

ELLIPSES FROM TRIANGLES

*M. Cicconet, K. Gunsalus**

Center for Genomics and Systems Biology
New York University

D. Geiger, M. Werman[†]

Courant Institute of Mathematical Sciences
New York University

ABSTRACT

We present an ellipse finding and fitting algorithm that uses points and tangents, rather than just points, as the basic unit of information. These units are analyzed in a hierarchy: points with tangents are paired into triangles in the first layer and pairs of triangles in the second layer vote for ellipse centers. The remaining parameters are estimated via robust linear algebra: eigen-decomposition and iteratively reweighted least squares. Our method outperforms the state-of-the-art approach in synthetic images and microscopic images of cells.

Index Terms— ellipse detection, ellipse fitting, pattern recognition, image analysis, cell counting

1. INTRODUCTION AND PREVIOUS WORK

Fitting multiple and overlapping ellipses on digital images remains an open research area, despite the large number of approaches that were published, from as far back as four decades ago [1], up until as recently as 2012 [2].

There are three main classes of methods. Least squares algorithms (e.g., [3, 4]) turn the ellipse-fitting problem into a restricted-optimization problem. Hough transform techniques (e.g., [5]) search for local maxima in a parameter histogram. Edge contour following methods (e.g., [6, 2]) provide many elliptical hypothesis from small “local” arcs before grouping similar hypothesis and discarding ones with low “weight.”

All methods have limitations. The first is very sensitive to outliers on the data, and thus only adequate in very low-noise conditions. Furthermore, in its standard form it only works for one ellipse. Hough transform techniques, although robust to outliers and able to deal with multiple and overlap, are computationally prohibitive due to the parameter space dimension. Contour following methods are sensitive to the grouping parameters and discarding of hypothesis, and thus not very adequate for shapes that deviate slightly from a perfect ellipse.

*This research is supported by NIH grants 5R01-GM085503 and 5R01-HD046236.

[†]The fourth author is a visiting scholar, from The Hebrew University of Jerusalem. His work was supported in part by the Israel Science Foundation and the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI).

In this paper we introduce a new, hybrid technique, that exploits some of the advantages from the first two classes of methods. The centers of the ellipses are obtained in a Hough transform fashion, but contrary to previous alternatives, voting is performed in two layers: first, pairs of points (with tangents) define triangles, and pairs of triangles cast votes. The estimation of the remaining parameters is based on the solution of a linear system that depends both on the points and the tangents—typical least square methods work on points only.

Our method outperforms the best available method in our databases of synthetic images and of mouse embryo cells.

2. THE ALGORITHM

The algorithm for ellipse finding has four steps, detailed in the following subsections.

2.1. Points and Tangents (Step 1)

Most shape-fitting methods are based on edges, and those who use tangent information usually compute the tangents from the edge image. In our method, a point (pixel) with its tangent is treated as the basic unit of information, and we capture them in one step from the image, using a bank of Morlet wavelet filters [7], keeping maximum angle responses larger than a threshold.

2.2. Location of Centers (Step 2)

The following property of points and tangents in ellipses (see Figure 1) is used to find ellipse centers. Let p, q be points in an ellipse, with tangents τ_p, τ_q . Let x_{pq} be the intersection of the lines passing through p with tangent τ_p and through q with tangent τ_q . Let m_{pq} be the midpoint of (p, q) . The line through x_{pq} and m_{pq} intersects the center of the ellipse. To see why this is true, observe that this trivially holds for a circle, and that an ellipse is a center-preserving linear transformation of a circle. It follows that two such lines intersect at the ellipse’s center.

This step is similar to what is performed in the Randomized Hough Transform [5], with the difference that we look at pairs of pairs, instead of a triplet of points. This makes the implementation easier, lets one chose more robust, close

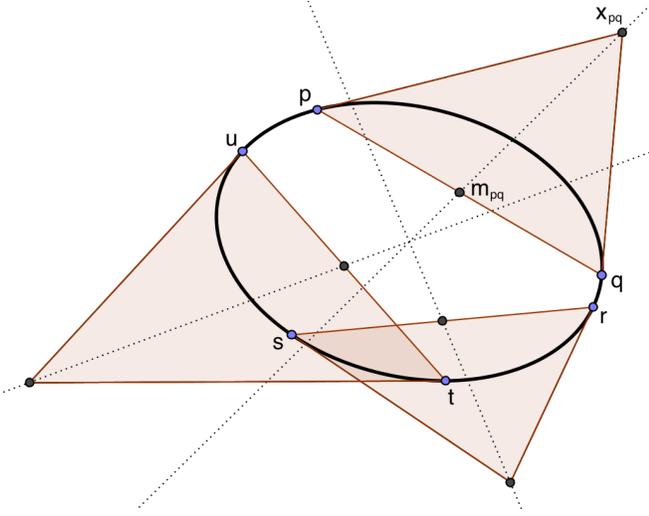


Fig. 1. Properties of triangles defined by pairs of points in an ellipse allow center recovery. In this figure, the triangles have edges tangent to the ellipse at the specified points; the dotted lines go through the midpoints of the edges defined by the pairs (p, q) , (r, s) , and (t, u) ; intersecting at the center of the ellipse.

to orthogonal, pairs of lines to intersect, and the approach is a “layered” architecture (the first layer considers pairs of points, the second considers pairs of triangles).

As in the RHT, we do not take into account all possible triangles, only a subset of N_T of them, chosen randomly. We use $N_T = c \cdot N_P$, where c is a small integer constant (2), and N_P is the number of pairs of points (for which the corresponding tangents intersect).

To reduce noise and computational load in locating the centers, we only accumulate votes when the distance between the estimated center c_{pqrs} and the boundary points p, q, r, s is within a range of *radii* we are looking for. Further restrictions can be applied: for instance, only accumulate centers that are within image boundaries.

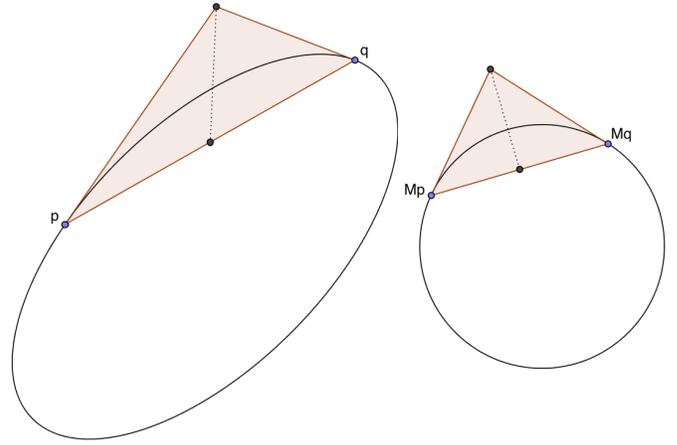
Local maxima in the accumulator matrix A correspond to ellipse centers. Since A is sparse, we first convolve it with a gaussian kernel before searching for local maxima. Furthermore, there are thresholds for the minimum value of a local maxima, and how far from a stronger one they should be for consideration.

Figure 3 (center column) shows examples of accumulator images for real and synthetic images.

2.3. Clustering (Step 3)

Pairs of triangles (quadruples of points) vote for centers they belong to.

Let $\{c_i : i = 1, \dots, N_C\}$ be the set of ellipse centers found as local maxima in A . Every quadruple p, q, r, s is associated to the ellipse center $c_{\tilde{i}}$, $\tilde{i} = \arg \min \|c_{pqrs} - c_i\|$, if $\|c_{pqrs} -$



(a) Before mapping. (b) After mapping.

Fig. 2. The triangle defined by p, q and their tangents becomes isosceles once the ellipse is mapped into a circle.

$c_{\tilde{i}}\|$ is less than a “proximity” threshold. (c_{pqrs} is defined in the previous subsection.)

Figure 4 (a,b) shows the output of this phase for a synthetic image containing two ellipses.

2.4. Remaining Parameters (Step 4)

Once the center of an ellipse is known, by subtracting the center from their boundary points (which are known from the previous step), the ellipse is translated to be centered at the origin, $(0, 0)$. Three parameters remain to be found: the major axis, the minor axis, and the angle between the major axis and the horizontal line. We compute these parameters by estimating the linear transformation M that maps the ellipse points and the tangents into a circle.

Let p and q be two points in the boundary of such an ellipse, and x_{pq}, m_{pq} as defined in Subsection 2.2. We define

$$v = q - p, \quad w = x_{pq} - m_{pq}, \quad (1)$$

and observe that Mv and Mw should be perpendicular, that is, the corresponding mapped triangle should be isosceles, see Figure 2.

Thus, M satisfies

$$0 = \langle Mv, Mw \rangle = \langle v, M^T Mw \rangle. \quad (2)$$

Let $B = M^T M$. Since B is symmetric, it can be written as

$$B = \begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix}. \quad (3)$$

Now, writing $v = (v_1, v_2)$ and $w = (w_1, w_2)$, Equation 2 becomes

$$\begin{pmatrix} v_1 w_1 & v_1 w_2 + v_2 w_1 & v_2 w_2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = 0. \quad (4)$$

At least 3 equations are needed to solve for α, β, γ . Since there are many pairs of points in the boundary of the ellipse, say $(p^i, q^i), i = 1, \dots, N$, we build a system of equations,

$$\begin{pmatrix} \vdots \\ v_1^i w_1^i & v_1^i w_2^i + v_2^i w_1^i & v_2^i w_2^i \\ \vdots \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = 0. \quad (5)$$

Let P be the $N \times 3$ matrix in the above equation, and $x = (\alpha, \beta, \gamma)$, we are then left to find a *non trivial* solution for the linear system $Px = 0$. This can be solved in the least squares sense using the eigen-decomposition of $P^T P$.

Once we have α, β, γ , and therefore B , we can find M that satisfies $B = M^T M$ by computing the eigenvalues and eigenvectors of B . (Some of the remaining ellipse parameters are obtained from B , other from M .)

M , we recall, takes points in the boundary of an ellipse to points in the boundary of a circle. The solution found via eigen-decomposition is a *scaling* of the plane in the direction of the major axis of the ellipse (which is the direction of the eigenvector or B corresponding to its largest eigenvalue— b , satisfying $\|b\| = 1$). See Figure 4 (c) for an example. In particular, the radius ρ of the transformed circle (the computation of which we describe in a moment) gives the length of the semi-minor axis, and the rotation of the ellipse is the angle that b makes with the horizontal line. The last parameter, the length of the semi-major axis, is equal to $\frac{\sqrt{e_2}}{\sqrt{e_1}} \rho$ (where e_1, e_2 are the eigenvalues of B , with $e_1 \leq e_2$). The norms of the transformed points (by the mapping $p \rightarrow Mp$) should be the same (since they are on a circle centered at the origin), except for outliers (boundary points that were not properly clustered to the corresponding ellipse center—see Figure 4 (a,b)). ρ is the radius of such circle. We estimate it via 1-dimensional Robust Regression [8] (to cope with outliers).

Figure 3 shows examples of our method in real and synthetic images.

3. EXPERIMENTS AND CONCLUSION

We compared our algorithm with the state-of-the-art method an edge contour following technique by Prasad *et al.* [2]. The authors report experiments with synthetic and real images (the later a subset from the Caltech-256 database [9]—the specification of this subset is currently unavailable, according to [2]’s project page). Their results outperform classical and modern methods [10, 11, 12, 13, 5].

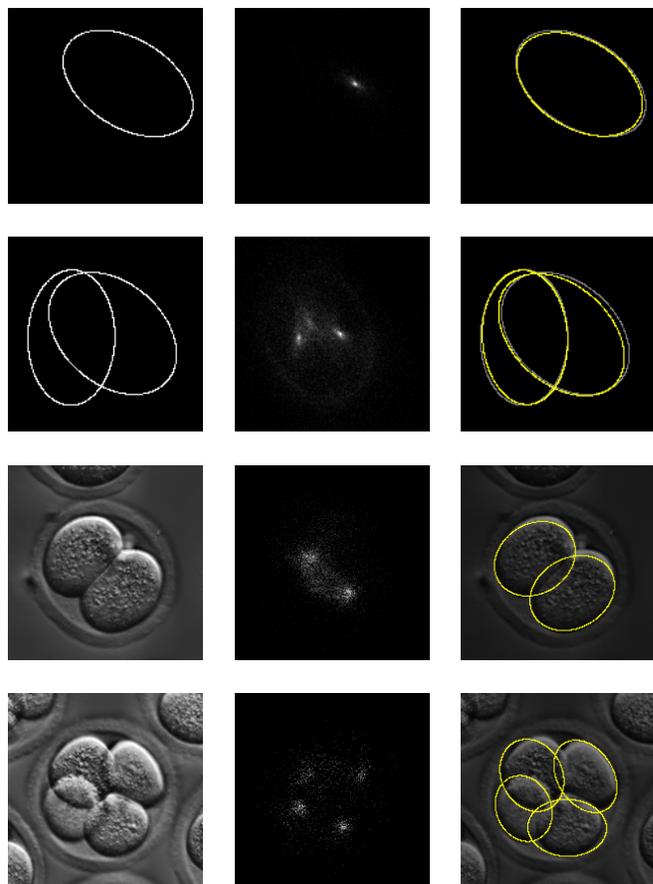


Fig. 3. Outputs of our method in synthetic and real images. The central column shows the accumulator space, from which the local maxima correspond to the ellipse centers.

We generated our own synthetic images: three sets of 100 images, containing one, two, or three ellipses per image (respectively). Images are of 256 by 256 pixels, ellipses are entirely contained in the images, the semi axis range from 20 to 60 pixels (uniformly distributed), orientation is random, and the distance between centers is of at least 32 pixels (so that the overlap is not arbitrarily high). For real world pictures we chose two sets of microscopic images of mouse-embryo, containing two or four cells per image (examples are shown in Figure 3), from the database in [14]. These sets contain 421 and 332 images each, respectively.

Table 1 shows the results we obtained. Detection is considered correct if the computed parameters are at most Γ away from the ground-truth parameters. If $n > 2$ ellipses are located at a distance $< \Gamma$ from a ground truth ellipse, $n - 1$ are considered false positives. False negatives correspond to lack of detection at a distance $< \Gamma$ from a true ellipse, and to ellipses located more than Γ pixels away from any true true ellipse. Since the ground truth values for embryo cels are based on a circle fitting algorithm, the angles of the computed ellipses were not considered during evaluation. We

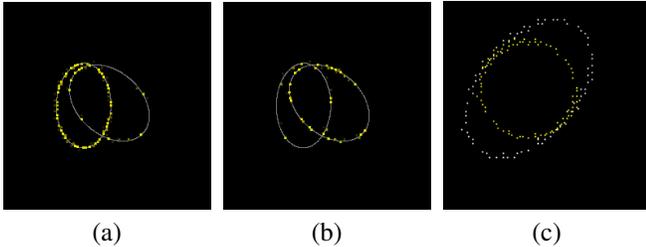


Fig. 4. (a,b) Points clustered in different ellipses; notice the outliers, which our method is able to cope with. (c) Points are mapped into a circle for the computation of the ellipse parameters other than its center.

used $\Gamma = 20$ for synthetic images, and $\Gamma = 15$ for images of 2 cells.

NI	NE	Method	Precision	Recall
100	100	Ours	1.00	1.00
		Prasad	0.98	0.98
		Prasad (Or.)	0.86	0.50
100	200	Ours	0.96	0.96
		Prasad	0.84	0.93
		Prasad (Or.)	0.83	0.27
100	300	Ours	0.86	0.82
		Prasad	0.74	0.86
		Prasad (Or.)	0.74	0.21
421	842	Ours	0.93	0.92
		Prasad	0.77	0.50
		Prasad (Or.)	0.82	0.13

Table 1. Results of our method, in comparison with Prasad’s [2]. NI and NE stand for number of images and ellipses (cells, in the cases of the last block of rows), respectively. The “Prasad (Or.)” rows correspond to the original implementation of salient elliptic hypotheses selection in [2].

Prasad’s method has a step to select “salient” ellipses from an initial set of hypotheses. We also tested the method with a modified selection step, which in general improved precision and recall rates: we discard ellipses whose centers are less than 20 pixels from already existing ones. As input to Prasad’s algorithm we used the binary images obtained using the Sobel edge detector (the built-in Matlab implementation). In all implementations (including our method), we set the maximum number N_C of ellipses to be found before the experiment (for the number of ellipses in each image was known), picking the N_C most prominent outputs.

One might ask: why computing the remaining parameters of the ellipse using the method of Subsection 2.4, and not a simpler method, say [3]? The least squares methods, such as [3], work on pixels only, whereas our method use points *and* tangents. Theoretically, more information should provide better results, we made an experiment to verify this comparing

our method with one replacing the last step (Subsection 2.4) by Fitzgibbon’s least squares approach [3]. In 100 synthetic images containing 2 ellipses each, our original method’s precision and recall rates are about 3 times those of the modified method. The same holds in 100 synthetic images containing 3 ellipses each.

In light of the results reported in [2], our experiments show that ellipse fitting methods are highly dependent on the input dataset. Although [2] outperformed competitors in particular sets of images, that didn’t imply best performance in our images.

Our new method not only gives the state of the art results in ellipse finding, it contributes to the existing literature of shape detection in three main ways. First, we use points and tangents, rather than just points, as the basic unit of information. Second, we analyze these units in a hierarchical pairwise fashion: in the first layer, triangles (each pair of points with respective tangents define a triangle), and in the second, pairs of triangles. This provides further evidence on the reliability of looking at pairs of data units for geometric data analysis, which we have already seen in other contexts, such as circle detection (reference submitted for publication). Third, the method we use to capture the ellipse parameters other than the center is novel, and relies on robust techniques from linear algebra (eigen-decomposition) and statistics (iteratively reweighted least squares).

4. ALGORITHM COMPLEXITY AND FINAL NOTES

Step 1 is wavelet filtering, thus $O(n)$, where n is the number of pixels in the image. Step 2 is quadratic in the number of outputs from Step 1, thus $O(n^2)$. Step 3 is of the order of number of outputs from Step 2 times the number m of ellipses found, thus $O(m \cdot n^2)$. The time spent on eigen-decompositions in Step 4 can be considered constant, since all involved matrices are at most 3×3 . Assymptotic complexity depends on the cost of building $P^T P$, which is $O(9 \cdot r_P)$, where r_P is the number of rows of P (the number of columns of P is constant, equal to 3). Since for each ellipse r_P is of the order of the number of triangles on its boundary (i.e., $O(n^2)$), we conclude that Step 4 is $O(m \cdot n^2)$. Therefore, overall the algorithm is $O(m \cdot n^2)$.

Our algorithm was implemented in Matlab (R2011b, 64-bit). It runs in less than 5 seconds on synthetic images containing 3 ellipses, and 20 seconds on images containing 4 cells. Images are of size 240×240 pixels. All experiments were performed on an iMac with 2.8 HGz Intel Core i7 processor and 16 GB RAM, running Mac OS 10.9.

The lists of mouse-embryo images used in the experiments are available at the project’s webpage [15].

Drawings in Figures 1 and 2 were made with the help of GeoGebra. This open-source software allows, in particular, to dynamically visualize the property described in Figure 1, by moving the points on the ellipse.

5. REFERENCES

- [1] R. O. Duda, *Pattern Classification and Scene Analysis*, Wiley Publishers, New York, 1973.
- [2] Dilip K. Prasad, Maylor K. H. Leung, and Siu-Yeung Cho, “Edge curvature and convexity based ellipse detection method,” *Pattern Recognition*, vol. 45, no. 9, pp. 3204–3221, 2012.
- [3] Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher, “Direct least square fitting of ellipses,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, 1999.
- [4] Paul L. Rosin, “A note on the least square fitting of ellipses,” *Pattern Recognition Letters*, vol. 14, pp. 799–808, 1993.
- [5] Robert A. Mclaughlin, “Randomized hough transform: Improved ellipse detection with comparison,” Tech. Rep., Univ. of Western Australia, 1998.
- [6] Alex Yong Sang Chia, Susanto Rahardja, Deepu Rajan, and Maylor Karhang Leung, “A split and merge based ellipse detector with self-correcting capability,” *IEEE Transactions on Image Processing*, , no. 7, pp. 1991–2006, 2011.
- [7] Joan Bruna and Stéphane Mallat, “Invariant scattering convolution networks,” *CoRR*, vol. abs/1203.1513, 2012.
- [8] P. W. Holland and R. E. Welsch, “Robust regression using iteratively reweighted least-squares,” *Communications in Statistics: Theory and Methods*, vol. A6, pp. 813–827, 1977.
- [9] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” <http://authors.library.caltech.edu/7694/1/CNS-TR-2007-001.pdf>, 2007.
- [10] F. Mai, Y.S. Hung, H. Zhong, and W.F. Sze, “A hierarchical approach for fast and robust ellipse extraction,” *Pattern Recognition*, vol. 41, no. 8, pp. 2512 – 2524, 2008.
- [11] Euijin Kim, Miki Haseyama, and Hideo Kitajima, “Fast and robust ellipse extraction from complicated images,” in *Proceedings of the IEEE International Conference on Information Technology and Applications*, 2002.
- [12] Xiangzhi Bai, Changming Sun, and Fugen Zhou, “Splitting touching cells based on concave points and ellipse fitting,” *Pattern Recognition*, vol. 42, no. 11, pp. 2434 – 2446, 2009.
- [13] Zhi-Yong Liu and Hong Qiao, “Multiple ellipses detection in noisy environments: A hierarchical approach,” *Pattern Recognition*, vol. 42, no. 11, pp. 2421 – 2433, 2009.
- [14] M. Cicconet and K. C. Gunsalus, “Mouse embryo tracking database,” <http://aquila.bio.nyu.edu/celltracking/>, 2014.
- [15] M. Cicconet, “Ellipses from triangles—project webpage,” <http://marceloc.net/science/eft.html>, 2014.