文章编号:1000 - 582X(2005)11 - 0043 - 03

图像处理中的圆分析算法*

张建恩,曹长修,金　琼
(重庆大学 自动化学院,重庆　400030)

摘　要:快速而准确地检测圆在计算机视觉、自动化检测领域有着广泛的应用前景. 传统的 Hough 变换有计算量大、占用内存多、提取的参数受参数空间的量化间隔制约等缺陷,针对在简单图像中检测圆时的实时性要求,提出了一种新的圆检测方法. 该方法在图像无相交的情况下,先对图像进行二值化处理,然后对其进行边缘检测,取得图像的边缘链码,在这基础上对图像的圆形度,圆半径等圆的基本要素进行分析计算. 该方法计算速度快、占用的内存小,检测性能比较高.

关键词:图像处理;链码;圆形度;半径
中图分类号:TP391　　**文献标识码**:A

在图像处理中,经常遇到圆的分析处理,例如医学中的细胞分析[1],玻璃瓶口检测[2]等. 这些应用中通常都要计算图像的圆形度和半径. 许多应用对实时性要求都很高,为了提高处理速度,现提出一种新的算法. 在256色灰度图的基础上对图像进行阀值分割[3],把图像转换为二值图,然后提取图像边缘链码,在边缘链码的基础上对图像的圆形度,圆半径进行分析计算.

1　圆分析算法

1.1　链码描述

链码描述是用曲线起始点的坐标和边界点方向代码描述曲线或边界的方法. 任何一条开曲线或闭曲线都可以写成

$$C_{i=1}^{n}a_i = a_1a_2\cdots a_n,\ a_i \in \{0,1,2,\cdots,7\},\quad i = 1,2,3,\cdots,n. \tag{1}$$

的形式,其中方向码的8个值的位置如图1所示.

0	1	2
7		3
6	5	4

图1　方向码值位置图

1.2　圆的检测分析

圆形度又称复杂度、分散度,其定义为:

$$C = L^2/A, \tag{2}$$

式中,L 为区域周长,A 为区域面积. 分析表明,当图像区域为圆时,C 有最小值 4π. 其他任何形状的图像区域,$C > 4\pi$. 且形状越复杂,值越大. 例如不管面积多大,正方形区域圆形度 $C = 16$,正三角形区域圆形度为 $12\sqrt{3}$,差异相当明显.

1.2.1　圆面积的计算

对图像某一区域 R_i,其面积(A_i)是 R_i 中象素点数. 对一帧图像,设有 k 个区域,即 $i = 1,2,3\cdots,k$. 其总面积(A)是各区域面积之和.

$$A_i = \sum_x \sum_y f(x,y),\text{其中}f(x,y) = \begin{cases}1, & (x,y) \in R_i\\ 0, & (x,y) \notin R_i\end{cases} \tag{3}$$

$$A = \sum_{i=1}^{k} A_i. \tag{4}$$

采用顺序扫描方式能够周密地计算面积 A_i 和 A. 已知分割后的图像 $f = \{R_1, R_2, \cdots, R_k\}$,设计计数器 $C_1, C_2, \cdots, C_k$. 扫描中辨认点的归宿并给响应的计数器计数. 扫描结束,各计数器的值即为目标区域 R_i($i = 1,2,3,\cdots,k$)的面积 A_i,累加 A_i 便得到帧内目标图像总面积 A.

利用上述的链码结构,提出了计算图像的面积的快速算法. 该算法只需对链码进行一次的扫描就能计算出图像的面积,极大地提高了图像处理的速度. 针对单目标区域图像,该算法的实现步骤如下:

* 收稿日期:2005 - 05 - 09
作者简介:张建恩(1979 -),男,广东中山人,重庆大学硕士研究生,主要研究方向为图像处理和模式识别.

1) 从链码起点开始,取其纵坐标为第一高度,取其横坐标为长度起点,初始化长度终点等于起点;

2) 取下一个点,判断它的纵坐标是否为一新的高度,若是则转到3),若不是则转到4);

3) 储存这一新高度,取此点的横坐标为此高度的长度起点,初始化长度终点等于起点;

4) 用此点的横坐标与此高度的起点和终点比较,若此横坐标小于起点,则用它作为新起点;若此横坐标大于终点,则用它作为新终点;否则直接转到5);

5) 判断是否到了链码的终点,若是则转到6),否则转到2);

6) 对每一个高度求其长度;

7) 面积等于每个高度的长度之和.

1.2.2 圆周长的计算

图像周长的计算也是采用链码. 基于链码的周长的计算一般有以下几种[4]:

定义1 区域 R 与背景的交界线的长度之和. 交界线有且只有水平和垂直两个方向,其表达式为:

$$L_1 = \sum_{i=1}^{N} l_{i\_stitch}, \tag{5}$$

式中的 l_{i_stitch} 是交界线单位长度,N 为交界线的数目.

定义2 将象素看作点,周长 L 定义为区域边界象素的8链码的长度之和,其表达式为:

$$L_2 = \sum_{i=1}^{Q} l_{i\_chain}, \tag{6}$$

式中的 l_{i_chain} 为链码长度,水平和垂直方向的链码长度为1,其他方向的链码长度为$\sqrt{2}$,Q 为边界线上的象素点数.

定义3 周长为区域 R 的边界点数,其表达式为:

$$L_3 = \sum_{i=1}^{Q} l_{i\_dot}, \tag{7}$$

式中的 l_{i_dot} 取1.

这几种计算周长的定义中,定义1求出的周长最大,定义3最小. 对圆周来说,定义2求出的周长误差最小. 但在圆分析中,即使用定义2也很难得到好的结果. 以半径为200象素的圆为例,周长理论值应为$400\pi \approx 1\ 256.637\ 0$,用定义2计算的结果是1 316.850 3,误差为4.79%. 这样的误差是不能满足实际应用的要求的,因此需要作出改进. 考察一个半径为200象素的圆,由于圆的对称性,只要考虑角度 θ 的两个区间:0~22.5°和22.5°~45°,得出这样一个结果:在0~22.5°之间,水平与垂直方向的象素之和为61个,其他方向的象素之和为15个;在22.5°~45°之间,水平与垂直方向的象素之和为20个,其他方向的象素之和为45个. 按定义2得到的弧长82.213 2,83.225 4与理论值(78.539 8)相比,误差为4.677%和5.966%. 可见定义2中的链码长度并不能真实反映圆周长度.

为了改进算法,首先了解一下圆在计算机上的存储情况. 生成圆或圆弧的常用算法有Bresenham算法和DDA算法[5]. 以Bresenham算法为例,其生成圆的方法是先生成0~45°的圆弧,然后在圆弧是圆的基础,对称地生成整个圆. 根据Bresenham算法,0~45°的圆弧的象素个数为$\frac{\sqrt{2}}{2}R$,设任意两个圆的半径为 R_1、R_2,水平与垂直方向象素的链码长度为 λ_1,其他方向象素的链码长度为 λ_2,其水平垂直方向的象素个数分别为 a_1、a_2,则有:

$$a_1\lambda_1 + \left(\frac{\sqrt{2}}{2}R_1 - a_1\right)\lambda_2 = \frac{2\pi R_1}{8}, \tag{8}$$

$$a_2\lambda_1 + \left(\frac{\sqrt{2}}{2}R_2 - a_2\right)\lambda_2 = \frac{2\pi R_2}{8}, \tag{9}$$

式(8)乘以 R_2 减去式(9)乘以 R_1 得:

$$(a_1R_2 - a_2R_1)\lambda_1 - (a_1R_2 - a_2R_1)\lambda = 0, \tag{10}$$

即

$$\lambda_1 = \lambda_2, \tag{11}$$

代入式(8):

$$\lambda_1 = \lambda_2 = \frac{\pi}{2\sqrt{2}} \approx 1.110\ 720\ 734\ 5 \tag{12}$$

由此可见,对于任意大小的圆都可以假定它的任何方向的链码长度为同一值. 用该方法来计算圆周长,其表达式为:

$$L = \sum_{i=1}^{Q} \lambda, \tag{13}$$

取 $\lambda = 1.121\ 997\ 376$,这里的理论值(1.110 720 734 5)和实际取值(1.121 997 376)的差异是由于象素个数的舍入误差造成的. 取常数 $\lambda = 1.121\ 997\ 376$ 来计算周长能得到更准确的值. 可以看出,式(13)类似于式(7),只是 λ 值取值不是1,而是1.121 997 376.

当圆的面积和周长计算出来后,就可以用定义式(2)计算圆形度. 为了得到一个更直观的数据,定义圆率为:

$$\phi = 4\pi/C, \tag{14}$$

式中 C 为圆形度. 理论上,圆率为0~1的小数,当图像为圆时,ϕ 有最大值1,为其他图像时,ϕ 是小于1的小数,所以能更直观地反映圆形的程度.

圆的半径可以用面积计算,也可以用周长计算,亦可以采用两者的均值. 这里采用的是两者的均值:

$$R = \frac{(L/2\pi + \sqrt{s/\pi})}{2}, \tag{15}$$

式中 R 为半径,L 为周长,S 为面积.

2 仿真分析结果

在计算机上对上述算法进行仿真测试. 测试中采用的是自定义图像,给出以下两组结果:第 1 组给出 5 个不同半径的圆,用前面提出的面积计算算法和定义式(13)计算其面积和周长,用式(14)和式(15)计算其圆率和半径,并给出半径误差见表 1. 第 2 组是几个不同形状的图像的圆率见表 2.

表 1　不同半径的圆的测试结果

测试图像	面积	周长	圆率	半径	半径误差/%
半径为 100 象素的圆	31 335	627.196 5	1.001 0	99.846 3	0.153 7
半径为 150 象素的圆	70 679	945.843 8	0.992 8	150.264 2	0.176 1
半径为 200 象素的圆	125 591	1 260.003 1	0.994 1	200.238 9	0.119 5
半径为 250 象素的圆	196 227	1 578.650 3	0.989 5	250.586 0	0.244 4
半径为 300 象素的圆	282 551	1 897.297 6	0.986 4	300.931 1	0.310 4

表 2　不同形状的图像的测试结果

测试图像	圆率
圆形(半径为 200 象素)	0.994 1
椭圆形 1(长轴为 600 象素, 短轴为 200 象素)	0.591 5
椭圆形 2(长轴为 350 象素, 短轴为 300 象素)	0.982 5
正方形(边长为 400 象素)	0.631 0
正三角形(边长为 400 象素)	0.591 7
不规矩图形 1(多边形)	0.622 9
不规矩图形 2(多边弧形)	0.550 2

3 结　论

圆的检测分析的算法还有 Hough 变换[6-7],但 Hough 变换在圆的检测中需要很大的计算量,在简单的应用环境中(如瓶口检测)不如本计算圆形度的方法快. 本算法在简单的图像处理中可以达到实时性要求,能达到很好的结果.

参考文献:

[1] 李树祥. 医学图象技术的发展与应用[J]. 中国图象图形学报,1996,1(4):281 - 286.

[2] 陈元琰,姜颖军. 基于计算机视觉的玻璃瓶裂纹在线检测系统[J]. 计算机应用,2001,21(11):48 - 49.

[3] 何斌. Visual C + + 数字图像处理(第二版)[M]. 北京:人民邮电出版社,2002.

[4] 傅德胜,寿益禾. 图形图像处理学[M]. 南京:东南大学出版社,2002.

[5] 潘云鹤. 计算机图形学:原理、方法及应用[M]. 北京:高等教育出版社,2001.

[6] 陈震,高满屯,杨声云. 基于 Hough 变换的直线跟踪方法[J]. 计算机应用,2003,23(10):30 - 32.

[7] 孙亦南,刘伟军,王越超,等. 一种用于圆检测的改进 Hough 变换方法[J]. 计算机工程与应用,2003,20:35 - 37.

Algorithm of Circle Analyse in Image Processing

ZHANG Jian-en, CAO Chang-xiu, JIN Qiong

(College of Automation, Chongqing University, Chongqing 400030, China)

Abstract: To detect circles fast and correctly has wide application prospect in the field of computer vision and automatic inspection. The conventional Hough transform has the limitation of using too much computation and memory space, and restricted extracting parameter. Aiming at this problem, the authors bring forward a new detection method in allusion to the real-time requirement of detecting circle in simple image. When the image does not intersect each other, they transform the image into binary image and then perform edge detection to it. After that the chain code of the image is acquired. Basing on that, the circularity of the image and radius of the circle more quickly are calculated and a satisfactory result is gotten.

Key words: image processing; chain code; circular degree; radius

(编辑　吕赛英)