Pattern Recognition 48 (2015) 411–421

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Randomized circle detection with isophotes curvature analysis

Tommaso De Marco [a,*], Dario Cazzato [a,b], Marco Leo [a], Cosimo Distante [a]

[a] Istituto Nazionale di Ottica - CNR, Viale delle Libertá n. 3, 73010 Arnesano, LE, Italy
[b] Universitá del Salento, 73100 Lecce, Italy

ARTICLE INFO

Article history:
Received 22 August 2013
Received in revised form
20 July 2014
Accepted 11 August 2014
Available online 20 August 2014

Keywords:
Circle detection
Sampling strategy
Isophotes
Density estimation

ABSTRACT

Circle detection is a critical issue in image analysis and object detection. Although Hough transform based solvers are largely used, randomized approaches, based on the iterative sampling of the edge pixels, are object of research in order to provide solutions less computationally expensive. This work presents a randomized iterative work-flow, which exploits geometrical properties of isophotes in the image to select the most meaningful edge pixels and to classify them in subsets of equal isophote curvature. The analysis of candidate circles is then performed with a kernel density estimation based voting strategy, followed by a refinement algorithm based on linear error compensation. The method has been applied to a set of real images on which it has also been compared with two leading state of the art approaches and Hough transform based solutions. The achieved results show how, discarding up to 57% of unnecessary edge pixels, it is able to accurately detect circles within a limited number of iterations, maintaining a sub-pixel accuracy even in the presence of high level of noise.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Circle detection is a critical issue in pattern recognition and computer vision [1]. Most of the current solutions are based on Hough transform [2], the well-known method to recognize complex models in computer vision. The Hough transform has been used for the first time to detect curves in images in [3] and since then different variants have been proposed [4,5]. The circular Hough transform (CHT) consists of a sequence of algorithmic steps: an edge map of the image is firstly computed and then the detected edge pixels are mapped into a three dimensional (Hough) space defined by the parameters necessary to represent univocally a specific circle (center coordinates and radius). The analysis of this Hough space is based on the construction of an accumulator array according to a specific voting strategy, which allows us to select circular objects with a number of edge pixels lying on the circumference higher than a threshold. CHT approaches are affected by relatively high memory requirements and computational load, but they can generally achieve an high degree of accuracy, anyway subject to the image quality and to the setup of the involved functional parameters (whose values have to be supplied by the user on the basis of the characteristics of the processed image). Additionally, CHT based approaches are generally affected by a large number of false positive detections, especially in case of incorrect settings of the input parameters and in the presence of noise.

To overcome these limitations, over the years, different improvements of the original CHT approach have been proposed: size invariant formulation [6], fuzzy HT [7], hypothesis filtering [8], and randomized Hough transform [9,10]. In the randomized CHT iteratively three edge points are randomly chosen to determine the parameters of a candidate circle, and then parameters are collected in the 3D accumulator; candidate circles corresponding to maxima in the accumulator are chosen as detected circles. Maintaining this iterative approach, in different solutions the high memory-demanding accumulator is replaced by less computational expensive voting strategy, in order to select among a set of candidate circles the best one(s). In [11] a LUT-based voting scheme is proposed. In the Randomized Circle Detection (RCD) method [12] at each iteration four edge pixels are selected: three pixels are used to define a possible circle, while the fourth one is used to check if it can be considered as a valid candidate or not; then, a voting process based on counting the number of edge pixels lying on the circumference (namely *inliers*) determines whether the candidate circle is a true circle or not. In [13], an improved RCD method is presented, GRCD-R, where a multiple-evidence-based sampling strategy is used to determine a restricted candidate circles set (starting from the edge pixels extracted by the Sobel operator): a fourth pixel, as in RCD, with additional evidences derived by the gradient of the image is used to check the validity of each candidate circle. Then a refinement stage is performed on the recognized circles, to reduce a bias effect due

* Corresponding author. Tel.: +39 0832321816; fax: +39 0832420553.
E-mail addresses: tommaso.demarco@ino.it (T. De Marco), dario.cazzato@ino.it (D. Cazzato), marco.leo@ino.it (M. Leo), cosimo.distante@ino.it (C. Distante).

http://dx.doi.org/10.1016/j.patcog.2014.08.007
0031-3203/© 2014 Elsevier Ltd. All rights reserved.

to the usage of only three edge pixels for the circle parameters' computation.

Different circle detectors have been developed based on other approaches, such as genetic algorithms [14,15] and nature-inspired computing [16]. Recently EDCircles has been presented, a parameter-free circle detection algorithm able to recognize circles and ellipses, which uses an *a contrario* validation step to reduce false positives [17]. EDCircles is based on the analysis of the edge segments (arcs) extracted by an edge segment detector: the Edge Drawing Parameter Free (EDPF) algorithm [18], that does not need parameters supplied by the user (different from traditional edge extractors).

In this work, we propose a randomized circle detection algorithm which uses an alternative multiple-evidence strategy to define valid circles set: maintaining the four edge pixels approach, we add further constraints based on the curvature of the isophotes. Isophotes are curves connecting pixels in the image with equal intensity, whose properties make them particularly suitable for objects detection [19]. Preliminarily, edge pixels with too high or too low local isophote curvature are discarded, then the remaining pixels are classified into subsets with the same isophotes curvature (and consequently belonging to possible candidate circles with the same radius). This leads to three improvements: firstly, the sampling process can be limited on each subset, so increasing the probability to sample edge pixels belonging to the same circle; secondly, candidate circles with radius not compliant with the subset under exam (false positives) can be discarded before the voting process and finally, dependency of the results from the used edge map is reduced. This last feature is particularly relevant considering the aforementioned problem of the input parameters: the proposed work-flow is not parameter free, because a traditional edge extractor is used, but isophotes analysis allows us to obtain similar detection performance starting from edge maps differently detailed or affected by noise. For each candidate circle a kernel density based estimation voting process is performed; this provides better results than simple counting of edge pixels, because *inliers* are automatically defined according to the distribution of the distances between each edge pixel and the circle center. Then, detected circles parameters are refined with an error linear compensation algorithm, in order to provide a better fitting with the recognized circle and the *inliers*.

Each step of the proposed algorithm is described in Section 2, while in Section 3 the obtained results and performance are shown; the method is compared with other circle detectors in the literature using common test images. Finally, achieved conclusions are given in Section 4.

2. Algorithm description

The presented algorithm is based on the analysis of the curvature of isophotes, curves connecting pixels in the image with equal intensity. Isophote's properties make them particularly suitable for objects detection and image segmentation, e.g. they have been used for face detection [20], for ridges seeking in CT/MRI scans of the human brain [21], and, recently, for accurate eye center location [22]. In particular, it has been demonstrated that their shapes are independent of rotation and varying lighting conditions, and, in general, isophote features result in better detection performance than intensities, gradients or Haar-like features [19].

2.1. Isophotes curvature estimation

Curvature κ of an isophote, which is the reciprocal of the subtended radius r, can be computed as

$$\kappa = \frac{1}{r} = -\frac{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}{(L_x^2 + L_y^2)^{3/2}} \qquad (1)$$

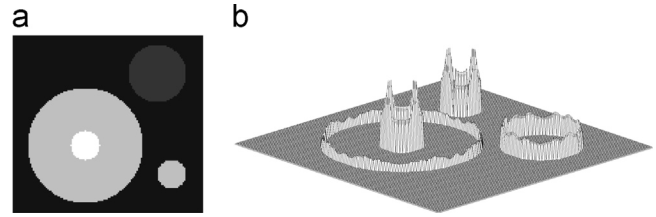

Fig. 1. (a) Original image. (b) 3D view of the isophote curvature at the edges.

where $\{L_x, L_y\}$ and $\{L_{xx}, L_{xy}, L_{yy}\}$ are the first- and second-order derivatives of the luminance function $L(x,y)$ in the x and y dimensions respectively (for further details refer to [23]). For our purpose, isophotes curvature is restricted to edge pixels V extracted using a Canny operator [24], as shown in the sample image in Fig. 1. Restriction of the curvature values on V is performed using a median filter centered on each edge pixel, in order to reduce the aliasing effect due to the image discretization that otherwise produces a crown-like effect along the edge. This effect can be also reduced by preliminarily applying a Gaussian smoothing to the original image; the convolution of the original image with a Gaussian kernel improves, in fact, isophote curvature estimation, but with high scales, i.e. standard deviations, the important features of the image can be discarded.

The obtained curvature map is then processed by filtering values between a lower-bound T_{min} and an upper-bound T_{max} ($T_{min} < \kappa < T_{max}$): this allows us to discard pixels with excessively low curvature (e.g. belonging to lines or in general to circles with radius bigger than the image size), or presenting a too high curvature (i.e. too small circles), shrinking V only to meaningful pixels. Eventually, (T_{min}, T_{max}) can be chosen in order to limit the search to circles of fixed radius, or to a desired sign of κ (the sign of isophote curvature, in fact, depends on the relative intensity of the outer/inner sides of the curve).

2.2. Edge pixels classification

The filtered map of κ values is then analyzed to detect the occurrence of most probable values, as it was a probabilistic distribution. This way to proceed arises from the fact that if a circle is present in the image, then there is an accumulation of edge pixels with the corresponding curvature. Calling M the unknown number of local maxima in κ distribution, Mean Shift is employed to detect local maxima κ_i, $i = 1:M$, in κ, assigning at each edge pixel a probability weighted by a 1D Gaussian kernel. The edge map V is then divided into subsets V_i, given by pixels with the same isophote curvature κ_i. The basic idea is that edge pixels located on the same circle, or on circles with equal radius, have equal isophote curvature; consequently, in a randomized iterative circle detector, limiting the sampling at each V_i requires less iterations than sampling between all edge pixels without a specific criteria. Additionally, further evidence constraints can be considered in the analysis of candidate circles, discarding circles with parameters not compliant with the V_i subset under examination. Referring to Fig. 1, three principal local maxima can be found in κ (two circles have in fact equal radius), corresponding to local modes $\kappa_i, i = 1:3$. Minor local maxima can be observed, due to the discretization aliasing previously introduced, but the relative number of edge points is generally very limited, so they can be easily detected and discarded.

2.3. Iterative circle detection

Each V_i is separately processed by an iterative randomized algorithm: following the four points approach [13], four pixels (v_j, v_k, v_l, v_m) are randomly sampled on V_i at each iteration.

Download English Version:

https://daneshyari.com/en/article/530002

Download Persian Version:

https://daneshyari.com/article/530002

Daneshyari.com