Robust Real-time Tracking of Ellipse Arcs

J. Biber, M. Vincze, M. Ayromlou, W. Ponweiser
Institute of Flexible Automation, Vienna University of Technology
1040 Vienna, Austria, e-mail: {jb,vm,ma,wp}@infa.tuwien.ac.at

Abstract:
Ellipse arcs are typical features of man-made objects and tracking of ellipse arcs is needed in robotics applications where partial occlusion is a standard problem that should be handled. The difficulty in tracking of ellipse arcs is to robustly locate the edge and to fit the arc, that is, the visible segment of the ellipse. Robust edge detection is obtained using an improved version of the approach using Edge Projected Integration of Cues (EPIC). It is shown how EPIC highly reduces the number of outliers and that a probabilistic fitting method obtains robust ellipse estimates for subtended angles larger than a half circle. The ellipse arc tracker is able to automatically detect occlusion. From the ellipses and a related object model the pose of the part can be reconstructed and this signal can be used for motion control.

Keywords: Tracking, ellipse arc, cue integration, occlusion, real-time

1 Introduction

Many man-made objects have visible features that are circular or elliptical or contain segments thereof. Circles are seen as ellipses unless viewed directly head-on. Using the model knowledge of the circle and the image of the ellipse, five of the six possible degrees of freedom of the pose of the object can be reconstructed [4, 7]. Therefore the ellipse and ellipse arcs (or segments) are significant features. Tracking of full ellipses at real-time has been presented in [1, 9]. However, in many cases only part of the ellipse is visible, or the object contains ellipse segments. Therefore the ability to track ellipse arcs is needed.

The difficulty in tracking of ellipse arcs is to robustly locate the edge and to fit the arc, that is, the visible segment of the ellipse. Fitting the arc is difficult because a few outliers will render the process instable for small subtended angles [6]. The ellipse parameters vary substantially and pose estimation erroneous. This work is based on [1, 14], which presented a real-time ellipse tracking scheme based on tracker lines and a simple version of Edge Projected Integration of Cues (EPIC) to find the correct edgels. The rationale is that selective edge trackers and a robust method of ellipse fitting combine nicely to obtain robust behaviour and smooth reconstruction of the ellipse parameters. The above work has been extended in three ways: (1) Automatic detection of occlusion and adaptation of the tracking scheme. (2)

Improved version of EPIC to better find the correct edgel and to improve the robust (smooth) behaviour. And (3), calculating the ellipse arc for subtended angles over 180 degrees using a RANSAC [5] approach.

Section 2 will describe the tracking method. The ellipse arc tracker supervises the performance of the edge trackers and uses this to automatically handle occlusion (Section 2.4). Finally, Section 3 shows the tracking performance and how the tracker is embedded in a generic tool for tracking pose estimation, Vision for Robotics (V4R).

1.1 Related Work

The most common technique to detect ellipses is the Hough Transform (HT). The HT is fairly robust but needs a five parameter space and is therefore computationally very expensive. Variations of the basic method use multi-step approaches to reduce calculation time, e.g., a modified HT using a 2-dimensional accumulator array [15], the Randomised Hough Transform [11], or selecting only the parameters needed for the application [2]. However, frame rate cannot be obtained with these methods. Another approach uses the geometric symmetry of ellipses from an edge image [10].This approach first locates candidates of ellipse and circle centres. In a second step, the geometric symmetry is exploited to extract ellipses. The algorithm works very fast but lacks robustness if ellipses of different sizes are to be detected. The most promising approach for a real-time application is the k-RANSAC algorithm for ellipse detection [3] (based on the RANSAC idea [5]) though presently execution at frame rate is not feasible. In [14] an ellipse tracker based on a cue integration method to find edgels (EPIC) and suited for real-time execution is presented. This work is extended to handle occlusion and edgel detection is made more robust with an improvement of EPIC that does not require to set parameters.

2 Real-time Ellipse Arc Tracker

The ellipse arc tracker is implemented as a state tracker. The states are the ellipse parameters and the edgel specifications of each tracker line. The functionality of using tracker lines has been first presented in [8]. Real-time performance is reached with the following four tracking steps: (1) Place tracker lines along the circumference of the ellipse. (2) Find edgels along the tracker lines using EPIC. (3) Fit ellipse to selected edgels with a random selection scheme, and (4) Detect occlusions and rearrange placement of tracker lines.

2.1 Placing Tracker Lines

To find edgels from the elliptical object edge tracker lines are used. Each of this one-dimensional lines can be placed in the image in terms of position and orientation and allows to search for the ellipse edge. As result of this search each tracker line returns edgels which

are likely to belong to the edge searched for. How we ensure to find correct edgels with high likelihood is described in Section 2.2.

Tracking of an object feature through an image sequence means to refind this feature in the current image using the knowledge of its pose within the previous image. To handle changes in size, position and orientation of the ellipse arc, we arrange the tracker lines along the circumference of the ellipse arc determined at the previous tracking step. All tracker lines are oriented orthogonal to the ellipse to enable handling of maximum changes of its pose. A configuration of tracker lines used for tracking a circle segment is shown in Fig. 1.

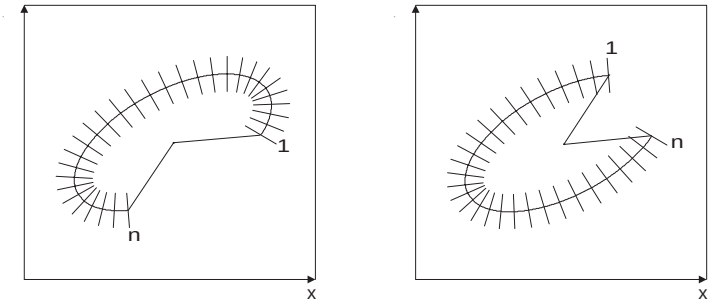

Figure 1: Arrangement of tracker lines used for tracking a circle segment. The tracker lines are distributed orthogonal to the ellipse estimated, to enable searching for the edge at the next tracking step. For tracking of circle segments we try to place the tracker lines only at the ellipse arc seen as a projection of the circle segment onto the image plane.

For tracking of circle segments of known size (subtended angel of the circle segment), it is useful to arrange the tracker lines only at the ellipse arc seen as a projection of the circle segment onto the image plane. Fig. 1 shows how the tracker lines are placed for different viewing anlges. The difficulty is to arrange the tracker lines only at the ellipse arc, although the subtended angle of the ellipse arc varies. This is done by distributing the centre of the tracker lines in the plane of the circle segment. The tracker lines are arranged with constant angle between each tracker line in the known range of the circle segment. Then the position of each tracker line in the image is determined by projecting the points from the circle plane onto the image plane. That ensures that the tracker lines are distributed in the right range given by the angle of the circle segment (see also Fig. 6). In all examples 30 tracker lines are placed along the ellipse arc.

2.2 Edge Projected Integration of Cues (EPIC)

The basic idea of EPIC is to integrate the values of region cues such as intensity, color, texture, or optical flow, at the nearest edgel. Integration also uses model knowledge to select which side of the edge belongs to the object and which side is background. The rationale of using EPIC to seek edgels is to reduce the number of edgels found in the window to the "good" edgels, that is, the edgels that indicate with high likelihood the very feature tracked in the last cycle. The result is a more robust fit of the feature geometry and an increase in effectiveness

rendering processing fast (see the next section).

The EPIC procedure is as follows. Cue values from the last tracking step are stored as mean and standard deviation values. New edgels are found in each line of the tracking window. Between the edgels in each line new cue values are calculated. The new cue value c_{side} for each interval is calculated from the maximum value of the histogram of the cue values. Experiments with median values gave results of similar robustness, however, the computations needed to calculate the median increase more than linearly with larger intervals.

The new cue values are weighted using the mean and standard deviation values from the last tracking cycle. The cue value of the present tracking cycle c^t_{side}, where the superscript denotes the time step for obtaining the values, is used to find the weighted values $C_{side,i}$ as follows

$$C_{side,i} \quad = \quad \exp\left(-\frac{\left(\mu^{t-1} - c^t_{side}\right)^2}{2\sigma^{t-1}}\right) \tag{1}$$

where the cue values from the last tracking cycle $t-1$ have been stored as the mean value μ^{t-1} and the standard deviation value σ^{t-1}. The index $side$ of the above values refers to the two possible sides of an edgel, $left$ and $right$. Knowledge of the object is used to select only the side that belongs to the target object while the background side is not regarded. This knowledge is found from the object model. It is used to set weights w_{side}. If the model indicates an object, the respective weight w_{side} is 1, otherwise it is 0. The likelihood l_k that an edgel k is a "good" edgel is evaluated to

$$l_k = \frac{1}{W}\sum_{i=1}^{n} w_{left}C_{left,i} + w_{right}C_{right,i} \quad \text{with} \quad W = \sum_{i=1}^{n} w_{left} + w_{right} \tag{2}$$

where $i = 1, ..,$ number of cues. The advantage of this scheme is that each value $C_{side,i}$ is calculated using the Gaussian distance measure of Eq. 1, which eliminates the need to set threshold parameters. To achieve adaptation to slowly varying lighting changes in the environment, the mean and standard deviation values are adapted after each tracking step. Using the result of feature detection, new values μ^t and σ^t are calculated from the c^t_{side} values of all the good edgels that voted for the finally selected edge (please refer to the next two Sections for details).

The EPIC scheme is very effective because the features contain attributes that give indications to select the cues (intensity, color, texture, ...) of the object. Based on the localization of edgels, the cues can be easily integrated and the list of cues given above can be easily increased with other cues. The principal idea is to use these cues to limit the selection of edgels. For example, trials of incorporating color added robustness to distinguish the correct edges from shadows and highlights. This limitation to good edgels renders the next step of fitting the feature geometry to the edgels very efficient.

2.3 Fitting Ellipse Geometry

After finding edgels with EPIC, the geometry of the target feature is fitted to the data points. Figure 2 shows two cases for an ellipse using the RANSAC principle [5] first presented for ellipse tracking in [1]. It has been adopted for ellipse arcs by using a better distance measure.

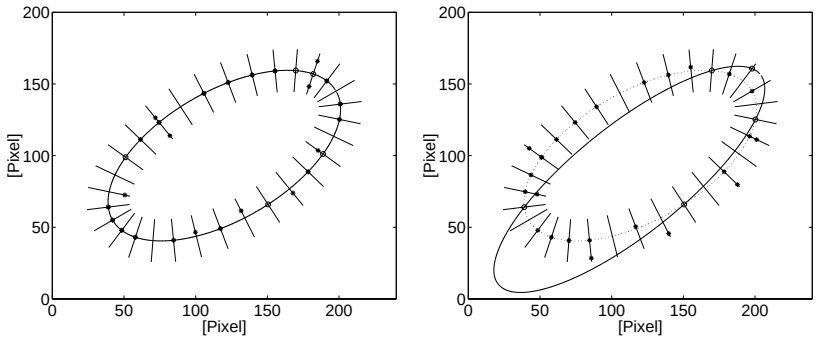

Figure 2: Selecting five good (left) and four good and one bad (right) edgels to fit the ellipse.

For the voting procedure the distance between each edgel and an ellipse hypothesis has to be calculated for every trial of the RANSAC algorithm. Hence, it is important to use a very efficient error measure that approximates that distance well enough. The gradient-weighted algebraic distance (analysed in [12]) is used, which is a very good approximation for edgels which lie close to the ellipse as can be expected in tracking. It allows execution at real-time. Stable ellipse fits are reached also for small ellipse arcs using the number of edgels voting and the sum of squared distances of the voting edgels as quality measures.

2.4 Handling Occlusion

As it is shown in the previous Section it is essential for robust ellipse estimates that most of the tracker lines find edgels from the ellipse edge. That means, that in case of ellipses with occlusion or ellipse segments, we have to try to place as many tracker lines as possible at the visible part of the ellipse. This is done by relocating the tracker lines after the ellipse is estimated, using information from the RANSAC voting scheme (Section 2.3).

For handling of occlusion, we have to find out at which area the elliptical edge is visible and where not. The tracker lines are used to get this information. After applying the RANSAC voting scheme to the edgels found by the use of EPIC (Section 2.2), each tracker line has the information if an edgel has voted for the ellipse found. Voting means, that one edgel from a certain tracker line lies close enough to the ellipse estimated. For relocation of the tracker lines we assume that the ellipse edge is visible at the area of a tracker line, if the tracker line includes a voting edgel, and will be called a "good tracker line" (otherwise a "bad tracker line").

If there is a number of neighbouring bad tracker lines this region is recognised as occluded.

Fig. 3 demonstrates the placement of tracker lines for the full ellipse and after occlusion is recognised. The occluded segment is not tracked anymore. At each end of the ellipse arc one tracker line is added to detect the recovery of the occlusion.

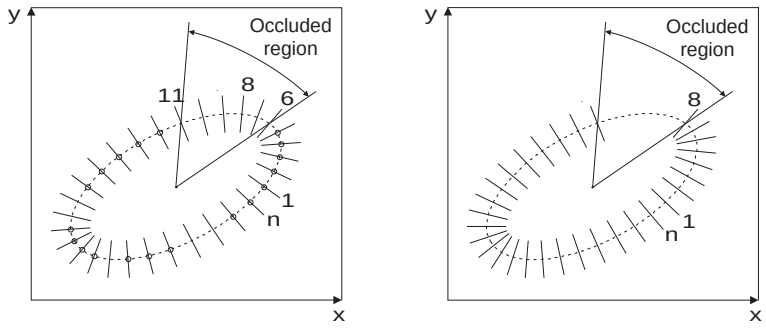

Figure 3: Principle of recognising and handling occlusion: Edgels voting for the ellipse, are marked with a small circle. If there is a large number of neighbouring tracker lines which have not voted for the ellipse found, an area of occlusion is recognised. Tracker lines are not placed on this occluded area at the next tracking step. Only at both ends of the arc one tracker line is placed onto the occluded area, to enable recognition of changes in occlusion or orientation of the arc.

To handle changes in occlusion we observe the tracker lines at both ends of the ellipse arc. Because we also want to recognise decreasing occlusion, the tracker lines are always relocated in a way, that at both ends one tracker line is placed at the occluded region. In case of more than one occluded region the biggest one should be excluded from tracking, that means no tracker lines should be placed on that region. Fig. 4 shows the case of an occlusion excluded from tracking (text marker), and a second increasing occlusion (white paper).

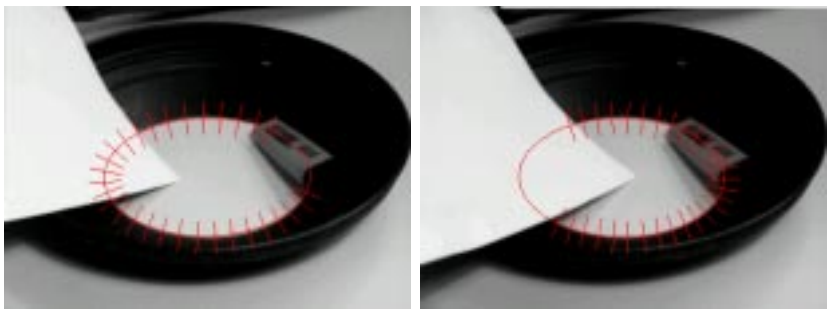

Figure 4: Handling of more than one occluded region: If during tracking of an ellipse arc an other part of the tracked ellipse arc is occluded too, and this occlusion becomes larger than the not tracked part, this occlusion is excluded from tracking at the next tracking step instead.

3 Experiments

Experiments are conducted using the intensity component of a color images taken at frame rate. The method is implemented on a Pentium PC with 400MHz. Fig. 5 demonstrates tracking of a full ellipse and automatic detection of occlusion. Please also observe how the reflection of the lamp shade changes but the tracker lines adapt and tracking is robust. Fig. 6

shows the tracking of an ellipse segment over cluttered background and with significant changes of the reflectivity of the tracked object, which are handled by EPIC.

Figure 5: Placing the tracker in case of occlusion. At both ends one tracker line is placed beyond the end to enable changes in occlusion or orientation of the arc.

Figure 6: Samples of tracking an ellipse arc. The ellipse tracked is displayed in yellow.

The ellipse tracker is part of the model-based tracking tool "Vision for Robotics" (V4R). V4R enables tracking of lines, junctions and regions and now ellipse arcs. V4R also provides the functions to project features into the image and to estimate the pose from the image data. Using the ellipse information in the image, the 3D centre point of the model circle and the surface normal of the ellipse plane are reconstructed. This information is then used to estimate the pose of the object. Pose estimation also uses line and junction information, detects outliers in case of a failure of tracking, and finally reports a least squares fit for the object pose. Fig. 7 shows the tracking of a monitor base. The estimated pose is re-projected into the image for better evaluation of the result obtained.

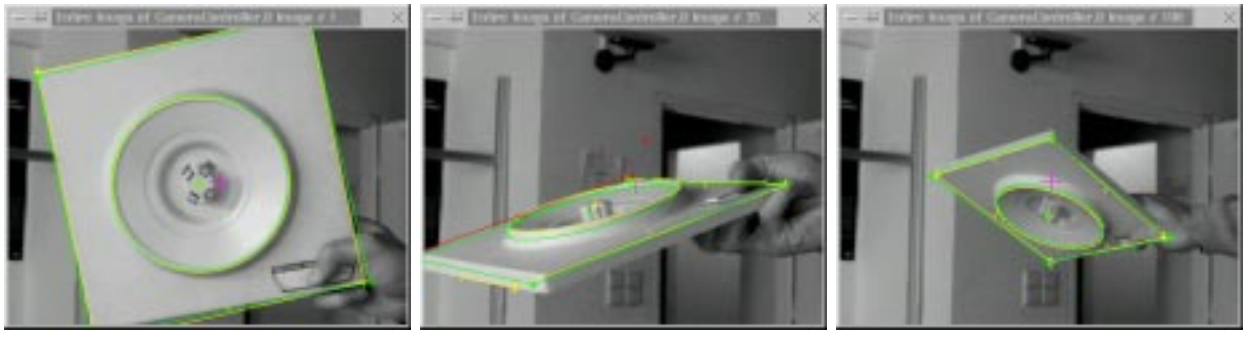

Figure 7: Tracking the base of a monitor. The pose estimated from the lines and the ellipse is re-projected into the image (green/dark). Tracking results are displayed in yellow/bright color.

4 Conclusion

An ellipse arc tracker has been presented that uses cue integration with EPIC to obtain robustness. The results show significant improvements over cluttered background with continuous adaptation to changing intensity of light reflected by the object. The method can track two elipses in field time ($20ms$) and is integrated into the model-based vision tool V4R.

Future work concerns ellipse arcs that subtend an angle less than 180 degrees, where ellipse fitting becomes unstable and other methods need to be investigated. However, the most significant need is to find the ellipses automatically to enable automatic initialisation of tracking.

Acknowledgements

This work is partly supported by the Austrian Science Foundation (FWF) under grant P13167-MAT and EU GROWTH Project GRD1-1999-10693 FlexPaint.

References

[1] Ayromlou, M., Vincze, M., Kubinger, W., Zillich, M.: "Robust Tracking of Ellipses at Frame Rate"; OAGM Workshop on Pattern Recognition, 155-164, Steyr, Austria, May 27-28, 1999.

[2] Bennett, N., Burridge, R., Saito,N., "A Method to Detect and Characterize Ellipses Using the Hough Transform", IEEE Transactions on Pattern Analysis and Machine Intelligence 21(7), pp. 652-657, 1999.

[3] Cheng, Y.S., Lee, S.C.: "A new method for quadrative curve detection using K-RANSAC with accelereration techniques", Pattern Recognition 28(5), pp. 663-682, 1995.

[4] Ferri, M., Mangili, F., Viano, G.: "Projective Pose Estimation of Linear and Quadratic Primitives in Monocular Computer Vision", CVGIP: Image Understanding 58(1), pp. 66-84, 1993.

[5] Fischler, M.A., Bolles, R.C.: "Random Sample Consensus: A Paradigm for Model Fitting", Communications of the ACM Vol.24(6), pp.381-395, 1981.

[6] Fitzgibbon, A., Pilu, M., Fisher, R.B.: "Direct Least Square Fitting of Ellipses", IEEE PAMI 21(5), 476-480, 1999.

[7] Forsyth, D., Mundy, J.L., Zisserman, A., Coelho, C., Heller, A., Rothwell, C.: "Invariant Descriptors for 3-D Object Recognition and Pose", IEEE PAMI 13(10), pp. 971-991, 1991.

[8] Hager, G., Toyama, K.: "The XVision-System: A Portable Substrate for Real-Time Vision Applications", Computer Vision and Image Understanding 69(1), pp. 23-37, 1998.

[9] Jrg, S., Langwald, J., et.al.: "Flexible Robot-Assembly using a Multi-Sensory Approach", IEEE ICRA, pp. 3687-3694, 2000.

[10] Lei, Y., Wong, K.C., "Ellipse detection based on symmetry", Pattern Recognition Letters 20(1), pp. 41-47, 1999.

[11] McLaughlin, R.A., "Randomized Hough transform - improved ellipse detection with comparison", Pattern Recognition Letters 19(3-4), pp. 299-305, 1998.

[12] Rosin, P.L.: "Analysing Error of Fit Functions for Ellipses", Pattern recognition Letters 17, 1461-1470, 1996.

[13] Rosin, P.L.: "Further Five-Point Fit Ellipse Fitting", Graphical Models and Image Processing 61, 245-259, 1999.

[14] Vincze, M.: "Robust Tracking of Ellipses at Frame Rate"; Pattern Recognition 34(2), 487-498, 2001.

[15] Yip, R.K.K., Tam, P.K.S., Leung, D.N.K.: "Modification of Hough transform for circles and ellipse detection using a 2-dimensional array", Pattern Recognition 25(9), pp. 1007-1022, 1992.