Memoirs of the Faculty of Engineering, Okayama University, Vol. 36, No. 1, pp. 107–116, December 2001

Automatic Detection of Circular Objects by Ellipse Growing

Kenichi KANATANI*
Department of Information Technology
Okayama University
Okayama 700-8530 Japan

Naoya OHTA
Department of Computer Science
Gunma University
Kiryu, Gunma 376-8515 Japan

(Received November 9, 2001)

We present a new method for automatically detecting circular objects in images: we detect an osculating circle to an elliptic arc using a Hough transform, iteratively deforming it into an ellipse, removing outlier pixels, and searching for a separate edge. The voting space is restricted to one and two dimensions for efficiency, and special weighting schemes are introduced to enhance the accuracy. We demonstrate the effectiveness of our method using real images. Finally, we apply our method to the calibration of a turntable for 3-D object shape reconstruction.

1. INTRODUCTION

Since a circle in the scene is projected into an ellipse in the image, circular objects in the scene can be detected by finding ellipses in the image. Already, many techniques have been proposed for detecting ellipses in images. However, not all of them are intended to detect circular objects. Their aims are classified roughly as follows:

1. To identify elliptic regions for object recognition or particle measurement in medical and biological applications.
2. To obtain a line drawing by segmenting edge chains and fitting lines to linear parts, and ellipses to curved parts [7, 16].
3. To detect a circular object in the scene for accurately computing the 3-D position of the object or the camera relative to it [19].

While most of the past studies were intended for the first aim, this paper focuses on the third: we seek a good algorithm for detecting ellipses to be used for quantitative measurement in robotics applications. Hence, *accuracy* is our most important concern.

The basic principle of ellipse detection is the Hough transform: an ellipse is specified by five parameters, so we vote in a quantized 5-dimensional Hough space the parameters of the ellipses that could pass through each edge pixel and pick out the value that wins the maximum number of votes. However, direct computation of this would require a long computation time and a memory space proportional to the fifth power of the quantization levels, so various techniques have been proposed to reduce the computation time and memory space. For example, the computation is divided into a cascade: voting is done in a 2-dimensional Hough space with the remaining three parameters fixed, and this is repeated for all quantized values of the three parameters. In this process, one can introduce various constraints for pruning unnecessary search, hierarchically change the resolution of the image and the Hough space, and do random sampling instead of exhaustive search, a variant of this being the genetic algorithm. Instead of using edge pixels alone, one can also use complex primitives and extra information such as point pairs, triplets, edge segments, and their orientations. See, e.g., [2, 4, 10, 11, 12, 15, 17, 18, 21, 22, 23, 24].

*E-mail kanatani@suri.it.okayama-u.ac.jp

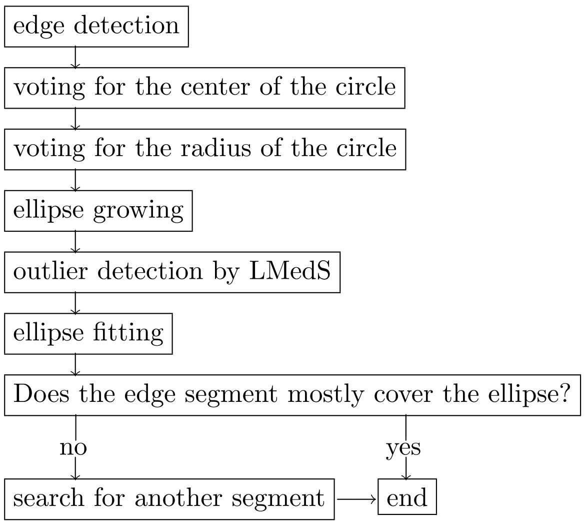

edge detection
voting for the center of the circle
voting for the radius of the circle
ellipse growing
outlier detection by LMedS
ellipse fitting
Does the edge segment mostly cover the ellipse?
no
yes
search for another segment
end

Figure 1: Flow chart of the procedure.

Another approach is to start with digital curves consisting of connected edge pixels. They are segmented into linear and curved parts, lines being fitted to the linear parts, and ellipses to the curved parts [7, 16]. The segmentation is based on the digital curvature, the residual of fitting, and miscellaneous heuristics, but the accuracy of the fit depends very much on segmentation. This may not be a great problem for visual applications but is crucial for precise measurement in robotics applications, with which we are concerned.

Thus, the Hough transform alone is not efficient enough, while the use of digital curves alone does not warrant sufficient accuracy. In this paper, we integrate these two: we detect a circle osculating to an edge segment by the Hough transform, find edge pixels tangent to it, refit an ellipse to the detected pixels, and repeat this process. We call these iterations *ellipse growing*.

A closely related method to ours was proposed by Asayama and Shiono [2], who detected via the 5-dimensional Hough transform the two circles osculating to an ellipse from both sides, from which they computed the parameters of the ellipse. This method works only when the entire ellipse is visible without occlusion, and a long computation time and a large memory space are required for the Hough transform. In contrast, our method is able to fit an ellipse to a partially occluded image of a circular object very efficiently with high precision.

The Hough transform we propose for detecting an osculating circle is made efficient by using, instead of the standard 3-dimensional Hough space, a 2-dimensional array for the center and a 1-dimensional array for the radius. We also introduce special weight functions for enhancing the vote peaks. We then iteratively deform the osculating circle into an ellipse. We also remove outlier edge pixels by LMedS, and search for another edge segment to combine. Fig. 1 shows the flow chart of the procedure.

Our system is not intended to have universal merits, whose pursuit is rather unrealistic. We have in mind robotics environments where

1. only a small number of circular objects exist in the field of view, and
2. the circular objects we are looking at have a clearly visible size in the image.

We do not consider such complicated scenes as aggregates of circular and elliptic particles or cells. Our method is strengthened by specifically exploiting this situation.

In the following, we describe each step separately and confirm the effectiveness of our method using real images. Finally, we apply our method to the calibration of a turntable for 3-D object shape reconstruction.

2. PROCEDURE

2.1 Edge Detection

In order to obtain long edges with few branches, we combined the zero crossing and the Sobel operator, cutting out from the zero-crossing edges those parts with high Sobel values. The resulting 8-connected edges were segmented

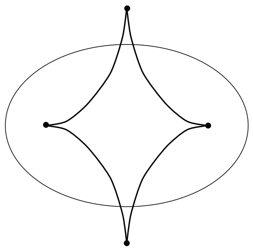


Figure 2: The evolute of an ellipse and its singularities.

at branch points. Loops were cut open at one point. Among the resulting edge segments, short ones (less than 60 pixels in our experiment) were discarded, and the remaining ones were registered in an edge segment list, from which we detected osculating circles.

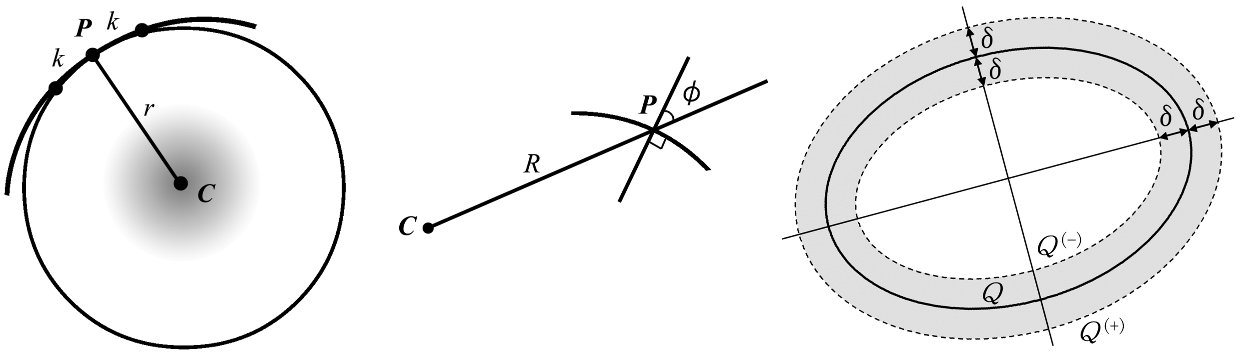
2.2 Voting for the Center

For a circular arc, its center is easily obtained by voting along all lines normal to the arc [8]. However, the lines normal to an ellipse envelop a star-like curve called an *asteroid* with four cusp vertices (Fig. 2). This is the *evolute* of the ellipse (of which the ellipse is the *involute*), also known as the *caustic* [1]. An osculating circle whose center is on the evolute is called the *circle of curvature*, which generally has a second degree contact to the curve. But if its center is at one of the four singularities of the evolute, the degree of contact becomes three [3].

It follows that if we vote along all lines normal to an elliptic arc, we obtain ridges along the evolute and peaks at its singularities. But the peaks are less conspicuous than for a circular arc. So, we directly vote on the evolute as follows.

For each pixel P on the edge segment, we compute the center C and the radius r of the circle that passes through P and two points on the segment k pixels away from it on both sides (Fig. 3(a)). Then, we vote around C with Gaussian diffusion of standard deviation γr in all directions (in our experiment, we set $k = 30$ and $\gamma = 1/10$; we did not vote at the k end points of the segment). We also weight each vote by $1/\sqrt{r}$ to give preference to edge segments of higher curvature if the length is the same and to edge segments of larger radii if the central angle is the same.

We do this for all the registered edge segments[1], detect peaks of the votes, and take as candidates the four pixels that win the largest numbers of votes (see Sec. 3 for real image examples). This procedure works well if the osculating circle is shared by multiple edge segments.



(a) (b) (c)

Figure 3: (a) Voting around the center of the circle passing through P and the two points away from it by k pixels on both sides. (b) A pixel away from the estimated center C by distance R votes the value R with weight $e^{-R^2/2s^2}\cos\phi$. (c) Fitting an ellipse to the longest edge segment inside the region within distance δ from the estimated ellipse.

[1] In our experiment, we voted every three pixels for efficiency. We also introduced various approximations such as limiting the voting region to a parallelogram with one edge orthogonal to the edge segment and the other parallel to one of the coordinate axes.

2.3 Voting for the Radius

For a circular arc, its radius is easily estimated by voting the distances of all edge points from the estimated center [8]. However, the number of false peaks increases for an elliptic arc. So, we adopt the following strategy.

We vote the distance R of each edge pixel P from the estimated center C with Gaussian diffusion of standard deviation 1 on both sides and the weight $e^{-R^2/2s^2}\cos\phi$, where ϕ is the angle between the line CP and the normal direction to P, which is computed from the output of the Sobel operator.

The exponential part[2] $e^{-R^2/2s^2}$ limits the voting to be approximately within distance s from C (in our experiment, we set s to be 1/4 of the image size). By the cosine part $\cos\phi$, edge segments making large angles from the circumference direction are given small weights. In particular, *the weight is 0 if the segment is oriented along the radius direction*, and edge segments with opposite gray-level gradients have weights of *opposite signs*, so the contributions from closed loops that do not include P inside are mutually canceled to *zero*, enhancing only the contributions from the edge segment to which the circle osculates (see Sec. 3 for real image examples).

After the voting, we choose the value that wins the maximum number of votes. We do this for each of the four candidates for the center and pick out the one that wins the largest number of votes for the radius. This procedure works well if the osculating circle is shared by multiple edge segments.

2.4 Ellipse Growing

Regarding the detected osculating circle as the initial ellipse, we make it grow into an ellipse that fits to the segment better. For the current ellipse, we define a ring region around it by expanding and contracting the ellipse by δ pixels along its major and minor axes (Fig. 3c).

Quadratic curves (including ellipses) have equations of the form

$$Ax^2 + 2Bxy + Cy^2 + 2f(Dx + Ey) + f^2F = 0, \tag{1}$$

where f is an arbitrarily fixed constant (e.g., the image size). This equation describes an ellipse when

$$AC - B^2 > 0. \tag{2}$$

If we define the vector $\boldsymbol{x}$ and the matrix $\boldsymbol{Q}$

$$\boldsymbol{x} = \begin{pmatrix} x/f \\ y/f \\ 1 \end{pmatrix}, \qquad \boldsymbol{Q} = \begin{pmatrix} A & B & D \\ B & C & E \\ D & E & F \end{pmatrix}, \tag{3}$$

eq. (1) is rewritten in the form

$$(\boldsymbol{x}, \boldsymbol{Q}\boldsymbol{x}) = 0, \tag{4}$$

where $(\boldsymbol{a}, \boldsymbol{b})$ denotes the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The matrices $\{\boldsymbol{Q}^{(+)}, \boldsymbol{Q}^{(-)}\}$ of the expanded and contracted ellipses are computed as follows:

1. Compute the matrix $\boldsymbol{S}$, the vector $\boldsymbol{c}$, and the scalar c as follows:

$$\boldsymbol{S} = \begin{pmatrix} A & B \\ B & C \end{pmatrix}, \qquad \boldsymbol{c} = \begin{pmatrix} D \\ E \end{pmatrix}, \qquad c = (\boldsymbol{c}, \boldsymbol{S}^{-1}\boldsymbol{c}) - F. \tag{5}$$

2. Compute the eigenvalues $\{\lambda_1, \lambda_2\}$ of $\boldsymbol{S}$ and the corresponding unit eigenvectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2\}$.

3. Compute $\lambda_1^{(\pm)}$ and $\lambda_2^{(\pm)}$ as follows:

$$\lambda_1^{(\pm)} = \frac{c}{(\sqrt{c/\lambda_1} \pm f\delta)^2}, \qquad \lambda_2^{(\pm)} = \frac{c}{(\sqrt{c/\lambda_2} \pm f\delta)^2}. \tag{6}$$

[2] We used a polynomial approximation of e^{-x^2} for efficiency.

4. Compute the matrices $\boldsymbol{S}^{(+)}$ and $\boldsymbol{S}^{(-)}$ by

$$\boldsymbol{S}^{(\pm)} = \boldsymbol{U} \begin{pmatrix} \lambda_1^{(\pm)} & \\ & \lambda_2^{(\pm)} \end{pmatrix} \boldsymbol{U}^\top, \tag{7}$$

where $\boldsymbol{U}$ is the 2×2 matrix consisting of vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ as its columns in that order.

5. The matrices $\{\boldsymbol{Q}^{(+)}, \boldsymbol{Q}^{(-)}\}$ are given by

$$\boldsymbol{Q}^{(\pm)} = \begin{pmatrix} \boldsymbol{S}^{(\pm)} & \boldsymbol{S}^{(\pm)}\boldsymbol{S}^{-1}\boldsymbol{c} \\ (\boldsymbol{S}^{(\pm)}\boldsymbol{S}^{-1}\boldsymbol{c})^\top & F + (\boldsymbol{c}, \boldsymbol{S}^{-1}(\boldsymbol{S}^{(\pm)} - \boldsymbol{S})\boldsymbol{S}^{-1}\boldsymbol{c}) \end{pmatrix}. \tag{8}$$

Using $\{\boldsymbol{Q}^{(+)}, \boldsymbol{Q}^{(-)}\}$, we can specify the ring region around the ellipse of eq. (4) as follows:

$$(\boldsymbol{x}, \boldsymbol{Q}^{(+)}\boldsymbol{x})(\boldsymbol{x}, \boldsymbol{Q}^{(-)}\boldsymbol{x}) < 0. \tag{9}$$

From among the registered edge segments, we choose the one that has the largest number of consecutive pixels inside this region and fit an ellipse to it. Many algorithms have been proposed for ellipse fitting with high precision [5, 6, 13, 14]. Here, we used a publicly available program package of a method called *renormalization*[3]. Around the fitted ellipse, we recompute a ring region, to which we apply the same procedure and repeat this to convergence.

The renormalization fits a general quadratic curve of the form of eq. (1), so it sometimes fits a hyperbola or a parabola when the segment is very short due to occlusion. In our experiment, we initially set $\delta = 4$ (pixels) and incremented it one by one if eq. (2) was not satisfied. If δ reached 10, we returned the initial circle and went on to the next process (see Sec. 3 for a real image example).

2.5 Outlier Removal

Even if the ellipse growing converges, all the edge pixels to which the ellipse is fitted need not be part of an ellipse. Since our aim is precise measurement in robotics applications, we need to distinguish elliptic parts from non-elliptic parts correctly. To this end, we remove "outliers[4]" by *LMedS* [20].

Let $\{\boldsymbol{x}_\alpha\}$, $\alpha = 1, ..., N$, be the vector representations (see eq. (3)) of the pixels to fit an ellipse. If all the pixels are "inliers " (i.e., if they constitute an elliptic segment), an ellipse is optimally fitted by minimizing the sum of the squared *geometric distances* to these pixels. If higher order terms are ignored, this amounts to minimizing the following function [5, 6, 13, 14]:

$$J = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{x}, \boldsymbol{Q}\boldsymbol{x})^2}{\|\boldsymbol{P}_k\boldsymbol{Q}\boldsymbol{x}\|^2}. \tag{10}$$

Here, we define $\boldsymbol{P}_k = \mathrm{diag}(1, 1, 0)$ (the diagonal matrix with 1, 1, and 0 as the diagonal elements in that order).

In the presence of outlier pixels, the LMedS procedure goes as follows [20]. We initialize the matrix $\boldsymbol{Q}_m = \boldsymbol{O}$ and the scalar $S_m = \infty$ and repeat the following computation until it converges:

1. Randomly choose five points from $\{\boldsymbol{x}_\alpha\}$.
2. Compute an ellipse $\boldsymbol{Q}$ that passes through them.
3. Compute the following median:

$$S = \mathrm{med}_{\alpha=1}^{N} \frac{(\boldsymbol{x}_\alpha, \boldsymbol{Q}\boldsymbol{x}_\alpha)^2}{\|\boldsymbol{P}_k\boldsymbol{Q}\boldsymbol{x}_\alpha\|^2}. \tag{11}$$

4. If $S < S_m$, update $\boldsymbol{Q}_m \leftarrow \boldsymbol{Q}$ and $S_m \leftarrow S$.

After convergence, we judge those points $\boldsymbol{x}_\alpha$ that satisfy the following condition[5] as inliers:

$$\frac{(\boldsymbol{x}_\alpha, \boldsymbol{Q}\boldsymbol{x}_\alpha)^2}{\|\boldsymbol{P}_k\boldsymbol{Q}\boldsymbol{x}_\alpha\|^2} < 10S_m. \tag{12}$$

In our experiment, we stopped the iterations if we have $S \geq S_m$ for 10 consecutive times. If hyperbolas or parabolas are fitted in the course of the iterations, they are ignored. Finally, we fit an ellipse $\boldsymbol{Q}$ to the detected inliers and cut out the longest segment e that covers all the inliers.

[3] `http://www.ail.cs.gunma-u.ac.jp/Labo/programs-e.html`

[4] In the statistical literature, outliers are the data generated by unmodeled random fluctuations [20]. Here, they are contiguous pixels on non-elliptic edge segments. In other words, "outlier removal" actually means "segmentation" of elliptic parts.

[5] We estimated the data standard deviation σ using the formula $\hat{\sigma} \approx 1.4826\sqrt{S_m}$ given in [20] and set the threshold to $(2.13\hat{\sigma})^2$.

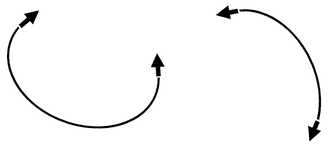


Figure 4: Judging if the edge segment covers more than the half of the ellipse.

2.6 Search for Another Segment

In order to cope with ellipses consisting of multiple segments, we do the following search.

First, we check if the detected segment e covers more than half of the entire ellipse. This is done by checking if the end parts of e are in converging orientations (Fig. 4). If so, we stop and return the ellipse $\boldsymbol{Q}$ we have obtained. Otherwise, we expand $\boldsymbol{Q}$ by $1+\gamma$. This is done by replacing eqs. (6) by

$$\lambda_1^{(+)} = \frac{\lambda_1}{(1+\gamma)^2}, \qquad \lambda_2^{(+)} = \frac{\lambda_2}{(1+\gamma)^2}. \tag{13}$$

(We let $\gamma = 1.2$). We randomly choose a segment e' that is within the expanded ellipse and is not too short (we ignored those of less than 20 pixels). Randomly choosing four pixels from e and one pixel from e', we fit an ellipse by LMedS as described earlier. We repeat this until the sum of the median for e and the median for e' converges (we stopped if no update occurred 100 consecutive times). We return the initial $\boldsymbol{Q}$ if the resulting sum of medians is larger than four times the initial median for e. Otherwise, we detect inliers by applying eq. (12) to e and e' separately and fit an ellipse to the detected inliers.

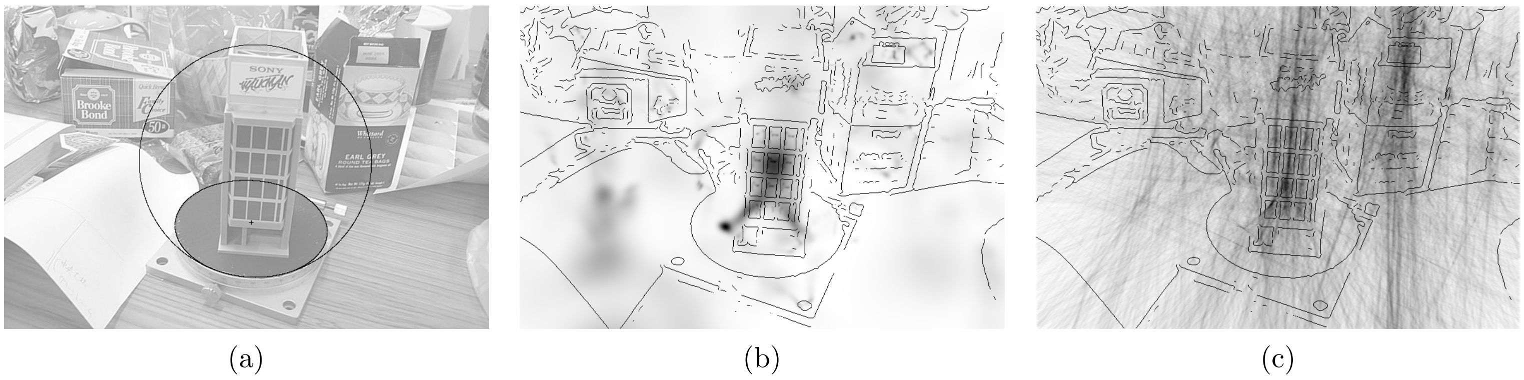
The procedure described so far is for detecting one ellipse. Multiple ellipses can be detected by removing the edge segments of already detected ellipses and repeating the same procedure.

3. REAL IMAGE EXAMPLES

Fig. 5(a) shows the detected osculating circle and the final ellipse superimposed on the original image. Fig. 5(b) is the initial edge image, on which the votes for possible centers of osculating circles are superimposed in gray levels. For comparison, Fig. 5(c) shows the result obtained by the standard Hough transform: we vote along the normal line to each edge pixel with Gaussian diffusion of one-pixel standard deviation on both sides [8]. As we can see, our scheme concentrates the votes along the evolute of the ellipse, in particular at its singularities.

Fig. 6(a) shows the histogram of the absolute values (recall that we vote positive and negative values in our scheme) of the votes for the radius from the center detected in Fig. 5(b). For comparison, Fig. 6(b) shows the histogram obtained by the standard Hough transform: the distance from the estimated center to each edge pixel is voted with Gaussian diffusion of one-pixel standard deviation [8]. We adjusted the scale of the two histograms so that the total number of votes is equal for both. As we can see, our scheme reduces the contributions from clustered edges to almost zero because of the cancellation of the signs, enhancing only the contributions from isolated arcs.

Fig. 7(a) shows an edge image of a partially occluded ellipse. For this image, a hyperbola was fitted in the ring region along the initial circle even for $\delta = 10$. So, we fitted an ellipse by LMedS and went on to search for another



(a) (b) (c)

Figure 5: Estimation of the center of the osculating circle. (a) The detected osculating circle and the fitted ellipse. (b) Voting for the center. (c) Standard Hough transform.

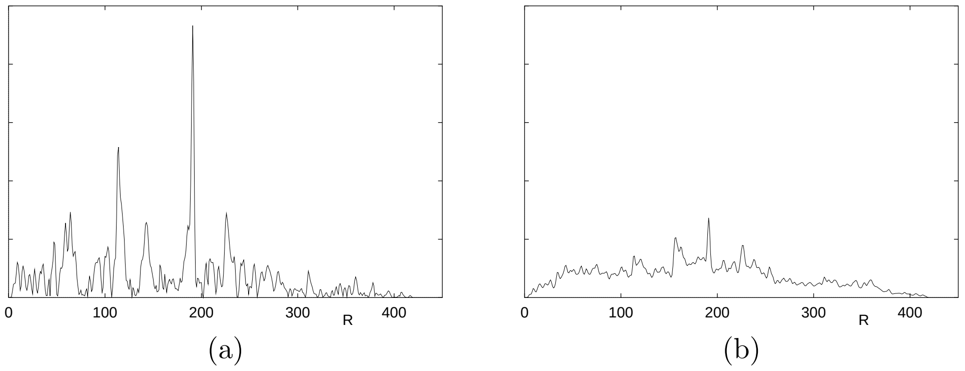


Figure 6: Voting for the radius of the osculating circle. (a) Proposed method. (b) Standard Hough transform.

segment. Fig. 8(b) shows 1. the initial circle, 2. the ellipse after the ellipse growing, 3. the ellipse fitted by LMedS, and 4. the ellipse fitted together with another segment detected. Thus, partially occluded ellipses can be detected by combining separate arcs.

Fig. 8 shows various examples, showing the edge images (upper rows) and the initial circles and the final ellipses superimposed on the original images (lower rows). In all cases, one or two iterations of the ellipse growing produced almost satisfactory shapes, and four or five iterations were sufficient for convergence. As we see, we can obtain good fits even when the initial segments are very short due to occlusion. For 500 × 333-pixel images, the average computation time was 18 seconds, most of which was spent on the preprocess for the edge segment list. We used Pentium III for the CPU and Linux for the OS. At present, each step is programmed separately. We expect that the computation time can be reduced significantly by integrating all the steps and refining the program further.

4. TURNTABLE CALIBRATION

As stated in the Introduction, our method is intended for 3-D reconstruction of objects. As a realistic example, we show an application to the calibration of a turntable.

One of the widely used methods for 3-D shape reconstruction is to place an object on a rotating turntable and take its images. Since this is equivalent to rotating the camera around the object and taking its images, the object shape can be reconstructed by the standard triangulation. To do this, however, we need to know in advance the position of the turntable relative to the camera. It can be computed from the image of the turntable if the size of the turntable is known [9].

We assume that the camera is calibrated and let the constant f in eqs. (1) and (3) be the focal length (in pixels) of the camera. In the image, we define an xy coordinate system such that the origin is at the principal point (the point through which the optical axis passes). Taking the x-axis pointing up and the y-axis pointing right, we define the z-axis in the depth orientation. Let r be the radius of the turntable, and $\boldsymbol{Q}$ its observed image. In order to remove the scale indeterminacy of $\boldsymbol{Q}$, we normalize it to $\det \boldsymbol{Q} = -1$ (if $\det \boldsymbol{Q} = 0$, we have non-elliptic figures such as two lines, one degenerate line, one point, or an empty set [9]).

Let λ_1, λ_2, and λ_3 be the eigenvalues of $\boldsymbol{Q}$, and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ the orthonormal system of the corresponding

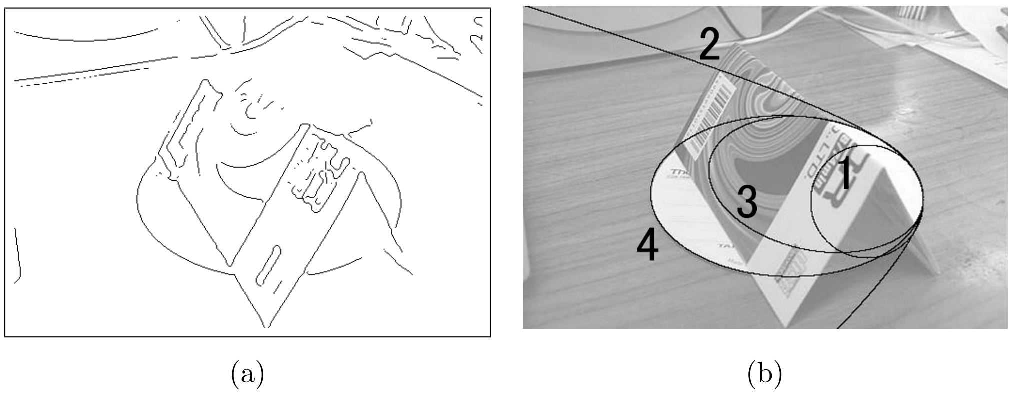


Figure 7: (a) The edge image used. (b) 1. the initial circle, 2. the ellipse after the ellipse growing, 3. the ellipse fitted by LMedS, and 4. the ellipse fitted together with another segment detected.

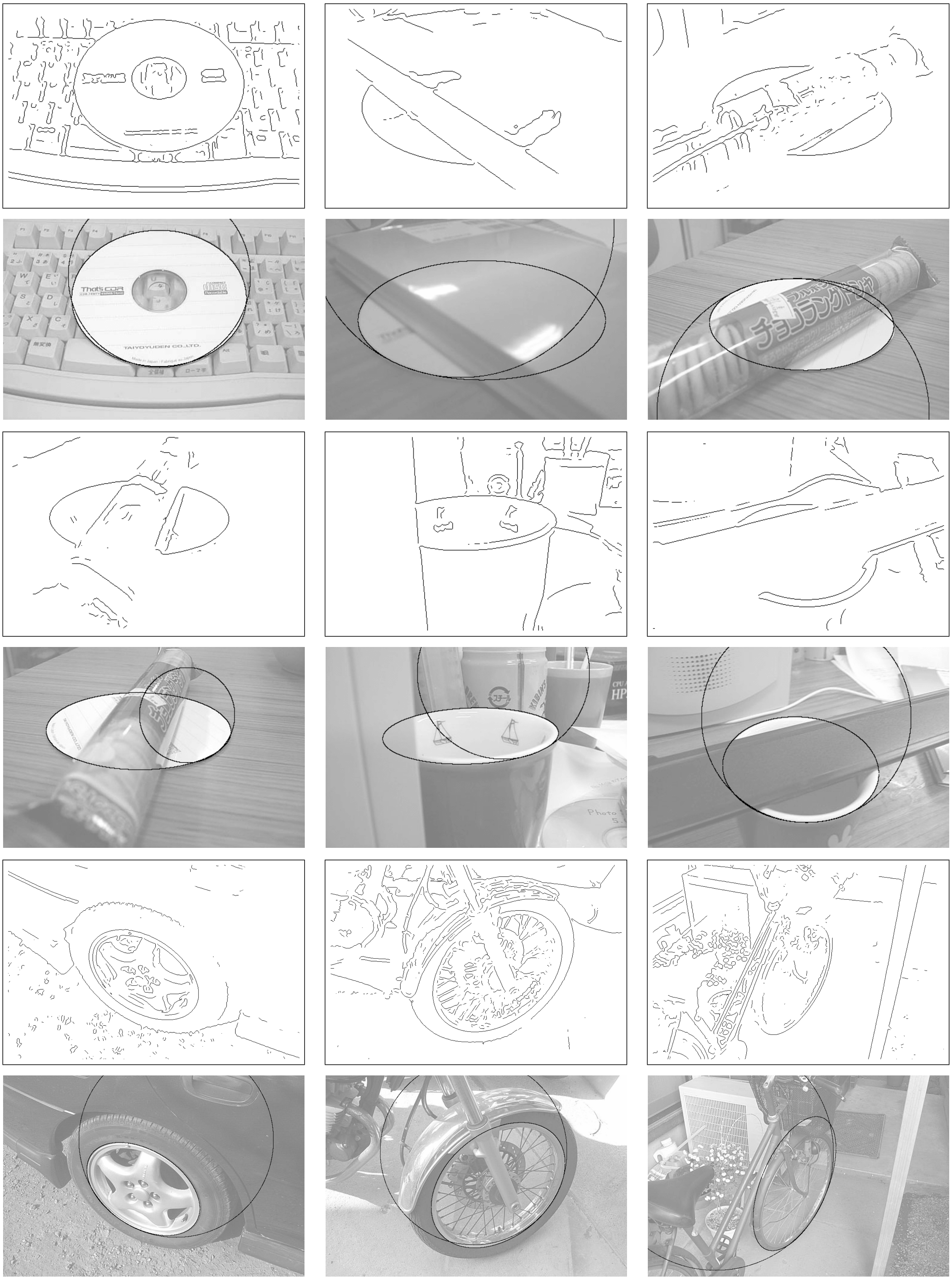


Figure 8: Examples of ellipse fitting. The upper rows are the edge images used. In the lower rows, the initial circles and the fitted ellipses are superimposed on the original images.

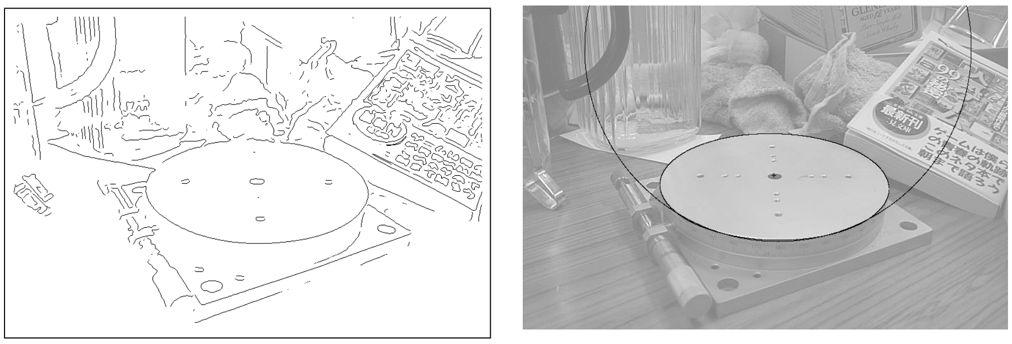


Figure 9: Calibration of a turntable.

eigenvectors. The eigenvalues are ordered in such a way that $\lambda_3 < 0 < \lambda_1 \leq \lambda_2$. The unit normal to the surface of the turntable is computed as follows [9]:

$$\boldsymbol{n} = N[\sqrt{\frac{\lambda_2 - \lambda_1}{\lambda_2 - \lambda_3}}\boldsymbol{u}_2 \pm \sqrt{\frac{\lambda_1 - \lambda_3}{\lambda_2 - \lambda_3}}\boldsymbol{u}_3]. \tag{14}$$

Here, $N[\,\cdot\,]$ designates normalization to a unit vector. We obtain two solutions, because looking down on the turntable and looking up at it would result in the same ellipse image. In the usual setup, we need to pick out the looking-down solution. This is done by selecting the sign for which $n_1 n_3 \leq 0$. The orthogonal distance of the turntable surface from the lens center is given by

$$d = \sqrt{\lambda_1^3}r. \tag{15}$$

The center of the turntable is projected onto the image at the position

$$\boldsymbol{x}_C = Z[\boldsymbol{Q}^{-1}\boldsymbol{n}], \tag{16}$$

where $Z[\,\cdot\,]$ designates normalization to make the z component 1. The 3-D position of the center of the turntable is

$$\boldsymbol{r}_C = \frac{d\boldsymbol{x}_C}{(\boldsymbol{n}, \boldsymbol{x}_C)} \tag{17}$$

with respect to the camera coordinate system. The angle between the optical axis of the camera and the rotation axis of the turntable is give by

$$\theta = \cos^{-1}|n_3|. \tag{18}$$

Fig. 9(a) is an edge image of a turntable. The focal length is 700 pixels, and the radius of the turntable is 6.45cm. In Fig 9(b), the initially detected circle and the finally fitted ellipse are superimposed on the original image (768 × 512 pixels). The angle between the optical axis of the camera and the rotation axis of the turntable is computed to be 59.4°; the distance between the center of the lens to the center of the turntable is computed to be 27.3cm.

5. CONCLUDING REMARKS

With the motivation of obtaining accurate 3-D measurement of circular objects, we presented a new method of automatically detecting ellipses from images: we detect the osculating circle to the observed ellipse by a Hough transform, iteratively improve the fit by our ellipse growing scheme, remove outlier pixels by LMedS, and search for separate arcs. We improved efficiency by limiting the Hough space to be one and two dimensions and enhanced accuracy by introducing special weighting schemes. Using real images, we demonstrated that our method allows us to detect partially occluded circular objects very effectively within a reasonable time. Finally, we illustrated an application of our method to the calibration of a turntable for 3-D object shape reconstruction.

Acknowledgements

The authors thank Mitsuo Okabe of Fujitsu Cadtech, Ltd., Japan, for real image experiments. This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C(2) (No. 13680432).

References

[1] V. I. Arnol'd, *Catastrophe Theory*, Springer, Berlin, 1984; 3rd Revised and Expanded Edition, Springer, Berlin, 1992.

[2] Y. Asayama and M. Shiono, A slanted ellipse detection by a circle detecting Hough transform using a par of arcs, *Proc.IAPR Workshop on Machine Vision Applications*, November 1998, Makuhari, Chiba, Japan, pp. 494–497.

[3] J. W. Bruce and P. J. Giblin, *Curves and Singularities*, Cambridge University Press, Cambridge, U.K., 1984.

[4] Y. C. Cheng and S. C. Lee, A new method for quadratic curve detection using K-RANSAC with acceleration techniques, *Patt. Recog.*, **28**-5 (1995), 663–682.

[5] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, On the fitting of surfaces to data with covariances, *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**-11 (2000), 1294–1303.

[6] W. Chojnacki, M. J. Brooks and A. van den Hengel, Rationalising the renormalisation method of Kanatani, *J. Math. Imaging Vision*, **14**-1 (2001), 21–38.

[7] D. B. Cooper and N. Yalabik, On the computational cost of approximating and recognizing noise-perturbed straight lines and quadratic arcs in the plane, *IEEE Trans. Comp.*, **25**-10 (1976), 1020–1032.

[8] D. Ioannou, W. Huda and A. F. Laine, Circle recognition through a 2D Hough Transform and radius histogramming, *Image Vision Comput.*, **17**-1 (1999), 15–26.

[9] K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, 1993.

[10] N. Guil and E. L. Zapata, Lower order circle and ellipse Hough transform, *Patt. Recog.*, **30**-10 (1997), 1729–1744.

[11] C.-T. Ho and L.-H. Chen, A fast ellipse/circle detector using geometric symmetry, *Patt. Recog.*, **28**-1 (1995), 117–124.

[12] C. L. Huang, Elliptical feature extraction via an improved Hough transform, *Patt. Recog. Lett.*, **10**-2 (1989), 93–100.

[13] Y. Leedan and P. Meer, Heteroscedastic regression in computer vision: Problems with bilinear constraint, *Int. J. Comput. Vision*, **37**-2 (2000), 127–150.

[14] B. Matei and P. Meer, Reduction of bias in maximum likelihood ellipse fitting, *Proc. 15th Int. Conf. Patt. Recog.*, September 2000, Barcelona, Spain, Vol.3, pp. 801–806.

[15] D. Pao, H. F. Li and R. Jayakumar, A decomposable parameter space for the detection of ellipses, *Patt. Recog.*, **14**-12 (1993), 951–958.

[16] P. L. Rosin and G. A. W. West, Nonparametric segmentation of curves into various representations, *IEEE Trans. Patt. Anal. Mach. Intell.*, **17**-12 (1995), 1140–1153.

[17] G. Roth and M. D. Levine, Extracting geometric primitives, *CVGIP: Image Understand.*, **58**-1 (1993), 1–22.

[18] G. Roth and M. D. Levine, Geometric primitive extraction using a genetic algorithm, *IEEE Trans. Patt. Anal. Mach. Intell.*, **16**-9 (1994), 901–905.

[19] C. A. Rothwell, A. Zisserman, C. I. Marinos, D. A. Forsyth and J. L. Mundy, *Relative motion and pose from arbitrary plane curves*, *Image Vision Comput.*, **10**-4 (1992), 250–262.

[20] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, Wiley, New York, 1987.

[21] S. Tsuji and F. Matsumoto, Detection of ellipses by a modified Hough transform, *IEEE Trans. Comp.*, **27**-8 (1978), 777–781.

[22] W.-Y. Wu and M.-J. J. Wang, Elliptical object detection by using its geometric properties, *Patt. Recog.*, **26**-10 (1993), 1449–1500.

[23] H. K. Yuen, J. Illingworth and J. Kittler, Detecting partially occluded ellipses using the Hough transform, *Image Vision Comput.*, **7**-1 (1989), 31–37.

[24] J. H. Yoo and I. K. Seth, An ellipse detection method from the polar and pole definition of conics, *Patt. Recog.*, **26**-2 (1993), 307–315.