

一种基于随机 Hough 变换圆检测的改进算法

赵桂霞, 黄 山

(四川大学电气信息学院, 四川 成都 610065)

摘 要:介绍了一种基于随机 Hough 变换(RHT)的圆检测的改进算法。该算法利用梯度方向信息来确定采样的三点是否进行累积,然后再利用确定候选圆范围的方法来缩小所要搜索的像素点的范围。此方法较好地解决了传统 RHT 中由于随机采样而造成的大量无效累积问题,并且改进后的算法使运行速度得到进一步的提高,检测性能也有较大的改善。该算法分别在加噪和不加噪的人工图像上做了实验,检测性能和处理速度方面都比传统的 RHT 有明显的改善和提高。

关键词:随机 Hough 变换;圆检测;梯度方向信息;窗口

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2008)04-0077-03

An Improved Randomized Hough Method of Circle Detection

ZHAO Gui-xia, HUANG Shan

(School of Electrical Engineering and Information, Sichuan University, Chengdu 610065, China)

Abstract: Proposes an improved algorithm for detecting circles which is based on randomized Hough transform. It uses gradient direction information to determine whether the parameter should be accumulated or not. And it uses the range of circle to reduce the range of pixels searching. The problem of useless accumulation in traditional RHT is well solved and the improved method has higher speed and better detection performance. Synthetic images with different level noises have been taken to test the detection performance and the speed. And this algorithm has been proved an efficient method.

Key words: randomized Hough transform; circle detection; gradient direction information; window

0 引言

在复杂图像中快速准确地检测出圆及其参数是计算机视觉和模式识别中一项重要的任务。Hough 变换^[1-3]常常被用于圆形特征的检测。Hough 变换的优点在于:它对图像的噪声不敏感,能有效地消除噪声的影响;这种变换便于并行计算,计算机视觉领域的一些问题相当复杂,需要很大的计算量,并行计算是提高计算速度的有效方法。

Hough 变换的基本思想是将图像空间中具有一定关系的像元进行聚类,寻找能把这些像元用某一解析形式联系起来的参数空间累积对应点^[4,5]。在参数空间超过 2 维时, Hough 变换几乎不可能应用于实际,这是由于 Hough 变换计算量和占用内存都很大。为了克服这一缺点, Xu 等^[2]提出了随机 Hough 变换(Randomized Hough Transform, RHT)。由于 RHT 采用了

随机采样和多对一映射,使得时间和空间的开销都大大减少。但是在处理复杂图像时,由于随机采样引入大量无效单元,从而造成大量无效累积。文中提出了一种改进的 RHT 用于圆检测。

1 传统的 RHT 算法

众所周知,半径为 r , 圆心坐标为 (a, b) 的圆的参数空间的表示为

$$(x - a)^2 + (y - b)^2 = r^2 \quad (1)$$

利用该方程可以用较简单的累加的方法实现对圆的检测。此方程包括 3 个参数 (a, b, r) , 根据 Hough 变换的思想,需要建立一个 3 维累加数组,可以让圆心 (a, b) 依次变化计算出 r , 即要通过降低维数来减少计算量。

1.1 传统 RHT 算法原理

用于圆检测的 RHT 具体算法:

(1) 构造边缘点集 D , 初始化参数单元集 $P = \text{NULL}$, 循环次数 $K = 0$;

(2) 从 D 中随机选取 3 点;

(3) 由这 3 点解特征参数 p ;

收稿日期:2007-07-02

基金项目:2007 年四川省重点技术创新项目计划(07XM020)

作者简介:赵桂霞(1981-),女,山东菏泽人,硕士研究生,研究方向为计算机视觉;黄 山,博士,教授,研究方向为模式识别与图像处理。

(4) 在 P 中找一个 p_c , 满足 $\|p - p_c\| \leq \delta$, 若找到则转(6); 否则转(5);

(5) 将 p 插入 P , 其对应计数值为 1, 转(7);

(6) 将 p_c 所对应的计数值加 1, 若小于指定阈值 N_t , 转(7); 否则转(8);

(7) $k = k + 1$; 若 $k > k_{\max}$, 结束; 否则, 转(2);

(8) p_c 为候选圆特征参数, 若该参数对应圆上的边缘点数 $M_{pc} > M_{\min}$, 转(9);

(9) p_c 为真实圆参数, 将落在参数 p_c 对应特征上的点从 D 中去掉, 判断已检测到的圆的数目是否达到规定数目, 若是则结束, 否则重置 $P = \text{NULL}$, $K = 0$, 转(2)。

其中 k_{\max} 为规定的检测一个圆过程中所允许采样的最大循环次数。 M_{\min} 是圆所必需的最小点数, 一般设为 $2\pi\lambda$, 其中 λ 为一个固定系数。 P 为参数空间的参数单元集, 是一个动态链表结构。 M_{pc} 为图像空间中落到候选圆上的点数。

1.2 传统 RHT 无效累积分析

当处理的图像中有多个圆存在时, RHT 由于随机采样会引起大量无效累积单元, 造成大量的无效累积 (见图 1)。假设图像中有几个非圆上的点, 有 N 个大小相同的圆, 且每个圆上有 q 个点, 则随机采样所得 3 点落在同一圆上的概率为:

$$p = \frac{NC_q^3}{C_{Nq+n}^3} = \frac{Nq(q-1)(q-2)}{(Nq+n)(Nq+n-1)(Nq+n-2)} \quad (2)$$

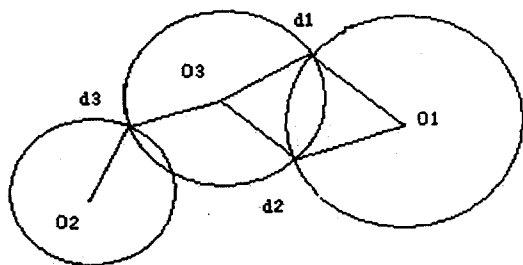


图 1 无效累积说明图

理想情况下, 假设所有点都在圆上, 即 $n = 0$ 时, 式(2)可简化为:

$$p = \frac{Nq(q-1)(q-2)}{Nq(Nq-1)(Nq-2)} \approx \frac{1}{N^2} \quad (3)$$

由式(3)知, 当图像中仅有大小相等的圆时, 随机采样 3 点落在同一圆上的概率和圆的总数的平方成反比。随着圆的数量的增多, 产生无效累积的概率也就随之猛增。

2 改进的 RHT 算法

针对 RHT 变换中的无效累积问题, 可以首先利用

梯度方向信息来确定随机选取 3 点是否在同一个圆上。如果在同一个圆上再对其进行参数计算, 否则就重新选取 3 点。根据圆的性质, 如果 3 点的梯度方向和各自的法线方向在允许的误差范围内时, 即认定 3 点在同一个圆上, 计算出圆的参数后再以圆心为中心, 以圆的直径为边长的正方形范围内依次取一点, 若该点到所解圆心的距离与半径之差在允许的范围时, 说明该点在这一圆上。

利用梯度方向信息可以有效减少动态链表的参数单元集, 确定参数后再通过缩小搜索范围来提高 RHT 变换的速度和效率。

2.1 梯度的求取

在对圆进行检测时, 往往只需要得到该圆的边界信息, 所以要用到边界检测。在数字图像处理中, 一阶导数可以用来检测图像中的一个点是不是边缘点, 而一阶导数是基于各种二维梯度的近似值。像素点 $f(x, y)$ 处的梯度定义为如下的向量:

$$\nabla f = [G_x, G_y]^T = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \quad (4)$$

梯度向量指向坐标 (x, y) 处 f 的最大变化率方向。在边缘检测中梯度的大小是一个非常重要的量, 且是一个标量, 用 ∇F 表示:

$$\nabla F = [G_x^2 + G_y^2]^{1/2} \quad (5)$$

梯度向量的方向也是一个重要的量, 令 $\alpha(x, y)$ 表示标量 ∇F 在 (x, y) 处的方向角, 有

$$\alpha(x, y) = \arctan\left(\frac{G_y}{G_x}\right) \quad (6)$$

这里, 角度是以轴为基准度量的, 边缘在 (x, y) 处的方向与此点的梯度方向垂直, 即此点的梯度方向与此点的法线方向重合。公式(5)和(6)分别给出了梯度大小和方向的计算公式, 两个公式都涉及到 G_x, G_y 的求取, 即图像上每个像素位置的偏导数 $\frac{\partial f}{\partial y}, \frac{\partial f}{\partial x}$ 。一般是通过 Roberts 算子、Soble 算子等经典边缘检测算子来求取 G_x, G_y 。文中采用计算量比较小的 Soble 算子来进行边缘检测。

2.2 加窗

文中采用加窗口的方法来减少运算时间, 提高计算速度。在候选圆周围加一个正方形窗口, 即以候选圆的直径加一定的余量为边长, 以候选圆的圆心为中心的正方形。当随机取点的时候先比较此点是否在所加窗口内, 如果是, 则计算该点到候选圆的距离, 若不是, 则放弃此点, 继续取下面的点进行判断。此方法可缩小搜索范围, 减少计算量的同时提高了计算速度。

2.3 改进的 RHT 算法的描述

(1) 存储所有边缘点 $d_i(x_i, y_i)$ 至集合 D , 并记录

各点的梯度方向,初始化参数单元集 $P = \text{NULL}$;

(2) 从 D 中随机选取 3 点;

(3) 计算这 3 点所确定的圆的参数 $p(a, b, r)$, 然后判断法线方向与梯度之差是否在允许范围内,若是则转(4);否则转(2);

(4) 在 P 中寻找 p_c , 满足 $\|p - p_c\| < \delta$, δ 是允许误差,若找到则转(6),否则转(5);

(5) 将 p 插入 P , 令其计数值为 1;

(6) 若 $a - r < x_i < a + r$ 且 $b - r < y_i < b + r$, 则计算 d_i 到候选圆心的距离,若在允许的误差范围内,则将计数值加 1,遍历正方形范围内所有的点;

(7) 若该参数对应的圆上的点数 $M_p > M_{\min}$, 转(8);否则,为虚假圆,将该参数特征从 P 中剔除;

(8) 检测到的为真实圆,判断已检测到的圆是否已达到规定的数目,若是,结束;否则,把圆上的点从边缘点集中去除,重置 $P = \text{NULL}$,转(2)。

3 实验结果

实验是在内存为 512MHz 的 P4 的计算机上进行的,所处理的图片是合成的圆图像(见图 2、图 4),其检测结果分别见图 3、图 5。实现语言为 VC++。



图 2 原图像 图 3 图 2 的检测结果

表 1 计算速度对比

	传统 RHT 算法(ms)	改进算法(ms)
图 2	362	215
图 4	827	466

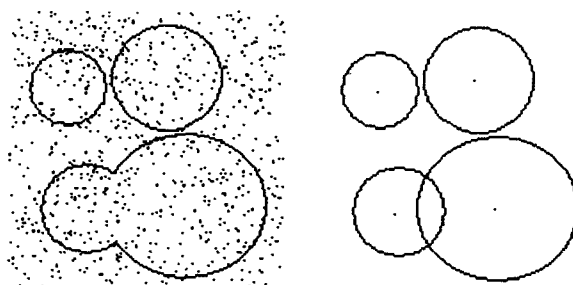


图 4 带噪声图像 图 5 图 4 的检测结果

4 结 语

文中的改进 RHT 算法和传统的 RHT 算法由表的速度对比中可以看出,改进的 RHT 算法在计算速度上有一定的优势。主要是由于通过对随机选取的 3 点的梯度信息判断和确定范围尽量消除无效累积使计算速度有进一步的提高。本算法对像素点多的图像处理的速度也比较快,有一定的使用价值。

参考文献:

- [1] Chen Teh - Chuan, Chung Kuo - Liang. An Efficient Randomized Algorithm for Detecting Circles[J]. Computer Vision and Image Understanding, 2001, 83: 172 - 191.
- [2] Xu L, Oja E, Kultanen P. Randomized Hough Transform (RHT): Basic Mechanisms Algorithms, and Computational Complexities[J]. Computer Vision Graphics Image Process, Image Understanding, 1993, 57(2): 131 - 154.
- [3] 林金龙, 石青云. 用点 Hough 变换实现圆检测的方法[J]. 计算机工程, 2003(11): 17 - 18.
- [4] 束志林, 戚飞虎. 一种新的随机 Hough 快速圆检测算法[J]. 计算机工程, 2003(6): 87 - 88.
- [5] 张兵权, 王建峰, 卢 军. 一种新的对随机 Hough 变换改进的检测圆的方法[J]. 计算机工程与应用, 2006(18): 53 - 54.

(上接第 76 页)

项目开发时,模型转换组件的开发有着十分重要的意义。

因为模型转换组件质量的高低决定着所生成的代码的质量,因此开发模型转换组件的工作应该由项目组中水平较高的成员来完成。好的模型转换组件所生成的代码,会使用恰当的设计模式,符合最佳的设计实践,因此具有很高的质量^[6]。由于项目的大部分代码都是由模型转换组件自动生成的,因此,各项目成员所交付的最终代码质量相近,风格一致。这对提高后期的测试效率,缩短项目的开发周期都有较好的影响。

参考文献:

- [1] 郑 涛. AndroMDA: 免费的 MDA 午餐[J]. 程序员: CSDN 开发高手, 2004(4): 94 - 97.
- [2] 徐晓钟, 谢康林. 基于 MDA 方法学软件开发方式的原理与实现[J]. 微机发展, 2004, 14(4): 74 - 77.
- [3] OMG. MDA Guide Version 1. 0. 1 [EB/OL]. 2003 - 06. <http://www.omg.org/cgi-bin/doc?omg/03-06-01>.
- [4] Crane D, Pascarello E. Ajax in Action [M]. [s. l]: Manning Publications Co., 2006.
- [5] 柯自聪. Ajax 开发精要——案例与框架 [M]. 北京: 电子工业出版社, 2006.
- [6] 刘发贵, 胡耀民. 基于 MDA 的模式化软件设计方法与应用[J]. 计算机应用, 2005, 25(4): 790 - 791.