

文章编号: 1006-2475(2001) 06-0001-04

一种求取物体凹点的算法研究

刘相滨, 王麓雅

(湖南师范大学图像识别和计算机视觉研究所, 湖南 长沙 410081)

摘要: 在图像识别的应用中,物体凹凸情况的分析是非常重要的,它既是描述物体形状的一个特征参数,也是一些重叠物体分离算法的前提。目前,求取物体边界凹点的算法已有多种^[1-3],但都存在一些不足之处,为此,本文介绍了一种简便算法,该算法在我们开发的微生物细胞图像自动判读系统中已得到应用。

关键词: 边界; 凹点; 算法

中图分类号: TP391

文献标识码: A

A Study on Computing the Concavity of Object

LIU Xiang-bin, WANG Lu-ya

(Institute of Image Recognition & Computer Vision, Hunan Normal University, Changsha 410081, China)

Abstract In the image recognition applications, the analysis on the concavity and convex of objects is very important. It is one shape feature parameter of objects and is also the basis of some splitting algorithms for the overlapped objects. There have been several algorithms for computing the concavity at moment, but they all have some problems. A facile algorithm is presented in the paper. This algorithm has been applied in our microbiology cell image automatic counting system and works well.

Key words boundary; concavity; algorithm

0 引言

在图像分析处理特别是图像识别的应用中,通常需要分析物体的凹凸情况,一方面可以作为描述物体形状的一个特征参数,另一方面也是图像进一步分析的重要前提。例如,在细胞图像的自动判读系统^[4,5]或农业选种应用^[6]中,许多重叠、粘连物体的分离算法都要用到物体的凹点信息。

求取物体边界凹点的算法目前已有多种,但这些算法都有一些不足之处,有些算法对凹点的描述比较准确,但计算量很大;而有些算法计算比较简单,但对凹点情况的描述不够准确。我们在开发微生物细胞图像自动判读系统的过程中,根据实际情况及具体需要,研究设计了一种简易的算法。该算法主要根据物体两个边界点间连线的中点是否落在物体的外部来判定检测

收稿日期: 2001-09-29

作者简介: 刘湘滨(1970-),男,江西吉安人,湖南师范大学图像识别和计算机视觉研究所讲师,硕士,研究方向: 计算机图形图像处理,模式识别。

点是否为凹点,因而只需进行判断、加法及除 2 等简单运算,从而避免了因计算夹角、法线而带来的大量乘、除及三角等复杂运算。该算法在我们的自动判读系统中已经得到成功应用,进一步说明了它是有效和可靠的。

1 已有算法介绍

1.1 计算边界点所成夹角

这种算法的思想是首先在物体边界上选择一个检测点及与其等距的前继点和后续点,然后计算前继点和后续点与检测点的连线所成的夹角,最后根据夹角的大小判定该点是否为凹点^[1]。

如图 1 所示, M 、 N 、 P 均为物体的边界点,其中点 P 为检测点,与其等距的前继点和后续点分别为 M 、 N ,步距为 S 。已知 M 、 N 、 P 三点的坐标,则我们可以计算出直线 PM 与 PN 所成夹角 θ 的大小,若 θ 小于设定的阈值 T ,则认为点 P 为一个凹点。其中步距 S 和角度阈值 T 均为算法参数。

该算法通过 θ 角的大小比较准确地描述了该点的凹凸程度,然而在实际应用中,物体的形态特征各异,图像中或多或少存在着噪声,因此很难设定合适的步长 S 和角度阈值 T 来寻找所有物体的凹点,若 S 设得过小或 T 过大,可能把噪声点当作了凹点;若 S 设得过大或 T 过小,又可能把一些局部凹点忽略掉了。实际上就如图像的阈值分割法一样,有时根本就不存在一个合适的 S 和 T 能够理想地找出所有物体的凹点,此外,该算法对边界上的每个点都要计算一次夹角,计算量是很大的。

1.2 计算边界点切线

这种算法主要通过检测物体边界点的切线是否落在物体区域内部来判定该点是否为凹点。

如图 2 所示, M 、 N 、 P 均为物体的边界点。设直线 l 为物体边界上某点的切线,则从图上我们可以看出:除切点外,如果切线上切点两边的点 1)落在物体区域外部,则该点为局部凸点;2)落在物体区域内部,则该点为局部凹点。如果直线 l 和物体边界多处相切,则这些切点为物体的最大凸点,并且相邻两个最大凸点之间必有一个凹点。例如图中直线 MN 为边界点 M 和 N 的共同切线,则 M 、 N 为物体的两个最大凸点,且边界点序列 MN 段中必有一个凹点 P 。根据这个思想,我们可以找出物体的所有凹点。这种算法对凹点的描述也是很准确的,但计算量是比较大的。

1.3 方向链码

方向链码是物体边界点相对走向的编码序列。通过方向链码来获取物体凹凸情况的算法有微分链码^[2]和高阶链码^[3]。

微分链码反映了相邻两个边界点走向的改变,改变越大,说明物体在该点凹陷或凸出程度越大。为了反映多个相邻边界点走向的改变情况,有学者提出了高阶链码算法,设边界上 i 点的一阶链码为 $Chain_1(i)$,则其 k 阶方向链码定义如下:

$$Chain_k(i) = k \times Chain_1(i) - \sum_{j=1}^{k-1} (k-j) [Chain_1(i-j) + Chain_1(i+j)]$$

这种算法很容易受噪声的影响,并且对物体凹凸情况的描述不是很准确。



图 1 计算边界点所成夹角

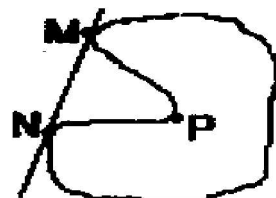


图 2 计算边界点切线

2 本文算法

2.1 算法分析

我们研究发现,如果物体边界上两个点间的连线全部落在物体内部,则该处肯定是物体的一个凸起区域;否则,如果全部落在物体外部,则该处肯定是物体的一个凹陷区域;如果部分落在物体外部、部分落在物体内部,则可缩小间距重新选点进行判断,直到符合上述两种情况为止。根据这个思想,我们设计了一种简便的求取物体凹点的算法。

算法实现的首要问题是如何判断两边界点间的连线和物体的位置关系。我们注意到这样一个事实,只有局部凹点才有可能是物体的凹点,而局部凸点决不会是物体的凹点。因此,我们首先从最简单的情况入手——步距为 1 的情况,找出所有的局部凹凸点,然后剔除局部凸点,然后只对局部凹点进行进一步的检测。

为便于说明,下面列出了步距为 1 时三个相邻边界点所有可能位置关系。如图 3 所示,共分为三类情况,图中 M N P 均为物体的边界点,其中点 P 为检测点, M N 分别为其前继和后续点。

对第一类情况,三点成一直线,我们认为点 P 为局部凸点;对第二、三类情况,如果图中“×”点为物体内部点,则点 P 为局部凸点,否则点 P 为局部凹点。由于步距为 1,所以计算“×”点的坐标非常容易。

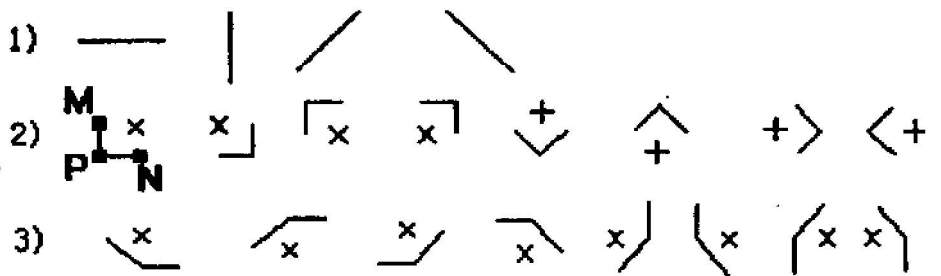


图 3 步距为 1 时三个相邻边界点所有可能位置关系

设 M N P 三点的坐标分别为 (x_1, y_1) (x_2, y_2) (x_0, y_0) , “×”点坐标为 (x, y) , 则有:

$$\begin{aligned} fx &= (x_1 + x_2) / 2.0f; \\ fy &= (y_1 + y_2) / 2.0f; \\ x &= (fx > x_0? 1: (fx < x_0? - 1: 0)); \\ y &= (fy > y_0? 1: (fy < y_0? - 1: 0)); \\ x &= x + x_0; \\ y &= y + y_0; \end{aligned}$$

完成第一次检测后,就可逐步增加步距继续进一步的检测,次数可以自行设定。为简化计算,也只需检测连线的中点即可。有些应用中,只需检测一次即可。

不论采用哪种凹点求取算法,不可避免都会遇到几个连续的边界点、或者相距很近的几个边界点均被检测为凹点的现象,此时,我们必须从中选取一个凹陷程度最大的点作为该区域的实际凹点。在本文算法中,每一次检测出的局部凹点都赋予一个凹陷程度权值: 1)步距为 1 时,第二类情况的凹点权值为 1,第三类情况的凹点权值为 2; 2)步距大于 2 时,如果该点符合凹点特征,则以连线的长度作为近似权值,否则不计权值; 3)步距为 1 和步距大于 2 的权值分别存放,且步距大于 2 时的权值进行累加。凹点合并时,综合考虑两种权值,首先对连续的凹点进行,然后对指定距离内的凹点进行合并。

2.2 算法描述

首先定义物体边界点数据结构:

```
struct    ContourPoint{
int      x, y;           /坐标值
bool     CavFlag;       /局部凹点标志
int      Weight1;       /步距为 1时凹陷程度权值
int      Weight2;       /步距大于时凹陷程度权值
}
```

限于篇幅,在此只给出算法的主控部分。

获取物体边界点序列,并初始化其中的 CavFlag Weight1 Weight2值;
对每个边界点执行(第一次检测)

```
{
取其前继与后续点,计算连线中点坐标 P(x y);
if( P属于物体 )
    CavFlag置 0;
else
    CavFlag置 1;
    if(属于第二类情况)
        Weight1置 1;
    else
        Weight1置 2;
}
```

对步距大于 2时的检测(只对 CavFlag为 1的边界点进行);

合并连续的局部凹点;

合并间距小于指定距离的凹点。

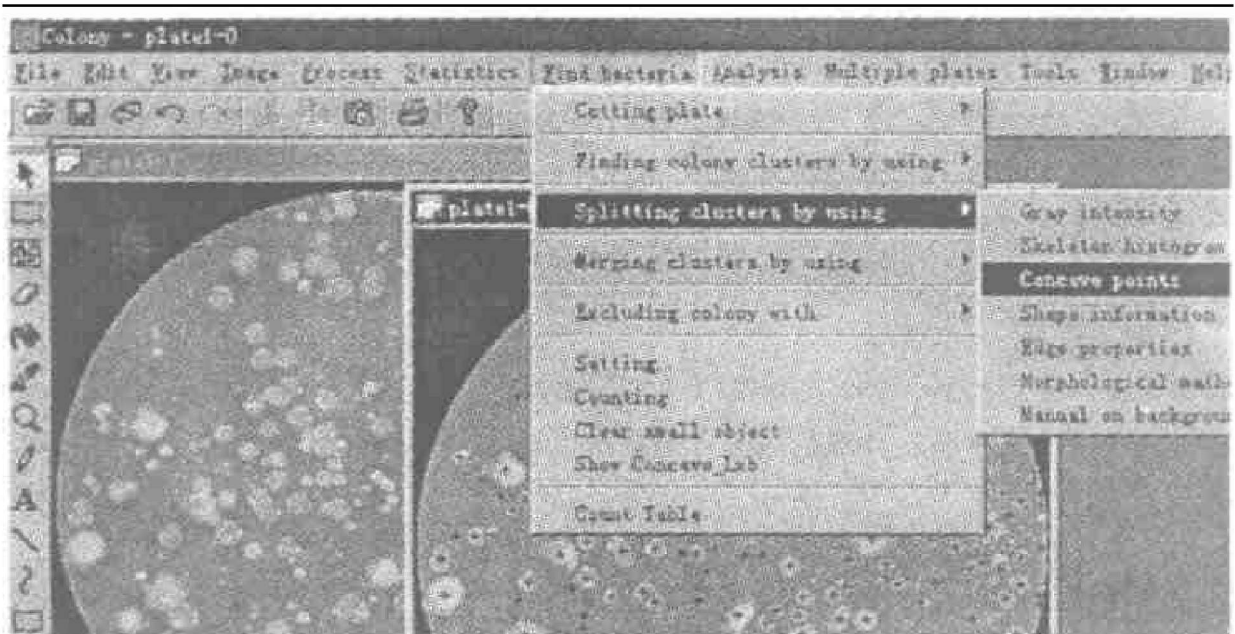


图 4 系统利用凹点信息进行细胞群分离的一个运行界面

(下转第 10页)

提示一个# 提示符。此时用户具有 Unix超级用户的权限。

后续步骤同本地用户。

5 结束语

Oracle是目前应用最广泛的大型数据库管理系统之一,在数据安全性、完整性、并发控制和恢复等方面具有突出的优势,随着信息化建设的不断发展,许多应用软件的数据库从 FoxPro升级到 Oracle,因此原有数据的继承、与其它系统数据交换等方面都有不同程度的影响。

Oracle本身带有数据转换工具如 SQL* LOAD,它需要手工编写控制文件,写出各字段的长度、数据类型和起始位置,也只能做到文本文件向 Oracle数据库的转换,不是很方便;采用 ODBC技术需要设置参数和编程,操作上更专业化,有一定难度。

我们在进行系统改造和数据库升级的同时还考虑到应用方式从 C/S向 B/S转换,对数据迁移的方法进行了深入的研究和比较,从实际情况出发主要采用中间文件和透明性迁移两种方法,达到了较好的应用效果。在迁移过程当中,必然会遇到数据空缺的问题,对此问题我们将在另文中研究。

参考文献:

- [1] C. J. DATE. 数据库系统导论 [M]. 孟小峰 王珊,等译.北京:机械工业出版社,2000, 10.
- [2] 刘志斌. Oracle数据库系统高级应用技术 [M]. 北京:清华大学出版社, 1994, 6.
- [3] 罗晓沛.数据库技术(中级) [M]. 北京:清华大学出版社, 1999, 8.
- [4] 杨勇勤,何志均.实现多系统间的数据交换 [J].计算机工程与应用, 2001(2).

(上接第 4页)

3 实验结果

我们用 Visual C⁺ 6.0实现微生物细胞图像自动判读系统,上述算法在系统中得到了成功应用,系统利用凹点信息进行细胞群分离的一个运行界面如图 4所示,从图中看出,细胞群分离效果是比较理想的。

参考文献:

- [1] Liang Ji. Intelligent Splitting in the Chromosome Domain[J]. Pattern Recognition, 1989, 22(5): 519~ 532.
- [2] Kenneth R. Castleman. Digital Image Processing[M]. Prentice Hall, 1996.
- [3] 刘 泉,傅祖芸,等.一种植物染色体图像核型分析系统 [J]. 计算机工程与应用, 2000, 36(3): 71~ 74.
- [4] 陆建峰,杨静宇,等.重叠细胞图像分离算法的设计 [J]. 计算机研究与发展, 2000, 37(2): 228~ 231.
- [5] Liu Xiangbin, Wang Luya, et al. A new separating algorithm for cell images [A]. The Sixth International Conference for Younger Computer Scientists[C]. 2001, Hangzhou, 997~ 999.
- [6] Talukder Ashit, et al. Modified binary watershed transform for segmentation of agricultural products [A]. Proc. SPIE[C]. 1999, 53~ 64.