

# 改进的随机圆检测算法

虞旦 韦巍

(浙江大学电气工程学院, 杭州 310027)

**摘要** 介绍了一种基于随机圆检测的改进算法。解决了随机圆检测算法随机采样阶段引入的大量无效累积和验证阶段假设圆之外的边缘点所进行的无效计算的问题,首先对边缘图像进行8邻接连通,3个随机采样点在一个连通曲线内选择,然后利用圆的性质求得圆的参数,再应用证据收集过程来进一步验证圆的真实性,该验证过程中只考虑假设圆的外切正方形和内接正方形之间的边缘点。实验结果表明,该算法计算量小,对比改进前,检测性能有所提高,且鲁棒性也相应提高,同时能够检测圆弧和有相互重叠的多个圆。

**关键词** 随机圆检测 8邻接连通 多圆检测

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2009)08-1590-04

## Improved Randomized Circle Detection Algorithm

YU Dan, WEI Wei

(College of Electrical Engineering, Zhejiang University, Hangzhou 310027)

**Abstract** An improved algorithm based on randomized circle detection is presented. The large amount of useless accumulations yielded by random sampling of randomized circle detection algorithm and the useless square computations resulted from the outlier edge points during the evidence-collecting phase are solved. First, the edge pixels with 8-adjacency connectivity are connected, and three edge pixels in the same connected curve which can exactly determine a possible circle with center and radius are randomly picked. When a possible circle is found, an evidence-collecting process is implemented to further verify whether the possible circle is a true circle or not, in which only the edge points locate between the circumscribed square and the inscribed square of the possible circle are considered. Compare to conventional detecting algorithm, the proposed algorithm consumes less computing resources, has better detection performance when comparing to other methods and can detect circular arc or partly overlapped circles by the experiments.

**Keywords** randomized circle detection, 8-adjacency connectivity, multi-circles detection

## 1 引言

图像中圆的检测是机器视觉中的一项重要任务。其中应用最广的是应用 Hough 变换<sup>[1-2]</sup>, Hough 变换是检测灰度图像中直线、圆和椭圆的一种方法。其基本原理是将图像空间具有一定关系的像元在参数空间进行聚类,但是这种方法在参数空间超过 2

维的时候,计算量和占用内存很大。为了解决这个问题,Xu 等人提出了随机 Hough 变换(RHT)<sup>[3]</sup>,对边缘点随机采样后进行多对一的映射,减少了计算量和内存的开销,但是基于 RHT 的方法其思想仍旧是进行参数空间的累积,无法从根本上解决耗时耗存储空间的问题,而且由于会进行参数空间的量化,这和检测结果的准确性直接相关,也就很难在图像中准确地定位圆,Chen 等人于是提出了随机圆检测

基金项目:浙江省自然科学基金项目(R105341)

收稿日期:2007-09-07;改回日期:2008-05-19

第一作者简介:虞旦(1983~),男。浙江大学电气工程学院控制理论与控制工程专业博士研究生。主要研究方向为机器视觉、智能机器人。E-mail:yiwuyudan@163.com

算法 (RCD)<sup>[4]</sup>。与 RHT 不同, RCD 并不是基于 Hough 变换的思想, 它采用了假设-验证的结构进行检测可能的圆, 由于不涉及参数空间的累积, 而且假设圆的计算是直接在原图中进行, 所以效率和准确性都较基于 RHT 的方法有所提高。但是即便如此, 当图像中存在多个圆特征时, RCD 在随机采样时针对的是整个图像空间中的边缘点, 在对假设圆进行验证的时候, 面对的也是所有边缘点, 这些均会消耗大量无效的计算时间, 针对这些问题对 RCD 算法进行了改进, 提高了检测的效率和鲁棒性。

## 2 随机圆检测算法

用于圆检测的 RCD 算法的具体步骤如下<sup>[4]</sup>:

(1) 构造包含图像中所有边缘点的集合  $\mathbf{V}$ , 初始化失败计数  $f$  为 0, 初始化  $T_f, T_{\min}, T_a, T_d, T_r$  5 个阈值。其中,  $T_f$  表示容许失败的次数。如果  $\mathbf{V}$  中剩余边缘点的个数少于  $T_{\min}$ , 圆检测也将停止。随机选取的 4 个点要求任意两个点之间的距离大于  $T_a$ 。为了验证第 4 个边缘点是否在前 3 个边缘点所决定的假设圆上, 定义了距离阈值  $T_d$ 。比例阈值  $T_r$  用于对假设圆的验证。此外定义了  $|\mathbf{V}|$ , 用于标识  $\mathbf{V}$  中剩余的边缘点数。

(2) 如果满足了循环中止条件  $f = T_f$  或者  $|\mathbf{V}| < T_{\min}$ , 则停止; 否则随机在  $\mathbf{V}$  中挑选 4 个边缘点, 记做  $v_i, i = 1, 2, 3, 4$ 。并且从  $\mathbf{V}$  中去掉这 4 个点。

(3) 检验 4 个点中任意两个点的距离是否大于  $T_a$  和第 4 个边缘点是否在另 3 个边缘点所决定的假设圆上, 如果满足, 进到下一步; 否则, 将 4 个点返回  $\mathbf{V}$  中,  $f = f + 1$ , 并返回步骤(2)。

(4) 假设  $C_{ijk}$  就是步骤(3)中的假设圆, 圆心  $(a_{ijk}, b_{ijk})$ , 半径  $r_{ijk}$ , 设置计数器  $c$  为 0。对于  $\mathbf{V}$  中的任意一点  $v_l = (x_l, y_l)$ , 如果与圆  $C_{ijk}$  的距离  $d_{l \rightarrow ijk}$  小于  $T_d$ , 即  $d_{l \rightarrow ijk} = \left| \sqrt{(x_l - a_{ijk})^2 + (y_l - b_{ijk})^2} - r_{ijk} \right| \leq T_d$ ,  $c = c + 1$ , 并且从  $\mathbf{V}$  中去掉点  $v_l$ 。扫描所有边缘点后, 记计数值  $c$  为  $n_p$ 。

(5) 如果满足  $n_p \geq 2\pi r_{ijk} T_r$ , 假设圆  $C_{ijk}$  就被验证为真实的圆, 将  $f$  值设为 0, 返回步骤(2); 否则, 把  $n_p$  个边缘点返回  $\mathbf{V}$ , 返回步骤(2)。

## 3 改进的检测算法

上述检测算法的每一次循环主要由确定假设圆和验证假设圆两个部分组成, 对于确定假设圆这部分, 可以通过提高随机采样的准确性来提高效率, 在包含了多个圆特征的图像中, 原有算法中步骤 2 的随机采样, 可能采样到的像元包括: 圆上的点, 噪声点, 其他曲线上的点, 只有采样的 4 个点来自同一个圆时, 才是有效的采样, 因而大量的采样都是无效的。有一个显而易见的结论是: 构成一个圆的所有边缘点必然是连通的或者是部分连通的, 如果先对边缘点进行 8 邻接的连通, 随机采样的 4 个点在同一连通曲线上选择, 这在很大程度上提高了这 4 个点来自同一个圆的概率。并且只需要对边缘点个数超过某个阈值的连通曲线进行可能圆的检测, 滤去了小噪声, 同样有助于减少计算量。注意到一个圆肯定处在其外切正方形之内, 内接正方形之外, 所以对于第 2 部分提高效率的方法就是: 在得到假设圆后, 先构造出假设圆的内接和外切正方形, 外切正方形之外和内接正方形之内的边缘点不进行验证计算, 这样可以在边缘点个数较多时显著地减少计算时间。

基于以上两点, 改进的随机圆检测算法如下:

对边缘图像进行 8 邻接的连通, 构造成链表  $L$ , 每一节点  $l_i$  表示一条连通曲线, 并且有相应的属性参数  $p_i$  表示每条曲线包含的边缘点数, 假设链表上共有  $n$  个节点, 算法遍历链表上的节点搜索可能的圆。初始化参数  $T_a, T_d$  和  $T_r$ 。由于对每个节点的处理都是相同的, 下面描述其中任一个节点的检测算法:

(1) 如果  $p_i$  小于某个阈值, 跳出本次循环, 本节点不作处理; 否则初始化失败计数  $f$  为 0, 初始化参数  $T_f$  和  $T_{\min}$ 。节点中剩余边缘点的个数少于  $T_{\min}$ , 则跳出本循环。注意与原有算法不同的是, 这里两个参数的设置是自适应的。是  $p_i$  的一个线性函数。此外定义  $|p_i|$  为用于标识节点  $l_i$  中剩余的边缘点数。

(2) 如果满足了循环中止条件  $f = T_f$  或者  $|p_i| < T_{\min}$ , 则停止; 否则随机在节点  $l_i$  中挑选 4 个边缘点, 并且从  $l_i$  中去掉这 4 个点。

(3) 检验 4 个点中任意两个点的距离是否大于  $T_a$  及第 4 个边缘点是否在另 3 个边缘点所决定的假设圆上, 如果满足, 进到步骤(4); 否则, 将 4 个点返回节点  $l_i$  中,  $f = f + 1$ , 并返回步骤(2)。

(4) 假设  $C_{ijk}$  就是步骤 (3) 中的假设圆, 圆心  $(a_{ijk}, b_{ijk})$ , 半径  $r_{ijk}$ , 设置计数器  $c$  为 0。根据圆心和半径计算内接正方形和外切正方形。对于外切正方形之内和内接正方形之外的边缘点  $v_l = (x_l, y_l)$ , 如果与圆  $C_{ijk}$  的距离  $d_{l \rightarrow ijk}$  小于  $T_d$ , 即  $d_{l \rightarrow ijk} = \left| \sqrt{(x_l - a_{ijk})^2 + (y_l - b_{ijk})^2} - r_{ijk} \right| \leq T_d$ ,  $c = c + 1$ , 如果点  $v_l$  属于节点  $l_i$ , 则从  $l_i$  中去掉点  $v_l$ 。统计完后, 记计数值  $c$  为  $n_p$ 。

(5) 如果满足  $n_p \geq 2\pi r_{ijk} T_r$ , 假设圆  $C_{ijk}$  就被验证为真实的圆, 将  $f$  值设为 0, 返回步骤 (2); 否则, 把  $n_p$  个边缘点返回节点  $l_i$ , 返回步骤 (2)。

## 4 实验结果

实验分为两部分, 程序用 C 语言在 Pentium IV 512M 内存的 PC 上实现。第 1 部分实验设计是为了检验本文算法在计算时间方面的效率, 以图 1 所示的 3 张自然图片为例, 分别采用 RCD 算法和本文改进算法进行检测, 分别进行 100 次实验, 然后对耗费的检测时间进行统计。

把检测得到的圆叠加到原图上, 得到图 2 所示的结果。

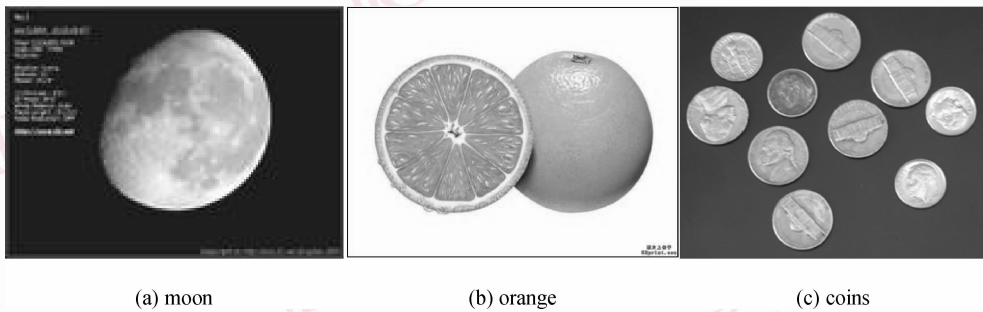


图 1 实验所用的自然图片  
Fig. 1 Nature images for test

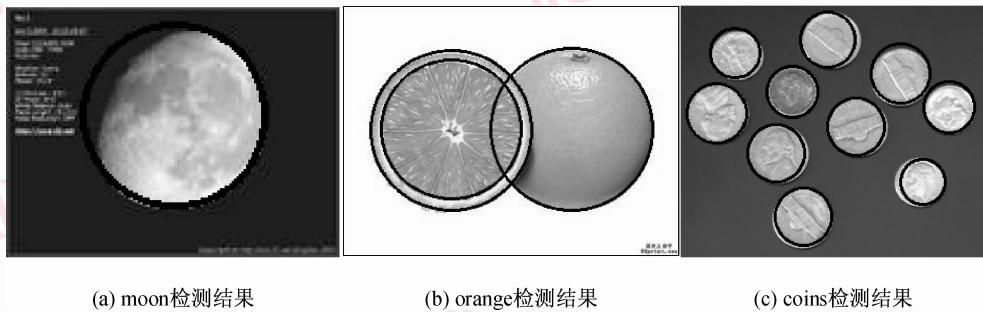


图 2 自然图片的处理结果  
Fig. 2 Test results of nature images

表 1 列出了两种算法对图 1 进行圆检测时在计算速度上的对比。从表 1 可以看出, 改进的算法在计算时间上较原有算法减少了一个数量级别, 这在实际应用中具有显著的意义, 要特别指出的是算法的计算时间与参数  $T_f, T_{\min}, T_d$  和  $T_r$  的取值有直接关系, 实验中参数值的选择依据的原则是使得检测成功率达到 90% 以上。

第 2 部分的实验则为了检验改进的算法在准确性上是否有提高, 以图 1(c) 为例, 图中有 10 个圆, 各圆的实际参数 (圆心和半径) 通过在图像处理软件中手工量测得到, 这为后面的准确性分析提供了

表 1 两种算法的耗时比较

Tab. 1 Contrast of consuming time by two algorithms

		图像		
		moon	orange	coins
圆个数		1	3	10
图像大小		140 × 105	320 × 240	300 × 246
参数	$T_{\min}$	400	400	1 000
	$T_f$	800	1 200	3 000
	$T_d$	1.2	1.5	1.5
	$T_r$	1.5	1.2	1.0
平均耗时 (ms)	RCD	702	1 233	2 232
	改进的 RCD	15	31	58

可能,实验同样采用两种算法分别进行 100 次的圆检测,表 2 列出了结果,其中准确性定义为所有 100 次实验准确检测圆个数除以全部需要检测的圆个数(10×100)。

表 2 两种算法对 coins 图像检测准确性比较

Tab.2 Contrast of accuracy by two algorithms

单位:%

RCD			改进的 RCD
$T_i = 6\ 000$	$T_i = 4\ 000$	$T_i = 3\ 000$	
$T_d = 1$	$T_d = 1.5$	$T_d = 1.8$	
86	79	74	100

被处理图片的边缘图片由图 3(a)给出,图 3(b)给出了误检测的情况,所检测出来的圆中有

3 个(图中标号为 1,5 和 8)是错误的,经过分析,其可能的原因如下:标准的随机圆检测算法针对的是整个图像平面的所有边缘点,图 1(c)存在多个圆和较多噪声点,在验证阶段,真实的圆的一部分就和相邻圆的部分及噪声点构成了一个错误的圆,例如标号 8 圆的大部分和标号 4,5 圆的小部分构成了一个错误的圆。这类误检测在采用改进算法时是不可能出现的,因为改进的算法在确定假设圆阶段在同一连通曲线上随机选择边缘点,而出现误检测的圆所包含的点显然不在一条连通曲线上,另外改进算法在验证阶段对可能的圆上的点进行了范围界定,就是改进算法中步骤 4 所提到的外切正方形之内和内接正方形之外,同样也就不可能出现如图 3(b)那样的误检测。

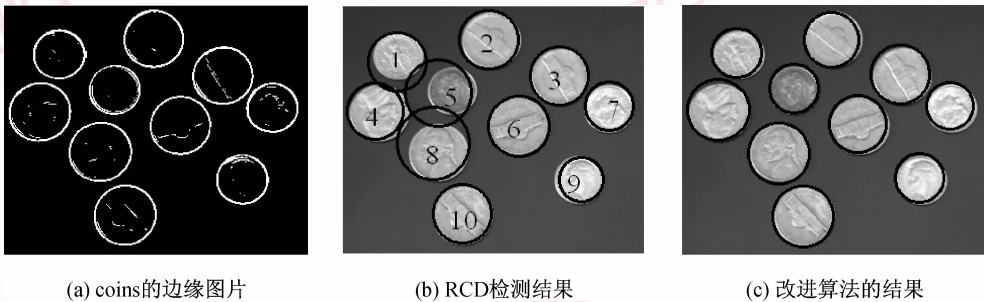


图 3 自然图片 coins 的实验结果  
Fig. 3 Test results of nature image 'coins'

表 2 给出的结果说明改进的算法在准确性上较之原有算法有很大的提高,特别是处理的图像存在多个圆和高噪声时,改进的算法并未出现如图 3(b)一样的误检测,准确性为 100%,实验结果有力地证明了这点。

## 5 结 论

本文对随机圆检测算法进行了分析,提出了改进的检测算法,改进的算法保留了原算法随机检测圆的基本思想,采样之前先进行边缘图像的连通,采样点的选择在同一连通曲线上,提高了针对性,因而大大减少了无效的采样,另外,验证阶段加入了范围界定,这样候选圆之外的点无需进行复杂的平方和开方运算,这些都提高了检测速度。新算法对噪声

也不敏感,而且在处理多个圆及圆重叠都有很好的准确度,由于阈值的设定是自适应的,因而鲁棒性也更好。

## 参考文献 (References)

- 1 Zheng Nan-ning. Computer Vision and Pattern Recognition[M]. Beijing: Science Press, 1987: 154-160. [郑南宁. 计算机视觉与模式识别[M]. 北京: 科学出版社, 1987: 154-160.]
- 2 Illingworth J, Kittler J. A survey of the Hough transform [J]. Computer Vision Graphics Image Process, 1988, 44(1):87-116.
- 3 Xu L, Oja E. Randomized Hough transform: basic mechanisms, algorithms, and computational complexities [J]. Computer Vision Graphic Image Process: Image Understanding, 1993, 57(2): 131-154.
- 4 Chen Teh-chuan, Chung Kuo-liang. An efficient randomized algorithm for detecting circles [J]. Computer Vision and Image Understanding, 2001, 83(2): 172-191.