像;

[0012] 步骤4:图像分割;

[0013] 对裂缝灰度图像进行降噪处理,然后将图像进行二值化处理;

[0014] 步骤5:轮廓提取;

[0015] 将图像中的裂缝图像以及噪声全部进行轮廓提取并标上序号,并获得提取出的所有轮廓的像素坐标;计算每个轮廓所围成的像素面积;

[0016] 步骤6:目标裂缝轮廓筛选;

[0017] 对所有裂缝的面积进行排序,通过面积数据筛选出目标裂缝的轮廓的序号,然后,获取目标裂缝轮廓的像素坐标和所围成的像素面积;

[0018] 步骤7:图像蒙板处理;

[0019] 取一张纯黑色的图像,将裂缝轮廓的像素坐标赋值到纯色图像上,并对轮廓内部进行颜色填充,填充颜色与背景颜色相反;然后,获取图像中裂缝所占的矩形区域,将其他多余的区域裁剪掉,获得路面裂缝数字图像。

[0020] 本发明还提供了一种路面裂缝数字图像中裂缝修补方法,基于获得的路面裂缝数字图像;其特征在于:对传统的闭运算进行改进,通过膨胀、滤波和腐蚀的方法,对路面裂缝数字图像中断裂的裂缝进行修补;

[0021] 改进后的闭运算基本步骤为:

[0022] (1)对路面裂缝的二值化图像进行膨胀处理,邻域大小选为 $(M1, M1)$,处理次数为 $N1$ 次, $M1$ 和 $N1$ 的取值由所需连接裂缝的间距决定;

[0023] (2)使用中值滤波器对膨胀后的图像进行滤波,中值滤波所选的模板大小由膨胀处理后的裂缝最窄宽度决定,要求滤波后的裂缝图像没有出现断裂带;

[0024] (3)针对步骤2中所述中值滤波的特性,对图像进行腐蚀处理,邻域大小选为 $(M2, M2)$,处理次数为 $N2$,其中 $M2$ 与 $M1$ 要相等, $N2$ 的取值需要比 $N1$ 小。

[0025] 本发明还提供了一种路面裂缝参数计算方法,基于获得的路面裂缝数字图像;其特征在于:所述路面裂缝参数包括目标裂缝的面积、裂缝长度和裂缝平均宽度计算;

[0026] 读取轮廓提取时轮廓围成的像素面积即可得到目标裂缝的面积;通过图像细化算法,提取出图像骨架,展现出裂缝的形态信息,经过细化处理后的二值化图像,其裂缝像素是黑色的,灰度值为0;背景是白色的,灰度值为255;然后,对图像进行遍历运算,按照骨架像素点的间距以及是否为背景像素点,将所有像素点进行分类,按下列公式进行裂缝长度计算:

[0027] $Length = 1 \cdot X + \sqrt{2} \cdot Y + 0 \cdot Z$;

[0028] 式中, $Length$ 表示裂缝的长度, X 表示与相邻骨架像素点距离为1的像素点总数, Y 表示与相邻骨架像素点的距离为 $\sqrt{2}$ 的像素点总数, Z 表示背景像素点的总数;

[0029] 裂缝平均宽度计算公式为:

[0030] $Width = \frac{Area}{Length}$;

[0031] 式中, $Width$ 表示裂缝的平均宽度, $Area$ 表示图像中裂缝的面积, $Length$ 表示裂缝的长度。

[0032] 本发明提供了一种能够在具有背景干扰项和噪点的情况下提取出裂缝轮廓的方法,可以避免滤波对图像边缘细节的破坏,使得裂缝图像具有更为完整的边缘细节,现有技术大多通过滤波算法去除图像中的背景干扰和噪点,本方法可以大大减少滤波造成的细节模糊,使得处理后的二值化裂缝图像的形态与原图更为接近,提升了裂缝面积、长度等计算

的精度。本发明还提供了一种修补裂缝断裂区域的方法,相对于常用的闭运算修补断裂带的方法,本方法可以修补相距更远的断裂带,得到边缘更为平滑的裂缝边缘,与原始裂缝图像的形态更为接近。

附图说明

- [0033] 图1是本发明实施例的技术路线图。
- [0034] 图2是本发明实施例中采集路面裂缝图像缩放示意图。(a)原图 (b)缩放 0.5倍图
- [0035] 图3是本发明实施例中采集路面裂缝图像伽马变换示意图。
- [0036] 图4是本发明实施例中采集路面裂缝图像灰度处理示意图。
- [0037] 图5是本发明实施例中样本图像经灰度处理前后像素统计图的对比。
- [0038] 图6是本发明实施例中采集路面裂缝样本图像经均值滤波、中值滤波、高斯滤波、双边滤波四种滤波方式与原样本图像的比较。
- [0039] 图7是本发明实施例中全局阈值法的处理效果图。
- [0040] 图8是本发明实施例中最大类间差法的处理效果图。
- [0041] 图9是本发明实施例中局部阈值法的处理效果图。
- [0042] 图10是本发明实施例中全局阈值法、最大类间差法、局部阈值法三种阈值法处理效果比较。
- [0043] 图11是本发明实施例中采集路面裂缝图像中值滤波后的二值化图像。(a) 原图 (b)中值滤波后。
- [0044] 图12是本发明实施例中由于中值滤波所选的二维模板过大造成二值化分叉裂缝发生断裂的图像。
- [0045] 图13是本发明实施例中对目标图像进行腐蚀和膨胀后的效果演示图。
- [0046] 图14是本发明实施例中对目标图像进行闭运算处理后的结果。
- [0047] 图15是本发明实施例中对目标图像改进后闭运算的处理过程示例。
- [0048] 图16是本发明实施例中轮廓提取时的边界类型表。
- [0049] 图17是本发明实施例中对目标图像进行轮廓提取后的图像。
- [0050] 图18是本发明实施例中多连通闭区域示意图。
- [0051] 图19是本发明实施例中闭区域所取坐标点示例。
- [0052] 图20是本发明实施例中对路面裂缝进行面积筛选后的结果图。
- [0053] 图21是本发明实施例中对网状裂缝进行面积筛选后的结果图。
- [0054] 图22是本发明实施例中对裂缝图像轮廓目标提取前后对比图。
- [0055] 图23是本发明实施例中对裂缝图像填充前后对比图。
- [0056] 图24是本发明实施例中网状裂缝图像填充处理过程。
- [0057] 图25是本发明实施例中对裂缝目标的最小矩形框选处理。
- [0058] 图26是本发明实施例中经过裁剪后的裂缝图像。
- [0059] 图27是本发明实施例中对裂缝图像颜色反转前后图像对比。
- [0060] 图28是本发明实施例中图像细化中八个相邻点的组成情况示意图。
- [0061] 图29是本发明实施例中图像细化算法的索引表。
- [0062] 图30是本发明实施例中经过索引表细化算法处理后,得出的裂缝图像骨架。

- [0063] 图31是本发明实施例中黑色像素的两种分布演示图。
 [0064] 图32是本发明实施例中八邻域的坐标关系图。
 [0065] 图33是本发明实施例中样本网状裂缝的骨架图像。

具体实施方式

[0066] 为了便于本领域普通技术人员理解和实施本发明,下面结合附图及实施例对本发明作进一步的详细描述,应当理解,此处所描述的实施例仅用于说明和解释本发明,并不用于限定本发明。

[0067] 请见图1,本实施例提供一种路面裂缝数字图像提取方法,包括以下步骤:

[0068] 步骤1:路面裂缝图像预处理;

[0069] 本实施例中,首先采集路面裂缝图像,然后读取路面裂缝图像,最后,将路面裂缝图像进行缩放处理,以适应屏幕大小。图像预处理为后续处理奠定了基础。具体为:

[0070] 本实施例中采集图像,是采用光电摄像头对沥青或者水泥路面裂缝进行现场采集,摄像头安装在车的底部,拍摄角度与路面垂直,同时采用车载计算机按照一定频率对路面图像进行自动拍摄,采集路面裂缝信息。

[0071] 本实施例中读取图像,是采用JPG格式和PNG格式作为路面裂缝图像读取的格式,JPG格式具有占用内存小、打开速度快的优势,可以提高系统识别裂缝、检测裂缝的效率,PNG格式具有防锯齿、图片质量高、无损压缩的优势,可以进一步提高裂缝提取的精确度,两种格式间支持相互转换。首先,将待处理图片放入程序指定的文件目录中,然后,程序读取图片并进行处理和计算,程序会依次显示出处理过程中的图片,最后,将处理后的图像保存在程序指定的文件夹中,面积、长度等参数将显示在运行栏中,便于检测图像处理过程和提高数据可视性。

[0072] 本实施例对图像的长和宽进行调整,以适应屏幕,其缩放表达式为:

[0073] $X_1 = fx * src.cols;$

[0074] $Y_1 = fy * src.rows;$

[0075] 式中,src.cols、src.rows分别表示原始图像的宽和长,fx表示水平轴上的比例因子,fy表示垂直轴上的比例因子, X_1 、 Y_1 分别表示处理后图像的长和宽,fx与fy相等时可实现对图像的等比例缩放。

[0076] 步骤2:基于伽马变换的图像增强;

[0077] 本实施例采用伽马变换算法对路面裂缝图片进行处理,将相机过曝或者曝光不足的图片进行修正,并且对裂缝图像进行增强突出,便于图像分割时的阈值选择;

[0078] 本实施例对整个图像的所有像素点的值使用伽马变换公式进行处理,伽马变换的彩色图像公式为:

[0079] $f(R,G,B) = C * h(R,G,B)^\gamma;$

[0080] 式中,C为常数,h(R,G,B)为原始图像像素点的R、G、B通道的亮度值,f(R,G,B)为处理后的像素点的R、G、B通道的亮度值, γ 为伽马系数。

[0081] 步骤3:图像灰度处理;

[0082] 本实施例根据RGB和YUV颜色空间的变化关系来建立亮度Y与R、G、B三个颜色分量的对应关系,以亮度值表达图像的灰度值,得到灰度图像;

[0083] 将RGB图像转化成为灰度图像,可以有效简化图片包含的信息且不会对图像特征造成破坏。裂缝图像灰度化后,可以根据裂缝和路面背景的亮度差异来对裂缝进行提取,将背景区域和裂缝区域分开。

[0084] 由于一张数字图像是由大量的像素点组成的,因此设一张图像的左上角为原点,水平向右为x轴正方向,垂直向下为y轴正方向,整个图像可以建立一个二维坐标轴,于是可以通过一个关于坐标的函数来表示图像中任意位置的像素点的像素值。一张彩色图片的表达式可以表示为:

$$[0085] \quad f(x, y) = (R, G, B);$$

[0086] 式中,R、G、B分别为图像坐标(x, y)处的红色、绿色、蓝色分量值,其中(0, 0, 0)表示黑色,(255, 255, 255)表示白色。通过R、G、B三个分量,可以采用加权平均法得到灰度图像。加权平均法是将R、G、B三个分量以不同的权值进行加权平均。由于人眼对绿色的敏感最高,对蓝色的敏感最低,因此,绿色的权重最高,而蓝色的权重最低,其灰度处理公式为:

$$[0087] \quad \text{Gray}(i, j) = 0.299 * R(i, j) + 0.578 * G(i, j) + 0.114 * B(i, j);$$

[0088] 式中,Gray(i, j)为灰度处理后的图像的像素点值,R(i, j)为像素点的红色通道亮度值,G(i, j)为像素点的绿色通道亮度值,B(i, j)为像素点的蓝色通道亮度值。

[0089] 步骤4:图像分割;

[0090] 本实施例采用中值滤波算法对裂缝灰度图像进行降噪处理,然后,使用自适应阈值法有效地将图像进行二值化处理;

[0091] 本实施例对比了高斯滤波、均值滤波、双边滤波和中值滤波算法,最终选择使用中值滤波作为图像平滑和降噪的工具。中值滤波是一种非线性滤波,在处理脉冲噪声和椒盐噪声时效果极佳,而且可以有效保护图像的边缘信息。中值滤波的基本原理是把数字图像或者数字序列中一点的值用该点的一个邻域中各点值的中值代替,将每个像素点的灰度值设置为该点某邻域窗口内的所有像素灰度值的中值,从而消除孤立的噪声点。方法是用某种结构的二维滑动模板,将板内像素按照像素值的大小进行排序,生成单调上升(或下降)的二维数据序列。二维中值滤波输出为:

$$[0092] \quad g(x, y) = \text{med}\{f(x-k, y-l), (k, l \in W)\};$$

[0093] 其中,f(x, y), g(x, y)分别为原始图像和处理后图像。W为二维模板,通常为3*3, 5*5区域,也可以是不同的形状,如线状,圆形,十字形,圆环形等。

[0094] 本实施例对比了最大类间差法、全局阈值法和自适应阈值法的处理效果,最终使用自适应阈值法将灰度化的路面裂缝图像进行二值化处理,这种算法不仅可以作为阈值分割,还可以用作边缘提取,这个主要由所选邻域决定如果所选邻域非常小(比如3x3),则很显然阈值的自适应程度就非常高,可以有效检测出图像的边缘;如果所选邻域非常大(比如49x49),则图像每次处理的邻域非常大,边缘检测的效果会大幅降低,多表现为二值化的效果。自适应阈值法是根据像素的邻域块的像素值分布来确定该像素位置上的二值化阈值,因此这种算法对光线有很强的适应性。

[0095] 自适应阈值法的算法步骤是:先使用步骤四的中值滤波器将图像平滑处理,并求出图像的梯度图,然后,运用Laplacian算子,找到具有局部最大阈值的点,这些点的原始灰度值便是候选的局部阈值,然后,采样候选点,将平滑图像中的候选点灰度值替换为原始图像中的灰度值或者更大一点的值,最后,插值灰度点,得到阈值表面,其表达式如下:

$$[0096] \quad \lim_{n \rightarrow \infty} P_n(x, y) = P_{n-1}(x, y) + \frac{\beta \cdot R_n(x, y)}{4};$$

$$[0097] \quad R(x, y) = P(x, y+1) + P(x, y-1) + P(x-1, y) + P(x+1, y) - 4P(x, y);$$

[0098] 式中 $R(x, y)$ 为Laplacian算子,强迫任意点 $R(x, y) = 0$ 的几何意义是使得灰度梯度的曲线变得光滑,从而让图像的灰度梯度是连续变化的,因而其二次导数为0。 β 可以调整阈值形成的光滑曲线与原图灰度曲线的收敛程度,当 $\beta = 0$ 时,残差消失,两条曲线完全重合, $1 < \beta < 2$ 时,曲线收敛更快。

[0099] 步骤5:轮廓提取;

[0100] 步骤四的中值滤波算法和自适应阈值法凸显出了裂缝的边缘,将图像中的裂缝图像以及噪声全部进行轮廓提取并标上序号,并获得提取出的所有轮廓的像素坐标;计算每个轮廓所围成的像素面积;

[0101] 本实施例的轮廓提取,是使用光栅扫描法对整个图像的像素点进行扫描,即从左往右、由上往下,先扫描完一行,再移至下一行起始位置继续扫描;

[0102] 从边界开始点 (i, j) 以边界跟踪算法得到一条边界,为每条新找到的边界 B 赋予一个新的唯一的编号,NBD表示当前跟踪的边界的编号;在光栅扫描的过程中,保存最近遇到(上一个)的边界 B' 的编号,记为LNBD;每当扫描到图片的新行的起始位置时,将LNBD重置为1;

[0103] 假设输入图像为 $F = \{f_{ij}\}$,将初始的NBD设为1,即把图像 F 的图像框架看成第一个边界;使用光栅扫描法扫描图像 F ,当扫描到某个像素点 (i, j) 的灰度值 f_{ij} 不为0时执行以下步骤;

[0104] (1)将 f_{ij} 分三种情况讨论:

[0105] (a)如果 $f_{ij} = 1$ 并且 $f_{i,j-1} = 0$,则 (i, j) 是外边界开始点,NBD+=1,

[0106] $(i_2, j_2) \leftarrow (i, j-1)$ (NBD加1,且将像素坐标 $(i, j-1)$ 赋给坐标 (i_2, j_2));

[0107] (b)如果 $f_{ij} \geq 1$ 并且 $f_{i,j+1} = 0$,则 (i, j) 是孔边界开始点,NBD+=1,

[0108] $(i_2, j_2) \leftarrow (i, j+1)$ (NBD加1,且令 $i_2 = i, j_2 = j+1$);如果 $f_{ij} > 1$,则LNBD $\leftarrow f_{ij}$

[0109] (则将 f_{ij} 的值赋给LNBD);

[0110] (c)其他情况,则执行步骤(4);

[0111] (2)根据上一个边界 B' 和当前新遇到边界 B 的类型,从边界类型表得到当前边界 B 的父边界;

[0112] (3)从边界开始点 (i, j) 开始按步骤3.1到步骤3.5进行边界跟踪;

[0113] (3.1)以 (i, j) 中心, (i_2, j_2) 为起始点,按顺时针方向查找 (i, j) 的4或8邻域是否存在非0像素点;若找到非0像素点,则令 (i_1, j_1) 是顺时针方向的第一个非0像素点;否则令 $f_{ij} = -NBD$,转到步骤(4);

[0114] (3.2) $(i_2, j_2) \leftarrow (i_1, j_1)$, $(i_3, j_3) \leftarrow (i, j)$ (将像素点的坐标 (i_1, j_1) 赋给坐标 (i_2, j_2) ,坐标 (i, j) 赋给坐标 (i_3, j_3));

[0115] (3.3)以 (i_3, j_3) 为中心,按逆时针方向, (i_2, j_2) 的下一个点为起始点查找 (i_3, j_3) 的4或8邻域是否存在非0像素点;

[0116] (3.4)分三种类型进行讨论:

[0117] (a)如果 (i_3, j_3+1) 是步骤3.3中已经检查过的像素点且是0像素点,则 $f_{i_3, j_3} \leftarrow -$

NBD(将-NBD的值赋给像素点 (i_3, j_3) 的灰度值 f_{i_3, j_3})；

[0118] (b) 如果 (i_3, j_3+1) 不是步骤3.3中已经检查过的0像素点,并且 $f_{i_3, j_3}=1$,则 $f_{i_3, j_3} \leftarrow -\text{NBD}$ (将-NBD的值赋给像素点 (i_3, j_3) 的灰度值 f_{i_3, j_3})；

[0119] (c) 其他情况,不改变 f_{i_3, j_3} ；

[0120] (3.5) 如果 $(i_4, j_4) = (i, j)$ 且 $(i_3, j_3) = (i_1, j_1)$,即回到了边界开始点,则转到步骤4;否则令 $(i_2, j_2) \leftarrow (i_3, j_3)$, $(i_3, j_3) \leftarrow (i_4, j_4)$ (将像素点的坐标 (i_3, j_3) 赋给坐标 (i_2, j_2) ,坐标 (i_4, j_4) 赋给坐标 (i_3, j_3)),转到步骤3.3；

[0121] (4) 如果 $f_{i_j} \neq 1$,则 $\text{LNBD} \leftarrow |f_{i_j}|$ (将 $|f_{i_j}|$ 的值赋给LNBD),从点 $(i, j+1)$ 继续光栅扫描;当扫描到图片的右下角顶点时结束。

[0122] 步骤4对图像进行了二值化处理,使得图像中只含有黑色(像素值为0)和白色(像素值为255)两种颜色,在保存裂缝特征的情况下进一步减小了计算机的处理量,但图像中依旧有很多噪声,以及被误判为裂缝的背景区域,如果此时采用邻域较大的滤波算法,可以有效去除噪声和杂点,但必将对裂缝目标图像的轮廓细节以及面积参数产生影响,从而使得识别和计算效果不佳。本发明在轮廓提取的过程中,先使用OpenCV中的findContours()函数,获取图像中裂缝以及噪声的轮廓信息,包括轮廓标号、轮廓像素点坐标,其中,由于轮廓提取算法采用光栅扫描法对图中像素点进行遍历扫描,因此标号顺序是从左至右、由上到下的顺序。然后,计算出图像中所有轮廓所围成区域的面积,这里使用OpenCV中的contourArea()获取轮廓面积,其原理是格林公式法,格林公式法得出的面积公式为:

$$[0123] \quad S_D \approx \frac{1}{2} \sum_{k=1}^n \int_{P_{k-1}P_k} -ydx + xdy = \frac{1}{2} \sum_{k=1}^n \begin{vmatrix} x_{k-1} & x_k \\ y_{k-1} & y_k \end{vmatrix};$$

[0124] 式中,设需测面积在xoy平面上,D是xoy平面上的多连通闭区域, P_k 是该平面上的第k个坐标点,其中k为整数,取值范围为: $1 \leq k \leq n$,该坐标点的坐标为 (x_k, y_k) ,按照坐标点进行连线,将曲线近似用直线段表示, $\overline{P_{k-1}P_k}$ 表示相邻顺序两点连接所得的直线段,n是坐标点的总数;

[0125] 步骤6:目标裂缝轮廓筛选;

[0126] 本实施例采用冒泡排序法对所有裂缝的面积进行排序,通过面积数据筛选出目标裂缝的轮廓的序号,然后,获取目标裂缝轮廓的像素坐标和所围成的像素面积;

[0127] 本实施例通过分析每个轮廓所占的面积大小,从而筛选出所需处理的目标裂缝轮廓。本实施例采用冒泡排序法,对所有轮廓对应的面积进行由大到小的排序,通过比较面积大小筛选出所需提取的目标裂缝轮廓。冒泡排序法的原理是将邻近的数字两两进行比较,按照从大到小或者从小到大的顺序进行交换,这样经过几次循环处理后,即可完成整个数字集的排序。本发明使用的是从大到小排序的冒泡排序法,一般情况下,图像中所要提取的裂缝所占面积应该最大,因此只需提取出排序后的第一个裂缝即可,若裂缝形态比较复杂(如网状裂缝等),则可能需要提取多条封闭轮廓才可以展现出完整的裂缝轮廓。

[0128] 本实施例采用的冒泡排序法,包括以下步骤:

[0129] (1) 比较相邻的数字,如果第一个比第二个小,则交换这两个数字;

[0130] (2) 对每一对相邻数字做同样的处理,从开始第一对到结尾的最后一对,处理完一遍后,最后一个数字会是最小数,这个数的排序已经完成;

[0131] (3)针对所有的数字重复以上步骤,除了最后已经排序好的数字;

[0132] (4)持续每次对越来越少的无序数字(无序数字)重复上面步骤,直到没有任何一对数字需要比较,则序列最终有序。

[0133] 步骤7:图像蒙板处理;

[0134] 取一张纯黑色的图像,将裂缝轮廓的像素坐标赋值到纯色图像上,并对轮廓内部进行颜色填充,填充颜色与背景颜色相反;然后,获取图像中裂缝所占的矩形区域,将其他多余的区域裁剪掉,获得路面裂缝数字图像。

[0135] 图像蒙板即是对蒙板选中区域之内的区域进行保护,它是一种特殊的选区,可以保护该区域不被操作,而对非掩盖的地方应用操作,本发明中使用图像蒙板处理,目的是限定新导入的纯色图像的可修改区域,便于对轮廓内部区域进行颜色填充,蒙板处理后,所需处理的裂缝目标将被提取出来并粘贴到新的背景上,从而去除所有噪声和背景干扰,得到一张只含有所需处理的裂缝的数字图像,很好地保护了裂缝的边缘和轮廓细节。该处理方法的步骤如下:

[0136] (1)选取步骤五中所需要的目标裂缝的轮廓,获取目标裂缝轮廓的像素坐标。

[0137] (2)导入一张黑色背景的纯色图,该图的长和宽都要比正在处理的路面裂缝图片大,将轮廓像素坐标按1:1的比例映射到纯色图上,并将这些点的像素值改为255,如果目标裂缝覆盖的面积中有封闭轮廓,则需要对轮廓目标进行分别提取,将所有裂缝轮廓目标整合到一张图片中。

[0138] (3)对纯色图上的轮廓内部进行颜色填充,填充颜色为白色(像素值为255),如果目标裂缝覆盖的面积中有封闭轮廓,则需依次对轮廓内部进行反色填充,以显示出裂缝的结构和形状。本发明采用四邻域算法实现对轮廓内部的填充,即先在图像中选择一个种子点,并用黑色表示。

[0139] (4)以(3)中选择的这个点为起点,将它压入栈中,所要填充的颜色为白色,因此将该点设置为白色,判断这个点的四邻域像素。然后,设置一个颜色阈值,以该阈值判断当前像素与四邻域像素的灰度差值,如果该差值小于阈值,则把这个像素点作为下一个种子点,压入栈中,否则继续判断。当栈为空时,则种子填充结束,否则重复(4)。

[0140] (5)采用最小矩形边框选择法:最小矩形边框是指用一个最小的矩形,把找到的形状包起来,使得目标像素完全存在于矩形边框内。该算法的原理是从图像左上角的像素点开始,依照高和宽绘制轮廓的垂直边界最小矩形,矩形是与图像的上下边界平行的,直到目标像素点不处在最小矩形外为止。

[0141] (6)使用ROI选择和图像裁剪,在最小矩形框选处理后,可以得到最小矩形的四个边界的坐标点,这些坐标点将作为ROI区域的坐标,根据ROI的坐标可直接将此矩形区域围成的图像从原图中扣出,从而获得长宽尺寸最小的含有目标裂缝的图像,使用ROI选择和图像裁剪,可以使得裂缝图像占据全图主要部位,便于后续处理,同时也缩小了图像尺寸,减少了后续操作的计算量。

[0142] 步骤8:图像形态学处理;

[0143] 对传统的闭运算进行改进,通过膨胀、滤波和腐蚀的方法,对图像中断裂的裂缝进行修补,通过图像细化算法,提取出图像骨架,展现出裂缝的形态信息。

[0144] 本实施例将图像进行形态学处理,本处所处理的图像时二值化图像,形态学处理

可以简化图像数据,同时保持它们基本的形状特性,并除去不相干结构。本实施例对形态学处理中的闭运算进行改进,且使用了索引表图像细化算法,可以实现对图像数据的简化和裂缝的修补。闭运算是先对图像膨胀后腐蚀,用来填充体内小空洞,连接邻近物体,连接断开的轮廓线。本实施例改进后的闭运算可以平滑裂缝边缘,消除闭运算产生的毛刺,增强连接邻近物体的能力,更好地抑制由于膨胀产生的明显噪声。

[0145] 实施例改进后的闭运算基本步骤为:

[0146] (1) 对路面裂缝的二值化图像进行膨胀处理,邻域大小选为 $(M1, M1)$,处理次数为 $N1$ 次, $M1$ 和 $N1$ 的取值由所需连接裂缝的间距决定,间距较小时, $M1$ 和 $N1$ 的取值也较小;

[0147] (2) 使用中值滤波器对膨胀后的图像进行滤波,中值滤波由于是把数字图像或数字序列中一点的值用该点的一个邻域中各点值的中值代替,因此在二值化图像中也具有修补断裂带的作用,其修补能力取决所选邻域大小,但中值滤波会使得图像目标边缘区域纳入背景,减少裂缝的面积和宽度;

[0148] (3) 针对步骤2中中值滤波的特性,对图像进行腐蚀处理,邻域大小选为 $(M2, M2)$,处理次数为 $N2$,其中 $M2$ 与 $M1$ 要相等, $N2$ 的取值需要比 $N1$ 小。

[0149] 本实施例在采用细化算法前对图像进行预处理:此处的预处理是对二值化图像进行颜色反转,彩色图像的颜色反转,比较简单的思路是使用255减去当前待处理像素值,对整个图像进行该处理后即可得到反转后的图像。对于二值化图像而言,由于二值化图像只有0(黑色)和255(白色)两类像素点,因此使用彩色图像的颜色反转算法依旧有效。颜色反转公式为:

[0150] $g(x, y) = 255 - f(x, y)$;

[0151] 式中, $g(x, y)$ 表示反转后的像素点灰度值, $f(x, y)$ 表示反转前的灰度值。颜色反转后,图像将满足细化算法的要求,且图像中裂缝仅有颜色发生了改变。

[0152] 本实施例采用索引表图像细化算法:细化算法的基本原理是在不破坏图像连通性的前提下逐次去除目标边界,通常选择一组结构元素对,不断在这些结构对中循环,如果所得结果不再变化,则终止迭代过程,随着迭代的进行,集合也不断细化。图像细化可以提取出裂缝的骨架,使用极少的像素点表示出裂缝的形状和长度,再次减少了计算量。关于结构对的选择,应该在不断重复的迭代细化过程中使用同一个结构对。

[0153] 细化过程需要满足两个条件,一是在细化过程中,图像应该有规律地缩小;二是在图像逐步缩小的过程中,应当使图像的连通性质保持不变。在图像细化中,要判断一个点能否去掉,是根据它的八个相邻点的情况来判断的,八个点可以组成六种不同的组成情况,总结上述情况,可以得出以下四个判据:

[0154] (1) 内部点不能删除;

[0155] (2) 孤立点不能删除;

[0156] (3) 直线端点不能删除;

[0157] (4) 如果 P 是边界点,去掉 P 后,如果连通分量不增加,则 P 可以删除。

[0158] 本实施例的索引表图像细化算法是指做出一张索引表,这个表含有需要细化的点的八个邻域的情况,该算法需要不断地遍历边缘,按照索引表对邻域点进行是否删除判断,直到边缘的每个点都不能再被删除为止。

[0159] 索引表的查表方法是:设白点为1,黑点为0,边缘点为 P ,则 P 的八连通区域为:

$$[0160] \quad \begin{bmatrix} P0 & P1 & P2 \\ P3 & P & P4 \\ P5 & P6 & P7 \end{bmatrix};$$

[0161] 对于这些点,需要对其赋予不同的价值,若P点八邻域的某个点为黑色,则认为其价值为0,如果为白色,则按照权值表赋予对应的价值,权值表为:

$$[0162] \quad \begin{bmatrix} P0=1 & P1=2 & P2=4 \\ P3=8 & P & P4=16 \\ P5=32 & P6=64 & P7=128 \end{bmatrix};$$

[0163] 该表经过计算后得到如下表达式:

$$[0164] \quad M=P0*1+P1*2+P2*4+P3*8+P4*16+P5*32+P6*64+P7*128;$$

[0165] 式中,M表示计算结果,通过该计算结果查找索引表中对应的数值,即可知道该点是否需要删除,用这个方法能把所有点的情况映射到0~255的索引表中。

[0166] 步骤9:目标裂缝的面积、长度和平均宽度计算;

[0167] 本实施例所用的面积计算方法为格林公式法,在步骤5时已经计算出目标裂缝的面积。

[0168] 本实施例所用的裂缝长度计算方法采用的分类思想,经过前面八个步骤的处理后,所得的裂缝目标图像大小已经很小,大大减少了之后算法处理的计算量,裂缝细节保存完整,背景区域没有杂质,裂缝的骨架也可以很好地展现出其结构,利于对长度进行计算。但是裂缝的骨架大多是比较复杂的,取裂缝骨架图像中黑色像素的中心点,这个像素上、下、左、右的四个点的中心,距离该黑色像素的中心点的间距为1个像素,而其左上、左下、右上、右下四个点的中心距离黑色像素的中心点的间距为 $\sqrt{2}$ 个像素。至于白色背景点,将不算入间距当中。

[0169] 本实施例读取轮廓提取时轮廓围成的像素面积即可得到目标裂缝的面积;经过处理的二值化图像,其裂缝像素是黑色的,灰度值为0;背景是白色的,灰度值为255;然后,对图像进行遍历运算,按照骨架像素点的间距以及是否为背景像素点,将所有像素点进行分类,按下列公式进行裂缝长度计算:

$$[0170] \quad \text{Length} = 1 \cdot X + \sqrt{2} \cdot Y + 0 \cdot Z;$$

[0171] 式中,Length表示裂缝的长度,X表示与相邻骨架像素点距离为1的像素点总数,Y表示与相邻骨架像素点的距离为 $\sqrt{2}$ 的像素点总数,Z表示背景像素点的总数。

[0172] 裂缝平均宽度计算公式为:

$$[0173] \quad \text{Width} = \frac{\text{Area}}{\text{Length}};$$

[0174] 式中,Width表示裂缝的平均宽度,Area表示图像中裂缝的面积,Length表示裂缝的长度,求出裂缝的平均宽度,将在评判路面损坏情况和修补方案上有重要作用。

[0175] 本实施例采用指针遍历算法对图像进行遍历运算,对图像中的像素点按照上述规则进行分类,将中心点为黑色的两种不同分类的像素点按照所对应间距进行累加求和,将会得到更为精确的裂缝长度。

[0176] 指针遍历法的算法步骤为:

[0177] (1) 令图像中的任一像素点的坐标为 $R[i, j]$, 其中 i 表示横坐标值, j 表示竖坐标值, 图像从左上角开始进行遍历运算; 则它的八邻域的像素点的坐标按照八邻域关系图表示;

[0178] (2) 根据八邻域的坐标关系, 取每个像素的中心点作为衡量像素距离的因素, 将像素点分为三类:

[0179] 中心点和邻域点的距离为1的点有: $R[i, j-1]$ 、 $R[i-1, j]$ 、 $R[i+1, j]$ 、 $R[i, j+1]$;

[0180] 中心点和邻域点的距离为 $\sqrt{2}$ 的点有: $R[i-1, j-1]$ 、 $R[i+1, j-1]$ 、 $R[i-1, j+1]$ 、 $R[i+1, j+1]$;

[0181] 中心点的八邻域区域没有黑色像素点, 这种情况则将该点视为背景点;

[0182] (3) 对以上三种情况进行分类统计, 并进行裂缝长度计算;

[0183] $Len = N_{str} * 1 + N_{sla} * \sqrt{2}$;

[0184] 式中, N_{str} 为中心点和邻域点距离为1的点数, N_{sla} 为中心点和邻域点距离为 $\sqrt{2}$ 的点数, Len 为裂缝的总长度。

[0185] 接下来通过现实案例对本发明做进一步阐述: 采用光电摄像头对沥青或者水泥路面裂缝进行现场图像采集。通过将现场采集回来的图像进行分类, 这些原始图像中的裂缝主要分为三种基本形态, 横向裂缝、纵向裂缝、分叉裂缝、网状裂缝。读取路面裂缝图像, 并对其进行缩放处理, 结果如图2所示。

[0186] 为对相机过曝或者曝光不足的图片进行修正, 采用伽马变换算法对路面裂缝图片进行处理, 并且对裂缝图像进行增强突出。

[0187] 伽马变换可以使得图像的像素值直方图中的R、G、B像素点向0(黑色)或255(白色)集中, 使得颜色较暗的裂缝在图像中更为明显, 拉开图像中亮暗像素点的灰度值, 利于后期对裂缝图像二值化的阈值选择。例如, 图3 为经过伽马变换后的裂缝图像。

[0188] 将原始图像转化为灰度图像, 对亮度值进行量化, 如图4所示。

[0189] 将一幅彩色图像变换为灰度图, 可以用两种方法实现。第一种方法是求出每个像素点的R、G、B三个分量的平均值, 然后将这个平均值赋予给这个像素的三个分量, 所得的每个像素点可表示为:

[0190] $f(x, y) = \frac{1}{3}(R + G + B)$;

[0191] 第二种方法是根据RGB和YUV颜色空间的变化关系来建立亮度Y与R、G、B三个颜色分量的对应关系, 以亮度值表达图像的灰度值。彩色图像可以由如下公式得到亮度信息:

[0192] $f(x, y) = 0.299R + 0.587G + 0.114B$;

[0193] 这里本实施例采用第二种方法, 图5为灰度处理前后的统计图。

[0194] 在裂缝病害图像的滤波处理中通过定性评价方法对比了均值滤波、中值滤波、高斯滤波、双边滤波四种滤波方法。图6为四种滤波方法与原图的比较。

[0195] 定性评价: 路面裂缝图像的滤波首先要具有保护边缘细节的能力, 此外, 图像需要一定的清晰度, 才能突出裂缝的形状和细节。均值滤波遍历图像使得滤波速度较慢, 而且将周围的景物的像素点给平均了, 使得图像变得模糊。因此均值滤波并不符合要求。中值滤波

可以有效保护图像的边缘信息,相比于高斯滤波和双边滤波,其对图像的模糊度影响稍高,但在处理脉冲噪声和椒盐噪声时效果极佳。

[0196] 通过定向分析,对于路面裂缝病害而言,使用中值滤波要优于其他滤波方法。故选择中值滤波。

[0197] 在裂缝图像的二值化方法对比了全局阈值法、最大类间差法(otsu法)、局部阈值法(自适应阈值法)等。三种二值化方法效果依次如图7、8、9所示。

[0198] 定性评价:通过试验分别采用三种不同裂缝的处理进行对比,三种阈值选取方法的试验结果如图10所示。三种阈值选取方法都可以清晰地分辨出目标裂缝,但最大类间差法和局部阈值法的背景中有明显的椒盐噪声,需要进行二次滤波。当图像的光线不均匀时,只有局部阈值法可以有效剔除裂缝光线较暗的部分(如竖向裂缝)。当图像中的背景颜色不均,噪声干扰较大时(如网状裂缝),只有局部阈值法可以有效减少噪声,并且分辨出裂缝轮廓。

[0199] 通过定向分析,局部阈值法最适合处理路面裂缝病害图片的二值化,不过局部阈值法来说,由于带有滤波效果,因此目标裂缝的细节会被平滑处理,使得部分细节丢失,在处理较细的裂缝时,要注意邻域的选取。

[0200] 为有效抑制椒盐噪声,在局部阈值法之后使用中值滤波。中值滤波后的二值化图像如图11所示。

[0201] 当背景的噪声过大、噪点分布密集时,小型的二维模板(如 $3*3$ 、 $5*5$ 等)已经不再适用,但如果中值滤波所选的二维模板太大,会使得图像中较细的裂缝中的像素点与背景像素点比值过小,使得细裂缝也被纳入背景,从而导致裂缝出现断裂(如图12所示)。

[0202] 因此,滤波后的照片大多需要进行形态学处理,在不改变目标特性的情况下对目标图像进行修补。形态学处理常用的方法有:a.腐蚀、b.膨胀、c.开运算、d.闭运算。其中,图像腐蚀膨胀结果如图13所示,图像中白色区域代表背景,图(a)中黑色区域表示被腐蚀区域,灰色区域表示腐蚀后的保留区域;图(b)中黑色区域表示原始图像的像素区域,灰色区域表示膨胀后新增的区域。

[0203] 图像后处理闭运算可以很好地修补裂缝连续性较好、只有部分区域近距离断裂的图像,当遇到断裂带之间距离过远,或者断裂区域较多较密的图像时,闭运算后的图片会产生明显的毛刺和干扰,如图14所示。

[0204] 基于此情况,对闭运算进行改进,在闭运算的图像膨胀之后使用中值滤波器,对图像进行平滑处理,再使用腐蚀还原图像。改进后的闭运算处理过程如图15所示。

[0205] 将图像中黑色或者白色区域视作目标,进行轮廓提取,根据上一个边界和当前新遇到边界的类型,可以通过查询边界类型表得到当前边界的父边界,边界类型表如表16所示,并将轮廓所在的像素点以数组的方式存储起来,不同目标的数组集是分开的。为便于观察,在提取轮廓数据之后,将所有轮廓的像素点赋予新的颜色,效果如图17所示。

[0206] 本处在计算面积时所用的是格林公式法,非常适合裂缝轮廓等闭合区域的面积计算,多连通闭区域示意图如图18所示,闭区域所取坐标点如图19所示。相比像素数统计求面积而言,这个方法不需要使用滤波器将除目标以外的杂质过滤,减少了滤波器对裂缝边缘的平滑影响,从而使得在求面积时裂缝细节保留完整,所得面积结果更加精确可信。

[0207] 由于在裂缝图像中,裂缝病害目标所占面积往往最大,因此用冒泡排序法将面积

排序,再选中面积最大的轮廓,完成对裂缝目标的提取。为便于观察,将裂缝病害的轮廓点赋予红色(图中断点所示),将其他干扰项的轮廓点赋予绿色(图中灰色所示),其处理结果如图20(b)所示。形如结构较复杂的网状裂缝的面积筛选如图21所示。经过格林公式的处理后,得到图像中所有轮廓所围区域的面积。

[0208] 取一张已完成二值化后的裂缝图像,对图像中的目标裂缝和杂质进行轮廓提取;对所有轮廓的面积进行计算,使用冒泡排序法对所有面积进行由大到小或者由小到大排序;在进行冒泡排序后,依照轮廓面积选取目标裂缝,一般情况下,目标裂缝所占面积应该在图像中最大,因此选取冒泡排序后第一个或者最后一个轮廓即为目标裂缝的轮廓,本发明采用由大到小的冒泡排序算法,一般第一个轮廓即为目标裂缝的轮廓;选取目标裂缝的轮廓的所有像素点,导入一张黑色背景的纯色图,其大小要比原图大,将轮廓像素点按1:1的比例映射到纯色图上,将在纯色图上被映射的点的灰度值赋值为255。图像目标轮廓提取的结果如图22所示。

[0209] 提取到裂缝病害图像的轮廓后,对轮廓内部进行颜色填充,使用四邻域算法,得到纯净的仅有裂缝病害的二值化图像。图像填充后的结果如图23所示。若图像是网状裂缝,由于裂缝目标内也有背景,因此需要按照a.提取外部轮廓b.按轮廓填充白色c.提取内部轮廓d.按轮廓填充黑色的顺序来处理图像,网状裂缝的处理过程如图24所示。

[0210] 在图像进行目标提取和填充之后,虽然目标裂缝的像素点数量没有改变,但图像的大小发生了改变,采用图像裁剪可以使得裂缝病害在图像中占据主要位置,减少了后期统计像素点的计算量。在裁剪之前对裂缝病害最小矩形边框和ROI区域进行选择,最小矩形框选前后对比如图25所示。最小矩形框选处理后,得到最小矩形的四个边界的坐标点,这些坐标点作为ROI区域的坐标,根据ROI的坐标直接将此矩形区域围成的图像从原图中扣出,得出的裂缝目标图像如图26所示。

[0211] 对图像进行颜色反转,以满足细化算法对图像的要求,经过颜色反转后的图像与原图像进行对比,结果如图27所示。

[0212] 即便得到了完整的二值化图像,且经过了图像的裁剪、颜色反转等操作,计算长度依旧是个难以解决的问题。采用对裂缝骨架的提取,使得计算裂缝长度的可操作性大大提高,通过统计组成骨架的像素点数,并进行分类处理,即可计算出裂缝的长度。提取图像骨架的过程一般被称为对图像细化的过程,细化过程需要满足两个条件:1.在细化的过程中,图像应该有规律地缩小;2.在图像逐步缩小的过程中,应当使图像的连通性质保持不变。在图像的细化中,要判断一个点是否能去掉,是根据它的八个相邻点的情况来判断的,八个相邻点的组成情况如图28所示。

[0213] 做出一张表,然后根据需要细化的点的八个邻域的情况进行查询,这便是索引表细化算法,算法所需要做的工作是不断地遍历边缘,依照索引表对邻域点进行是否删除判断,直到边缘的每一个点都不能再被删除。

[0214] 查表方法是:设白点为1,黑点为0,边缘点为P,P0至P7为边缘点P的八个相邻点,则P的八连通区域可以表示如下:

$$[0215] \begin{bmatrix} P0 & P1 & P2 \\ P3 & P & P4 \\ P5 & P6 & P7 \end{bmatrix}$$

[0216] 对于这些点,我们需要对其赋予不同的价值,若P点八邻域的某个点为黑色,则认为其价值为0,如果为白色,则按照权值表赋予对应的价值,权值表如下:

$$[0217] \begin{bmatrix} P0=1 & P1=2 & P2=4 \\ P3=8 & P & P4=16 \\ P5=32 & P6=64 & P7=128 \end{bmatrix}$$

[0218] 那么,该点在细化处理时的状态数值就是加权值的和,例如有下面一种邻域组合:

$$[0219] \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[0220] 则计算后得到它的值 $=2+8+128=138$,它对应着索引表中第139项的数值,用这种方法,把所有点的情况映射到0~255的索引表中。图像细化的索引表如表 29所示。

[0221] 经过索引表细化算法处理后,得出的裂缝图像骨架如图30所示。

[0222] 通过处理后,所得的裂缝目标图像大小以达到最小,减小了之后算法的计算量,裂缝的细节保存完整,背景区域没有杂质,裂缝的骨架可以很好地用于计算。因此,只需统计像素点的个数,就可以完成对裂缝长度的计算。但是裂缝的形状和骨架是比较复杂的,取裂缝骨架图像中的黑色像素的中心点,对于如图31所示的(a)类情况,其点间距为一个像素,对于(b)类情况,其点间距为 $\sqrt{2}$ 个像素,因此如果只是简单地将像素点累加,所得的裂缝长度会有很大误差。

[0223] 本发明中对裂缝所组成的像素进行的遍历归类处理,有效提升了计算精度。采用指针访问图像中的像素点数据遍历图像。

[0224] 根据图32八邻域的坐标关系,取每个像素的中心点作为衡量像素距离的因素,将像素点分为两类:中心点和邻域点的距离为1的点有: $R[i, j-1]$ 、 $R[i-1, j]$ 、 $R[i+1, j]$ 、 $R[i, j+1]$;中心点和邻域点的距离为 $\sqrt{2}$ 的点有: $R[i-1, j-1]$ 、 $R[i+1, j-1]$ 、 $R[i-1, j+1]$ 、 $R[i+1, j+1]$ 。此外,中心点的八邻域区域还可能没有黑色像素点,这种情况则将该点视为背景点。对以上三种情况进行分类统计。并进行裂缝长度计算,其公式为:

$$[0225] L = 1 \cdot X + \sqrt{2} \cdot Y + 0 \cdot Z ;$$

[0226] 式中,L表示裂缝的长度,X表示与相邻骨架像素点距离为1的像素点总数,Y表示与相邻骨架像素点的距离为 $\sqrt{2}$ 的像素点总数,Z表示背景像素点的总数。

[0227] 图33为待处理的裂缝骨架图像,通过本发明技术方案处理后计算,得到了裂缝病害目标的面积和长度,因此可以轻易地计算出裂缝病害的平均宽度,其公式为:

$$[0228] \text{Width} = \frac{\text{Area}}{\text{Length}} ;$$

[0229] 式中,Width表示裂缝的平均宽度,Area表示图像中裂缝的面积,Length 表示裂缝的长度。

[0230] 本发明提供了一种能够在具有背景干扰项和噪点的情况下提取出裂缝轮廓的方法,可以避免滤波对图像边缘细节的破坏,使得裂缝图像具有更为完整的边缘细节,现有技术大多通过滤波算法去除图像中的背景干扰和噪点,本方法可以大大减少滤波造成的细节

模糊,使得处理后的二值化裂缝图像的形态与原图更为接近,提升了裂缝面积、长度等计算的精度。本发明还提供了一种修补裂缝断裂区域的方法,相对于常用的闭运算修补断裂带的方法,本方法可以修补相距更远的断裂带,得到边缘更为平滑的裂缝边缘,与原始裂缝图像的形态更为接近。

[0231] 应当理解的是,上述针对较佳实施例的描述较为详细,并不能因此而认为是对本发明专利保护范围的限制,本领域的普通技术人员在本发明的启示下,在不脱离本发明权利要求所保护的范围情况下,还可以做出替换或变形,均落入本发明的保护范围之内,本发明的请求保护范围应以所附权利要求为准。

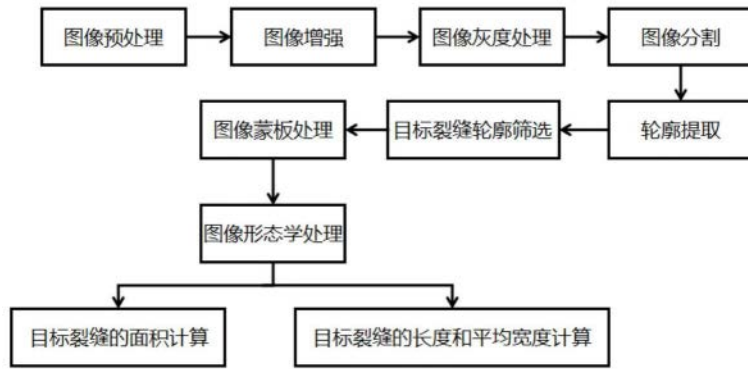


图1

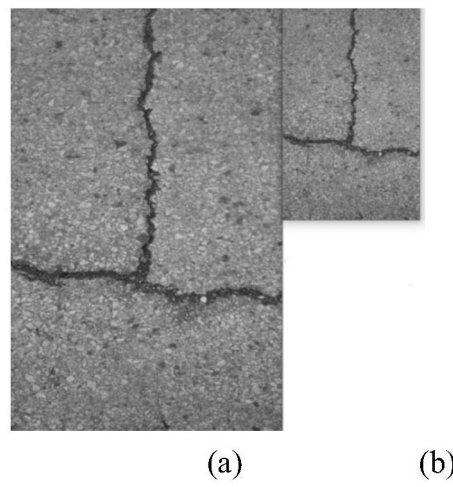


图2

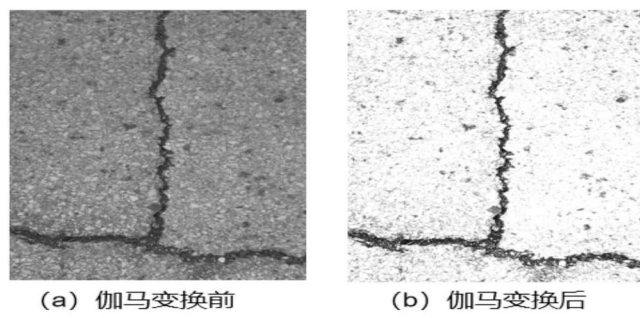


图3

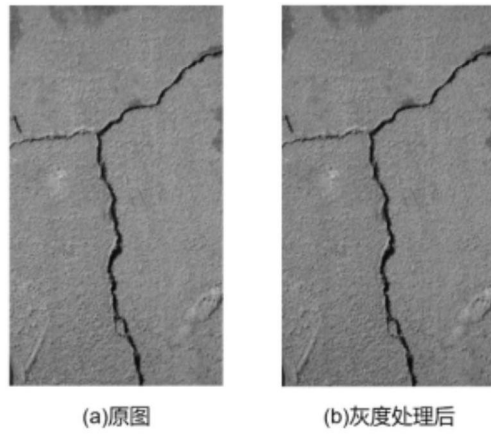


图4

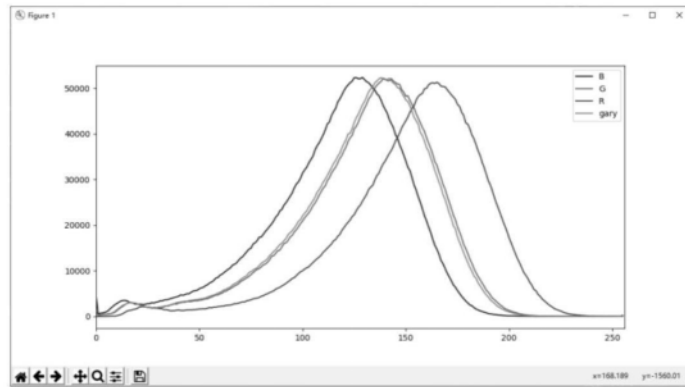


图5

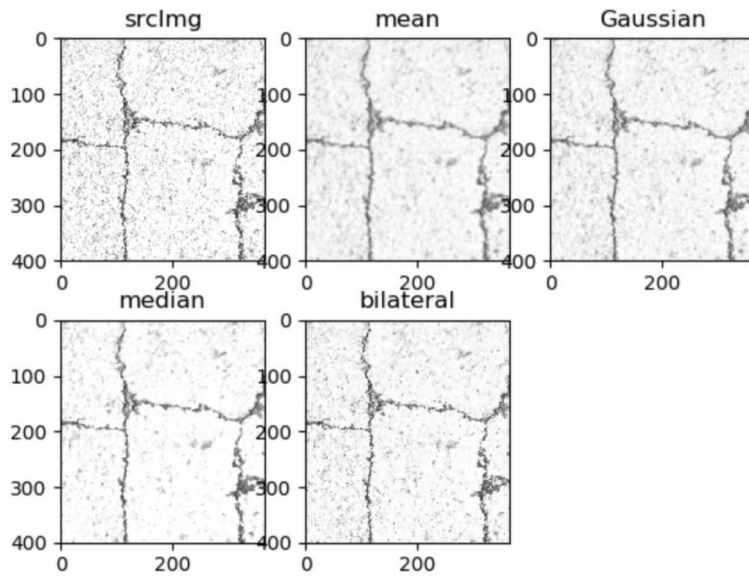


图6

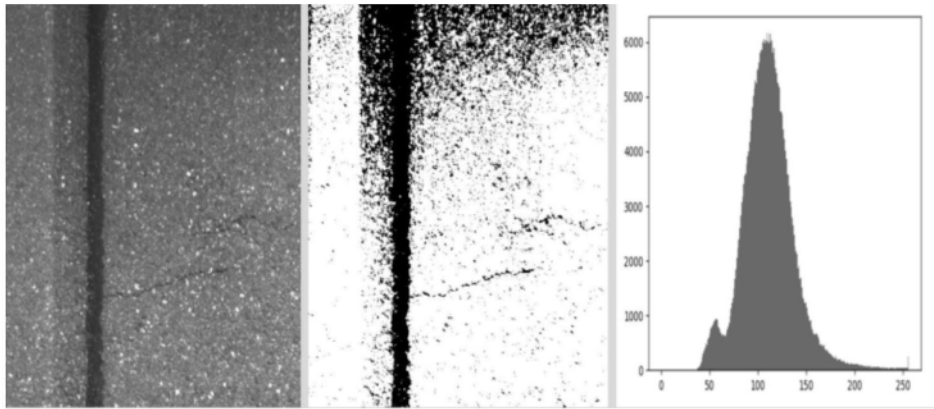


图7

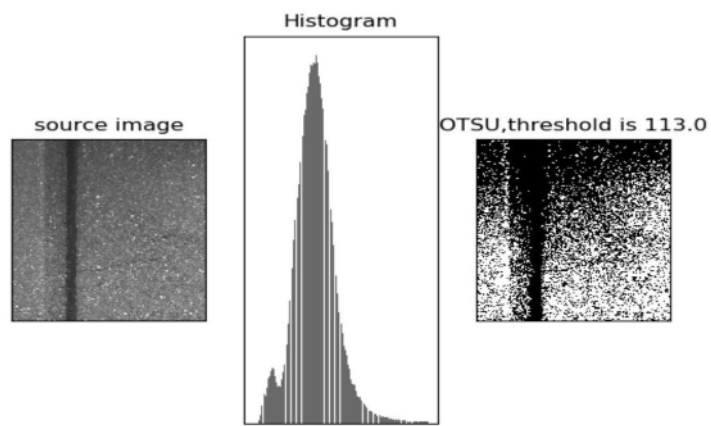


图8

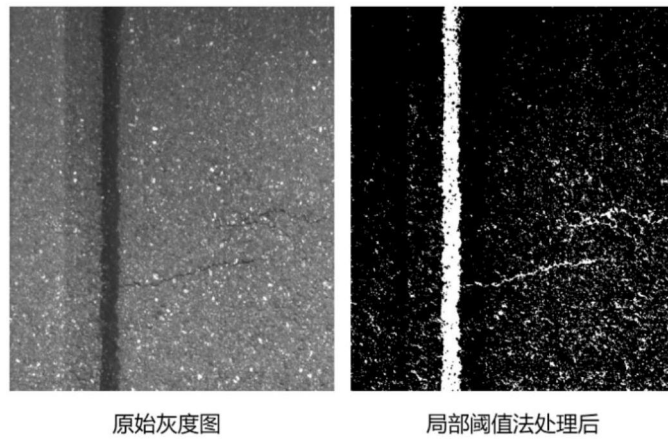


图9

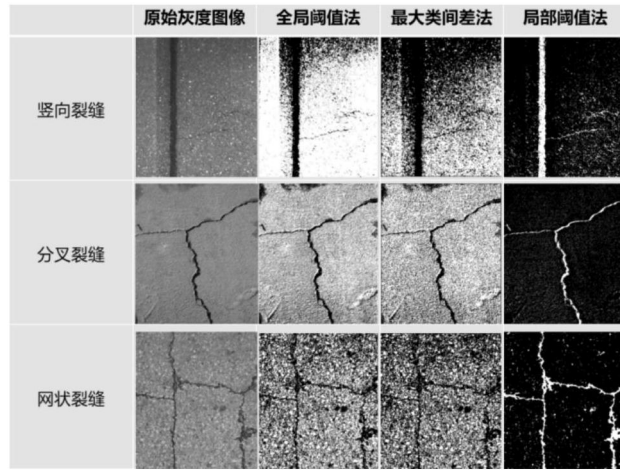


图10

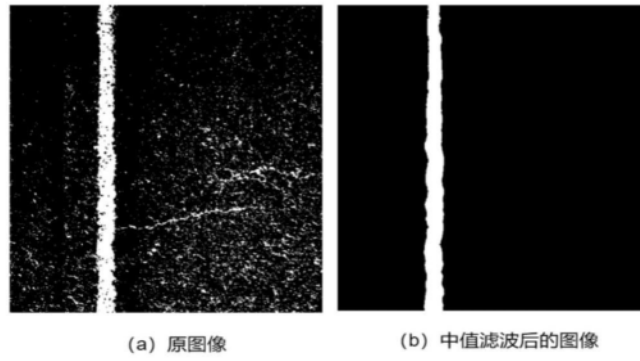


图11



图12

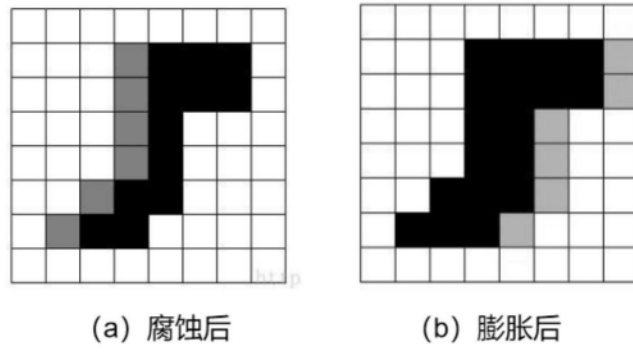


图13

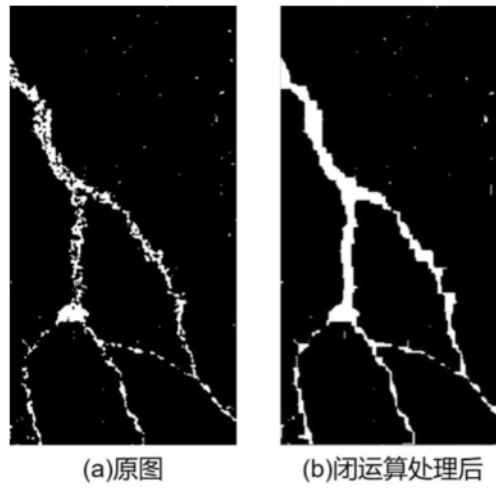


图14

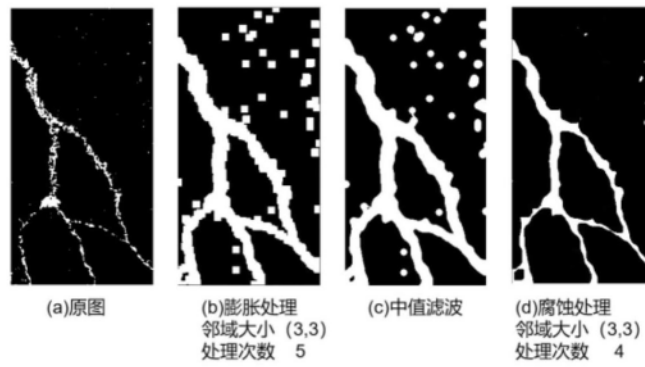


图15

当前边界 **B** 的父边界:

当前边界 B 的类型 \ 上一个边界 B' 的类型	外边界	外边界	孔边界
	孔边界	边界 B' 的父边界	边界 B'
外边界	边界 B' 的父边界	边界 B'	边界 B' 的父边界
孔边界	边界 B'	边界 B' 的父边界	边界 B' 的父边界

图16

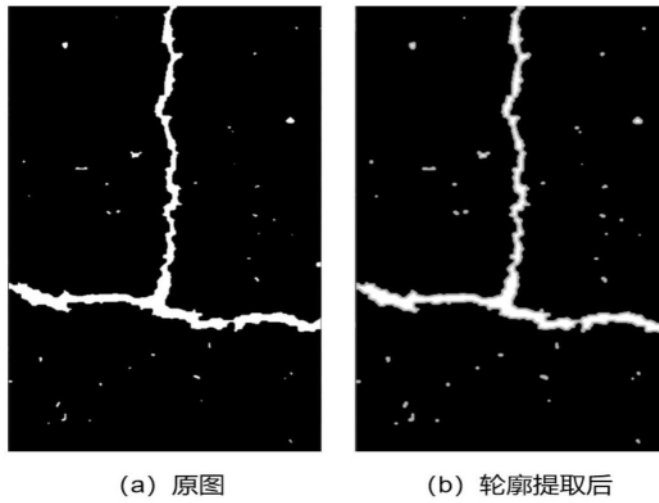


图17

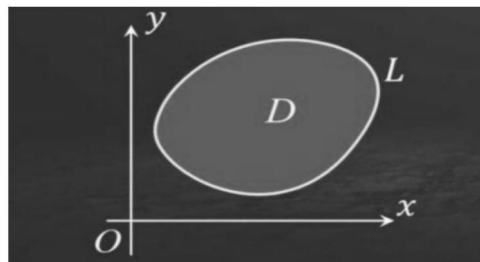


图18

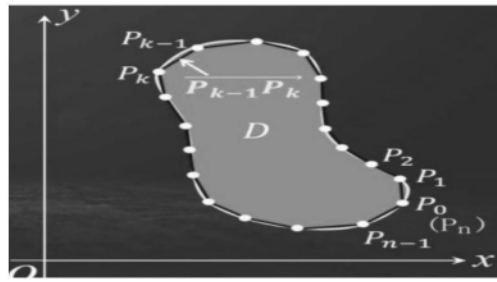
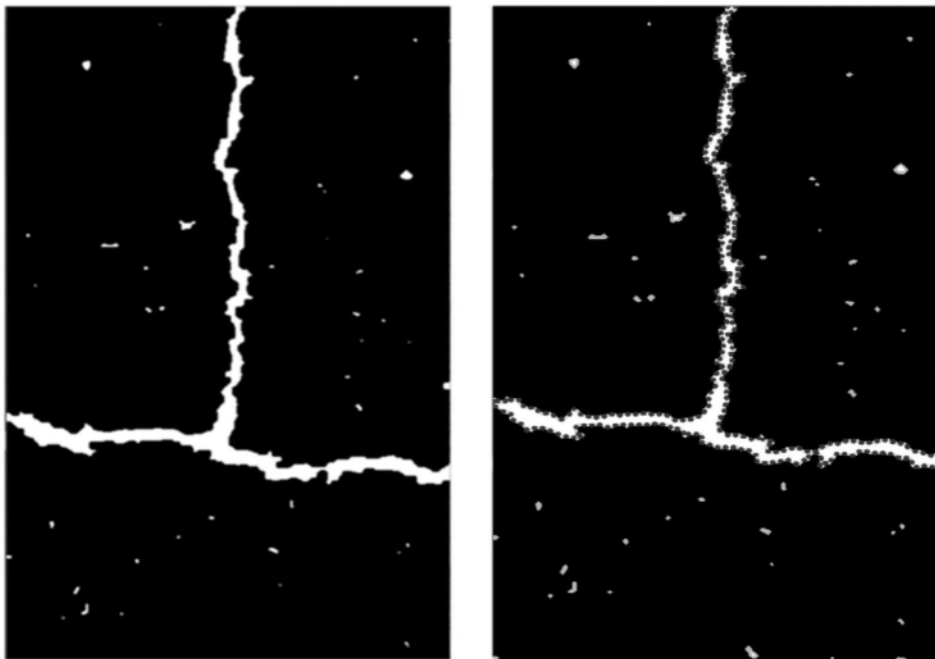


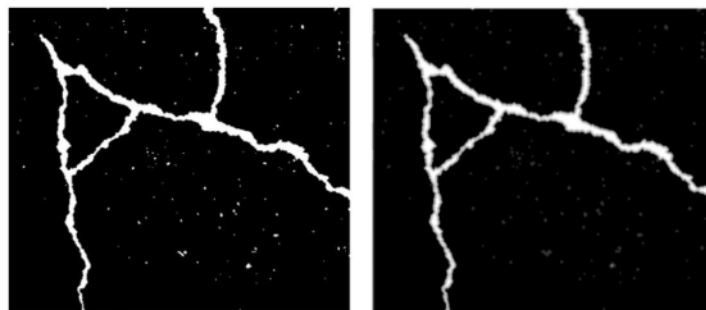
图19



(a) 原图

(b) 面积筛选后

图20



(a) 面积筛选前

(b) 面积筛选后

图21

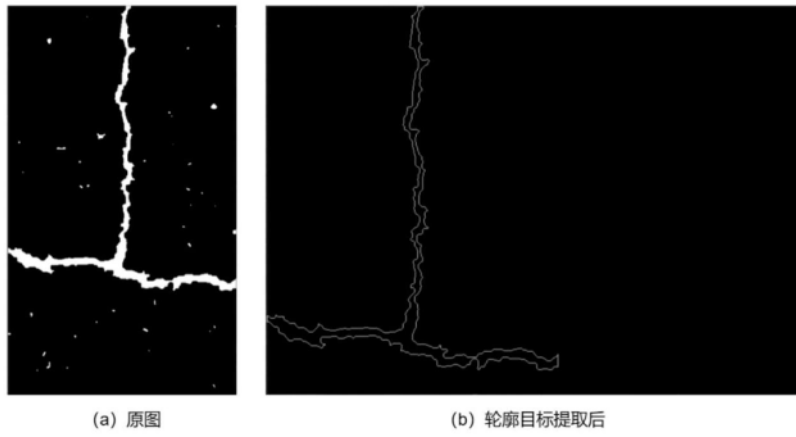


图22

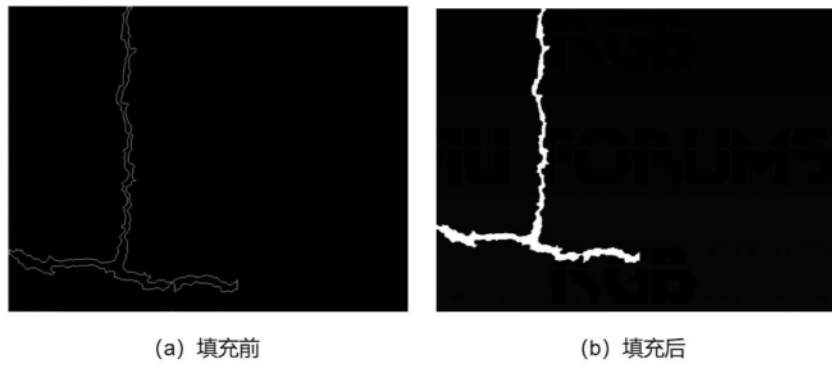


图23

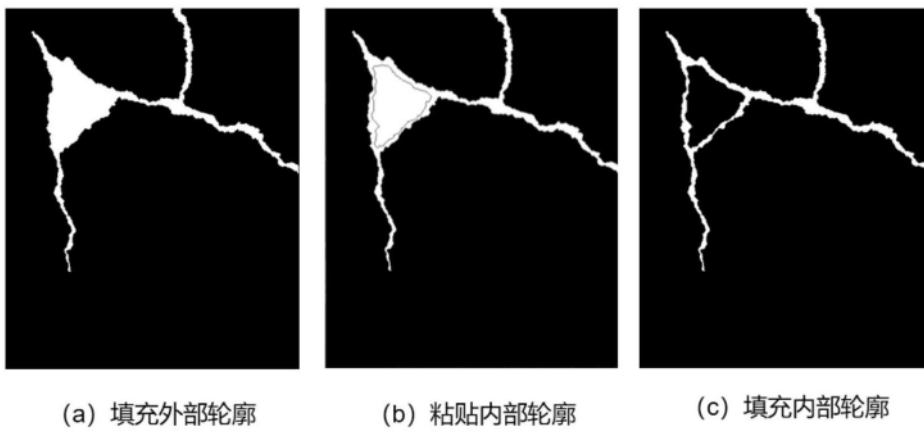


图24

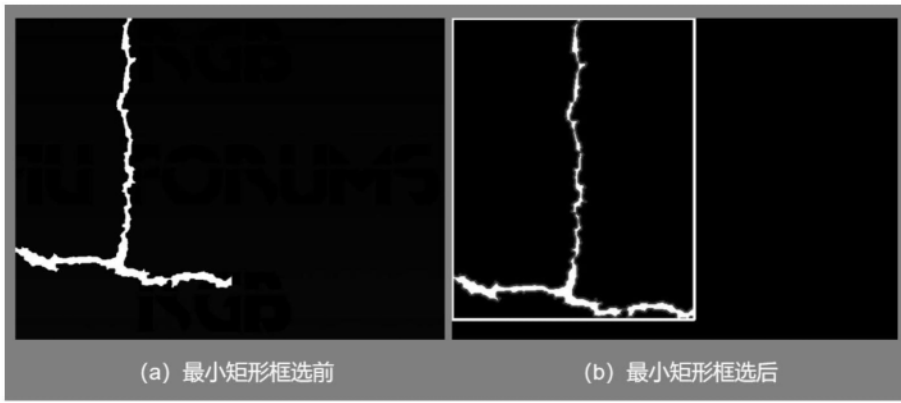


图25



图26

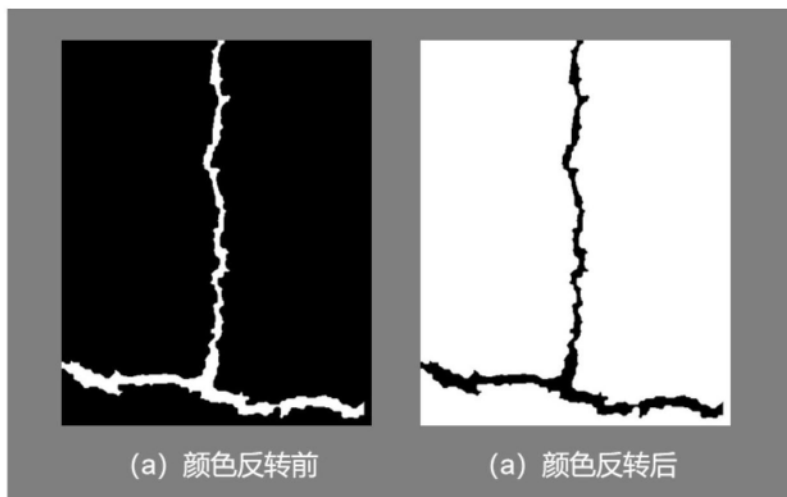


图27

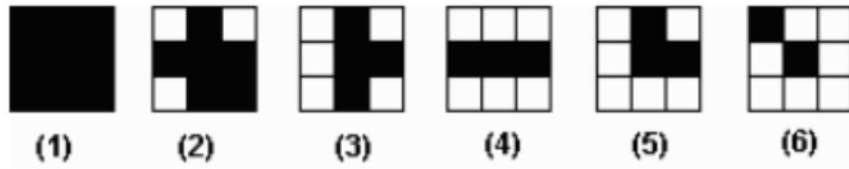


图28

```

static int erasetable[256]= {
0. 0. 1. 1. 0. 0. 1. 1.      1. 1. 0. 1. 1. 1. 0. 1.
1. 1. 0. 0. 1. 1. 1. 1.      0. 0. 0. 0. 0. 0. 0. 1.
0. 0. 1. 1. 0. 0. 1. 1.      1. 1. 0. 1. 1. 1. 0. 1.
1. 1. 0. 0. 1. 1. 1. 1.      0. 0. 0. 0. 0. 0. 0. 1.
1. 1. 0. 0. 1. 1. 0. 0.      0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0.      0. 0. 0. 0. 0. 0. 0. 0.
1. 1. 0. 0. 1. 1. 0. 0.      1. 1. 0. 1. 1. 1. 0. 1.
0. 0. 0. 0. 0. 0. 0. 0.      0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 1. 1. 0. 0. 1. 1.      1. 1. 0. 1. 1. 1. 0. 1.
1. 1. 0. 0. 1. 1. 1. 1.      0. 0. 0. 0. 0. 0. 0. 1.
0. 0. 1. 1. 0. 0. 1. 1.      1. 1. 0. 1. 1. 1. 0. 1.
1. 1. 0. 0. 1. 1. 1. 1.      0. 0. 0. 0. 0. 0. 0. 0.
1. 1. 0. 0. 1. 1. 0. 0.      0. 0. 0. 0. 0. 0. 0. 0.
1. 1. 0. 0. 1. 1. 1. 1.      0. 0. 0. 0. 0. 0. 0. 0.
1. 1. 0. 0. 1. 1. 0. 0.      1. 1. 0. 1. 1. 1. 0. 0.
1. 1. 0. 0. 1. 1. 1. 0.      1. 1. 0. 0. 1. 0. 0. 0
};

```

图29

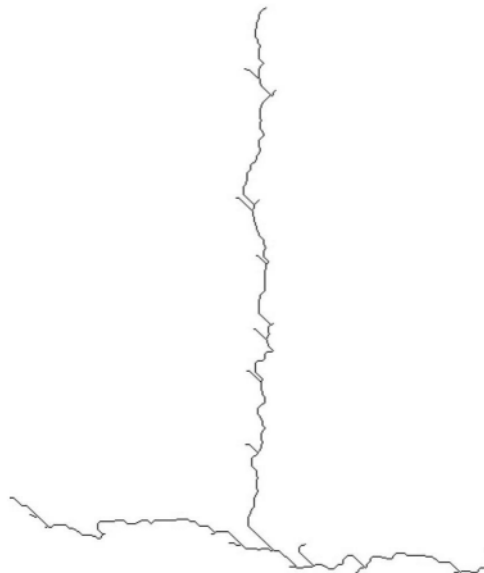


图30

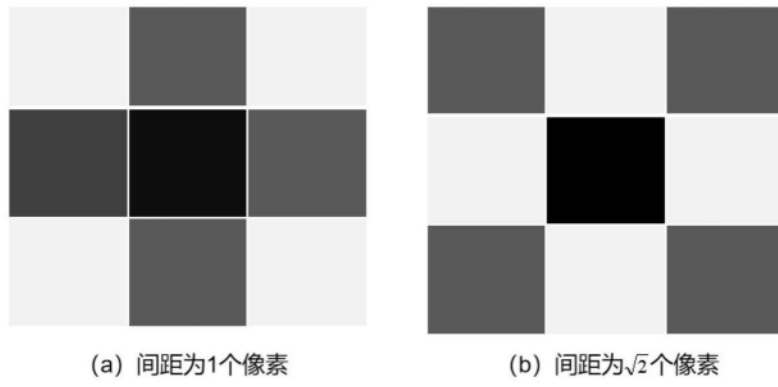


图31



图32



图33