

文章编号: 0258-7025(2004)02-0185-05

目标检测的区域编码和区域逻辑运算

聂守平¹, 刘峰², 王弘³

¹ 南京师范大学物理科学与技术学院光电技术江苏省重点实验室, 江苏 南京 210097;

² 南京邮电学院图像处理与图像通信江苏省重点实验室, 江苏 南京 210003, ³ 北京大学计算机科学与技术系, 北京 100871)

摘要 研究了图像逐行扫描、逐行编码的方法, 提出了存储区域信息的新的数据结构, 在传统的图像逻辑运算的基础上, 研究了区域逻辑运算, 并将区域编码和区域逻辑运算应用于目标检测, 并给出了实验结果。

关键词 图像处理; 逻辑运算; 区域逻辑运算; 目标检测

中图分类号 TN 911.73 文献标识码 A

Region Encode and Region Logical Algorithm for Object Detection

NIE Shou-ping¹, LIU Feng², WANG Hong³

¹ Department of Physics, Nanjing Normal University, Nanjing, Jiangsu 210097, China

² Image Processing & Image Communication Lab, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China

³ Department of Computer of Science and Technology, Peking University, Beijing 100871, China

Abstract The method of image scanning and encoding for each row is studied, and the new data structure for storing the image information is proposed. Compared with the conventional image logical algorithm, the region logical algorithm is studied. The region encoding and region logical algorithm have been used for object detection, and the results are given.

Key words image processing; logical algorithm; the region logical algorithm; object detection

1 引言

目标检测^[1]在机器人视觉、工业自动化、产品分类等领域的应用十分广泛,其目的是从多个目标中分离出所需要的目标,并完成对目标的形状、数目等特征的分析。由于目标与背景之间的复杂程度不同,目标的形状、数目、相互关系不同,所采取的检测方法也不同。通常可以遵循的处理思路是,通过单阈值、多阈值或自适应阈值^[2]将目标与背景分离,分离的结果只是从灰度的大小上将目标提取出来,这些目标中有些是真正需要提取的目标,也有的是由于阈值选取不当产生的假目标,在此基础上,可以采用边界跟踪、模板匹配等方法来实现从众多目标中将感兴趣的目標分离出来。

边界跟踪^[3]对于目标和背景有明显差异的情况是十分有效的,它是从某一点出发按一定的跟踪方向,搜索以边界起始点为中心的一个邻域,依次迭代直至扫描完整幅图像。若一幅图像中无噪声存在,这个算法将得到最大梯度边界,但是少量的噪声有可能使边界跟踪失败,并偏离到图像的外面。另外从得到的边界点坐标重构原来的图像则要对边界轮廓内部重新填充。

在已知目标形状的前提下,可以利用固定或可调的目标模板^[4]和图像中不同区域依次进行相关运算,根据相关值的大小来判断所检测的区域是否为目标。当要同时检测几种不同形状和大小的目标时,此方法变得相当繁琐。

在对目标检测时,要求每个目标都应该有一个

收稿日期 2002-09-24;收到修改稿日期 2002-11-11

基金项目 江苏省自然科学基金(BK2003100)和江苏省教育厅基金(02KJB460007)资助项目。

作者简介 聂守平(1967—),男,南京师范大学物理科学与技术学院副教授,博士,主要从事光信息处理和图像处理研究。

E-mail nieshuoping@jnet.njnu.edu.cn

序号,同时要确定哪些像素属于这个目标,这对于目标的测量是至关重要的。这一要求可以通过建立目标链码来完成。用链码来存储目标时,通常是存储目标的边界起始点以及每个边界点的坐标,这样虽然比存储整个目标所需要的空间少了许多,但只存储边界对于重构目标以及对目标中心、面积等特征的计算带来了不便。

为了既对目标进行分割,又有效地存储目标的像素特征,本文对通常的边界链码结构进行了改进,提出逐行扫描、逐行存储的区域链码结构,同时为了检测和背景具有相同灰度值的目标,提出了目标区域逻辑运算的概念,并给出了目标检测的结果。

2 目标区域链码结构

这里所采用的编码形式不是对目标进行边界跟踪,而是对连通区域进行编码,每一个区域表示一个目标,用唯一的一个序号来表示。按照从上到下、从左到右的顺序对图像进行扫描,扫描的过程中将图像信息进行编码,建立一个链表。链表的每一个结点数据结构及其含义如表1所示。区域编码过程为:

在每一行中,对连续的非零像素按照表1所示的数据结构依次存储,即记录目标序号、所在行号、连续非零像素段的起始列和终止列、目标灰度。若处理的是二值图像,目标的灰度值都相同,结点的最后一个元素可以忽略。

表1 结点数据结构

Table 1 Node data structure

Symbols	Meanings
N	Number of object
Row	Number of row
Start	Number of first column
End	Number of last column
Gray	Gray of object

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

(a)

N	1	1	1	2	2
Row	1	2	3	4	5
Start	2	3	4	2	1
End	5	4	4	2	5
Gray	1	1	1	1	1

(b)

图1 二值图像及区域编码结果

Fig. 1 Binary image and result of region encode

则,判断当前行的目标像素与上一行的目标像素之间是否处于连通状态,若是连通的,则当前像素和其邻域像素属于同一个目标,那么目标序号保持不变。

逐行扫描的编码形式可以用图1来说明。图1(a)是一幅二值图像灰度分布,用“1”表示目标;“0”表示背景。图1(b)则是采用上述方法所建立的区域编码结果。

从图1(b)可以看出,对图像进行一次扫描后,所有的目标像素都被存储在按照表1所定义的数据结构中。它说明整幅图像包含两个目标,第一个目标包含第1至第3行的连通区域,第二个目标包含第4至第5行的连通区域,同时每一个区域所在列的起至范围也得到了。根据这些值可以直观地得出目标的数目、目标所占区域,同时可以迅速地重构目标,避免了根据边界点坐标来填充目标内部的过程。这也就是区域编码和边界编码的区别。

3 目标区域逻辑运算

图像之间的逻辑运算包括与、或、非等运算,就逻辑与运算而言,其可以用来确定图像的公共区域,特点是对参与逻辑运算的两幅图像中对应像素进行判断。这里提出图像目标区域之间的逻辑运算,参与运算的不是对应的像素,而是两个目标区域。下面以二值图像为例来比较这两种运算的特点。

设 A 和 B 分别表示两幅图像,像素间的逻辑与运算表示为

$$C(i, j) = \begin{cases} A(i, j) & A(i, j) = B(i, j) = 1 \\ 0 & \text{其他} \end{cases} \quad (1)$$

其中 C 为经过逻辑与运算得到的图像。这种运算的特点是逐个像素进行比较,若两个对应像素灰度都为“1”,则运算的结果中保留此像素,否则像素置为“0”。

对于目标区域逻辑与运算,首先要经过图像逐行扫描得到类似于图1(b)所示的各个目标区域。在此基础上假设图像 A 中第 m 个目标区域中的一个像素为 $A(i, j)$,如果图像 B 中对应坐标像素 $B(i, j)$ 也属于图像 B 中第 n 个目标区域,则区域逻辑与运算的结果中直接保留图像 A 中第 m 个区域的所有像素,用式子表示为

$$C_m = \begin{cases} A_m & A(i, j) \in A_m \text{ 和 } B(i, j) \in B_n \\ 0 & \text{其他} \end{cases} \quad (2)$$

即只要图像 A 中第 m 个区域中存在至少一个这样的像素,它在图像 B 中对应的像素也属于图像 B 中某

对于行与行之间,按照四邻域或八邻域的连通规

一个目标区域,则图像 A 中第 m 个区域在运算结果中完全保留。

和区域逻辑与运算相似的还有区域逻辑非与运算,它表示若图像 A 中第 m 个区域中的任何一个像素,它在图像 B 中对应的像素都不属于任何一个目标区域,这时逻辑非与运算的结果中保留图像 A 中第 m 个区域,即

$$C_m = \begin{cases} A_m & A(i, j) \in A_m \text{ 和 } B(i, j) \notin B_n \\ 0 & \text{其他} \end{cases} \quad (3)$$

这两种情况虽然都是保留图像 A 中第 m 个区域,但含义不同,要根据检测的目标特性来选取。

如图 2 所示 (a)和 (b)是两幅二值图像 (c), (d)和 (e)为与运算、区域与运算、区域非与运算的结果。需要说明的是,像素之间的与运算是对称的,即 A 和 B 相与,或 B 和 A 相与结果相同,但对于区域运算则不具有对称性。如图 2(f)和 (g)为图像 (b)和 (a)进行区域与运算、区域非与运算的结果。

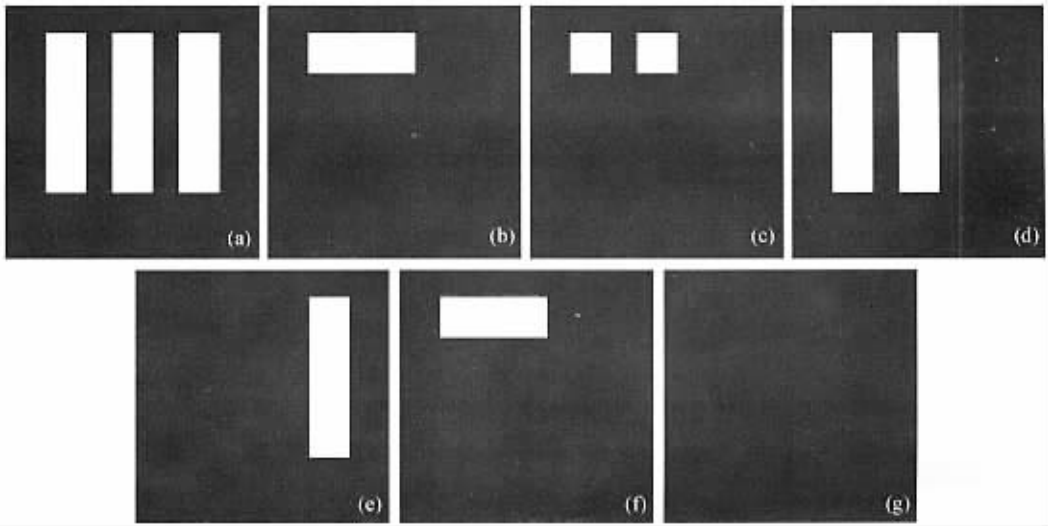


图 2 区域逻辑运算

Fig. 2 Region logical algorithm

4 目标区域检测实验

依据前面提出的目标区域链码结构和区域间逻辑运算来进行目标检测、目标二值化处理。

4.1 目标检测

图 3 所示为待检测的目标图像,其特点是目标孔和背景具有相同的灰度值,同时有两个未闭合的开孔和背景相通,这两个未闭合的开孔不属于通常意义上的孔径,由于和背景相通只能作为背景处理。现在的目的是要将 4 个圆孔和 1 个矩形孔提取出来。处理步骤如下:

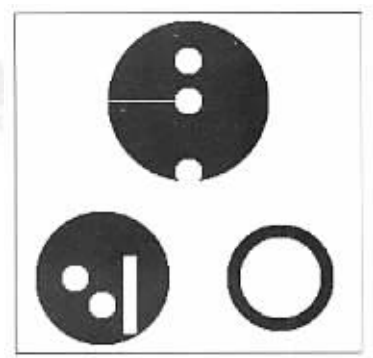


图 3 原始目标图像

Fig. 3 Original object image

N	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6	6
Row	0	...	255	26	...	43	166	...	225	179	...	238	186	...	206	206	...	225
Start	0	...	0	123	...	124	188	...	189	79	...	79	39	...	40	60	...	60
End	255	...	255	133	...	132	202	...	203	89	...	48	48	...	48	67	...	68
Gray	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

图 4 原始图像区域链码

Fig. 4 Region encode for original image

1) 对待测图像进行逐行扫描,所建立目标区域的链码结构如图 4 所示。从中可以得出整个图像被分成 6 个区域,第 1 个区域由背景、两个与背景相连通的开孔组成,其余 5 个区域分别表示圆孔和矩孔。

2) 建立一幅和被测图像同样大小的边界区域图像,如图 5 所示,其只有四周边界像素为“1”,其余像素为“0”(由于 1 个像素的边界显示时不容易分辨,故图 5 中将白色的边界放大了,图 3 和图 5 的黑色外框是为了表示图像的范围大小)。按相同的方法对图 5 进行扫描,建立目标区域的链码结构如图 6 所示。由于只有四周边界像素为“1”且是连通的,因此目标区域个数为 1。

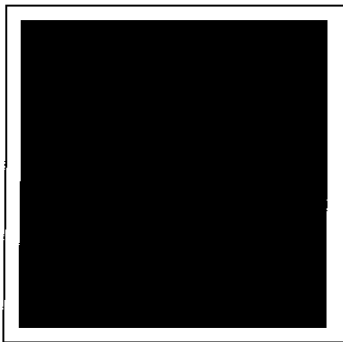


图 5 边界区域图像
Fig. 5 Boundary region image

3) 利用区域逻辑非与运算,对图 4 中 6 个目标区域依次进行判断,目标区域 1 包含了整个背景,当然也包含了整个边界,即目标区域 1 中存在边界像素,其对应像素包含在图 6 所示的区域中,则由式(3)可知运算结果中目标区域 1 被置为“0”。相反,目标区域 2 至 6 中不存在任何一个像素,其对应像素包含在图 6 所示的区域中,则运算结果中目标区域 2 至 6 得到保留。整个检测结果如图 7 所示。

N	1	1	1	1
Row	0	1	...	255
Start	0	0	...	0
End	255	0	...	255
Gray	1	1	1	1

图 6 边界图像区域编码

Fig. 6 Region encode for boundary image

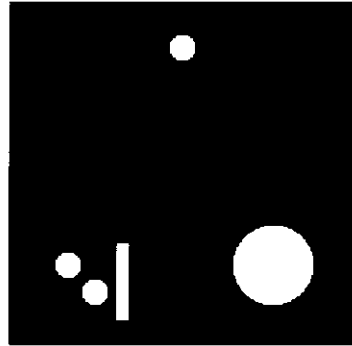


图 7 目标检测结果

Fig. 7 Result of object detection

4.2 目标二值化

图 8(a)所示为待处理的图像,对于多灰度图像很难选取一个恰当的阈值实现二值化,这里构造两幅图像采用区域逻辑运算来进行。先选取一个较小的阈值和较大的阈值分别对图像进行二值化,结果如图 8(b)和(c)所示。在图 8(b)中,由于阈值选取得较小,二值化的结果中引入了不属于目标的像素,而在图 8(c)中,由于阈值选取得较大,一些原来属于目标的像素也变成了背景。由于图 8(c)是由较大阈值进行二值化得到的,因此图中的每一个像素都属于目标本身,这时如果能够判断出图 8(b)的区域哪些是和图 8(c)中的目标区域连通的,并将这些区域保留,其他的区域置为背景,则可以实现恰当的图像二值化。这也就是应用区域逻辑与运算的原理。

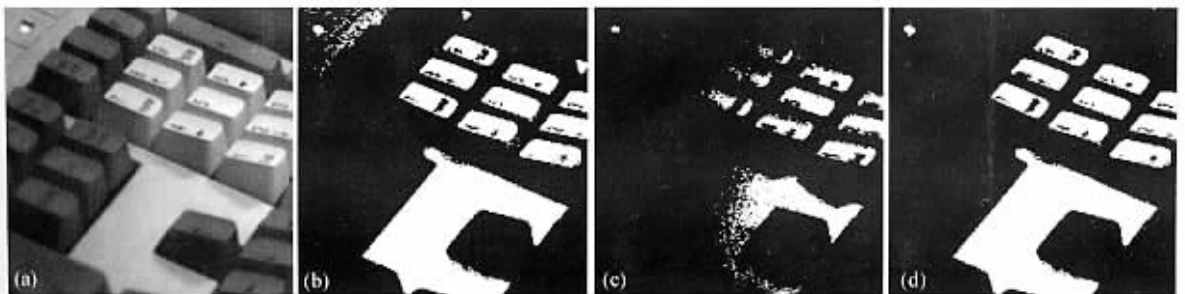


图 8 区域逻辑运算二值化图像

Fig. 8 Binary image with region logical algorithm

对图 8(b)和(c)分别进行逐行扫描,建立区域链码,利用区域逻辑与运算的方法得到的结果如图 8(d)所示。从结果中不难发现,图 8(b)左上角的众多小区域中的像素,因为不属于图 8(c)中任意一个目标区域,因此被置为了背景。

5 结 论

与边界跟踪建立边界链码不同,提出了逐行扫描、逐行编码的目标区域数据存储方式,既保存了目标区域信息,又有利于特征提取和目标重构;在此基础上提出了区域逻辑运算的概念,利用区域逻辑运算实现了目标的区域分割。实验结果表明,所采用的方法能够有效地用于目标检测和目标二值化。需要说明的是区域逻辑运算要求参与运算的是两幅图

像,即除了待检测图像之外,还要构造另外一幅图像,这是在实际应用中需要加以考虑的。

参 考 文 献

- 1 Ma Zhaomian, Tao Chunkan. Regional fractal and artificial object detection [J]. *Acta Physica Sinica*, 1999, **48**(12) 2202 ~ 2206
马兆勉,陶纯堪. 区域分形与人工目标检测[J]. 物理学报, 1999, **48**(12) 2202 ~ 2206
- 2 K. V. Mardia, T. J. Hainsworth. A spatial thresholding method for image segmentation [J]. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1988, **10**(6) 919 ~ 927
- 3 T. Pavlidis, Y.-T. Liow. Integrating region growing and edge detection [J]. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1990, **12**(3) 225 ~ 233
- 4 Marie-Pierre Dubuisson Jolly, Sridhar Lakshmanan, Anil K. Jain. Vehicle segmentation and classification using deformable templates [J]. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1996, **18**(3) 293 ~ 308

田 万方数据
WANFANG DATA