

带标记矫正的二值图象连通域像素标记算法

张修军¹ 郭霞² 金心宇¹

¹(浙江大学信息与电子工程系, 杭州 310027) ²(中国科学院大气物理研究所中层室, 北京 100029)

摘要 分析了连通域像素标记算法以及游程连通性算法的不足, 提出了一种带标记矫正的二值图象连通域像素标记算法, 该算法首先采用标记矫正来减少图象扫描次数, 然后再对标记采用 RLE 游程编码来提高合并效率, 这种改进算法对大多数形状目标可以在一次扫描中完成像素的标记, 从而使像素标记算法得到优化, 此外, 还指出了改进算法在处理向上分叉图形时所遇到的问题, 同时给出了解决方法, 实际运行效果较好, 有一定的使用价值。

关键词 计算机图象处理(520·6040) 连通域 像素标记 二值化图 标记合并 标记矫正

中图分类号: TP391.41 TP242.62 **文献标识码**: A **文章编号**: 1006-8961(2003)02-0198-05

The Pixel Labeled Algorithm with Label Rectified of Connecting Area in Binary Pictures

ZHANG Xiu-jun¹, GUO Xia², JIN Xin-yu¹

¹(Department of Information Science and Electronical Engineering, Zhejiang University, Hangzhou 310027)

²(Institute of Atmospheric Physics, Chinese Academic Sciences, Beijing 100029)

Abstract To distinguish different object areas, run length connectivity algorithm as well as pixel labeled method of connecting area are usually adopted in binary pictures in the image identification. The present paper points out some limitations of the pixel labeled algorithm and presents the corresponding improved one. It adopted the pixel labeled algorithm to reduce the numbers of scanned picture and adopt RLE coding method to improve efficiency of the area's combination. This improved algorithm, which greatly optimizes the original one can be applied to most kinds of shapes of objects and label all pixels with one scanning. Furthermore, this paper indicates that some problems will come out when the improved method is applied to handling images with up-fork area. The author proposes solutions to these problems. Finally it shows that the improved algorithm can be used to sum up-fork points in the up-fork pictures.

Keywords Computer image processing, Connected area, Pixel label, Binary picture, Label merge, Label rectified

0 引言

关于机器人视觉方面算法的研究, 目的就是从图象采集卡获取的图象中提取出各个机器人小车和球的位置以及运动方向的信息, 首先对图象进行二值化处理, 并通过对这个二值化图的目标区域进行像素标记来区分不同的小车或球, 进而统计出各个物体的坐标位置, 也就是每个目标物体区域的重心, 对二值化图象(0表示背景, 255表示前景物体区域)进行连通域标记就是将二值化图象中的各目标

区域用不同的数值表示(一个目标区域中的所有像素点都同一个数值)。通常使用的连通域标记算法主要有像素标记算法和游程连通性算法^[1]。本文主要对像素标记算法进行了研究分析, 指出了该算法的一些不足, 并给出了改进的像素标记算法, 从而大大简化了程序算法设计。对大多数不同形状的图象而言, 利用该算法得到的计算结果其等价标记非常少, 基本上只需一次扫描即可完成连通域标记, 因而而不一定需要再对标记进行扫描, 而运用传统的像素标记算法则必须要对标记再进行扫描, 而且标记合并算法比较复杂。同时由于算法的改进, 在必须进行等

价标记合并的时候,也能非常方便地管理这些等价标记,并且处理速度非常快.

实际证明,将改进的像素标记算法应用在机器人视觉程序中,结果比未经改进的算法有较大的提高,既简化了算法的程序设计,又提高了运算速度.

1 像素标记算法

1.1 简介

一幅图象由不同灰度的点组成.为了对图象的连通域进行标记,需要对一幅图象作从左到右,从上到下的水平扫描.要检测当前被扫描到的点是不是和周围的点连通,只能检查当前的像素和以前标记过的邻近像素的值是否一样.如果当前像素的值和邻近像素的值一样,就表示它们连通,反之,就表示和此邻近像素不连通,此时当前点就要给一个新的标记,同时标记保留在一个与原二值图等像素点个数的二维数组中.

这里,当前像素点用 $c5$ 表示, $c1, c2, c3, c4$ 表示 $c5$ 邻近已标记过的点,其中, $c1$ 表示其左上点, $c2$ 表示其上面的点, $c3$ 表示其右上点, $c4$ 表示其左边的点,同时 $label(c5)$ 表示 $c5$ 这个点的标记,对 $label(c1), label(c2), label(c3)$ 和 $label(c4)$ 也做同样的说明,由于采用从左到右,从上到下的水平扫描方式,所以当前点的这几个邻近点都已经被标记过.

邻近像素点的关系有 4-连通和 8-连通两种情况.如果考虑 4-连通,那么扫描过的邻近像素只有上($c2$)和左($c4$)两个点.如果当前点 $c5$ 的像素值为 255,则 $c5$ 点需要标记.若这时邻近点 $c2, c4$ 的像素值都为 0,则 $c5$ 点需要用一个新的标记,实际上就是将记录标记的变量 $label$ 值加 1,这就是说,新的标记值比原来的标记值大;若 $c2, c4$ 的像素值其中有一个不为 0,则用这个不为 0 的点对应标记值给 $c5$ 点赋值;若 $c2, c4$ 的像素值都不为 0,则用 $c2$ 和 $c4$ 的标记中任一标记值给 $c5$ 点赋值即可,同时记录这两个标记等价.在这里,等价就是指两个区域是连通的.当扫描完成后,会产生很多等价的标记,所以还要将由标记组成的二维数组再扫描一遍,以便将这些等价标记用一个标记来代替,这样才能完成区域的合并.同样的算法可推广到 8-连通情况,只不过 8 连通考虑的邻近像素为左上($c1$)、上($c2$)、右上($c3$)和左($c4$),所以考虑等价标记时,要考虑到这 4 个邻近的点.

1.2 算法存在的缺点

在实际编程中,考虑 8-连通域,一个简单的区域会出现图 1(a)这种情况,图中不同的纹线密度表示不同的标记.按标记算法,当前扫描到 $c5$ 这个点时,只要 $c5$ 这条线比上面的线向左多伸出两个点或两个点以上(多一个点没有这个问题),那么与 $c5$ 这个像素点对应的邻近像素点 $c1, c2, c3, c4$ 的值就全部为 0,于是需要添加一个新的标记.扫描结束后,在一个区域中可能会有好几根扫描线会出现这种情况,结果将导致一个简单的目标域却被分成几个连通区,见图 1(b),因此必须进行等价标记合并处理.此外,如果图象中的目标区域比较多,这种等价标记的数目将很大.在标记合并算法中,对大量等价的动态的标记所做的存储、管理和计算都是比较困难的.如果同时标记的个数大于 255 个,那么为了记录这些标记及标记的位置,就必须申请较大的空间,因为这时每个点的标记都要用一个双字节来表示.



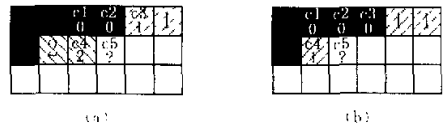
(a) 当前点与标记过的邻近点 (b) 不同连通区

图 1 像素点标记

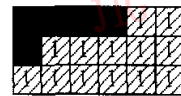
为了减少这些等价标记的个数,就要想办法解决图 1 这种情况,使同一区域中一条扫描线和上面一条扫描线的标记尽量保持一致,也就是用上面一条线的标记来矫正当前的标记.实例证明,应用这种方法就能有效减少等价标记的数量.

2 改进的像素标记算法

考虑图 2 的情况,图中不同的纹线密度表示不同的标记,白色的小方框表示未标记的像素.依据原来未改进的算法,图 2(a)中第 1 条扫描线像素标记



(a) (b)



(c)

图 2

为 1,第 2 条扫描线起点像素标记为 2.当扫描到 c_5 点的时候,由于 c_1, c_2 的值为 0,所以只考虑 c_3, c_4 2 个邻近点. 因为 c_3 比 c_4 先扫描到, c_3 对应的标记为 1,而 c_4 对应的标记为 2,所以 c_3 的标记值比 c_4 的要小. 为了使 c_3 的标记能矫正 c_4 这一条扫描线的标记,就要把当前的标记更新为 c_3 的标记,同时将 c_5 这条扫描线的起点的标记重新赋值为当前的标记,并且重新从 c_5 这条线的第 2 个点开始扫描,见图 2(b). 于是图 2(b)中 c_5 的标记就可以用 c_4 的标记,也就是 c_5 的标记也给定为 1. 这样就完成了标记的矫正,结果目标区域都标记为 1,见图 2(c).

从算法上考虑,在图 2 这种情况下,只重新扫描了一条线的前面几个点,就能去掉一个标记,如图 2 中重新扫描了第 2 条线的前 3 点. 如果图中没有向上分叉的区域,就可以在—次扫描中完成整个标记过程,而且标记的数量是最少的,也不存在等价标记的问题. 如果图象中的目标区域个数少于 255 个时,还可以用 1 个字节来表示一个点的标记,也就能得到一个不同灰度等级的连通域图.

当然还要考虑到图象的一些细节. 比如要是图象分叉了怎么办? 由于采用从左到右,从上到下的水平扫描方式,因此对那些向左、向右以及向下分叉的图形,处理效果同没有分叉的效果一样. 对这些图形区域进行处理的时候,当前扫描到的像素的标记肯定是和上—条线的像素标记—样,因为都能被矫正,所以这样的区域都只会—个标记值,见图 3(a)、(b)、(c). 在图 3(a)中,当扫描到第 3 条线的时候就做了一次标记矫正.

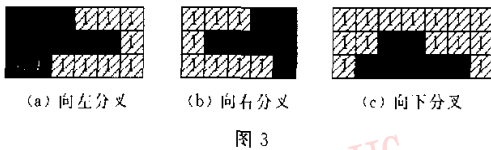


图 3

但是对于那些向上分叉的图象会出现什么样的情况呢? 见图 4(a)和图 4(b),当扫描到第 3 条线的时候,图 4(a)开始的标记是 2,而图 4(b)开始的标记是 1. 因为每次在做标记的时候都要比较当前像

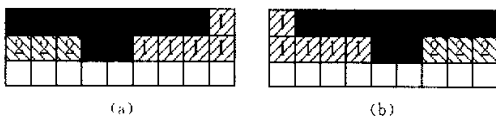


图 4

素点的 c_3 和 c_4 2 个邻近像素的标记,所以按改进算法,就要求在 $label(c_3) < label(c_4)$ 的时候,用 $label(c_3)$ 去矫正当前这一条扫描线的标记,同时从这条扫描线的第 2 点开始扫描. 以下将对这两种情况分别讨论.

先考虑左边标记比右边标记大的情况,见图 5(a),这时 c_5 这个点对应的 c_3 和 c_4 满足 $label(c_3) < label(c_4)$ 关系式. 也就是需要矫正的条件,所以对第 3 条线要重新扫描,并且第 3 条线的第 1 个点的标记被矫正为 1(见图 6).

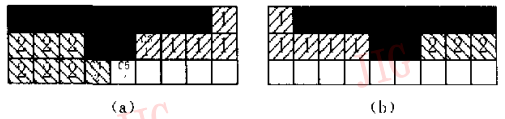


图 5



图 6

图 6 中,指针指向 c_5 这个点,仍然要判断 $label(c_3)$ 和 $label(c_4)$. 因为这时候 $label(c_3) > label(c_4)$,所以不需要标记矫正,而 c_5 这个点要—个像素标记,且只能给标记 1 或标记 2,但给标记 2 肯定不行,因为这样第 3 条线的第 2、3 和 4 这 3 个点都会标记成 2. 等扫描到第 5 个点时候就会再次出现图 5(a)的情况,又要求矫正,显然,就会出现死循环,所以只能给图 6 中的 c_5 赋标记 1. 在算法上,为了能用循环语句处理,在 c_1, c_2, c_3, c_4 中,取最小的标记给 c_5 就可以了,那么这种情况就用 c_4 的标记直接赋给 c_5 . 处理的结果就是第 3 条线的标记全为 1. 见图 7(a).

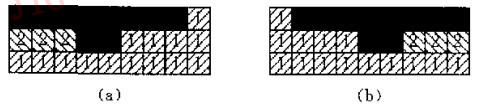


图 7

再考虑左边标记比右边标记小的情况,见图 5(b). 在对第 3 条线扫描,扫描到 c_5 这个点时,比较 $label(c_3)$ 和 $label(c_4)$. 因为 $label(c_3) > label(c_4)$,所以不需要矫正, c_5 的标记在 c_1, c_2, c_3, c_4 中取最小的标记即可. 这样处理的结果就是第 3 条线的标记全为 1,也就是说,图 5(b)这种情况是不需要重新扫描的,也就是不需要矫正,结果见图 7(b).

总结上面两种向上分叉的图形情况如下:在 $label(c3) < label(c4)$ 时,需要进行标记矫正,也就是用 $c3$ 的标记去矫正 $c4$ 这条线的标记;如果 $label(c3) > label(c4)$ 时,不需要进行标记矫正,而 $c5$ 的标记取 $c1, c2, c3, c4$ 中最小的标记.当然,考虑到当前点对应的 4 个邻近点有可能在目标区域之外,这时它们的标记值可能会是 0,则不考虑.因此 $label(c3)$ 和 $label(c4)$ 关系式比较的前提要求是两个标记都不为 0,同时 $c5$ 的标记取 $c1, c2, c3, c4$ 中不为 0 的最小标记值,即

```

IF label(c3)和label(c4)都不为 0,同时
label(c3) < label(c4)
THEN 进行标记矫正
ELSE c5 的标记取 c1, c2, c3, c4 中不为 0 的最小标记值
(label(c3)和label(c4)相等的情况已经包括在 ELSE 语句
中了)

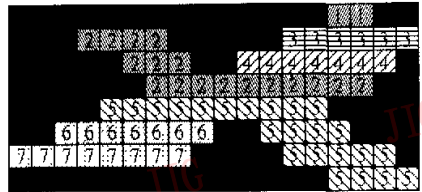
```

采用标记矫正后对图 4 的处理结果如图 7(b) 所示.从结果来看,对这种向上分叉的图形仍然不能在一次扫描中完全合并,所以还要考虑标记合并.

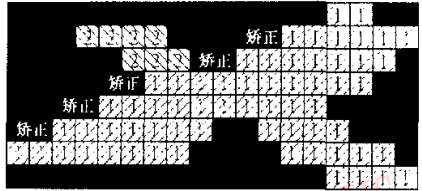
3 标记合并算法

对向上分叉的图形,如果按这种改进的标记算法处理的话,仍然会出现两个等价标记(从图 7 可以看出),每出现一个向上的分叉就有一对等价的标记,但总体来讲,等价标记大大减少了,进一步的解决方法有两个:一是搜索标记大的分支,一直搜索到这个分支的最高点左端为止,用小的标记给起点赋值,然后对这个分支重新扫描;二是按原来的方法,将标记的二维数组再扫描一遍,去掉等价的标记.对第 1 种方法,如果有一个向上的多级分叉的形状,比如一棵树的形状,那么就会出现多次扫描的过程,从而在速度上没有优势(当然是这棵树倒着长的话,那么就能节约很多计算量,同时省掉大量的存储等价标记的空间和烦琐的算法).如果采用第 2 种方法,相对于传统的方法,改进算法只能说将等价标记减少了,而有多少个向上的分叉就有多少个等价标记对,其优势在于对等价标记的管理和计算方便了.图 8(a) 示出了未采用矫正算法的标记结果,图上出现了 7 个等价的标记.图 8(b) 为采用了矫正算法的标记结果,只有两个等价标记,同时等价标记对的个数为 1,表示只有一个向上的分叉,且等价标记的坐标即为分叉的位置.

为了后续处理的快速和方便,采用第 2 种方法,



(a) 未采用矫正算法



(b) 采用了矫正算法

图 8

但不需将标记数组再扫描一遍,也不为这些标记申请与像素点一样多个数的标记空间来存储每个像素的标记,而是在做像素标记的同时,对标记采用 RLE 游程编码^[2].由于毗邻的像素点变化不太快,所以采用这种方法可以节省很大的存储空间,同时在以后的计算中,可以几个点一块计算,从而进一步提高了运算速度.对于采用传统的像素标记来说,由于等价标记太多,而且等价关系会出现一个大的标记和几个小的标记等价的情况,同时也会出现几个大的标记和一个小的标记等价的情况,这样的等价关系复杂,所以在进行合并时,要采用较繁杂的算法,比如基于树的合并查找(tree-based union find)算法.由于采用了带标记矫正的标记算法,因此等价标记就比较少,一般是一个大的标记等价一个小的标记,最复杂的情况也只是几个大的标记和一个小的标记等价(这是改进算法所保证的,见图 9 和图 10),所以等价关系简单得多,实现起来也很方便.如果等价标记个数为 0 的话,还可以不做标记合并这一步,那么游程编码的结果就是标记的结果.

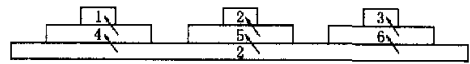


图 9 原标记等价算法的标记等价关系



图 10 改进标记算法的标记等价关系

在具体组织这些 RLE 编码时,考虑到这时等价标记不会很多,因此可采用链表的算法结构,用每一个链

表存储一个标记的 RLE 编码,而每个链表节点存储每个目标扫描线的起点 (x, y) 和长度 length. 对图 10 采用 RLE 编码,用链表实现的结果如图 11 所示.

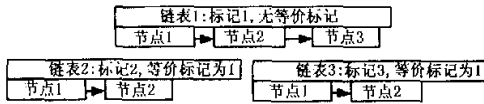


图 11 合并前的链表结构

由于链表结构的特点,将链表 2 和链表 3 合并到链表 1 只需做指针的赋值操作就可以了(图 12),所以非常快捷和方便.但是这些链表的建立需要去申请节点空间,同时计算结束后(比如计算出合并后区域的面积、质心等参数),还要将节点空间释放,以便清空链表,重新开始标记新的二值图象.对操作系统来讲,这些反复进行的操作是比较耗时的,所以又建立了一个临时链表.这个链表的作用就是保留这些将要释放的节点空间.以后需要申请新的节点空间时,就可以到这个临时链表中去取.因为对临时链表做节点存取时,都是指针赋值的操作,所以效率非常高.这样又进一步提高了算法的效率.同时因为采用了带矫正的像素标记算法,所以不会出现一个大的标记和几个小标记等价的情况,最多是大的标记和一个小的标记等价,当然几个大的标记可能和一个小的标记等价,这是改进算法所保证的.于是在做合并算法的时候就从大的标记开始查找等价标记,并做标记合并操作.这样做的好处是只需对这些链表做一次查找即可完成标记的合并过程.

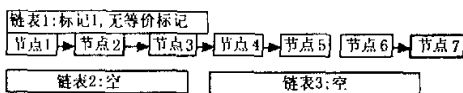


图 12 合并后的链表结构

4 结论

本算法对不包含向上分叉的图形进行标记,效果相当好,都能在一次扫描的过程中完成整幅图的标记,而且速度快.更普适的方法是对标记进行 RLE 游程编码,并用链表结构实现,这就解决了包含单纯向上分叉图形的标记等价问题.在机器人足球赛中,对 CCD 拍摄到的彩色图象取 R、G、B 的平均来进行图象分割,这样分割处理后得到的图象并不会出现向图象内部凹进去的现象,所以能在一次

扫描的过程中迅速完成标记,为处理争取了时间.如果对图象采用完全的彩色分割,比如利用 YUV 颜色空间,采用上下阈值的方法来进行分割^[5],是有可能出现等价标记.即便如此,对像素标记采用游程编码,并采用链表结构来实现的方法,仍然能快速实现标记的合并,同时本算法为统计图象中的分叉提供了一个有意义的解决途径.要统计不同方向的分叉,只需改变一下扫描的顺序就可以了.比如统计向下的分叉,扫描方式为从左到右,从下到上;要统计向左的分叉,扫描方式为从上到下,从左到右;同样,统计向右的分叉,扫描方式为从上到下,从右到左.

综上所述,改进的像素标记算法对传统的像素标记算法,进行了优化,对不包含向上分叉的图形可以进行快速的连通域标记,而且效果很好.此外,对仍包含等价标记的图形,由于改进算法的保证,仍能用链表实现 RLE 编码,完成快速标记合并,同时,改进算法还可以用来统计图形中的分叉个数.

参考文献

- 1 章毓晋. 图象工程(图象处理和分析)(上)[M]. 北京:清华大学出版社,1999:205~206.
- 2 Kenneth R. Castleman digital image processing[M]. New York: Prentice Hall, 1979:317~318.
- 3 James Bruce, Tucker Bulch, Manuela Veloso. Fast and inexpensive color image segmentation for interactive robots[A]. In: 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems [C], Kagawa UTM, Takamatsu, Japan, 2000:1~3;2061~2066.

张修军 1973 年生,浙江大学信息与电子技术工程系电路与系统专业 2000 级硕士研究生.主要研究方向为图象及信息处理、多媒体信息检索.



郭霞 1974 年生,1999 年获南京大学大气科学系应用气象学硕士学位,现为中国科学院大气物理研究所中层室 2000 级博士生.主要研究 GIS 在气象中的应用、全球臭氧反演.



金心宇 1956 年生,浙江大学信息与电子技术工程系电路与系统教授.主要研究图象及信息处理、无线局域网、智能网管网系统.

