

处理彩色图像的椭圆检测算法

张子皓, 钟宝江*, 王子楷, 陈翀

苏州大学计算机科学与技术学院, 江苏 苏州 215006

摘要 当前的椭圆检测算法均是针对灰度图像设计和实现的, 在处理彩色图像时会抛弃很多有用信息, 不利于获取更高质量的检测结果。为此, 提出一种针对彩色图像的椭圆检测算法。首先, 对图像的 3 个色彩通道以及加权灰度图分别进行椭圆检测, 融合多通道的检测结果得到一个椭圆集合; 然后, 将同一椭圆在不同图像通道中的多重响应结果进行聚类, 并合并成一个椭圆; 最后, 提出一种椭圆有效性的验证技术, 该技术通过 DiZenko 算子融合图像的色彩信息, 由此提取椭圆支撑线段来判断椭圆的有效性, 过滤掉无效的椭圆, 得到最终的检测结果。实验结果表明, 与现有算法相比, 所提算法充分使用图像的色彩信息, 显著提高了椭圆检测的效率, 在标准数据集上的 F-score 明显优于当前检测算法。

关键词 图像处理; 椭圆检测; 彩色图像; 椭圆聚类; 椭圆验证

中图分类号 TP391.4

文献标志码 A

DOI: 10.3788/LO220722

Ellipse Detection Algorithm for Color Image Processing

Zhang Zihao, Zhong Baojiang*, Wang Zikai, Chen Chong

School of Computer Science and Technology, Soochow University, Suzhou 215006, Jiangsu, China

Abstract Current ellipse detection algorithms have been designed and implemented for grayscale images. When processing color images, much useful information is discarded, which is inconducive to obtaining higher quality detection results. Therefore, an ellipse detection algorithm for color images is proposed. First, the three color channels and weighted grayscale image of the image are detected for ellipses, and a set of ellipses is obtained by fusing the detection results of multiple channels. Then, the multiple response results of the same ellipse in different image channels are clustered and combined into an ellipse. Finally, an ellipse validity verification technique is proposed. The technology fuses the color information of the image using the DiZenko operator, extracts the ellipse support line segment to determine the effectiveness of the ellipse, filters invalid ellipses, and obtains the final detection results. Experiments show that compared with existing algorithms, the proposed algorithm fully uses the color information of the image, thereby significantly improving the efficiency of ellipse detection, and the F-score on a standard dataset is significantly better than that of the current detection algorithms.

Key words image processing; color images; ellipse detection; ellipse clustering; ellipse validation

1 引言

椭圆是图像中常见的几何体, 检测图像中的椭圆是图像分析中重要的一步。椭圆检测可以应用于图像分析、图像匹配和物体检测^[1]。目前, 椭圆检测算法大致有 3 种: 1) 基于最小二乘(LS)的方法; 2) 基于霍夫变换(HT)的方法; 3) 基于弧段提取的方法。LS 是一种数学方法, 其过程相对简单, 是一种常见的椭圆检测算法。Fitzgibbon 等^[2]提出的直接最小二乘法(DLS), 可以实现非迭代的椭圆拟合并生成唯一的闭式解。然而, 由于矩阵存在奇异性, 该算法有时得不到最终的最

优解。Halir^[3]用更优的公式替换避免了上述问题。

Hough transform (HT)^[4]是一种特征检测方法, 在图像分析中被广泛使用。经典的 HT 被用于检测图像中的直线。经过数十年的发展, HT 得到了许多改进以满足各种约束, 其中包括椭圆。在这一过程中, Duda 等^[5]将 HT 用于检测图像中的圆锥曲线。

近年来, 研究人员提出基于弧段提取的椭圆检测方法, 即舍弃离散的边缘点, 使用弧段作为椭圆检测的数据。从图像中获取与椭圆检测相关的边缘弧段, 然后将可能为同一椭圆的弧段放在同一集合中, 再进行筛选、参数估计等工作。在这一过程中, 提取图像的边

收稿日期: 2022-02-14; 修回日期: 2022-04-09; 录用日期: 2022-07-05; 网络首发日期: 2022-07-15

基金项目: 江苏省高等学校自然科学基金项目(21KJA520007)、江苏高校优势学科建设工程资助项目、软件新技术与产业协同创新中心部分资助

通信作者: *bjzhong@suda.edu.cn

缘是一项重要的工作。Soh等^[6]使用提取短直线的方法获取相似弧段。Prasad等^[7]将椭圆弧段用于拟合椭圆。之后,Wang等^[8]使用一种新的方法获取椭圆弧段。Lu等^[9]提出一种基于弧支持线段(ARC)的算法,使用直线段检测(LSD)算法检测弧支持线段,具有较高的检测精度。

受限于硬件设备的计算能力与存储能力,传统算法在检测椭圆时通常只使用灰度图像。这一做法延续至今,所以目前的椭圆检测算法只适用于灰度图像。传统算法在检测椭圆时会丢失大量色彩信息,这会使得椭圆检测出现漏检、错检等问题。近年来,硬件设备的性能得到大幅提高,当前已经可以处理相对复杂的彩色图像,从而获得更好的检测结果。

本文提出一种基于色彩通道的椭圆检测(CAED)算法,该算法可以在彩色图像上取得更好的检测结果。对红、绿、蓝(R,G,B)这3个色彩通道以及加权灰度图分别进行检测,再对检测结果通过聚类进行合理融合,利用DiZenko算子求出3个色彩通道的梯度矢量;之后,根据平均梯度强度与方向得到图像边缘点,再使用椭圆弧支持线段检测算法检测弧支持线段;最后,基于

椭圆的弧支持线段判断每个椭圆的有效性并去除无效的椭圆。所提算法通过合理聚类与椭圆有效性判断融合相似椭圆并舍弃无效椭圆,可以显著提高召回率,同时使精确率损失降到最低,从而得到更优的检测结果。

2 彩色图像的椭圆检测

2.1 技术思想

一张图像中存在多个椭圆时,当图像灰度化后不同的像素点可能得到相同或相近的灰度值,这可能使得某些椭圆难以被检测。图1给出了一个直观的例子,图1(a)为一幅包含多个椭圆的彩色图像,其中,每一个像素点的灰度值相同。图1(b)是当前算法的检测结果,因为使用当前算法得到的灰度图的每一个像素点拥有相同的灰度值,所以未能检测出椭圆。为此,从R、G、B这3个色彩通道分别进行椭圆检测再通过聚类得到一个椭圆集合,结果如图1(c)所示,由于色彩信息被充分利用,图中椭圆被正确标出。一般来说,当图像中的像素点拥有相近的灰度值时,尽管人类能够识别图像中的椭圆,但是目前的椭圆检测算法却总是漏检,导致召回率较低。

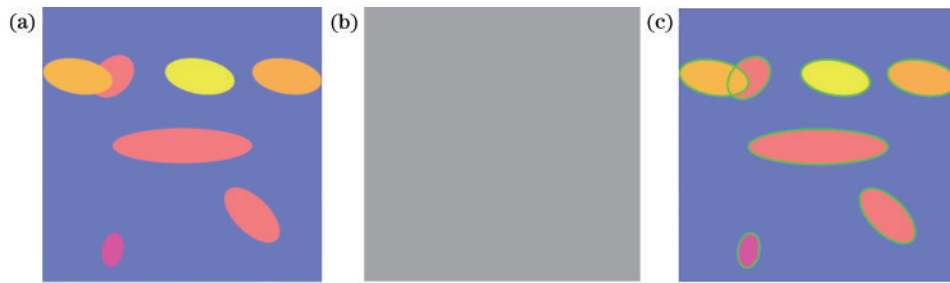


图1 现有椭圆检测算法的问题。(a) 测试图像;(b) 现有算法的检测结果;(c) 所提算法的检测结果

Fig. 1 Problem of existing ellipse detection algorithms. (a) Test image; (b) detection result of existing algorithm; (c) detection result of proposed algorithm

为了解决在彩色图像中椭圆检测经常漏检的问题,要充分获取图像的色彩信息,实现召回率的提高。具体地,有一张彩色图像 I ,该彩色图像由3个色彩通道构成,用 I_r 表示红色通道的图像层, I_g 表示绿色通道的图像层, I_b 表示蓝色通道的图像层。对3个通道图像以及根据加权法得到的灰度图像分别进行椭圆检测可以得到一个椭圆集合,再将相似椭圆进行聚类保证精确率不会严重损失。然而,图像不是由3个通道图简单叠加而成的,因此直接将原始图像转换为3个通道图像会造成色彩信息的错误从而导致一些无效的椭圆被检测出。为此,提出一种创新性的方法来过滤无效的椭圆。

色彩信息错误可能导致椭圆被错误检测,为了避免这个问题,需要先获得一幅能够综合色彩信息的图像。Akinlar等^[10]提出一种边缘检测算法,这种算法在彩色图像上可以获得较好的检测结果。该算法使用DiZenko算子计算张量梯度,可以获得更小的算法复杂度。基于DiZenko算子提出一种过滤无效椭圆的技

术。具体来说:首先,通过DiZenko算子得到一幅综合图像色彩信息的边缘图 W ,由此检测出图像中的椭圆弧支持线段;然后,对于先前检测出的椭圆,对照弧支持线段计算出一个椭圆有效值,记为 $\text{Goodness}(e)$;最后,通过合理配置 $\text{Goodness}(e)$ 的阈值,判断出对应椭圆的有效性。

2.2 算法模型

彩色图像拥有3个色彩通道,分别为红、绿、蓝。当前的椭圆检测算法均直接将图像转换为灰度图再进行后续操作,这样会损失大量的色彩信息,使得很多人眼清晰可见的椭圆消失或者被减弱。为此,将彩色图像分裂成3个子图像,每个子图像分别存储1个色彩通道,保证每个色彩通道的信息都被储存,从而避免色彩信息的丢失。

1幅图片由3个通道构成,将这3个通道分别记做 I_r 、 I_b 、 I_g ,令 F 表示3个色彩通道的叠加。由此, I 可被表示为

$$I = F(I_r, I_b, I_g). \quad (1)$$

目前的算法直接将彩色图像转换为灰度图,再对此灰度图进行操作。用 $N(x, y)$ 表示新构造的图像 N 上的任一点像素, Detect 表示椭圆的检测算法。具体做法为

$$N(x, y) = 0.299I_r(x, y) + 0.587I_g(x, y) + 0.114I_b(x, y), \quad (2)$$

$$E_{\text{Ellipses}} = \text{Detect}(N). \quad (3)$$

不难发现这种加权分量法会舍弃大量的色彩信息,将 1 幅图像的 3 通道值简单加权相加,不能充分利用图像的色彩信息。因此,将图像分裂成为 3 个子图像,每个子图像分别保留原图像 1 个色彩通道的信息。用 f 表示将一幅图像分裂的操作,那么图像的分裂过程用数学模型表示如下:

$$(I_r, I_g, I_b) = f(I). \quad (4)$$

对每个子图像以及通过分量加权法得到的灰度图进行椭圆检测,可以得到 4 个椭圆集合。利用分量加权法将图像转换为灰度图再检测椭圆会损失大量色彩信息,有些人眼清晰可见的椭圆在某一颜色通道可以被检测出来,但是在单一的灰度图中却难以检测。因此,在 3 个色彩通道进行检测得到的 3 个椭圆集合包含了一些原本检测不出来的椭圆。然而,这样做也会导致色彩信息的冗余。对于信息冗余,因为 3 个色彩通道代表的图像有一定相似性,得到的 4 个椭圆集合里的椭圆有许多重复椭圆,因此要对得到的椭圆进行聚类,将相同或相近的椭圆归为 1 个椭圆。

Prasad 等^[11]提出一种简单、计算量小的椭圆聚类方法,可以很好地检测相似椭圆并进行聚类。设两个椭圆的参数为 $\{x_i, y_i, a_i, b_i, \theta_i\}$, $i=1, 2$, 其中, x_i, y_i 分别为椭圆在 x 方向与 y 方向上的像素总数, a_i, b_i 分别为椭圆的长半轴、短半轴的像素总数,定义以下参数:

$$D_x = \frac{|x_1 - x_2|}{X}, \quad (5)$$

$$D_y = \frac{|y_1 - y_2|}{Y}, \quad (6)$$

$$D_a = \frac{|a_1 - a_2|}{\max(a_1, a_2)}, \quad (7)$$

$$D_b = \frac{|b_1 - b_2|}{\min(b_1, b_2)}, \quad (8)$$

式中: X, Y 分别是图像中 x 方向与 y 方向上的像素总数; D_x, D_y, D_a, D_b 分别为不同方面的相似度。定义以下方法来比较两个椭圆的相似性:

$$D_\theta = \begin{cases} 0, & \frac{b_1}{a_1} \geq 0.9, b_2/a_2 \geq 0.9 \\ 1, & \frac{b_1}{a_1} \geq 0.9, b_2/a_2 < 0.9 \\ 1, & \frac{b_1}{a_1} < 0.9, b_2/a_2 \geq 0.9 \\ \frac{\angle(\theta_1, \theta_2)}{\pi}, & \frac{b_1}{a_1} < 0.9, b_2/a_2 < 0.9 \end{cases}, \quad (9)$$

式中: $\angle(\theta_1, \theta_2)$ 表示两个椭圆长轴的最小角度。除了一般的椭圆,以上参数也适用于接近圆形的椭圆。与这些参数相对应,设定阈值 $\tilde{D}_\theta, \tilde{D}_x, \tilde{D}_y, \tilde{D}_a, \tilde{D}_b$ 。这些阈值在 $[0, 1]$ 之间,根据实验经验,将这些阈值都设为 0.1。最终的相似性指标定义为

$$D = \text{AND} \left[(D_x < \tilde{D}_x), (D_y < \tilde{D}_y), (D_a < \tilde{D}_a), (D_b < \tilde{D}_b), (D_\theta < \tilde{D}_\theta) \right], \quad (10)$$

式中: D 是布尔型变量,若两个椭圆的 D 值为真,就将这两个椭圆归为一个聚类。将所有检测到的椭圆按照得分降序排序,按顺序逐个比较两个椭圆,并将簇中评分最高的椭圆作为新的簇中心继续进行比较。当出现不属于任何簇的椭圆时,将这个椭圆作为新的聚类中心。遍历所有的椭圆,最终得到目标椭圆集合。上述过程可表示为

$$E_{\text{Ellipses}} = \text{Clustering} \{ x | x \in \text{Detect}(I_r) \cup \text{Detect}(I_g) \cup \text{Detect}(I_b) \}, \quad (11)$$

式中: Clustering 是聚类函数。

由于图像并不是由 3 个色彩通道图像简单叠加而成的,因此,直接将图像转换为 3 个通道图像会导致色彩信息的错误,从而使得一些错误的椭圆被检测出。为了解决这一问题,首先要获得一幅可以综合色彩信息的图像。DiZenzo 算子是一种可以综合图像色彩信息的算子,可将 DiZenzo 算子与椭圆弧支持线段用于无效椭圆的过滤。使用 DiZenzo 算子得到彩色图像的三通梯度,使不同颜色分量的相反矢量相互加强,从而充分获得彩色图像的色彩信息。设 u 和 v 为输入图像在 x 和 y 方向上的偏导数,DiZenzo 算子定义为

$$g_{xx} = u \cdot u = \left| \frac{\partial I_r}{\partial x} \right|^2 + \left| \frac{\partial I_g}{\partial x} \right|^2 + \left| \frac{\partial I_b}{\partial x} \right|^2, \quad (12)$$

$$g_{yy} = v \cdot v = \left| \frac{\partial I_r}{\partial y} \right|^2 + \left| \frac{\partial I_g}{\partial y} \right|^2 + \left| \frac{\partial I_b}{\partial y} \right|^2, \quad (13)$$

$$g_{xy} = u \cdot v = \left| \frac{\partial I_r}{\partial x} \right| \left| \frac{\partial I_r}{\partial y} \right| + \left| \frac{\partial I_g}{\partial x} \right| \left| \frac{\partial I_g}{\partial y} \right| + \left| \frac{\partial I_b}{\partial x} \right| \left| \frac{\partial I_b}{\partial y} \right|. \quad (14)$$

由式(12)~(14)可得,梯度幅度 $g(x, y)$ 和梯度方向 $\text{dir}(x, y)$ 分别为

$$g(x, y) = \left\{ \frac{1}{2} \left[(g_{xx} + g_{yy}) + (g_{xx} - g_{yy}) \cos(2\theta) + 2g_{xy} \sin(2\theta) \right] \right\}^{\frac{1}{2}}, \quad (15)$$

$$\text{dir}(x, y) = \theta = \frac{1}{2} \arctan \left(\frac{2g_{xy}}{g_{xx} - g_{yy}} \right). \quad (16)$$

在椭圆检测中,弧支持线段的提取至关重要。图像中的线段主要来自两种情况:1) 如果一个区域中的点拥有几乎相同的水平线角度且这些点总体呈直线分

布,那么这些点的集合就是线段;2)另一种线段来自于圆弧支撑区域,区域中某些点的分布类似于曲线,称其为椭圆弧支持线段。在之前的算法中,椭圆弧支持线段的提取往往在加权灰度图中进行,由于加权灰度图本身抛弃了大量色彩信息,因此弧支撑线段的检测可能会出现漏检、错检的问题。通过 DiZenko 算子可以得到综合图像色彩信息的边缘检测图像 W ,再使用 LSD 算法可以检测出更多正确的弧支持线段。

由于椭圆曲线可能由多个弧支持线段组构成,将多个弧支持线段连接成为一组,链接的任意两个弧支持线段应满足合理的连续性和凹凸性。对于连续性条件,一个弧支持线段的头部与另一个弧支持线段尾部之间的距离应足够小;对于凹凸性条件,弧支撑线段组中的线段应沿同一方向顺时针或逆时针变化。如果某个弧支持线段组包含 n 个线段,那么这个弧支持线段组有 $n-1$ 个角度间隔。假设这些角度为 $\{\theta_1, \theta_2, \dots, \theta_{n-1}\}$,那么这个弧支持线段组的角度和为 $\sum_{j=1}^{n-1} \theta_j$ 。弧支持线段组的角度和越大,它越有可能构成某个椭圆。因此有 $S = \sum_{j=1}^{n-1} \theta_j / 360^\circ$,其中, S 是弧支持线段组的分数。

对于先前检测出的任意椭圆 e ,假设椭圆 e 由 n 个弧支持线段组构成,定义 $\text{Goodness}(e)$ 为

$$\text{Goodness}(e) = \frac{C(e)}{\text{SI}(e)} \cdot (\sum_{i=1}^n S_i) / n, \quad (17)$$

式中: $\text{SI}(e)$ 表示在图像 W 中该椭圆的弧支持线段总数量; $C(e)$ 表示此椭圆被检测出的正确弧支持线段数量,正确的弧支持线段指该线段应在图像 W 中被检测出; S_i 表示构成椭圆 e 的第 i 个弧支持线段组。 $\text{Goodness}(e)$ 值越大,这个椭圆越可能是真实的椭圆。当 $\text{Goodness}(e)$ 大于 0.6 时,召回率会显著降低从而导致 F-score 降低,因此将 $\text{Goodness}(e)$ 值设为 0.6。

3 算法实现

所提算法的主要步骤包括图像输入、通道图像生成、通道图像上的椭圆检测、椭圆聚类 and 椭圆有效性判断。

3.1 通道图像生成

彩色图像拥有红、绿、蓝这 3 个色彩通道。为了保证色彩信息不丢失,将图像分裂成 3 个子图像,每个子图像分别储存 1 个色彩通道的信息,再通过加权平均算法得到原图像的灰度图。对某一图像 I ,得到 3 个子图像并求得加权灰度图,图 2 为通道生成示意图。

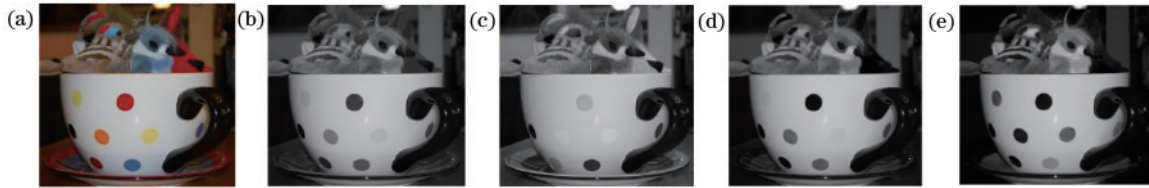


图 2 通道图像生成。(a) 测试图像;(b) 灰度图;(c)~(e) R、G、B 色彩通道

Fig. 2 Generation of image channels. (a) Test image; (b) grayscale image; (c)~(e) R, G, and B color channels

3.2 椭圆检测

Lu 等^[9]提出的 ARC 算法可以得到较高的得分,且花费的时间较少。鉴于 ARC 算法的优越性能,使用该算法在每个图像通道上分别进行椭圆检测,检测结果

如图 3 所示。从图 3 可以看到,3 个色彩通道上检测出的椭圆都不相同,这是因为每个色彩通道所蕴含的色彩信息不同,导致有的椭圆在某一个色彩通道清晰可见而在另一色彩通道却几乎看不到。

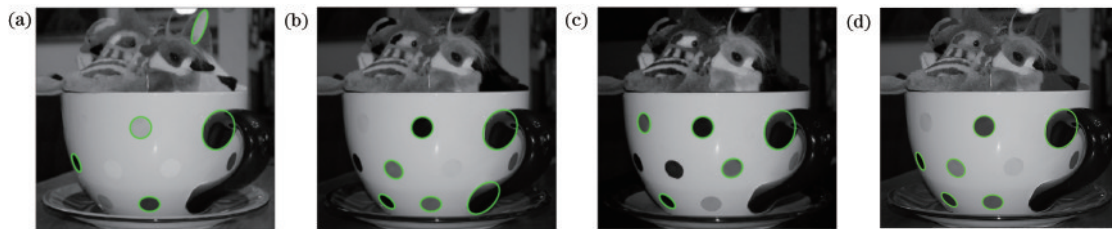


图 3 通道图像上的椭圆检测。(a)~(d) R、G、B 色彩通道及灰度通道上检测椭圆结果

Fig. 3 Ellipse detection on image channel.(a)~(d) Ellipse detection results on R, G, B color channels and grayscale channel

3.3 椭圆的聚类与有效性判断

对每个子图像进行椭圆检测,可以得到 4 个椭圆集合。将图像利用分量加权法转换为灰度图再检测椭圆会损失大量色彩信息,有些人眼清晰可见的椭圆在某一个色彩通道可以被检测出来但是在加权灰度图中却难以检测。为此,在色彩通道上得到的 3 个椭圆集合包含了一些原本检测不出来的椭圆。然而,这样做也会导

致色彩信息的冗余,因为 3 个色彩通道代表的图像有一定相似性,得到的 3 个椭圆集合里的椭圆有许多重复椭圆,因此需要对得到的椭圆进行聚类,将相同或相近的椭圆归为一个椭圆。使用 DiZenko 算子得到综合图像色彩信息的图像,对于先前检测出来的椭圆,求得每一个椭圆的 $\text{Goodness}(e)$,去除掉 $\text{Goodness}(e)$ 值小于 0.6 的椭圆。所提算法与 ARC 算法的结果对比如图 4 所

示。从图 4 可以看到,由于 ARC 算法对图像进行了色彩空间转换,部分椭圆无法被有效检测。而所提算法通过对色彩信息的保护,成功检测了大部分被遗漏的椭圆,展现了更优秀的检测性能。

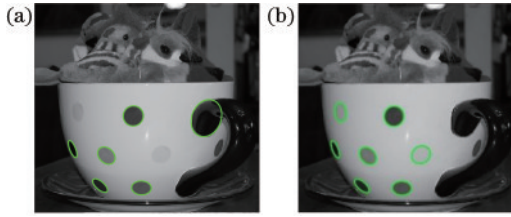


图 4 椭圆检测结果对比。(a) ARC 算法检测结果;(b) 所提算法检测结果

Fig. 4 Comparison of ellipse detection results. (a) detection results of ARC algorithm; (b) detection results of proposed algorithm

4 实验结果

4.1 实验配置

为了系统地评估所提算法的性能,在标准数据集上进行测试,并将所提算法的检测结果与当前主流的椭圆检测算法进行对比。测试数据集包括 Prasad 等使

用的数据集^[12-14]、Fornaciari 等使用的数据集^[15]、PCB 数据集^[16]和 Smartphone 数据集^[15]。作为对比的椭圆检测算法包括 Prasad 等^[12]提出的算法、AAMED^[17]、YAED^[18]、CNC^[18]、ELSDc^[14]及 ARC^[9]等 6 种。所有参与测试的算法均使用各自的参数设置。Prasad 等使用的数据集包含 198 张图片,包含来源于现实世界的椭圆形目标,例如人脸、小番茄等。Fornaciari 等使用的数据集包含 400 张图片,这些图片的最初来源为 MIR 和 LabelMe 数据库,其中, MIR 图片为一些高清的图片,而 LabelMe 图片大部分是噪声比较多而且分辨率比较低的图像。PCB 数据集包括 100 张具有各种干扰的工业图片,每张图片至少包含一个圆形或椭圆形。所有 PCB 图片均手动精确标记。Smartphone 数据集包含 629 张图像,这些图片使用智能手机拍摄,且大部分图片由同一物体从不同角度拍摄得到。

4.2 主观测评

从 Prasad 等使用的数据集与 Fornaciari 等使用的数据集中选取 6 张图像来进行对比。主观对比结果如图 5 所示,其中,第 1 行是测试图像,第 2 行是人工标注 (GT) 结果,其余为 6 种现有算法及所提算法的检测结果。从图 5 的对比结果可以看出,所提算法在图像色彩信息复杂时可以检测出更多正确的椭圆,表现最好。

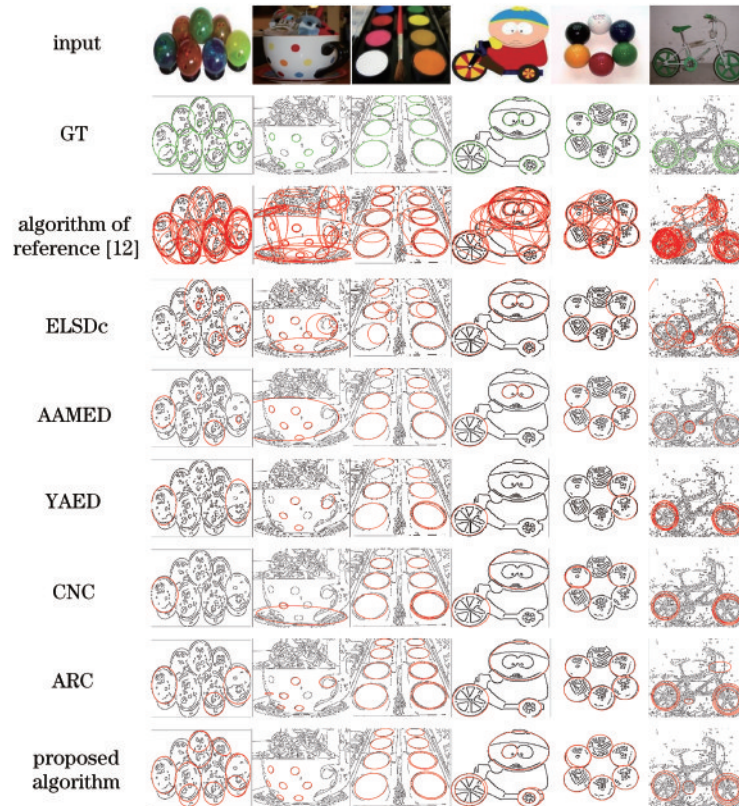


图 5 不同算法的椭圆测试结果

Fig. 5 Ellipse detection results of different algorithms

由于上述图像均来自标准数据集,当算法应用于实拍图像时可能会出现实用性差的问题。为了验证所提算法应用于实拍图像时的有效性与实用性,选取 3 张实拍图像,这些图像均来自不同的真实场景。由

于实拍图像存在像素低、噪声大、背景复杂等问题,检测椭圆将变得较为困难。所提算法检测结果如图 6 所示。从图 6 的检测结果可以看出,所提算法在实拍图像上也可以取得很好的检测结果。



图 6 实拍图像的椭圆检测结果
Fig. 6 Ellipse detection results of real images

4.3 客观测评

记 T_p 为正确检测出的椭圆个数; F_p 为错误检测的椭圆个数; F_N 为漏检的椭圆个数。采用以下 3 种指标评价各算法性能。

1) 精确率 (Precision): 检测到正确椭圆占所有检测到椭圆比例, 计算方法为

$$P = \frac{T_p}{T_p + F_p} \quad (18)$$

2) 召回率 (Recall): 检测到正确椭圆占人工标注椭圆的比例, 计算方法为

$$R = \frac{T_p}{T_p + F_N} \quad (19)$$

3) F-score 为精确率和召回率的加权调和平均, 计

算方法为

$$F_\alpha = \frac{P \times R}{\alpha \times P + (1 - \alpha) \times R} \quad (20)$$

式中: α 值决定了精确率和召回率这两个指标谁更重要。在实验中, 将 α 值设置为 0.5, 表明二者在 F-score 的计算中同样重要。

表 1 是几种椭圆检测算法的精确率、召回率和 F-score 的对比结果。从表中可以看到, 所提椭圆检测算法的召回率明显高于对比算法, 也获得了更高的 F-score。这是因为所提算法能够充分利用色彩信息, 检测出在某一通道比较明显而在另一通道不明显的椭圆。即所提算法在检测性能上优于现有的椭圆检测算法。

表 1 不同算法的客观评测结果
Table 1 Objective evaluation results of different algorithms

Dataset	Index	Algorithm of reference [12]	YAED	CNC	ELSDc	AAMED	ARC	Proposed algorithm
Prasad	Precision	0.5719	0.7029	0.7136	0.6283	0.5201	0.7389	0.6352
	Recall	0.2764	0.2275	0.2395	0.3657	0.2112	0.3571	0.4395
	F-score	0.3729	0.3437	0.3586	0.4623	0.3003	0.4815	0.5195
Random	Precision	0.0975	0.5953	0.6266	0.4473	0.3641	0.5936	0.5844
	Recall	0.2789	0.4094	0.3902	0.3649	0.1926	0.4725	0.5472
	F-score	0.1445	0.4852	0.4809	0.4019	0.2520	0.5262	0.5652
Smartphone	Precision	0.0425	0.4729	0.5708	0.4092	0.3364	0.5463	0.5755
	Recall	0.5419	0.5406	0.5457	0.3429	0.1888	0.7113	0.7642
	F-score	0.0788	0.5405	0.5579	0.3731	0.2418	0.6180	0.6565
PCB	Precision	0.7645	0.7117	0.7944	0.4168	0.9800	0.9294	0.9670
	Recall	0.7115	0.7692	0.8769	0.9154	0.7538	0.9115	0.9447
	F-score	0.7371	0.7393	0.8336	0.5728	0.8522	0.9203	0.9521
Average	Precision	0.3691	0.6207	0.6734	0.4754	0.5502	0.7021	0.6901
	Recall	0.5099	0.5124	0.5540	0.5413	0.3846	0.6600	0.7161
	F-score	0.3333	0.5272	0.5578	0.4525	0.4116	0.6365	0.6733

5 结 论

提出一种新的椭圆检测算法。所提算法考虑到彩色图像的 3 个色彩通道, 将图像转换为 3 个通道图像进行处理, 再将得到的椭圆进行聚类, 最后筛选掉无效的椭圆。所提算法可以充分利用彩色图像的色彩信息,

在色彩信息特别复杂的图像中, 可以获得更好的检测结果。将所提算法与当下流行的几种椭圆检测算法进行对比, 用 Prasad 等使用的数据集、Fornaciari 等使用的数据集、PCB 数据集和 Smartphone 数据集的图像进行了主观视觉测评以及精确率、召回率的比较。实验结果表明, 所提算法可以获得更高的 F-score。

参 考 文 献

- [1] 吴斌, 叶东, 郭玉波, 等. 基于航天应用的多圆特征识别和姿态估计[J]. 光学学报, 2017, 37(9): 0915004.
Wu B, Ye D, Guo Y B, et al. Multiple circle recognition and pose estimation for aerospace applications[J]. Acta Optica Sinica, 2017, 37(9): 0915004.
- [2] Fitzgibbon A, Pilu M, Fisher R B. Direct least square fitting of ellipses[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 21(5): 476-480.
- [3] Hailr R. Numerically stable direct least squares fitting of ellipses[EB/OL]. [2022-02-04]. <http://autotrace.sourceforge.net/WSCG98.pdf>.
- [4] Hart P E. How the Hough transform was invented [DSP History][J]. IEEE Signal Processing Magazine, 2009, 26(6): 18-22.
- [5] Duda R O, Hart P E. Use of the Hough transformation to detect lines and curves in pictures[J]. Communications of the ACM, 1972, 15(1): 11-15.
- [6] Soh Y, Bae J, Kim D, et al. A new method for ellipse fitting in satellite images[C]//2009 Second International Conference on Intelligent Computation Technology and Automation, October 10-11, 2009, Changsha, China. New York: IEEE Press, 2009: 502-506.
- [7] Prasad D K, Leung M K H, Quek C. ElliFit: an unconstrained, non-iterative, least squares based geometric Ellipse Fitting method[J]. Pattern Recognition, 2013, 46(5): 1449-1465.
- [8] Wang Y T, He Z Q, Liu X C, et al. A fast and robust ellipse detector based on top-down least-square fitting[C]//Proceedings of the British Machine Vision Conference 2015, September 7-10, 2015, Swansea, UK. London: British Machine Vision Association, 2015: 1-12.
- [9] Lu C S, Xia S Y, Shao M, et al. Arc-support line segments revisited: an efficient high-quality ellipse detection[J]. IEEE Transactions on Image Processing, 2020, 29: 768-781.
- [10] Akinlar C, Topal C. EDLines: a real-time line segment detector with a false detection control[J]. Pattern Recognition Letters, 2011, 32(13): 1633-1642.
- [11] Prasad D K, Leung M K H. Clustering of ellipses based on their distinctiveness: an aid to ellipse detection algorithms[C]//2010 3rd International Conference on Computer Science and Information Technology, July 9-11, 2010, Chengdu, China. New York: IEEE Press, 2010: 292-297.
- [12] Prasad D K, Leung M K H, Cho S Y. Edge curvature and convexity based ellipse detection method[J]. Pattern Recognition, 2012, 45(9): 3204-3221.
- [13] Jia Q, Fan X, Luo Z X, et al. A fast ellipse detector using projective invariant pruning[J]. IEEE Transactions on Image Processing, 2017, 26(8): 3665-3679.
- [14] Pătrăucean V, Gurdjos P, von Gioi R G. Joint A contrario ellipse and line detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(4): 788-802.
- [15] Fornaciari M, Prati A, Cucchiara R. A fast and effective ellipse detector for embedded vision applications[J]. Pattern Recognition, 2014, 47(11): 3693-3708.
- [16] Lu C S, Xia S Y, Huang W M, et al. Circle detection by arc-support line segments[C]//2017 IEEE International Conference on Image Processing, September 17-20, 2017, Beijing, China. New York: IEEE Press, 2017: 76-80.
- [17] Meng C, Li Z X, Bai X Z, et al. Arc adjacency matrix-based fast ellipse detection[J]. IEEE Transactions on Image Processing, 2020, 29: 4406-4420.
- [18] McLaughlin R A. Randomized Hough Transform: improved ellipse detection with comparison[J]. Pattern Recognition Letters, 1998, 19(3/4): 299-305.