A Mobile Vision System for Fast and Accurate Ellipse Detection

Michele Fornaciari, Rita Cucchiara
University of Modena and Reggio Emilia
Via Vignolese, 905, Modena, Italy

Andrea Prati
University IUAV of Venice
Santa Croce 1957, Venice, Italy

Abstract

Several papers addressed ellipse detection as a first step for several computer vision applications, but most of the proposed solutions are too slow to be applied in real time on large images or with limited hardware resources, as in the case of mobile devices. This demo is based on a novel algorithm for fast and accurate ellipse detection. The proposed algorithm relies on a careful selection of arcs which are candidate to form ellipses and on the use of Hough transform to estimate parameters in a decomposed space. The demo will show it working on a commercial smart-phone.

1. Introduction

The scientific community has recently found large interest to image/video processing on smart-phones, often called *embedded* or *mobile vision*. Possible applications range from real time object/person tracking, content-based retrieval of framed scene with markerless object recognition, face detection (and possibly recognition), blind people aid for movement, etc.

This demo represents a real-time implementation of ellipse detection capable to find very fast the ellipses in images captured by a commercial smart-phone. This can result to be the starting point for many computer vision applications, since ellipse shape is very common in nature and in hand-made objects. For instance, ellipse detection has been used in detecting wheels, road sign detection and classification [7], object segmentation for industrial applications, automatic segmentation of cells from microscope imagery [9], pupil/eye tracking [8], etc..

The proposed solution works at arc-level instead of pixel-level. Thanks to an innovative selection strategy, we choose arcs belonging to the same ellipse very quickly and very reliably. The ellipse center is estimated thanks to the property of the midpoints of parallel chords, and remaining parameters are estimated accumulating votes in a decomposed parameter space. The excellent trade-off between efficiency (in the order of 10 ms per image on a desktop PC and 40 ms on a smartphone) and accuracy makes this approach a superb candidate for implementation on mobile devices.

The importance of ellipse detection in image processing is witnessed by the large amount of works present in the literature. The estimation of the 5 parameters of the ellipse is very demanding in terms of computation and memory requirements. Most of the methods rely on the Hough Transform (HT) or its variants to estimate the parameters [2, 1].

A better characterization could be achieved using sets of connected edge pixels, i.e. *arcs*. Many approaches rely on this concept of edge following [3, 6] and share common considerations. They make an effort to build “meaningful” arcs by forming line segments and merging them according to different criteria. Then, they group the arcs as belonging to the same ellipse if they satisfy a set of constraints. Least square ellipse fitting is applied to each group to obtain ellipse parameters. These approaches usually achieve good detection accuracy but at the cost of slowness.

We present a novel algorithm for fast ellipse detection that: i) reduces the total amount of computation, evaluating only selected arc combinations, and ii) reduces the memory usage estimating the parameters via the HT in a decomposed parameter space.

2. Algorithm Description

The proposed ellipse detection algorithm first finds the subsets of *arcs*, formed by connected edge points sharing the same coarse gradient direction, of the input image that best fit an elliptic shape. Then, for each subset, it estimates the ellipse parameters applying the HT in a decomposed parameter spaces.

Like most, also the proposed method works on the edge points (and their direction) of a given image. Therefore, as baseline approach, we smooth the gray–scale image with a Gaussian filter and then apply the Canny edge detector. Among other edge detectors, Canny is well suited because, thanks to the non–maxima suppression, edges are very thin and the Sobel derivatives (needed to obtain the edge points direction) are already computed.

The resulting edge points are connected based on the edge direction. Connected points of the same class are labeled together to form the arcs. In order to remove at an early stage all those arcs that are not useful for further processing, very short arcs (noise) and straight arcs (not belonging to the curved edge of an ellipse) are discarded. Straight arcs are quickly detected thresholding the axes–ratio of the corresponding oriented bounding boxes.

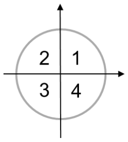


Figure 1. Classes of edge pixels.

Each arc is then classified in four classes as depicted in Fig. 2, according to both the direction and the convexity of its edge points.

An ellipse is found if at least three arcs of different class lying on the same elliptic boundary exist. A meaningful triplet of arcs must satisfy the following criteria (otherwise is discarded to save time): (1) coherent classes of adjacent arcs (i.e., with reference to Fig. 2 (4,1,2) or (1,2,3) are correct matches - counterclockwise, while (4,2,3) is not); (2) the convexity and respective position of the three arcs must be coherent with the fact that the convexity is toward the center; (3) for each triplet satisfying the above two criteria, the center of the ellipse is estimated for the two pairs and the two resulting ellipse centers must be close enough.

Given two arcs, we estimate the ellipse center relying on the following properties: *for an ellipse, the midpoints of a set of parallel chords and its center are collinear*. This property holds regardless accurate angle estimations and is thus well suited for real world images.

In order to estimate the slope of the line passing through the ellipse center and the midpoints of the chords a fast variant of the robust Theil-Sen estimator [4] has been used.

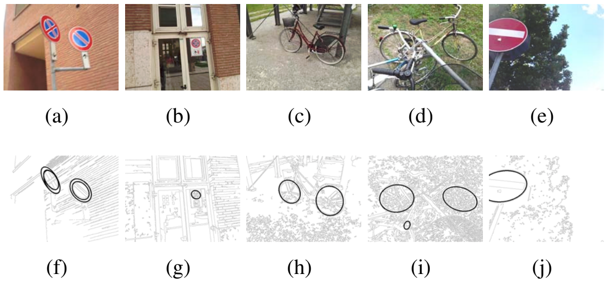
The ellipse center coordinates x_c, y_c are computed as the midpoint of the previously found ellipse centers. Similarly to [1] the ellipse parameter space is decomposed to find the semi–axes ratio and the orientation, but in two different 1D accumulators. Then, the value of the minor and major semi–axes can be easily derived.

Once all the parameters have been estimated, we assign a score to each ellipse that summarize how well the points of the three arcs fit the contour of the estimated ellipse and lower score ellipses are removed from the solution. Moreover, since there may be multiple instances of the same ellipse with slightly different parameters, we cluster the results following the criteria suggested in [5].

3. Experiments

The demo will show live results of the algorithm implemented for an Android platform on two commercial smartphones: a rather old one - Samsung Galaxy S - and a new one - Sony Xperia Z. Given the space constraints no qualitative results are reported here (but they will be shown at the demo), but some qualitative results are shown in Fig. 2:

The demo is developed on Android using OpenCV which calls the C++ function for detecting ellipses using our method via the Java Native Interface (JNI). The execution time is on average around 40 msec, depending on the complexity of the scene and the smart-phone used.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 2. Examples of results: first row reports input images, second row reports our results.

References

[1] A. S. Aguado, E. Montiel, and M. S. Nixon. On using directional information for parameter space decomposition in ellipse detection. *Pattern Recognition*, 29(3):369–381, 1996.

[2] C. Basca, M. Talos, and R. Brad. Randomized hough transform for ellipse detection with result clustering. In *Computer as a Tool, 2005. EUROCON 2005.The International Conference on*, volume 2, pages 1397 –1400, nov. 2005.

[3] L. Libuda, I. Grothues, and K.-F. Kraiss. Ellipse detection in digital image data using geometric features. In J. Braz, A. Ranchordas, H. Arajo, and J. Jorge, editors, *Advances in Computer Graphics and Computer Vision*, volume 4 of *Communications in Computer and Information Science*, pages 229–239. Springer Berlin Heidelberg, 2007.

[4] J. Matousek. Randomized optimal algorithm for slope selection. *Information Processing Letters*, pages 183–187, 1991.

[5] D. Prasad and M. Leung. Clustering of ellipses based on their distinctiveness: An aid to ellipse detection algorithms. In *Proc. of ICCSIT*, volume 8, pages 292 –297, july 2010.

[6] D. K. Prasad, M. K. Leung, and S.-Y. Cho. Edge curvature and convexity based ellipse detection method. *Pattern Recognition*, 45(9):3204 – 3221, 2012.

[7] A. Soetedjo and K. Yamada. Fast and robust traffic sign detection. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 2, pages 1341 –1346, oct. 2005.

[8] L. Swirski, A. Bulling, and N. A. Dodgson. Robust real-time pupil tracking in highly off-axis images. In C. H. Morimoto, H. O. Istance, S. N. Spencer, J. B. Mulligan, and P. Qvarfordt, editors, *ETRA*, pages 173–176. ACM, 2012.

[9] C. Teutsch, D. Berndt, E. Trostmann, and M. Weber. Realtime detection of elliptic shapes for automated object recognition and object tracking. In *Proc. SPIE 6070*, 2006.