

基于轮廓的孔洞填充算法

张德才，周春光，周强，池淑珍，王甦菁

(吉林大学 计算机科学与技术学院，长春 130012)

摘要:提出了一种基于轮廓的二值图像孔洞填充算法。该算法首先为区域添加外围轮廓标记，然后以区域内部一点作为种子进行区域生长来实现孔洞填充，最后再将外围轮廓的标记消除。实验表明，该算法具有很高的效率，并适用于任意类型的孔洞填充。

关键词: 区域生长； 孔洞填充； 外围轮廓； 运动目标检测

中图分类号:TP391 文献标识码: A

Hole Filling Algorithm Based on Contour

ZHANG De-cai, ZHOU Chun-guang, ZHOU Qiang, CHI Shu-zhen, WANG Su-jing

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract: A new hole filling algorithm based on contour was proposed in this paper. First of all, the circumscribed contour to the area is marked, and then a point in the area is selected as the seed to fill the hole by applying the region-growing method. At last, the algorithm eliminates all the marks of the circumscribed contour. Experimental results show that the algorithm has high efficiency and is adapt to all kinds of holes.

Key words: region growing; hole filling; circumscribed contour; moving object detection

运动目标检测是智能监控系统中的重要组成部分，也是运动行为分析的基础^[1]。因此，提高运动目标检测的准确性以及检测到的目标信息的完整性具有重要的意义。然而，在运动目标检测过程中，由于环境复杂或检测算法的缺陷，通过“二值化”得到的二值图像中往往存在孔洞。这些孔洞不仅影响了运动目标的检测效果，还破坏了运动目标信息的完整性^[2,3]。

作者简介: 张德才(1985~)，男，汉族，硕士研究生，从事模式识别的研究，E-mail:zhangdecai_qsd@163.com。
通讯作者: 周春光(1947~)，男，汉族，教授，博士生导师，从事计算智能和生物信息学的研究，E-mail:cgzhou@jlu.edu.cn。

基金项目: 国家自然科学基金(批准号: 60873146、60973092、60903097)、国家高技术研究发展计划(863)计划项目(课题编号: 2009AA02Z307)、教育部“符号计算与知识工程”重点实验室项目基金、吉林省生物识别新技术重点实验室(编号: 20082209)和吉林大学“211 工程”三期建设项目。

二值图像的孔洞消除算法正是针对这一类问题而提出的。目前的此类算法，往往存在时间、空间和适用性等方面的矛盾。如传统的算法^[4, 5, 6]首先要对整个边界区域进行区域生长，如果边界区域所占比例很大或被图像中的物体分割，那么边界的区域生长将耗费很多时间。文献^[7]提出了基于边界信息的孔洞填充算法，但是该算法仍然需要用区域生长扫描所有的边界像素；而且如果孔洞的边界和目标区域的边界重合时，将很难区分各个边界，因此该算法的不能适用于所有类型的孔洞。文献^[8]提出了一种基于目标外接矩形的孔洞填充算法，该算法在计算外接矩形时需要对目标区域内的坐标进行排序，而且外接矩形中可能包括其他的目标区域，因此该算法会出现重复处理的情况。

针对目标区域在整个二值图像中所占比例较小的一类图片，本文提出了基于轮廓的孔洞填充算法，直接对目标区域进行操作。该算法首先为区域添加外围轮廓标记，然后以区域内任意一点作为种子进行区域生长来实现孔洞填充，最后再将外围轮廓的标记消除。实验表明，该算法具有很高的效率，并适用于任意类型的孔洞填充。

1 基于轮廓的孔洞填充算法

1.1 算法描述

图 1 是该算法的一个应用实例，(a) 为含有孔洞的运动人体的二值图像。(b) 是算法为人体区域添加的外围轮廓标记示意图。(c) 以图(a) 中的任意一点作为种子点，以图(b) 为界进行区域生长来填充孔洞，并在最后消除轮廓标记，得到孔洞填充后的效果图。



图 1 孔洞填充过程

Fig. 1 Procedure of the hole filling

算法详细步骤如下：

- (1) 从上到下逐行扫描二值图像，搜索图像中第一个灰度值为 255 的像素点 s，该点也是第一个目标区域的起始点。
- (2) 将点 s 放在线性序列 G 中，并以 s 为种子利用区域生长的方法将当前连通区域内

所有点加入到 G 中。算法要求 s 不处于二值图像的边缘，如果 s 在图像边缘，则令 s 指向其 8 邻域内的非边缘点。

(3) 为当前连通区域添加外围轮廓。扫描 G 中的所有点，对扫描到的每一个点 p，如果 p 的 8 邻域内有灰度值为 0 的点 p_k ($k=1, 2, \dots, 8$)，将这些点标记为“2”。

(4) 对目标区域的外围轮廓进行标记。由 s 搜索其 8 邻域内标记值为 2 的一轮廓点 P_{con} 。

将 G 清空，并把 P_{con} 放入 G 中，然后以 P_{con} 为种子点并沿外围轮廓进行区域生长。

生长准则：对于每次从 G 中取出的当前点 p，扫描其 4 邻域内的点 p_i ($i=1, 2, 3, 4$)，设 p_i 的 8 邻域内各像素的灰度值为 n_1, n_2, \dots, n_8 ，用 $T(p_i)$ 表示 p_i 的标记值。若 p_i 是图像的边缘点，令 $T(p_i)=3$ ；否则，如果下面两个条件成立，则将 p_i 作为外围轮廓点放入 G 中，令 $T(p_i)=3$ 。

a) $T(p_i) \neq 0, T(p_i) \neq 3$ 。

b) $n_1 \& n_2 \& \dots \& n_8 = 0$ 或 $n_1 \& n_2 \& \dots \& n_8 = 1$ 且 $T(p_i) = 2$ 。

(5) 以 s 为种子点，外围轮廓为界进行区域生长，并消除孔洞。在生长过程中，对于取出的当前点 p，扫描其 8 邻域内的所有点 p_i ($i=1, 2, \dots, 8$)。若 p_i 满足 $T(p_i) \neq 3$ 且 $T(p_i) \neq 4$ ，将 p 放在 G 中，并令 $T(p)=4$ ；同时将灰度值为 0 的点的灰度值改为 255。

(6) 将所有的轮廓标记“3”修改为“0”，然后将所有的非“0”标记修改为“1”。

(7) 搜索图像中下一个目标区域上的一点 s，如果找到，转(2)；否则，算法结束。

1.2 标记的状态变化过程

图2给出了该算法处理某个目标的局部区域时标记变化情况。(a) 表示各标记初始状态。(b) 给区域的边界添加轮廓之后各标记的状态。添加的轮廓用“2”标记。(c) 搜索外围轮廓后各个标记的状态。上一步骤中的操作，保证了即使原始二值图像的外围轮廓和孔洞边界重合时，本步骤在寻找外围轮廓时也不会错误的搜索到孔洞的边界，这样算法总是能够精确的找到目标区域的外围轮廓。外围轮廓用“3”标记。(d) 以图中下划线处的像素点作为种子点，以外围轮廓为界进行区域生长来消除孔洞后各标记的状态，扫描过的像素点标记为“4”。(e) 将所有的轮廓标记“3”修改为“0”，然后将所有的非“0”标记修改为“1”。

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0

(a) 初始状态

2	2	2	2	2	2	2	2	0	0
2	1	1	1	1	1	1	2	2	0
2	1	2	2	2	2	2	1	2	0
2	1	2	0	0	0	2	1	2	2
2	1	2	2	0	0	2	2	1	2
2	2	1	2	2	2	2	2	1	2
0	2	2	1	1	1	1	1	1	2
0	0	2	2	2	2	2	2	2	2

(b) 添加轮廓

3	3	3	3	3	3	3	3	0	0
3	1	1	1	1	1	1	3	3	0
3	1	2	2	2	2	1	3	0	0
3	1	2	0	0	0	2	1	3	3
3	1	2	2	0	0	2	2	1	3
3	3	1	2	2	2	2	1	3	3
0	3	3	1	1	1	1	1	1	3
0	0	3	3	3	3	3	3	3	3

(c) 查找外围轮廓

3	3	3	3	3	3	3	3	0	0
3	4	4	4	4	4	4	3	3	0
3	4	4	4	4	4	4	4	3	0
3	4	4	4	4	4	4	4	3	3
3	4	4	4	4	4	4	4	4	3
3	3	4	4	4	4	4	4	4	3
0	3	3	4	4	4	4	4	4	3
0	0	3	3	3	3	3	3	3	3

(d) 孔洞填充

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0
0	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0

(e) 修正标记

图 2 标记的变化

Fig. 2 Variations of the marks

2 实验与分析

2.1 实验过程

实验中的样本全部来自于运动目标检测中产生的二值图像，如图3中(a)~(d)。每类样本均取2万张。本文提出的算法和传统的孔洞填充算法已经用c++实现，基于Windows XP操作系统，硬件环境为Intel Pentium D CPU，2.80GHz，1GB RAM。(e)~(h)是样本的处理结果。

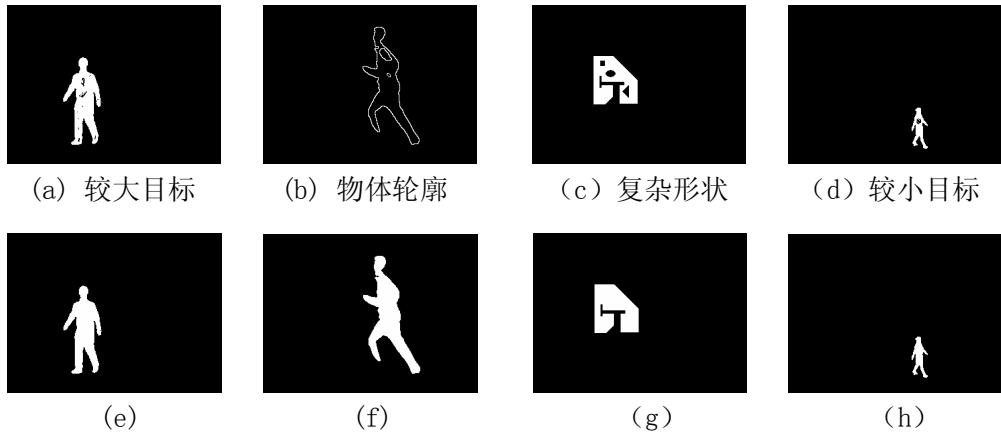


图 3 孔洞填充示例

Fig. 3 Example of the hole filling

实验中针对每一类样本分别用传统的孔洞填充算法和本文提出的算法进行处理，计算出每一张图像的平均处理时间，如表1所示：

表1 两类算法的运行效率

Table 1 Efficiency of the two algorithms (ms)

样本类型	传统算法平均处理时间	本文算法的平均处理时间
较大目标	32	30
目标轮廓	23	19
复杂形状	21	19
较小目标	22	16

2.2 结论

实验证明,本文提出的基于轮廓的孔洞填充算法比传统的孔洞填充算法具有更高填充效率;并且随着外围轮廓复杂度和目标区域面积的减小,本文提出的算法的效率也会相应提高。将该算法应用在运动目标检测系统中,能够显著提高系统的运行效率。进一步的研究工作是如何使得算法能够根据运动目标的信息对图像中的孔洞进行有选择的填充。

参考文献

- [1] Isaac Cohen, Gerard Medioni. Detecting and Tracking Moving Objects for Video Surveillance [J]. IEEE Proc. Computer Vision and Pattern Recognition, 1999:1–6.
- [2] Li Zengh, Lida Xu. Moving Multi-Object Tracking Algorithm Based on Wavelet Clustering and Frame Difference [J]. IEEE International Conference, 2009:3646–3647.
- [3] Wang Junqing, Shi Zelin, and Huang Shabai. Detection of Moving Targets in Video Sequences [J]. Opto-Electronic Engineering, 2005:5–8.
- [4] 冈萨雷斯等. 数字图像处理 [M]. 电子工业出版社, 第二版, 2007:496–500.
- [5] 章毓晋. 图像处理和分析基础 [M]. 高等教育出版社, 2002:196–198.
- [6] 姚敏. 数字图像处理 [M]. 北京: 机械工业出版社, 2008:250–254.
- [7] LI Wen-bin, WANG Chang-song. Hole Filling Algorithm Based on Contours Information [J]. Computer Engineering and Design, 2008, 29 (15): 58–3962. (李文斌, 王长松. 基于边界信息的孔洞填充算法 [J]. 计算机工程与设计, 2008, 29(15): 58–3962.)
- [8] ZHANG Yan, ZENG Li-bo, WU Qiong-shui, et al. A Rapid Hole Filling Algorithm for Arbitrary Shape [J]. Application Research Of Computers, 2004, 12:155–156. (张燕, 曾立波, 吴琼水, 等. 一种适用于任意形状区域的快速孔洞填充算法 [J]. 计算机应用研究, 2004, 12:155–156.)