



# 一种基于区域生长进行边缘跟踪测量路面损坏面积的方法

檀柏红<sup>1</sup>, 王娟<sup>2</sup>, 林挺<sup>1</sup>

(1. 天津科技大学经济与管理学院, 天津 300222; 2. 中国人民武装警察部队学院, 廊坊 065000)

**摘要:** 在基于单幅图像进行交通测绘处理中, 通常要在现场放置一个尺寸已知的标盘, 获得现场的数字图像后利用数字图像处理技术计算出所需的各种数据. 对路面的功能性破损如坑槽、沉陷、及油污污染面积的测量可以根据区域生长的规则进行边缘跟踪获得闭合边界, 从而计算出待测区域的像素面积, 再根据待测区域与标盘区域的比例关系计算出实际面积. 实验证明: 该方法时间复杂性低, 鲁棒性高, 能为路面损坏检测提供有利依据.

**关键词:** 面积测量; 边缘跟踪; 区域生长

中图分类号: TP317.4

文献标识码: A

文章编号: 1672-6510 (2008) 01-0063-04

## A Method to Measure Destructive Pavement Area Based on Region Growing and Boundary Tracking

TAN Bai-hong<sup>1</sup>, WANG Juan<sup>2</sup>, LIN Ting<sup>1</sup>

(1. College of Economics and Management, Tianjin University of Science & Technology, Tianjin 300222, China; 2. The People's Armed Police Academy, Langfang, 065000, China)

**Abstract:** A calibrated plate will be placed on the spot before digital pictures are taken to calculate all kinds of useful data with a distance measure system of traffic based on single view. For measuring pavement distress area such as groove, depressed, gasoline stained etc., a method based on region growing and boundary tracking was given in this paper. The closed boundary can be achieved with utilizing the algorithms of region growing and boundary tracking, furthermore the pixels in this area can be calculated. Finally the practical area can be calculated through the ratio relationship between the area and the calibrate plate. The method has low time complexity and high robust.

**Keywords:** area measurement; boundary tracking; region growing

随着我国经济实力的迅速增强, 公路建设取得了快速发展. 公路在使用过程中, 由于行车荷载作用和自然因素的影响, 路面会逐渐产生各种破损. 为了更好地监测公路的运营性能, 进行科学化的管理和维护, 必须运用快速、精确、高效、无损的现代化道路检测手段<sup>[1-3]</sup>. 数字摄影测量技术在测绘领域中的应用已基本成熟, 国内外已经开始将其引入道路交通现场测绘中, 利用单幅图像即可获得测绘所需数据, 基本思想是: 在现场放置尺寸已知的标盘, 利用数码相机拍摄现场, 利用图像处理技术计算出处理测绘所需的各种数据<sup>[4]</sup>. 目前世界上已经开发出一系列先进的路

面图像检测系统, 如英国 Transport Research Laboratory Ltd. 开发的 Harris<sup>[5]</sup>, 澳大利亚 ARRB 的 HAWKEYE2000<sup>[6]</sup>等, 系统可以对路面裂缝长度、损坏区域周长、面积进行检测, 但这些系统技术并未公开, 开发引进的费用昂贵.

由于利用数字图像处理技术可以对图像面积进行测量, 即利用数码相机先获得待测面积的图像, 统计出图像中目标的像素数, 乘以每个像素所占的实际面积就可以计算出图像的实际面积, 那么路面损毁面积也可以借此进行测量. 问题的关键在于如何获得待测区域的边缘, 获得边缘后可应用面积公式 (1) 计

收稿日期: 2007-09-06; 修回日期: 2007-12-05

基金项目: 天津科技大学科学研究基金资助项目 (20060304); 河北省交通厅公路设施损坏勘测数字化处理系统研究 (j-060115)

作者简介: 檀柏红 (1977—), 女, 河北丰南人, 讲师, 硕士.

算曲线包围的内部像素计数来求解<sup>[7]</sup>:

$$S = \frac{1}{2} \sum_{i=1}^n (x_i y_{i+1} - y_i x_{i+1}) \quad (1)$$

(当  $i=n$  时,  $i+1=1$ )

一般可以采取 Sobel 算子、Prewitt 算子、拉普拉斯算子等方法进行边缘提取,但这些边缘提取的算法很难获得闭合边界,从而难以统计出该区域的面积.此外,可以通过区域分割的方法获得待测区域的轮廓,如,胡小兵通过最优阈值对目标图像做分割<sup>[8]</sup>,通过计数独立轮廓区域内的像素来算面积.但是,它需要人工动态调整使独立轮廓区域边缘封闭,最优阈值也需要预先求取;赵鹏等给出了基于动态轮廓线的图像面积测量方法<sup>[9]</sup>,该方法应用动态轮廓线模型,使用基于 B 样条曲线的封闭曲线计算面积,适用于具有光滑且不规则形状边缘的目标面积的精确计算.这些方法均基于区域的方法对图像进行分割,为获得目标区域的轮廓,要处理所有目标轮廓中的像素,有较大的时间复杂度.

因此,本文针对交通测绘中对路面的功能性损坏如坑槽、凹陷、油污污染等面积测量的需要,通过对边缘提取算法和区域分割算法的研究,给出了一种利用数字图像处理技术中基于区域生长规则的边缘跟踪算法,以确定路面损坏的闭合边界,从而进一步获得该区域的面积,为道路路面损坏检测提供有利依据.

### 1 基于区域生长规则的边缘跟踪算法

物体边缘通常是连续的,边缘点之间有一定的约束关系.在某个边缘点附近沿边缘方向总能找到另一个边缘点,且这两个边缘点的灰阶和方向走向不会相差太大.如果在边缘提取算法中按照边缘点的连续关系进行跟踪就可以只处理物体边缘附近的像素获得物体轮廓,避免基于一般区域分割的算法对整体图像进行扫描<sup>[8,9]</sup>,从而具有很小的时间复杂度.

#### 1.1 区域生长算法的基本原理和实现

在区域生长算法中,通常有一个扩充邻域  $N$  和一个判断扩充邻域内两像素是否相似的判决规则  $C$ <sup>[10]</sup>.令集合  $R$  代表整个图像区域,  $T$  代表一个已经检测出的目标区域,  $B$  代表不属于这个目标区域的所有其他像素.可以找到满足以下条件的集合  $I$  和  $O$ :

- (1)  $I$  是  $T$  的子集,  $O$  是  $B$  的子集;
- (2) 如果有一个元素  $p, p \in T$  且  $p$  的扩充邻域内含有属于  $B$  的元素,则定有  $p \in I$ ;

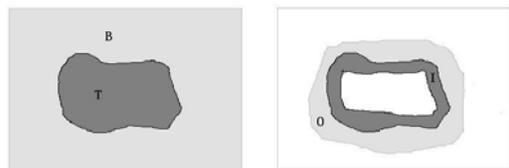
(3) 如果有一个元素  $p, p \in B$  且  $p$  的扩充邻域内含有属于  $T$  的元素,则定有  $p \in O$ ;

(4) 对  $I$  中的每个元素  $p, p$  的扩充邻域内的所有属于  $B$  的元素都位于集合  $O$  中;

(5) 对  $O$  中的每个元素  $p, p$  的扩充邻域内的所有属于  $T$  的元素都位于集合  $I$  中.

显然,上述条件(2)决定了  $I$  中包含了构成目标区域  $T$  的边缘的所有像素.如果能够找到这样一个  $I$ ,也就能找到目标区域  $T$  的边缘.本文即基于这一点,把区域生长算法中对  $T$  中所有元素的处理转变为对  $I$  和  $O$  中元素的处理,大大减少所需处理的元素,从而降低算法的时间和空间复杂度.

区域生长算法中像素集合关系见图 1.由图 1 可知,集合  $I$  使得  $T$  中的所有其他元素与  $B$  中的元素“隔离”开来,而集合  $O$  使  $B$  中的所有其他元素都与  $T$  中的元素“隔离”开来,所以只要对  $I$  和  $O$  中元素进行考察,当目标区域  $T$  已经无法继续扩张下去就知道了区域边缘的位置,进而再进行边缘跟踪,即可获得闭合的区域边缘.



(a) 图像区域与目标区域的关系 (b) 目标区域子集与图像区域子集的关系

图 1 区域生长算法中像素集合关系  
Fig. 1 Relationships of pixel sets in region growing algorithm

下面给出区域生长算法实现的步骤:

(1) 将集合  $I$  和  $O$  置空,将已知的一对边缘出发点中属于目标区域的那个元素放入集合  $I$  中,另一个放入  $O$  中.

(2) 对不满足相互“隔离”条件的  $I$ 、 $O$  进行扩充:对  $I$  或  $O$  中的任意一个元素  $p$ ,如果在  $p$  的扩充邻域内有一个不属于  $I \cup O$  的元素  $w$ ,且  $w$  与  $p$  连线穿过  $O$  (或  $I$ ) 中的任何元素或者是与  $O$  (或  $I$ ) 中的任意相邻元素的连线相交,则对  $w$  进行“判别”后放入集合  $I$  或  $O$  中.判别规则如下:如果在  $w$  的扩充邻域内能找到一个元素  $p, p \in I$  且  $p$  与  $w$  满足相似规则  $C$ ,则将  $w$  放入  $I$  中,否则将  $w$  放入集合  $O$  中.重复步骤(2)直到  $I$ 、 $O$  满足相互“隔离”.

(3) 生长  $I$  和  $O$ :记所有  $I$  中元素的扩充邻域所包含的元素所构成的集合为  $N(I)$ ,同理有

$N(O)$ . 对集合中任何一个不属于  $I \cup O$  的元素  $w$ , 如果  $w$  与  $N(O)$  中任何一个元素满足相邻关系, 则按照 (2) 中的判别规则将  $w$  放入集合  $I$  或者  $O$  中. 转到 (2) 继续执行.

上述步骤 (2) 保证在扩充时  $I$  和  $O$  始终满足互相“隔离”. 因而最终得到的集合  $I$  和  $O$  能够满足前面论述中所提到的条件 (4)、(5), 步骤 (2)、(3) 保证最终得到的集合  $I$  和  $O$  能够满足前面论述中所提到的条件 (2)、(3).

在步骤 (2) 所提到的判别规则中, 只对  $w$  与当前  $I$  中的元素进行相似规则  $C$  的判断后即将某些  $w$  放入了集合  $O$  中, 而当前  $I$  中可能并未完全包含  $w$  的扩充邻域内所有应该属于集合  $I$  的元素, 因此可能出现误将  $w$  放入集合  $O$  的情况. 为此, 引入错误恢复算法:

(4) 在将任意一个元素  $p$  放入集合  $I$  时, 对  $p$  的扩充邻域内任一元素  $w$ , 如果有  $w \in O$  且  $p$  与  $w$  满足相似规则  $C$ , 将  $w$  从集合  $O$  中删除, 并将它放入临时集合  $temp$  中.

(5) 从  $temp$  中取出一个元素  $p$ , 按照步骤 (4) 将  $p$  放入集合  $I$  中; 重复 (5) 直到  $temp$  为空. 转到 (2) 继续执行.

引入错误恢复算法后, 可以使最终得到的集合  $I$  和  $O$  满足前面所提到的条件 (1).

## 1.2 基于区域生长的边缘跟踪算法

基于上面所给出的区域生长算法的原理, 可以得到与其对应的边缘跟踪算法<sup>[11,12]</sup>. 由于路面损坏面积的测量通常是在与用户交互的环境下进行, 因此可以由用户首先选取所关心的区域中的任意一点, 然后根据这点找到第一对边缘点  $i$  和  $o$  后进行扩充. 并且规定扩充邻域  $N$  为 8, 两个像素  $p_1, p_2$  相似当且仅当满足公式 (2) 成立.

$$|p_1 - p_2| < Range(Max(p_1, p_2)) \quad (2)$$

其中,  $Range(I) = (W_{max} - W_{min})(I/I_{max})^n + W_{min}$

公式 (2) 中  $p_1, p_2$  为两个像素对应的灰度值,  $I_{max}$  为图像中最大灰度值,  $I$  为当前像素灰度值,  $[W_{min}, W_{max}]$  为用户选取的一个最小、最大权值范围,  $n$  值为 1, 2 或 3, 分别表示式 (2) 可以是线性式、平方式或立方式, 这里取  $n=1$ , 即采用线性式.

通过分析可知, 当扩充邻域  $N$  为 8 时, 只要集合  $I$  和  $O$  分别满足集合  $I$  和  $O$  中的所有元素都能按四邻域相邻的原则相连成一条线就能实现互相“隔离”. 因此, 可以得到与 1.1 节对应的边缘跟踪算法:

(1) 初始化: 标记所有像素为 UNPROCESSED

(未处理), 将第一对边缘发起点  $i$  和  $o$  的副本压入边缘堆栈  $edge$ , 标记  $i$  点为 INSIDE,  $o$  点为 OUTSIDE.

(2) 开始扩充边缘: 对当前的边缘发起点  $i$  和  $o$  寻找下一对边缘发起点  $i_1$  和  $o_1$ , 要求其满足:

a)  $i$  和  $i_1$  四邻域相邻;  $o$  和  $o_1$  四邻域相邻. (在这里认为像素到自身也满足四邻域相邻, 即  $i$  与  $i_1$  或者  $o$  与  $o_1$  可以相同)

b)  $i_1$  点的标号不为 OUTSIDE,  $o_1$  点的标号不为 INSIDE, 且  $i_1$  和  $o_1$  点中至少有一个点的标号为 UNPROCESSED.

c)  $i$  和  $i_1$  点满足上述“相似”条件;  $o_1$  和它扩充邻域内所有标号为 INSIDE 的点不满足“相似”;  $o_1$  和  $i_1$  不满足“相似”, 把  $i_1$  点标记为 INSIDE,  $o_1$  点标记为 OUTSIDE, 并设置当前边缘发起点为  $i_1$  和  $o_1$ , 转到 (2) 继续执行.

(3) 如果在 (2) 中没有找到满足条件的  $i_1$  和  $o_1$ , 则寻找这样一对  $i_1$  和  $o_1$ :  $i_1$  与  $i$  八邻域相邻, 其余条件与 (2) 中相同, 标记  $i_1$  点为 INSIDE,  $o_1$  点为 OUTSIDE. 在设置当前边缘发起点时应注意: 由于  $i$  与  $i_1$  不满足四邻域相邻, 因而可能对  $O$  集合不能形成“隔离”, 所以此处要分情况考虑:

a) 如果  $i_1$  与  $i$  扩充邻域的交集含有标号为 0 的元素  $w$ , 则对  $w$  进行判断: 如果  $w$  与它扩充邻域内所有标号为 INSIDE 的点都不满足相似, 标记  $w$  点为 OUTSIDE, 将当前边缘发起点  $i$  和  $o$  压入边缘堆栈  $edge$ , 设置当前边缘发起点为  $i$  和  $w$ ; 否则, 标记  $w$  点为 INSIDE, 设置当前边缘发起点为  $i_1$  和  $o_1$ .

b) 如果  $i_1$  与  $i$  扩充邻域的交集不含有标号为 0 的元素, 则设置当前边缘发起点为  $i_1$  和  $o_1$ , 转到 (2) 继续执行.

(4) 如果 (2)、(3) 均未找到  $i_1$  和  $o_1$ , 则从堆栈  $edge$  中弹出一对边缘发起点并设置它们为当前边缘发起点, 转到 (2) 继续执行.

需要注意的是: 当任何元素  $p$  设置标号为 INSIDE 后, 必须对  $p$  与  $p$  的扩充邻域内所有标号为 OUTSIDE 的元素进行“相似”性的判断, 如果它们满足“相似”, 则把当前边缘发起点压入堆栈  $edge$  中, 将  $p$  压入错误堆栈  $wrong$  中, 然后转到错误恢复算法 (5).

(5) 错误恢复算法: 从堆栈中弹出一个元素  $p$ , 设置  $p$  的标号为 INSIDE, 检查  $p$  与  $p$  扩充邻域内的所有标号为 OUTSIDE 的元素的“相似”性, 把满足“相似”的元素压入错误堆栈  $wrong$  中. 重复

上述操作直到堆栈 wrong 为空, 然后转到 (2) 继续执行.

利用此算法即可获得闭合区域的边缘于堆栈 edge 中.

### 2 实验分析

由于数码相机拍摄所得数字图像实际尺度很难度量, 必须在待测面积附近放置已知面积的平面物体作为参照, 本文实验中在损坏路面附近放置的是边长 32 cm 的正方形标盘, 其中心圆直径为 20 cm, 根据图像中标盘中心圆形成的图像的面积与真实面积之间的比例关系, 即可测得损坏路面的图像面积. 假设, 已知标盘中心的圆形在图像中所占的像素数为  $N_I$ , 目标像素数为  $N_p$ , 而标盘中心圆形所代表的实际面积为  $S_I$ ,  $S_p$  表示目标图像的面积, 这里指路面损坏的面积, 那么目标图像的像素所占的实际面积可近似地表示为

$$S_p = N_p \cdot S_I / N_I \quad (3)$$

由于路面破坏后的情况经数码相机成像到面阵 CCD 上, 该图像实质上是二维的点阵, 成像后的各点灰度不同, 另外如果图像为彩色图像, 可以经过灰度变换, 转换成灰度图像, 灰度图像已经没有光照强度影响的问题, 在此基础上用户只需在待测路面区域和标盘中心区域交互各确定任意一个点, 然后根据这两个点分别找到各自区域的一对边缘点  $i$  和  $o$  后进行扩充, 就可以根据前述算法获得路面损坏区域闭合边缘. 根据式 (1) 计算出待测损坏路面的像素面积  $N_p$  和标盘的像素面积  $N_I$ , 进而根据式 (3) 即可统计出目标区域的实际面积.

本文算法编程是在 Pentium4 PC 机 Windows 2000 操作系统 VisualC++6.0 版本的集成环境下完成的. 实验图像拍摄采用数码相机 Nikon Coolpix995 (2048 × 1536). 如图 2 所示, 图 2 (a) 为现场拍摄的路面被油污破坏的情况, 图 2 (b) 为采用本文算法对图 2 (a) 进行区域生长获得的实验结果, 图 2 (c) 为按本文边缘跟踪算法获得的实验结果, 该图中灰色像素区域即为集合  $I$ , 黑色像素区域即为集合  $O$ , 而位于灰色区域中最外侧被集合  $O$  所包围的点构成了目标区域的外轮廓. 因此可以非常方便地统计出检测出的闭合边缘所包含的像素点, 即  $N_I$ ,  $N_p$ , 计算结果如表 1 和表 2.

该路面被油污破坏部分的面积按照现场测量的结果为 9 281.30 cm<sup>2</sup>, 该结果是按照油污面积取四边

的切线所成的矩形 (长 100.2 cm, 宽 92.6 cm), 由此可以看出, 自动测量的结果其精度完全可以满足交通现场测量的要求, 而且更为准确.

从表 2 中区域生长法和区域生长边缘跟踪法的数据对比可以看出由于区域生长算法需要处理目标区域的所有元素, 因此其计算时间是与目标区域的面积相关的, 设目标区域的直径为  $n$ , 那么区域生长算法的时间复杂度为  $O(n^2)$ . 而基于区域生长规则了的边缘跟踪算法只需要处理目标边缘附近的部分元素, 因此其时间复杂度为  $O(n)$ .

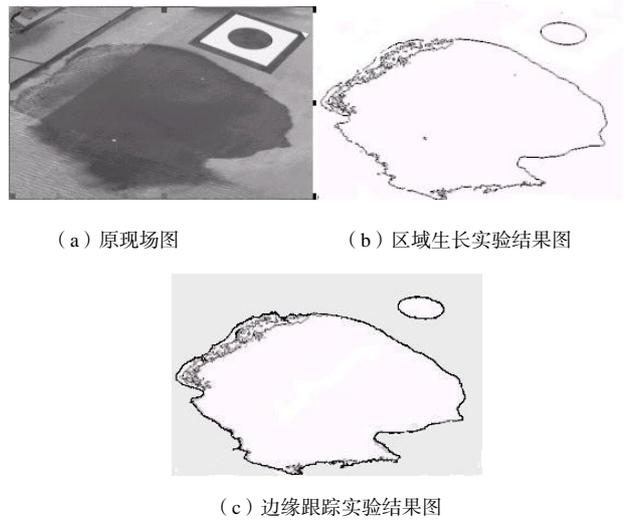


图 2 区域生长进行边缘跟踪测量路面损坏面积实验结果图  
Fig. 2 Results of measuring pavement distress area with region growing and boundary tracking method

表 1 基于区域生长边缘跟踪测量路面损坏面积实验数据  
Tab. 1 Result of pavement distress area with boundary tracking algorithm based on region growing

$N_I$ /pixel	$N_p$ /pixel	$S_I$ /cm <sup>2</sup>	$S_p$ /cm <sup>2</sup>
54 120	1 532 768	314	8 892.48

表 2 区域生长法和区域生长边缘跟踪法实验数据对比  
Tab. 2 Experiment comparison of region growing and boundary tracking

方 法	时间/ms	最大堆栈长度
区域生长法	1 780	3 012
区域生长边缘跟踪法	282	2 934

### 3 结 论

本文实现了用户交互点取目标区域发起点的基础上, 利用基于区域生长规则快速边缘跟踪算法以可 (下转第 75 页)