Fitting an Ellipse is Equivalent to Find the Roots of a Cubic Equation

Luis Gerardo de la Fraga and César Cruz Díaz
Cinvestav, Computer Science Department.
Av. Instituto Politécnico Nacional 2508. 07360 México City, México
Email: fraga@cs.cinvestav.mx

***Abstract*—In this paper will be prove that fitting an ellipse, according to the minimization of the squares of algebraic distances, is equivalent to calculate the roots of three cubic equations. The programming code to implement this solution is very small, without to need any special numeric library. Futhermore, posible numeric problems of the implementation are analized and several simulation results are shown to validate the proposed implementation.**

I. Introduction

The fitting of geometric features is an important problem in several fields of science and engineering. In particular, circle and conic are the most common geometric features for the aplication of image processing. Fitting an ellipse is an important task in Computer Vision, e.g. circle projections may become ellipses, and it has many practical applications as face detection [1], quality control [2] and analysis of grain [3], to mention just a few.

The simplest algorithm to fit an ellipse to a given set of points is based in the minimization of the sum of algebraic distances, from those given points to the ellipse [4]. Although this algorithm is also very easy to code, in fact in [4] is implemented in ten lines of Matlab code, it is not very robust and in [5] authors conclude –erronously, as will be shown in this article– that "the method does not work".

The goal of this work is to implement a program to solve the minimization problem without to need special numeric libraries. The method in [4] needs a procedecure to solve the generalized eigenvalue problem of a 6×6 matrix. Also, it is required that our solution must be robust, this is, the program must give a correct result even if five input points are used. The solution in [4] does not give a result when only five points are used.

Futhermore, having a procedure to fit an ellipse with the minimum set of points (five) is basic to the robust fit using the RANdom SAmple Consensus (RANSAC) algorithm. RANSAC is the state of the art algorithm to solve a robust estimation problem (with outliers and noise in the input data) in the field of computer vision [6]. The way which RANSAC solves an robust ellipse fitting problem is as follow: (1) Select randomly five points from the whole set of input points, (2) use the LS procedure to estimate a model with that five points, (3) evaluate the model (e.g, counting the number points on the model), (4) if the number of points on the model is gretear that a previous iteration, accept the ellipse model. This model is called the consensus set, (5) Go to step (1) until a certain number of iterations is reached.

In addition, the direct procedure in [4] (in fact, it is a least-square procedure) can be user as the initial solution for non-linear fitting methods [7].

The paper is organized as follows; in next Sec. II the linear algorithm in [4] and the solution in [8] are described; in Sec. III our solution is explained; in Sec. IV simulations results are shown, and a brief discussion is given in Sec. V. Finally in Sec. VI conclusions of this work are drawn.

II. Description of the linear algorithm

The method [4] fit a set of points to an ellipse, it is based in the minimization of the sum of square of algebraic distances from the the points to the ellipse, this is, the method minimizes the function:

$$g_a(\mathbf{a}) = \sum_{i=1}^{n} (\mathbf{a}^{\mathrm{T}} \mathbf{v}_i)^2 \tag{1}$$

where $\mathbf{a} = [a, b, c, d, e, f]^{\mathrm{T}}$ is the vector of coeficients of the general conic equation $ax^2 + bxy + cy^2 + dx + ey + f = 0$, and $\mathbf{v}$ is the vector $[x^2, xy, y^2, x, y, 1]^{\mathrm{T}}$ that is built from the coordinates of each point. The general conic equation represents an ellipse if $b^2 - 4ac < 0$. The term $\mathbf{a}^{\mathrm{T}}\mathbf{v}_i$ in (1) represents the *algebraic distance* from point (x, y) to the ellipse represented by vector $\mathbf{a}$.

Applying Lagrange multiplier to (1) and fixing the constrain scale to $4ac - b^2 = 1$, we obtain the system of equations:

$$\begin{aligned} S\mathbf{a} &= \lambda C\mathbf{a}, \quad \text{subject to:} \\ \mathbf{a}^{\mathrm{T}} C \mathbf{a} &= 1, \end{aligned} \tag{2}$$

where $S = D^{\mathrm{T}}D$, $D = [\mathbf{v}_1^{\mathrm{T}}; \mathbf{v}_2^{\mathrm{T}}; \ldots; \mathbf{v}_n^{\mathrm{T}}]$, and C is the constraint matrix; S and C are in $\mathbb{R}^{6\times 6}$. C is a matrix of zeros, except $C(2,1) = C(1,2) = 2$ and $C(1,1) = -1$.

Equation (2) can be rearranged by taking submatrices in $\mathbb{R}^{3\times 3}$ as in [8]:

$$\begin{bmatrix} S_1 & S_2 \\ S_2^{\mathrm{T}} & S_3 \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix}, \text{subject to:} \tag{3}$$

$$\mathbf{a}^{\mathrm{T}} C_1 \mathbf{a} = 1 \tag{4}$$

where

$$C_1 = \begin{bmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \tag{5}$$

Developing (3) we arrive to:

$$(S_1 - S_2 S_3^{-1} S_2^{\mathrm{T}})\mathbf{a}_1 = \lambda C_1 \mathbf{a}_1 \tag{6}$$

$$\mathbf{a}_1^{\mathrm{T}} C_1 \mathbf{a}_1 = 1 \tag{7}$$

$$\mathbf{a}_2 = -S_3^{-1} S_2^{\mathrm{T}} \mathbf{a}_1 \tag{8}$$

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \tag{9}$$

III. Proposed solution to the problem

Equation (6) represents the generalized eigenvalues and eigenvectors problem [9]. We can see more clearly this if it is changed in (6) $M = S_1 - S_2 S_3^{-1} S_2^{\mathrm{T}}$, and also doing X the eigenvectors matrix, and E the diagonal matrix of eigenvalues, giving

$$MX = C_1 XE \tag{10}$$

The eigenvectors of a symmetric matrix are all real (or no one is a complex number) [10]. Now, can be solved (10) without lost the symmetry propierties of matrices M and C_1?, or in other words, can be changed the general eigenproblem to a standard eigenproblem? [10, Ch. 15]. Answer is yes. The solution is obtained rewriten (10) in the form

$$C_1 X = MXE^{-1}, \tag{11}$$

and if we calculate the eigenvectors of $M = PE_2^2 P^{\mathrm{T}}$ (it will proved later M is symmetric, and then P is orthonormal), then we obtain:

$$C_1 X = PE_2 E_2 P^{\mathrm{T}} XE^{-1},$$
$$P^{\mathrm{T}} C_1 X = E_2 E_2 P^{\mathrm{T}} XE^{-1},$$
$$E_2^{-1} P^{\mathrm{T}} C_1 X = E_2 P^{\mathrm{T}} XE^{-1}.$$

If we do $X = PE_2^{-1} Y$ results

$$E_2^{-1} P^{\mathrm{T}} C_1 PE_2^{-1} Y = E_2 P^{\mathrm{T}} PE_2^{-1} YE^{-1}.$$

In the right part of this last equation we can see that $E_2 P^{\mathrm{T}} PE_2^{-1} = I$ and if we let $E_2^{-1} P^{\mathrm{T}} C_1 PE_2^{-1} = N$ we obtain

$$NY = YE^{-1}, \tag{12}$$

that is the standar eigenproblem.

A. Some details about our implementation

The eigenvalues of a symmetric matrix, for example A, are all real [9], and then we can calculate directly its eigenvalues by solving the cubic equation that results from $\det(A - \lambda I) = 0$. Therefore, to solve the ellipse fitting problem, three times can be used the eigendecomposition of a symmetric matrix of size 3×3: the first one to find the inverse of matrix S_3 in (6). The second one to find the eigendecomposition of matrix M, in expression (11). And the third one to find the final eigendecomposition of matrix N in (12).

Matrix S_3 is symmetric because of the form that it is built, but it is not evident how M and N are. From the expression to calculate $M = S_1 - S_2 S_3^{-1} S_2^{\mathrm{T}}$, we can see that S_1 is symmetric, and $S_2 S_3^{-1} S_2^{\mathrm{T}}$ is symmetric too[1] therefore M is symmetric.

From the expression to calculate $N = E_2^{-1} P^{\mathrm{T}} C_1 PE_2^{-1}$, E_2^{-1} is diagonal, C_1 is symmetric (see (5)) then $P^{\mathrm{T}} C_1 P$ is also symmetric, and the multiplication by E_2^{-1} does not afect (because it is diagonal), therefore N is finaly also symmetric.

Therefore, in order to solve the algebraic fitting problem of an ellipse we only need a very simple subroutine that calculates the eigenvectors and eigenvalues of a matrix of size 3×3, as the one presented in [11] or to solve a cubic equation as in [12].

In our imprementation we use function `dsyevv3()` [11] to calculate the eigendecomposition of matrices S_3, M and N. With the first eigendecomposition $S_3 = PDP^{\mathrm{T}}$, we calculate $S_3^{-1} = PD^{-1} P^{\mathrm{T}}$. Here, S_3 is also positive definite, then its three eigenvalues are greater than zero and there is not any problem to calculate D^{-1}.

With the second eigendecomposition, $M = PDP^{\mathrm{T}} = PE_2^2 P^{\mathrm{T}}$, we only know M is symmetric and we cannot calculate $E_2 = \sqrt{D}$ if one if its eigenvalues is zero or negative. In practice we observe that M is positive definite, and the ellipse cannot be calculated if two eigenvalues are negative or one of them is greater than 10^{10}. Also, if only one eigenvalue is very small and negative, its sign is changed.

Neither exist any problem to calculate the eigendecomposition of the third matrix N. Only we need to remember that the solution of the fitting problem is one of the eigenvectors $X = PE_2^{-1} Y$, and if any of them does not satisfy the constraint $\mathbf{a}_1^{\mathrm{T}} C_1 \mathbf{a}_1 > 0$, then the ellipse cannot be calculated.

IV. Simulations

To test our implemented program we perform the following experiment: four points were fixed at positions $(120, 120)$, $(480, 120)$, $(480, 280)$ and $(120, 280)$. The values of the fifth point (x, y) were taken inside ranges $0 \leq x < 600$ and $0 \leq y < 400$. The algebraic distance was measured for each point over the canonical ellipse in the origin as $f(a, b, x, y) = x^2 b^2 + y^2 a^2 - a^2 b^2$. The image in Fig. 1(a) show a white pixel where the algebraic distance of any of the five points was greater than 500; black pixels show where was posible to calculate an ellipse and the algebraic distances to any point was less than 500. For this image in Fig. 1(a) can be considered that fitted ellipse pass through the points, such as the ellipse shown in Fig. 2(a) (with its fifth point at $(300, 50)$). Images in Figs. 1(b), (c), (d), (e) and (f) were generated in the same manner that image in Fig. 1(a), but to algebraic distances 10^5, $5 \cdot 10^7$, $3 \cdot 10^8$, $5 \cdot 10^8$ and $1 \cdot 10^9$, respectively. Distance $3 \cdot 10^8$, corresponding to image in Fig. 1(d) could appers very big, but two exemplas of such ellipses fitted with two distint point on the white zones (where distance is greater than a $3 \cdot 10^8$): ellipse in Fig. 2(b) was calculated with the fifth point at $(200, 200)$, and ellipse in Fig. 2(c) was calculated with the fifth point at $(60, 60)$; both ellipses are correct, such as

[1]The inverse, if it exists, of a symmetric matrix is also symmetric and for any matrix A and a symmetric matrix B (both with real elements), the product ABA^{T} is a symmetric matrix.

they resulted of our proposed program as visually as can be observed on the images.

Finally, there are two cases where an ellipse cannot be calculated. First is when two of the eigenvalues of matrix M are negative or when any of them is greater to 10^{10}; this result form the parallel lines on image Fig. 1(f), and in fact two parallel lines is a denegerated solution for an ellipse. Second case is when the eigenvectors of matrix X in (11) do not satisfy the constraint $\mathbf{a}_1^{\mathrm{T}} C_1 \mathbf{a}_1 = 1$, these points form the central cross (+) in image of Fig. 1(f). This second case can be visualized when four points are on the corners of a square and the fifth point is on the square's center: the fitted curve is a circle and not an ellipse.

V. Discussion

The Matlab subrutine presented in [4] fails to calculate the ellipse with input points $(120, 120)$, $(480, 120)$, $(480, 280)$, $(120, 280)$ and $(60, 60)$ (the solution in [4] to eq. (2) is the eigenvector corresponding to the unique negative eigenvale, but in this case two eigenvalues are negative). Our solution gives a result, as is shown in Fig. 2(c).

Besides, our solution can be used perfectily to measure the ellipticity [5] of a bidimensional form. Some 2D curves are the ones used in [5] are shown in Fig. 3 (every form is composed from 150 to 240 points); the corresponding fitted ellipses can be observed in Fig. 4. Clearly, we can observe that our proposed solution gives a good result, and contrarily to authors say in [5], our proposed solution to the linear algorithm in [4] does work.

In summary, our implementation performs:

1) For each point calculate incrementally matrices S_1, S_2 and S_3
2) Invert matrix S_3 (using its eigendecomposition)
3) Calculate matrix M
4) Calculate the eigendecomposition of matrix M
5) If two eigenvalues of M are negative or one of them is greater than 10^{10}, stop.
6) If the smallest eigenvalue of M is negative, invert its value.
7) Calculate matrix N
8) Calculate the eigendecomposition of matrix N
9) Calculate the eigenvectors X
10) If eigenvectors of X do not satisfy the constraint, stop.
11) Optionally calculate the ellipse parameters $[a, b, X_c, Y_c, \alpha]$.

Step 1 is performed, using example matrix S_1, as $S_1 \leftarrow S_1 + \mathbf{v}_i^{\mathrm{T}} \mathbf{v}_i$, where vector $\mathbf{v}_i$ is $[x_i^2, x_i y_i, y_i^2]$ for each point i in the input set of points.

In Step 11, a and b are the semi-major and semi-minor ellipse lenght axes, respectively; (X_c, Y_c) is the ellipse center respect the global coordinate system, and α is the rotation angle respect the semi-major ellipse axis. These are calculated from the general conic coefficients a, b, c, d, e and f as:

$$\alpha = \arctan\left[2b/(a-c)\right],$$

$$X_c = \frac{cd - bf}{b^2 - ac}, \quad Y_c = \frac{af - bd}{b^2 - ac}.$$

Considering these auxiliary values:

$$u_1 = 2(ae^2 + cd^2 + fb^2 - 2bde - acf),$$
$$u_2 = (b^2 - ac)\left[(c-a)\sqrt{1 + \frac{4b^2}{(a-c)^2}} - (c+a)\right],$$
$$u_3 = (b^2 - ac)\left[(a-c)\sqrt{1 + \frac{4b^2}{(a-c)^2}} - (c+a)\right],$$

a and b are calculated as:

$$a = \sqrt{u_1/u_2}, \quad b = \sqrt{u_1/u_3}.$$

And only if $a < b$, these values are interchanged and α is updated as $\alpha \leftarrow \alpha + \pi/2$ if $\alpha < 0$, or $\alpha \leftarrow \alpha - \pi/2$ otherwise.

VI. Conclusions

A new implementation of the general algorithm in [4] to fit an ellipse according the minimization of algebraic distance was presented. We prove that it is only neccesary to solve three cubic equations to solve the fitting problem. We consider that our implementation generate the smallest executable program because no external numerical library (such as LAPACK) is required. Futhermore, our implementation is more stable and robust that the solution presented in [4].

The source code of our implementation is available at http://cs.cinvestav.mx/˜fraga/Softwarelibre/DirectFitting.tar.gz.

References

[1] R.Y.D. Xu and J.S. Jin. Camera control and multimedia interaction using individual object recognition. *Journal of Multimedia*, 2(3), Jun 2007.

[2] R.L. Karkkainen, B.V. Sankar, and J.T. Tzeng. A direct micromechanical approach toward the development of quadratic stress gradient failure criteria for textile composites. *Journal of Composite Materials*, 41(16):1917–1937, Jul 2007.

[3] N.D.G. White, D.S. Jayas, and G. Zhang. Separation of touching grain kernels in an image by ellipse fitting algorithm. *Biosystems Engineering*, 92(2):135–142, Oct 2005.

[4] A. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least square fitting of ellipses. *IEEE Patt. An. & Mach. Intell.*, 21(5), May 1999.

[5] M. Stojmenovic and A. Nayak. *Direct Ellipse Fitting and Measuring Based on Shape Boundaries*, volume 4872 of *LNCS*, pages 221–235. 2007.

[6] O. Chum and J. Matas. Optimal randomized RANSAC. *IEEE Trans. PAMI*, 30(8):1472–1482, Aug 2008.

[7] S.J. Ahn, W. Rauth, and H-J. Warnecke. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognition*, 34(12):2283–2303, Dec 2001.

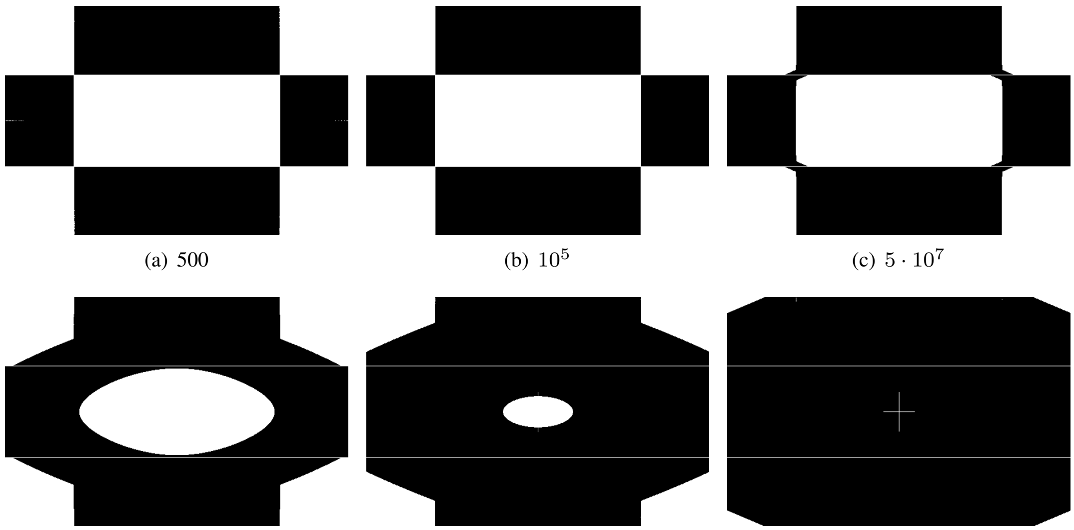
[8] R. Halir and J. Flusser. Numerically stable direct least squares fitting of ellipses, 1998. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.7559.

[9] G.H. Golum and C.F. van Loan. *Matrix computacion, 3rd ed.* Johns Hopkings Uni. Press, 1996.

[10] B.N. Parlett. *The symmetric eigenvalue problem*. Prentice Hall, 1980.

[11] J. Kopp. Efficient numerical diagonalization of hermitian 3x3 matrices. *Int. J. Mod. Phys. C*, 19:523–548, 2008.

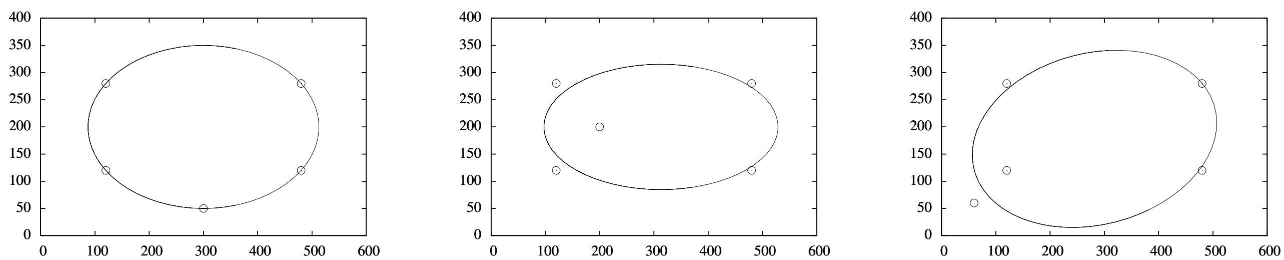
[12] R.W.D. Nickalls. A new approach to solving the cubic: Cardan's solution revealed. *The mathematival gazette*, 77:354–359, 1993.



(a) 500 (b) 10^5 (c) $5 \cdot 10^7$

(d) $3 \cdot 10^8$ (e) $5 \cdot 10^8$ (f) $1 \cdot 10^9$

Fig. 1. Images resulted of our simulation. The value on each image mean that for black pixels, the algebraic distance of any of the five used points is less than the given value



(a) Fifth point in $(300, 50)$ (b) Fifth point in $(200, 200)$ (c) Fifth point in $(60, 60)$

Fig. 2. Three ellipses fitted with the proposed algorithm. The other four points used in the fit are $(120, 120)$, $(480, 120)$, $(480, 280)$ and $(120, 280)$.

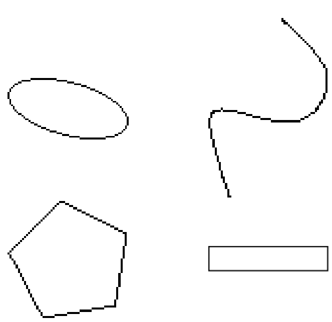


Fig. 3. Four images of 2D forms

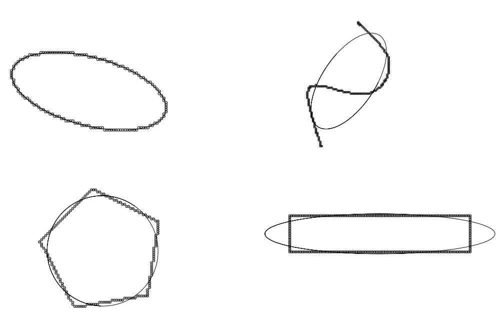


Fig. 4. Ellipses fitted to the 2D forms in Fig. 3.