

二值图像连通域标记优化算法

罗志灶, 周赢武, 郑忠楷

(闽江学院 电子系, 福建 福州 350108)

摘 要: 在分析现有二值图像像素扫描连通域标记算法的基础上, 提出像素扫描连通域标记优化算法。本算法具有速度快, 算法简单, 易于实现的特点, 仅需两次扫描, 即可实现像素的多目标标记。本算法将背景也作为目标加以标记, 分两步扫描图像和临时连通域标号矩阵完成连通域的标记和合并。采用顺序存储结构存储和处理等价标号, 算法速度快, 节约内存。

关键词: 二值图像; 连通域; 像素扫描; 标记

中图分类号: TP391

文献标识码: A

文章编号: 1007-4260(2010)04-0034-06

0 引 言

二值后的图像, 是由白像素点(通常二值图像用“1”, 灰度图像用“255”来表示)和黑像素点(通常用“0”来表示)组成的点阵图像^[1]。连通区域标记是指将图像中符合某种连通规则(4 邻域连通、8 邻域连通或 m 邻域)的像素标识为同一目标^[2], 设计合适的数据结构保存每个像素点所属的目标的标号, 和保存相关的目标的属性^[3], 例如目标的面积即目标的像素点的数量、重心、二阶矩等参数。连通域标记法是计算机视觉和目标识别的重要步骤, 直接影响目标识别的准确性和速度^[4]。

影响连通域标记算法的性能主要有两个方面: a) 图像扫描方式及等价连通域标号冲突处理的方法; b) 存储连通域信息的数据结构。提高连通域标记算法性能也着眼于这两方面: 减少图像的扫描次数, 尽可能减少回溯扫描, 一次扫描尽可能多地提取连通域的信息; 设计合适的数据结构, 尽可能减少连通域信息访问的时间, 提高算法效率。

1 现有算法分析

图像的连通域标记算法有多种, 根据扫描方法不同可分为像素扫描即点标记法^[5]和线段扫描即线标记法^[6]。像素点扫描方式有顺序扫描法^[7]、递归标记法^[8]、区域增长法^[9]等。线段扫描算法主要有基于游程的标记算法^[11]等。各算法均能准确标记目标连通域, 其主要区别在于等价标号的处理。最终的标记要把等价标号标记为同一标号。优化算法的目的是在正确解决等价标记和冲突标记基础上, 降低算法的复杂度和提高算法的性能。

像素点扫描法是比较常用的算法, 其特点是直观, 数据结构简单, 易于实现^[5]。二值图像扫描后, 会有大量的冲突标号。通常的做法是对二值图像迭代扫描, 以完全消除等价标号^[8], 算法运行时间不稳定。算法常用链表结构、树型结构或堆栈结构等指针型结构存储等价标号^[7], 大量的运行时间消耗在递归调用和指针传递上。

本文针对像素点扫描法的缺陷, 提出了改进像素点扫描法的算法。首先, 针对常用的存储等价标号的链表结构的缺点, 改用顺序存储结构以减少单个像素点的内存访问次数; 其二, 用与二值图像大小相同的二维数组存储标号, 标记每个像素点的连通域标号, 方便后续图像处理; 其三, 用一维数组存储等价

* 收稿日期: 2010-04-26

基金项目: 福建省重点学科项目(闽教高[2006]48号)资助。

作者简介: 罗志灶, 男, 福建三明人, 闽江学院电子系讲师, 硕士, 主要从事数字图像和嵌入式系统设计与开发。

标号的共同连通域标号及目标属性, 用于解决冲突标号的处理; 最后, 改进等价标号合并算法, 重新排序标号, 获得合理的目标标号。

2 算法描述

2.1 术语定义

连通域标号: 图像像素点所在的连通域的序号, 用与图像大小相同的二维数组保存每个像素点的连通域标号。连通域标号在本算法有两个阶段: 第一阶段, 对二值图像扫描取得的临时连通域标号, 此阶段, 会有不同的连通域标号属于同一目标。因此, 将此阶段的像素点连通域标号称为待合并连通域标号或临时连通域标号。第二阶段, 合并等价连通域标号, 即利用共同连通域标号替换第一阶段的临时连通域标号, 此阶段的像素点连通域标号就是最终的目标标号, 称为目标连通域标号。

等价标号: 在待合并连通域标号矩阵中, 会有不同的连通域标号属于同一目标, 则将此类连通域标号称为等价标号, 也称为冲突标号。

共同连通域标号: 指示待合并连通域标号所标记的连通域所属的目标的标号, 用一维数组保存, 以待合并连通域标号为下标, 该值指示临时连通域标号所标记的连通域属于哪个目标。

2.2 算法简介

本算法分为两个阶段。

第一阶段, 对二值图像进行一次扫描, 按某种连通(4 邻域或 8 邻域)规则, 标记所有像素点的待合并连通域标号, 同时, 按一定的规则, 标记待合并连通域标号的共同连通域标号。由于 4 邻域和 8 邻域的模板核过小, 无法一次正确标记所有的目标, 会有大量等价标号存在。用共同连通域标号标记各等价标号所属的共同连通域。

第二阶段, 扫描临时连通域标号矩阵, 纠正矩阵中的临时连通域标号, 即用共同连通域标号替换各像素点的临时连通域标号, 实现连通域的合并。合并时, 按共同连通域标号出现的次序, 重新定序, 确保目标连通域标号连续。合并后, 矩阵中的像素点连通域标号即是最终所得的目标连通域标号。

2.3 算法原理

本算法的 8 邻域和 4 邻域的处理方法相似, 以 4-邻域为例, 对算法阐述。

设某像素点 $f(x, y)$, 则 $f(x-1, y)$, $f(x+1, y)$, $f(x, y-1)$, $f(x, y+1)$ 为其四邻域左、右、上、下像素点。设 $\text{merge}(x, y)$ 是 $f(x, y)$ 像素点连通域标号。当扫描 $f(x, y)$ 时, 已完成了 $f(x-1, y)$ 和 $f(x, y-1)$ 扫描, $\text{merge}(x-1, y)$ 和 $\text{merge}(x, y-1)$ 为已知数。因此在 4-邻域内, $f(x, y)$ 像素点连通域标号 $\text{merge}(x, y)$ 仅和像素点 $f(x-1, y)$ 、 $f(x, y-1)$ 及其像素点连通域标号 $\text{merge}(x-1, y)$ 、 $\text{merge}(x, y-1)$ 有关, 如式(1)所示。

$$\text{merge}(x, y) = \begin{cases} \text{merge}(x-1, y) & \text{若 } f(x, y) = f(x-1, y) \text{ 且 } f(x, y) \neq f(x, y-1) \\ \text{merge}(x, y-1) & \text{若 } f(x, y) \neq f(x-1, y) \text{ 且 } f(x, y) = f(x, y-1) \\ \text{merge}(x, y-1) & \text{若 } f(x, y) = f(x-1, y) \text{ 且 } f(x, y) = f(x, y-1) \\ \text{Newlabel} & \text{若 } f(x, y) \neq f(x-1, y) \text{ 且 } f(x, y) \neq f(x, y-1) \end{cases} \quad (1)$$

式(1)表示:

(1) 当像素点 $f(x, y) = f(x-1, y)$ 且 $f(x, y) \neq f(x, y-1)$, 即 $f(x, y)$ 和左邻域像素点值相等时, 表明 $f(x, y)$ 与左邻域像素点是连通的, 则 $f(x, y)$ 的像素点连通域标号 $\text{merge}(x, y)$ 与 $\text{merge}(x-1, y)$ 标号相同。

(2) 当像素点 $f(x, y) \neq f(x-1, y)$ 且 $f(x, y) = f(x, y-1)$ 时, 表示 $f(x, y)$ 与上邻域连通, 但与左邻域不连通, 则 $f(x, y)$ 的像素点连通域标号 $\text{merge}(x, y)$ 与 $\text{merge}(x, y-1)$ 标号相等。

(3) 当像素点 $f(x, y) = f(x, y-1)$ 且 $f(x, y) = f(x-1, y)$ 时, 则 $f(x, y)$ 与上邻域、左邻域在同一个连通域内, 则需考虑:

(i) 若 $\text{merge}(x, y-1) = \text{merge}(x-1, y)$, 则表明 $f(x, y)$ 的上邻域和左邻域连通域标号一致, 仅需 $\text{merge}(x, y) = \text{merge}(x, y-1)$ 即可。

(ii) 若 $\text{merge}(x, y-1) \neq \text{merge}(x-1, y)$, 则表明 $f(x, y)$ 的上邻域和左邻域连通域标号冲突, 需

按 2.4 节处理冲突标号, 然后 $\text{merge}(x, y) = \text{merge}(x, y - 1)$ 。

(4) 若 $f(x, y) \neq f(x - 1, y)$ 且 $f(x, y) \neq f(x, y - 1)$ 时, 表明像素点 $f(x, y)$ 属于新的连通域, 因此连通域标号自动加 1, 即 $\text{Newlabel} = \text{Newlabel} + 1$, 并将新的连通域标号 Newlabel 赋予 $\text{merge}(x, y)$ 。

2.4 等价标号处理

本算法的实现难点主要是等价标号的处理和共同连通域的标记。如上节所述, 在算法的第一阶段, 扫描图像后获得临时连通域标号 merge , 其中会出现大量的等价标号即标号冲突, 用一维数组记录各临时连通域标号的共同连通域标号, 数组的下标为临时连通域标号, 其值为共同连通域标号。当遇到标号冲突时, 要合并等价连通域标号, 即对共同连通域数组扫描一遍, 将等价标号的共同连通域标号标记成一致。

设一维数组 common , 其下标为临时连通域标号, 即 $\text{merge}(x, y)$ 的值, common 元素的值表示某个共同连通域标号。 $\text{common}(\text{merge}(x, y))$ 表示像素点 $f(x, y)$ 的共同连通域标号。

扫描二值图像时, common 按如下方法处理:

(i) 当 $f(x, y) \neq f(x, y - 1)$ 且 $f(x, y) \neq f(x - 1, y)$ 时, 表明像素点 $f(x, y)$ 属于新的连通, 则共同连通域标号 common 新增一个元素, 即 $\text{common}(\text{merge}(x, y)) = \text{merge}(x, y)$ 。

(ii) 扫描图像时, 当出现 $f(x, y) = f(x, y - 1)$ 且 $f(x, y) = f(x - 1, y)$ 和 $\text{merge}(x - 1, y) \neq \text{merge}(x, y - 1)$ 时, 则说明遇到标号冲突, 需扫描 common 数组一遍, 对于任一元素 i , 若 $\text{common}(i) = \text{common}(\text{merge}(x - 1, y))$ 则修改其共同连通域标号 $\text{common}(i) = \text{common}(\text{merge}(x, y - 1))$ 。

扫描图像时, (i) 表明出现新的孤点, 共同连通域标号 common 新增元素标记该元素。(ii) 表明元素 $f(x, y)$ 与它的左邻域 $f(x - 1, y)$ 和上邻域 $f(x, y - 1)$ 连通, 且连通域标号 $\text{merge}(x - 1, y)$ 与 $\text{merge}(x, y - 1)$ 不一致, 即左邻域和上邻域冲突, 需要合并, 因此要对共同连通域标号 common 中所有的值等于 $\text{merge}(x - 1, y)$ 的元素改为 $\text{merge}(x, y - 1)$ 。

经合并处理后, merge 的元素仍是临时连通域标号, 有大量的冲突标号, 但图像的每个像素点 $f(x, y)$ 可通过 $\text{common}(\text{merge}(x, y))$, 得到共同连通域标号, 且是唯一的。此时, common 的值是断续的, 对后续的处理很不利, 需要对 common 及 merge 的标号调整。

调整方法如下:

定义临时一维数组 temp 及变量 nIndex , temp 大小与 common 相同。 temp 元素初始化为 -1, nIndex 初始化为 0。

扫描 merge 数组, 对任一元素 $\text{merge}(x, y)$ 作如下操作:

(i) 若 $\text{temp}(\text{common}(\text{merge}(x, y))) < 0$, 即该共同连通域标号首次扫描到, 则

$\text{nIndex} = \text{nIndex} + 1$;

$\text{temp}(\text{common}(\text{merge}(x, y))) = \text{nIndex}$;

$\text{merge}(x, y) = \text{temp}(\text{common}(\text{merge}(x, y)))$;

(ii) 若 $\text{temp}(\text{common}(\text{merge}(x, y))) \geq 0$, 即该共同连域已出现过, 则

$\text{merge}(x, y) = \text{temp}(\text{common}(\text{merge}(x, y)))$;

上述调整主要完成合并图像等价连通域即用唯一的标号标记连通域, 并按出现的先后次序, 标记连通域。操作(i)表示: 该等价标号所指的共同连通域标号首次出现, 因此标号自动加 1。操作(ii)表示: 该等价标号所指的共同连通域标号至少已出现过 1 次, 因此只需 temp 的已分配的标号直接赋给 $\text{merge}(x, y)$ 即可。至此, 对于每个像素点 $f(x, y)$, 均可在 merge 的相同位置检索到连通域标号 $\text{merge}(x, y)$ 。

图像的第一行像素点没有上邻域, 第一列没有左邻域, 需特殊处理。如下所示:

(i) 二值图像左上角的像素 $f(0, 0)$, 由于是第 1 个扫描的像素, 无需考虑相邻点连通性。

(ii) 二值图像第 1 行(最上行)的像素 $f(x, 0)$, 只需考虑左相邻像素的连通性。

(iii) 二值图像第 1 列(最左列)的像素 $f(0, y)$, 只需考虑上相邻像素的连通性。

除此之外的所有像素都要考虑左、上 2 个相邻像素的连通性来确定自己的连通性。

2.5 算法实现

算法运行分为两个阶段, 设 $f(x, y)$ 是图像的像素点:

第一阶段:

(1) 提取图像参数, 如宽 $nWidth$ 和高 $nHeight$ 等。

(2) 初始化 $merge(0, 0) = 0$, $common(0) = 0$, $nIndex = 0$;

(3) 扫描图像第1行, x 从1开始, 标记临时连通域标号 $merge(x, 0)$ 和共同连通域 $common$ 。按如下处理:

若 $f(x, 0) = f(x-1, 0)$, 则 $merge(x, 0) = merge(x-1, 0)$, 否则 $nIndex = nIndex + 1$,

$merge(x, 0) = nIndex$, $common(merge(x, 0)) = merge(x, 0)$;

(4) 扫描图像其余行

(i) 标记图像每行的第1个像素 $f(0, y)$, 若 $f(0, y) = f(0, y-1)$, 则

$merge(0, y) = merge(0, y-1)$; 若 $f(0, y) \neq f(0, y-1)$, 则 $nIndex = nIndex + 1$, $merge(0, y) = nIndex$, $common(merge(0, y)) = merge(0, y)$;

(ii) 标记其余像素点 $f(x, y)$ ($x > 0, y > 0$),

若 $f(x, y) = f(x-1, y)$ 且 $f(x, y) \neq f(x, y-1)$, 则 $merge(x, y) = merge(x-1, y)$;

若 $f(x, y) \neq f(x-1, y)$ 且 $f(x, y) = f(x, y-1)$, 则 $merge(x, y) = merge(x, y-1)$;

若 $f(x, y) = f(x-1, y)$ 且 $f(x, y) = f(x, y-1)$, 则比较 $merge(x-1, y)$ 和 $merge(x, y-1)$;

若 $merge(x-1, y) = merge(x, y-1)$, 则 $merge(x, y) = merge(x, y-1)$;

若 $merge(x-1, y) \neq merge(x, y-1)$, 则扫描 $common$, 对任一元素 i ; 若元素 $common(i) = common(merge(x-1, y))$, 则 $common(i) = common(merge(x, y-1))$;

若 $f(x, y) \neq f(x-1, y)$ 且 $f(x, y) \neq f(x, y-1)$, 则

$nIndex = nIndex + 1$, $merge(x, y) = nIndex$, $common(merge(x, y)) = merge(x, y)$ 。

第二阶段:

合并图像等价连通域, 并按出现的先后次序, 标记合并后的连通域。

(1) 定义临时一维数组 $temp$ 及变量 $index$, $temp$ 大小与 $common$ 相同。 $temp$ 元素初始化为 -1。 $index$ 初始化为 0。

(2) 扫描 $merge$ 数组, 按如下操作:

若 $temp(common(merge(x, y))) < 0$, 则 $temp(common(merge(x, y))) = index$;

$index = index + 1$;

若 $temp(common(merge(x, y))) \geq 0$, 即该共同连域已出现过, 则

$merge(x, y) = temp(common(merge(x, y)))$ 。

2.6 算法特点分析

(1) 初次扫描图像过程中, 在标记临时的连通域的同时, 对冲突标号进行初步标记, 即用一维数组记录等价标号的共同连通域标号, 一方面, 当遇到等价标号时, 无需回溯扫描; 另一方面, 在扫描图像时, 完整保留标号冲突的信息, 标记等价连通域的共同标号。

(2) 临时连通域标号扫描, 集中处理等价标号, 一方面, 合并等价连通域, 用连续的、唯一的共同连通域标号替代临时的待合并连通域标号, 用二维数组标记各像素点的连通域标号, 以利于后续图像处理, 可根据像素点坐标, 检索连通域标号; 另一方面, 重新排列各连通域标号, 以扫描过程中出现的次序, 重新标记各连通域。

(3) 本算法采用顺序存储结构保存待合并连通域标号和共同连通域标号, 利用像素点连通域标号的存储空间, 在不同阶段分别存储待合并连通域标号和像素点的最终目标连通域标号, 既节约算法所需内存, 又提高连通域标号索引的速度。一次扫描图像后, 即可通过临时连通域标号索引共同连通域标号, 获得目标连通域标号, 但将给后续的目标处理带来麻烦, 因此本算法采用二次扫描方式, 第二次扫描, 重新排列各连通域标号, 用二维数组标记各像素点的目标连通域标号。

3 实验分析与结论

本算法与迭代扫描算法^[15]、改进型迭代算法^[7]、常用的链表方式保存等价对的算法^[14]、游程标记算法^[11] 进行比较。这几类算法均具有代表性, 是常用的算法。所选算法中, 均有本算法相应的改进部分, 具有可比性。

本文选取不同结构的图像(512×512), 在 Matlab 环境下, 比较各算法的执行时间。链表型算法, 采用模拟指针方法。实验结果如表 1 所示。多数情况下, 本算法远优于基于游程标记算法和多次扫描算法, 甚至可缩短 10 倍时间, 同时本文算法优于基于链表的标记算法和改进型多次扫描算法。但任何算法都有局限性, 对于多细节和纹理的图像, 本文算法劣于改进型多次扫描算法, 主要原因是图像目标过小、数目过多, 且同一目标的连通域的冲突标号过多, 将导致重复遍历共同连通域标号数组急剧增加; 而由于目标小, 则多次扫描算法, 所需迭代扫描图像的次数反而更少, 因此表中对纹理型图像的标记, 表 1 中的

表 1 各算法不同图像的运行统计表

图像名称	本文算法	多次扫描算法	改进型多次扫描算法	基于链表算法	基于游程算法
laine. 512. pgm	0. 109	2. 484	0. 204	0. 219	4. 594
2. pgm	0. 094	0. 516	0. 219	0. 141	3. 5
7. pgm	0. 094	0. 703	0. 204	0. 172	3. 516
13. pgm	0. 062	0. 516	0. 172	0. 094	3. 094
1. 1. 03. tiff	1. 891	4. 141	0. 313	5. 453	8. 766
5. 2. 09. pgm	0. 204	3. 078	0. 234	0. 469	4. 047
cmpndd. pgm	0. 047	0. 297	0. 172	0. 063	3. 11

1. 1. 03. tiff 图像, 本算法却劣于改进型多次扫描算法。

本文将 512×512 图像, 按 16 *n×16 *n 划分, 将图像分成 32 幅不同大小, 以分析不同尺度图像的各算法的执行时间, 并随机抽取其中一组数据, 结果如图 1、图 2 所示。

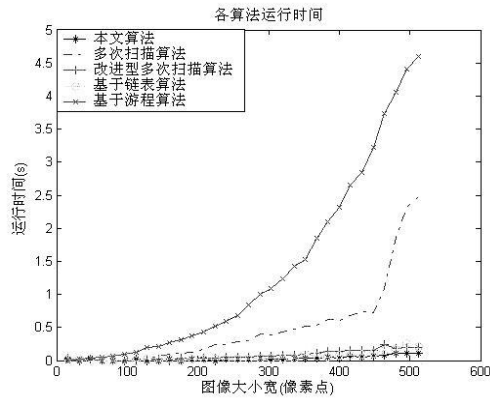


图 1 不同尺度图像下五种算法的执行时间(秒)

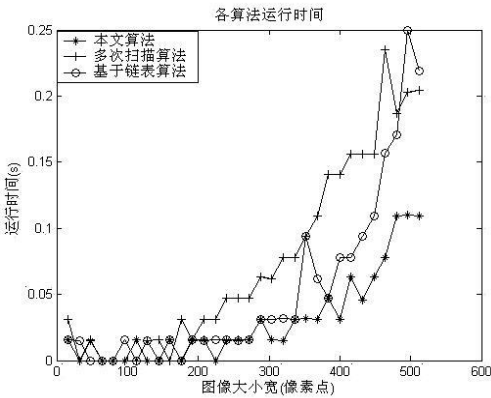


图 2 不同尺度图像下三种算法的执行时间(秒)

图 1 将 5 种算法进行比较, 图 2 则将相对执行时间较少的 3 种算法进行比较。由于 Matlab 计时不够精确, 执行时间在很少时, 会有图像更大, 但时间反而更少的现象, 但这不影响对算法执行时间的整体分析。

实验表明在不同尺度图像下, 本文算法均优于其它算法, 且本文算法执行时间相对较稳定, 且图像尺寸越大, 本文算法优势越明显。

参考文献:

[1] Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing(2nd ed.) [M] . 北京: 电子工业出版社, 2007.

[2] Sonka M, Hlavac V, Boyle R. Image Processing, Analysis, and Machine Vision(2nd ed.) [M] . 北京: 人民邮电出版社, 2002.

[3] Bulgarelli A, Stefano L D. A Simple and Effeient Connected Component Labeling Algorithm [J] . In: Proc of the 10th Intemational conference on Image Analysis and Processing, Venice, Italy, 1999; 322— 327.

[4] Yang, X. D. An Improved Algorithm for Labeling Connected Components in a Binary Image [J] . International Journal of Computer Vision, Graphics, and Image Processing, 1992; 555— 569.

[5] Fast Connected Component Labeling Algorithm Using A Divide and Conquer Technique [R] . TR— 2000— 04, Department of Computer Science, The University of Alabama.

- [6] IRANIM, ANANDAN P, BERGEN J. Efficient representations of video sequences and their applications[J]. Signal Processing: Image Communication, 1996, 8(4): 327—351.
- [7] Kenji Suzuki, Isao Honba, Noboru Sugie. Linear-time connected-component labeling based on sequential local operations[J]. Computer Vision and Image Understanding, 2003, 89(1): 1—23.
- [8] 徐正光, 鲍东来, 张利欣. 基于递归的二值图像连通域像素标记算法[J]. 计算机工程, 2006, 32(24): 186; 225.
- [9] REDDY B S, CHATTERJIB N. A FFT-based technique for translation, rotation, and scale invariant image registration[J]. IEEE Transactions on Image Processing, 1996, 5(8): 1266—1271.
- [10] BARTOLIA, DALALN, BOSE B. From video sequences to motion panoramas[M]. USA: IEEE Computer Society, 2002. 201—207.
- [11] 朱云芳, 叶秀清, 顾伟康. 视频序列的全景图拼接技术[J]. 中国图象图形学报, 2006, 11(8): 1150—1155.
- [12] 王钊旋, 李志林, 庞云阶. 一个二值图像连通成份标记的快速算法[J]. 工程图学学报, 1998, 3(13): 80—86.
- [13] 张修军, 郭霞, 金心宇. 带标记矫正的二值图像连通域像素标记算法[J]. 中国图象图形学报, 2003, 8(2): 198—201.
- [14] 宋斌. 一种新的图像连通域快速标号算法[J]. 电子测量技术, 2009, 32(9): 67—73.
- [15] R. M. Haralick, Some neighborhood operations, in: Real Time/ Parallel Computing Image Analysis[M]. Plenum Press, New York, 1981: 11—35.
- [16] 何妮. 当议 MATLAB6.1 的应用功能[J]. 安庆师范学院学报: 自然科学版, 2003, 9(2): 86—89.

Optimizing Algorithm of Labeling Connected Components in Binary Images

LUO Zhi-Zao, ZHOU Ying-wu, ZHENG Zhong-kai

(Department of electronic engineering, Minjiang University, Fuzhou 350108, China)

Abstract Based on analysis of existing algorithms of labeling connected components through scanning pixels in binary images, the improved algorithm is proposed. The algorithm has more advantages such as faster, simpler and easier to realize than the others, only need two-scans to label multi-objects which pixel belong to. The algorithm could label background as objects, labels and merges connected components via the two procedure of scanning binary image and scanning the matrix of provisional labels. It adopts sequential storage structure to store and resolve the equivalent labels, runs more rapidly, occupies much less memory than the unimproved algorithm.

Key words: binary images; connected components; scanning by pixels; labeling

(上接第 18 页)

例 2 设函数 $f(x) = e^x - 2$, $x \in [1, 2]$ 。显然 $f([1, 2]) = [e - 2, e^2 - 2] \supset [1, 2]$ 且 $f'(x) = e^x \geq e > 1$ 。根据推论 2 知 f 在 $[1, 2]$ 中有不动点。

参考文献:

- [1] 姚永红, 陈汝栋. Banach 空间中伪压缩映象不动点的迭代逼近[J]. 数学研究与评论, 2008, 28(1): 169—176.
- [2] 张海, 舒阿秀. Banach 不动点定理的注记及应用[J]. 安庆师范学院学报: 自然科学版, 2005, 11(4): 94—97.
- [3] 陈汝栋, 宋义生. Browder—Petryshyn 型的严格伪压缩映射的粘滞迭代逼近方法[J]. 系统科学与数学, 2006, 26(6): 651—657.
- [4] P. Vijayaraju, B. E. Rhoades, R. Mohanraj. A fixed point theorem for a pair of maps satisfying a general contractive condition of integral type[J]. Int. J. Math. Math. Sci., 2005(15): 2359—2444.
- [5] 张恭庆, 林源渠. 泛函分析讲义[M]. 北京: 北京大学出版社, 2005.
- [6] 程其襄, 张奠宙, 等. 实变函数与泛函分析基础[M]. 北京: 高等教育出版社, 2003.
- [7] Ravi P. Agarwal, Maria Meehan, DONal O'Regan. Fixed Point Theory and Applications[M]. Cambridge university press, 2001.

The Fixed-Point Theorems for Expansive and Non-contractive Mappings

WU Cui-lan¹, WANG Yun-jie²

(1. School of Mathematical Science, 2. Kewen Institute of Xuzhou Normal University, Xuzhou 221116, China)

Abstract In order to develop and improve the fixed-point theorems in metric spaces and extend their applications, some new fixed-point theorems are presented for expansive and non-contractive mappings by using Banach contraction principle in this paper.

Key words: expansive map; non-contractive mappings; Banach contraction principle; fixed-point theorem