A NOVEL APPROACH TO ORTHOGONAL DISTANCE LEAST SQUARES FITTING OF GENERAL CONICS

Sudanthi Wijewickrema, Charles Esson
Colour Vision Systems, 11, Park Street, Bacchus Marsh 3340, Australia
{*sudanthi, charlesess*}*@cvs.com.au*

Andrew Papliński
School of Information Technology, Monash University, Clayton 3800, Australia
app@csse.monash.edu.au

Keywords: Conic Fitting, Orthogonal Distance Least Squares Fitting.

Abstract: Fitting of conics to a set of points is a well researched area and is used in many fields of science and engineering. Least squares methods are one of the most popular techniques available for conic fitting and among these, orthogonal distance fitting has been acknowledged as the 'best' least squares method. Although the accuracy of orthogonal distance fitting is unarguably superior, the problem so far has been in finding the orthogonal distance between a point and a general conic. This has lead to the development of conic specific algorithms which take the characteristics of the type of conic as additional constraints, or in the case of a general conic, the use of an unstable closed form solution or a non-linear iterative procedure. Using conic specific constraints produce inaccurate fits if the data does not correspond to the type of conic being fitted and in iterative solutions too, the accuracy is compromised.
The method discussed in this paper aims at overcoming all these problems, in introducing a direct calculation of the orthogonal distance, thereby eliminating the need for conic specific information and iterative solutions. We use the orthogonal distances in a fitting algorithm that identifies which type of conic best fits the data. We then show that this algorithm requires less accurate initializations, uses simpler calculations and produces more accurate results.

1 INTRODUCTION

Conic fitting is a well known problem and has applications in many fields. Among the methods available for this, the most common are: the Hough transform (Hough, 1962), the moment method (Chaudhuri and Samanta, 1991), and least squares fitting (Gauss, 1963). The two former methods become computationally inefficient when a higher number of parameters are involved, and hence, least squares methods have received more attention in recent years. The objective of least squares fitting is to obtain the curve that minimizes the squared sum of a defined error measure.

$$\min\ \sigma^2 = \sum_{i=1}^{n} d_i^2 \tag{1}$$

where, d_i is the distance measure from of the i^{th} point and σ^2 is the squared sum of the errors over n points.

Depending on the distance measure that is minimized, least squares fitting falls into two main categories: algebraic and orthogonal (geometric/euclidian) distance fitting. The algebraic distance from a point to a geometric feature (*eg.* curve or conic) is defined by the following equation.

$$d_a = f(\mathbf{p}, \mathbf{x}) \tag{2}$$

where, $\mathbf{p}$ is the vector of parameters of the geometric feature, $\mathbf{x}$ is the coordinate vector and f is the function that defines the geometric feature or conic.

Although algebraic fitting is advantageous with respect to computing cost and simplicity of implementation, it has many disadvantages, the most serious of which is the lack of accuracy and the bias of the fitting parameters (Ahn, 2004; Fitzgibbon and Fisher, 1995). Changes have been suggested in an effort to improve accuracy and one such error measure is the first order approximation of the orthogonal distance or the normalized algebraic distance (Taubin, 1991).

$$d_n = \frac{d_a}{\| \bigtriangledown d_a \|} \tag{3}$$

where, d_a is the algebraic distance, d_n is the normalized algebraic distance and $\bigtriangledown = \frac{\partial}{\partial \mathbf{x}} = [\frac{\partial}{\partial x} \ \frac{\partial}{\partial y}]^T$

Although using the normalized algebraic distance gives better results than algebraic fitting, it also displays most of the drawbacks of the latter. In addition, unlike algebraic fitting, it cannot be solved in closed form, increasing the complexity of the calculations.

Orthogonal distance, which is agreed to be the most natural and best error measure in least squares techniques (Ahn, 2004), can be used to overcome problems related to algebraic fitting. Although a closed form solution exists for the calculation of the orthogonal point for a general conic, numerical instability can result from the application of the analytic formula (Fitzgibbon and Fisher, 1995; Press et al., 1992). Therefore, either non-linear optimization techniques for the general geometric feature (Ahn, 2004; Boggs et al., 1987; Helfrich and Zwick, 1993) or conic specific characteristics such as semi-axes, center and rotation angle for ellipses (Ahn et al., 2001; Gander et al., 1994; Spath, 1995) are used to calculate the orthogonal distance.

In contrast, the algorithm discussed here employs a novel and direct method of calculating the orthogonal distance, thereby overcoming the above mentioned problems. It also uses a simple procedure for calculating the Jacobian matrix and determines the best fit conic, irrespective of its type.

The rest of the paper is organized as follows: Section 2 gives a brief review of conics and section 3 introduces the proposed algorithm, while section 4 discusses experimental results.

2 REVIEW OF CONICS

A conic is expressed in the form of a 3×3 symmetric matrix, C. If a point $\mathbf{x} = [x \ \ y \ \ 1]^T$, given in homogeneous coordinates is on the conic, it satisfies:

$$\mathbf{x}^T C \mathbf{x} = 0 \tag{4}$$

where,

$$C = \begin{bmatrix} \tilde{C} & \mathbf{c} \\ \mathbf{c}^T & c \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{12} & c_{22} & c_{23} \\ c_{13} & c_{23} & c_{33} \end{bmatrix}, \tag{5}$$

$\tilde{C}$ is a 2×2 symmetric matrix, $\mathbf{c}$ is a two element vector and c is a scalar.

We can extract the five independent parameters of the conic from C by making c_{33} equal to a constant, assuming it's not zero (for example, we use $c_{33} = -1$). Then, the parameter vector $\mathbf{p}$ is as follows:

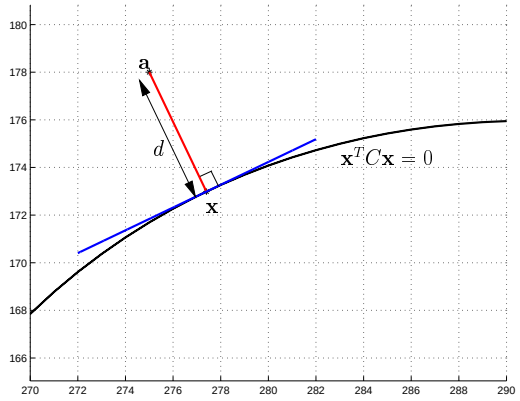


Figure 1: Orthogonal Distance between a Point and a Conic

$$\mathbf{p} = [c_{11} \ \ c_{12} \ \ c_{13} \ \ c_{22} \ \ c_{23}] \tag{6}$$

For any point $\mathbf{x_p}$ not on the conic, there is a unique line $\mathbf{l_p}$ called the polar of the pole $\mathbf{x_p}$ and is defined as in equation (7) (Hartley and Zisserman, 2003). $\mathbf{l_p}$ is the vector that satisfies the line equation ${\mathbf{l_p}}^T \mathbf{x} = 0$ for any point $\mathbf{x}$ on it. If the pole $\mathbf{x_p}$ is on the conic, the polar $\mathbf{l_p}$ becomes its tangent at $\mathbf{x_p}$ (Semple and Kneebone, 1956; Young, 1930).

$$\mathbf{l_p} = C\mathbf{x_p} \tag{7}$$

3 PROPOSED ALGORITHM

This section describes the orthogonal distance conic fitting algorithm. First, in section 3.1, the orthogonal distance is formulated, while its calculation is explained in section 3.2. Section 3.3 describes the complete conic fitting algorithm.

3.1 Orthogonal Distance from a Conic

The orthogonal distance is the shortest distance from a point to a conic, as shown in figure 1. The closest point on the conic from the given point is called the *orthogonal point*. Note that any point in space (on, inside or outside the conic C) is represented by $\mathbf{a}$, the orthogonal distance by d, and the corresponding orthogonal point on C by $\mathbf{x}$, and that the points are given in homogeneous coordinates. For such a point $\mathbf{a}$, the orthogonal distance, is given by equation (8).

$$d = \| \tilde{\mathbf{x}} - \tilde{\mathbf{a}} \| \tag{8}$$

where, $\mathbf{a} = [\tilde{\mathbf{a}} \ \ 1]^T$, $\tilde{\mathbf{a}} = [a_1 \ \ a_2]^T$, and $\tilde{\mathbf{x}} = [x \ \ y]^T$ are the non-homogeneous representations of the points $\mathbf{a}$, and $\mathbf{x}$ respectively.

Calculating the orthogonal distance involves the determination of point $\mathbf{x}$ on the curve for a given point $\mathbf{a}$. Since the orthogonal distance is the shortest distance from a point to a conic, the line connecting the points $\mathbf{x}$ and $\mathbf{a}$ is normal to the conic at $\mathbf{x}$. Therefore:

$$\mathbf{n}_1 = \bar{C}\mathbf{x} \tag{9}$$

where, $\mathbf{n}_1$ is the normal vector, $\bar{C} = [\mathbf{c}_1 \ \ \mathbf{c}_2]^T$ is the 2×3 matrix formed by the first two rows of C, and $\mathbf{c}_1$ and $\mathbf{c}_2$ are the first and second columns of C respectively.

The vector connecting the two points is given by $\mathbf{n}_2 = \tilde{\mathbf{x}} - \tilde{\mathbf{a}} = [x - a_1 \ \ y - a_2]^T$. The same equation can be written in the following form, to be consistent with equation (9).

$$\mathbf{n}_2 = \bar{A}\mathbf{x} \tag{10}$$

where, $\bar{A} = [\mathbf{a}_1 \ \ \mathbf{a}_2]^T$, $\mathbf{a}_1 = [1 \ \ 0 \ \ -a_1]^T$, and $\mathbf{a}_2 = [0 \ \ 1 \ \ -a_2]^T$.

Vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ given in equations (9) and (10) are in the same direction, and lead to equation (11).

$$\begin{aligned} &\bar{C}\mathbf{x} = \alpha\bar{A}\mathbf{x} \\ \Longrightarrow &\frac{\mathbf{c_1}^T\mathbf{x}}{\mathbf{a_1}^T\mathbf{x}} = \frac{\mathbf{c_2}^T\mathbf{x}}{\mathbf{a_2}^T\mathbf{x}} \\ \Longrightarrow &\mathbf{x}^T(\mathbf{c_1}\mathbf{a_2}^T - \mathbf{c_2}\mathbf{a_1}^T)\mathbf{x} = 0 \\ \Longrightarrow &\mathbf{x}^T B\mathbf{x} = 0 \end{aligned} \tag{11}$$

where, $B = \mathbf{c}_1\mathbf{a}_2^T - \mathbf{c}_2\mathbf{a}_1^T$, and α is a scalar parameter.

The relationship obtained in equation (11) is that of a conic, with one exception: the matrix representing the conic B is not symmetric. Without loss of generality, B can be manipulated to get the conventional form of a conic matrix D, which is symmetric but also satisfies the same relationship, as follows:

$$D = \frac{B + B^T}{2} \tag{12}$$

Therefore, for the orthogonal point $\mathbf{x}$ of any point $\mathbf{a}$, the following equations are satisfied simultaneously.

$$\begin{aligned} \mathbf{x}^T C\mathbf{x} &= 0 \\ \mathbf{x}^T D\mathbf{x} &= 0 \end{aligned} \tag{13}$$

These two quadratic equations represent the intersection of two conics and is solved directly using the method discussed in section 3.2. Out of the four possible solutions, the orthogonal point is the point closest to $\mathbf{a}$, a shown in figure 2.

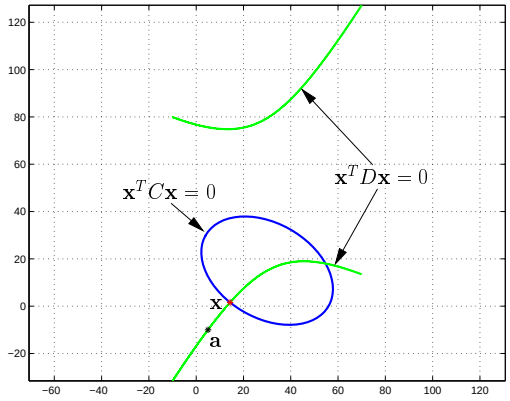


Figure 2: Orthogonal Point for a given Point on the Plane with respect to a Conic

3.2 Solving for the Orthogonal Point

To find the orthogonal point $\mathbf{x}$, two quadratic equations with two parameters should be solved. This results in the solution of a quartic equation whose closed-form solution is known to be numerically unstable (Fitzgibbon and Fisher, 1995; Press et al., 1992). Therefore, iterative methods and/or the introduction of conic specific information as additional constraints are widely accepted as the norm in this calculation (Ahn, 2004; Ahn et al., 2001; Faber and Fisher, 2001). As explained above, this has undesirable properties such as complex calculations, and inaccurate results when the fitted conic does not resemble the distribution of the data points.

To overcome these problems, we use the method used in (Miller, 1988) to solve for the intersection of two conics, as discussed next. (Semple and Kneebone, 1956) and (Young, 1930) show that there exist an infinite number of conics that go through the intersection points of two conics, and that they can be represented by a pencil of conics as follows:

$$C_f = C + \lambda D \tag{14}$$

where, λ is a scalar parameter

(Semple and Kneebone, 1956) further explain that there are three degenerate members (intersecting, parallel or coincident line pairs) in this pencil of conics, and that they go through the common poles of C and D. Like any member of the family, the degenerate conics also share the common intersection points of C and D. Equation (15) gives such a degenerate conic and figure 3 illustrates the relationship between the base conics and a degenerate member.

$$C_d = C + \lambda_d D \tag{15}$$

where, λ_d is the scalar that defines the degenerate conic

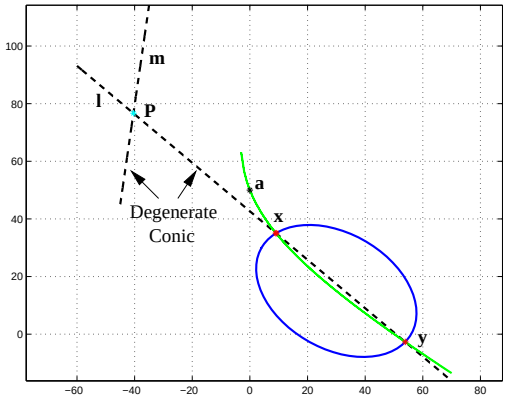


Figure 3: A Degenerate Member of the Conic Family

To obtain the degenerate members, we need to calculate the common poles of C and D. As shown by (Semple and Kneebone, 1956), the three degenerate members of a pencil go through one pole each, and therefore can be calculated by solving the generalized eigensystem $(C-\lambda D)\mathbf{x_p}=0$. This results in the common poles ($\mathbf{x_p}$). We choose a finite, real common pole out of these in our calculations. (Note that for any point in space, an orthogonal point exists on the conic C. Therefore, there exists at least one intersection point between C and D. This implies that we are able to find at least one finite, real common pole to be used in the calculation of the degenerate conic).

Since we know that C_d goes through the chosen common pole $\mathbf{x_p}$, it satisfies ${\mathbf{x_p}}^T C_d \mathbf{x_p}=0$. This gives the value for λ_d as shown in equation (16).

$$\lambda_d = -\frac{{\mathbf{x_p}}^T C \mathbf{x_p}}{{\mathbf{x_p}}^T D \mathbf{x_p}} \tag{16}$$

Now that C_d is known, the next step is to extract the line pair $\mathbf{l}$ and $\mathbf{m}$ that form it. The relationship between the lines and the conic matrix is given in equation (17). $\mathbf{l}$ and $\mathbf{m}$ can be extracted by using singular value or eigen decomposition.

$$C_d = \mathbf{l}\mathbf{m}^T + \mathbf{m}\mathbf{l}^T \tag{17}$$

Once the lines are obtained, the calculation of intersection points is done by solving one of the conic equations ($\mathbf{x}^T C\mathbf{x}=0$ or $\mathbf{x}^T D\mathbf{x}=0$) and the line equations ($\mathbf{l}^T\mathbf{x}=0$ and $\mathbf{m}^T\mathbf{x}=0$). This gives four solutions (two each for the two lines), and we select the (real) point closest to $\mathbf{a}$ as the orthogonal point. In figure 3, $\mathbf{x_p}$ is the common pole, $\mathbf{x}$ and $\mathbf{y}$ are the common intersection points of the pencil of conics and $\mathbf{l}$ and $\mathbf{m}$ are the extracted lines.

3.3 Conic Fitting Algorithm

From the method discussed above, the orthogonal distance from the i^{th} point (say $\mathbf{a}_i$) of a set of n points, can be determined directly for a general conic. The next step is to minimize the squared sum of all such distances as shown in equation (1), to find the best fit conic to the given set of points. Note that this is the same quantity that is minimized in the distance based algorithm of (Ahn, 2004) but assumes that the weighting matrix or the error covariance matrix is identity (Ahn et al., 2001). This results in an unconstrained non-linear optimization procedure.

Out of the algorithms available for solving this non-linear optimization problem, we select the Levenberg-Marquardt method (Ma et al., 2004) due to its robustness over other methods such as Gauss-Newton. For each step of the iteration for the Levenberg-Marquardt method, the update is done as shown in equation (18).

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \boldsymbol{\delta}_k \tag{18}$$

where,

$$\boldsymbol{\delta}_k = -(H_k + \alpha_k I)^{-1} J_k^T \mathbf{d}_k \ , \tag{19}$$

J_k is the Jacobian matrix for the k^{th} step, $H_k = J_k^T J_k$ is an approximation to the Hessian matrix, I is the 5×5 identity matrix, δ_k is the update vector, $\mathbf{d}_k$ is the orthogonal distance vector for all points at step k, and α_k is a scalar.

An initial guess for the parameter vector $\mathbf{p}_0$ has to be given, to start the iterative procedure. The selection of this initial guess is done as discussed at the end of this section. The values for the constant α_k is initially set to a small constant (*eg.* $\alpha_0 = 0.01$). For the consequent steps, if the error increases or if the calculated step yields an imaginary conic, the step is repeated with $\alpha_k = n_{const} \times \alpha_k$. If the error is decreased, the next step is calculated with $\alpha_k = \alpha_k / n_{const}$, where, n_{const} is a scalar constant (*eg.* $n_{const} = 10$).

To calculate the Jacobian matrix for each step, in the conventional way, it is required that the orthogonal distance be expressed as a function of the conic parameters. The method explained in section (3.1), although very convenient for conics with known parameters, makes it difficult, if not impossible for those with unknown parameters. Therefore, an alternative method is used to find the Jacobian matrix, or the matrix of first derivatives of the orthogonal distances with respect to the conic parameters.

First, equation (8) is expressed in the form of equation (20), the derivative of which, with respect to the conic parameter vector $\mathbf{p}$, leads to equation (21).

$$
\begin{aligned}
d^2 &= \| \tilde{\mathbf{x}} - \tilde{\mathbf{a}} \|^2 \\
&= (\tilde{\mathbf{x}} - \tilde{\mathbf{a}})^T (\tilde{\mathbf{x}} - \mathbf{a})
\end{aligned} \tag{20}
$$

$$
d\frac{\partial d}{\partial \mathbf{p}} = (\tilde{\mathbf{x}} - \tilde{\mathbf{a}})^T \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{p}} \tag{21}
$$

Note that for each step, the orthogonal distance and orthogonal point is known. Therefore, the only unknowns in equation (21) are $\frac{\partial d}{\partial \mathbf{p}}$ and $\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{p}}$. To find values for the latter in terms of known quantities, the equations in (13) are differentiated with respect to the parameter vector $\mathbf{p}$, resulting in:

$$
2 \begin{bmatrix} \mathbf{x}^T C \\ \mathbf{x}^T D \end{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} = - \begin{bmatrix} \mathbf{x}^T \frac{\partial C}{\partial \mathbf{p}} \mathbf{x} \\ \mathbf{x}^T \frac{\partial D}{\partial \mathbf{p}} \mathbf{x} \end{bmatrix} \tag{22}
$$

Since $\mathbf{x} = [\tilde{\mathbf{x}} \;\; 1]^T$, equation (22) can be rewritten to get equation (23).

$$
2 \begin{bmatrix} \mathbf{x}^T \bar{C}^T \\ \mathbf{x}^T \bar{D}^T \end{bmatrix} \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{p}} = - \begin{bmatrix} \mathbf{x}^T \frac{\partial C}{\partial \mathbf{p}} \mathbf{x} \\ \mathbf{x}^T \frac{\partial D}{\partial \mathbf{p}} \mathbf{x} \end{bmatrix} \tag{23}
$$

where, $\bar{C}$ and $\bar{D}$ are the 2×3 matrices consisting of the first two rows of C and D respectively.

The solution of equations (21) and (23) gives an expression for the first derivative of the orthogonal distance of a point with respect to the parameter vector as follows.

$$
\frac{\partial d}{\partial \mathbf{p}} = \begin{cases} \dfrac{(\tilde{\mathbf{x}} - \tilde{\mathbf{a}})^T}{d} S^{-1} \mathbf{s} & , \quad \forall \;\; \tilde{\mathbf{x}} \neq \tilde{\mathbf{a}} \\ \mathbf{0} & , \quad \text{otherwise} \end{cases} \tag{24}
$$

where, $S = 2 \begin{bmatrix} \mathbf{x}^T \bar{C}^T \\ \mathbf{x}^T \bar{D}^T \end{bmatrix}$, and $\mathbf{s} = - \begin{bmatrix} \mathbf{x}^T \frac{\partial C}{\partial \mathbf{p}} \mathbf{x} \\ \mathbf{x}^T \frac{\partial D}{\partial \mathbf{p}} \mathbf{x} \end{bmatrix}$

Equation (24) gives the 1×5 partial derivative vector of the orthogonal distance at each point. By stacking all such vectors corresponding to n points in a matrix, the $n \times 5$ Jacobian matrix is obtained. The $(k+1)^{th}$ step can then be calculated by substituting the value of the Jacobian at the k^{th} step in equation (18). An iterative minimization is then carried out until the update vector $\boldsymbol{\delta}_k$ reaches some threshold, indicating that a minimum is reached, and that further iteration does not significantly affect the results.

As an initialization to start the iteration, we suggest the use of the RMS (root mean squared) circle (as used in the circle fitting algorithm of (Ahn et al., 2001)). It uses the root mean squared central distances as its radius r and the center of gravitation as its center $\tilde{\mathbf{x}}_c$, as shown in the following equations.

$$
C_0 = \begin{bmatrix} I & -\tilde{\mathbf{x}}_c \\ -\tilde{\mathbf{x}}_c^T & \tilde{\mathbf{x}}_c^T \tilde{\mathbf{x}}_c - r^2 \end{bmatrix} \tag{25}
$$

where, $\tilde{\mathbf{x}}_c = \frac{1}{n} \sum_{i=1}^{n} \tilde{\mathbf{a}}_i$, $r = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \| \tilde{\mathbf{a}}_i - \tilde{\mathbf{x}}_c \|^2}$, and $\tilde{\mathbf{a}}_i$ is the i^{th} point in non-homogeneous coordinates.

As the proposed general conic fitting method does not require much precision in the initialization, the circle shown above can very conveniently be used as the initial guess. An example of a conic fitting, with this initialization is shown in figure 4(a), along with the square root of the squared sum of orthogonal distances, from the points to the estimated conic at each step of the optimization in figure 4(b).

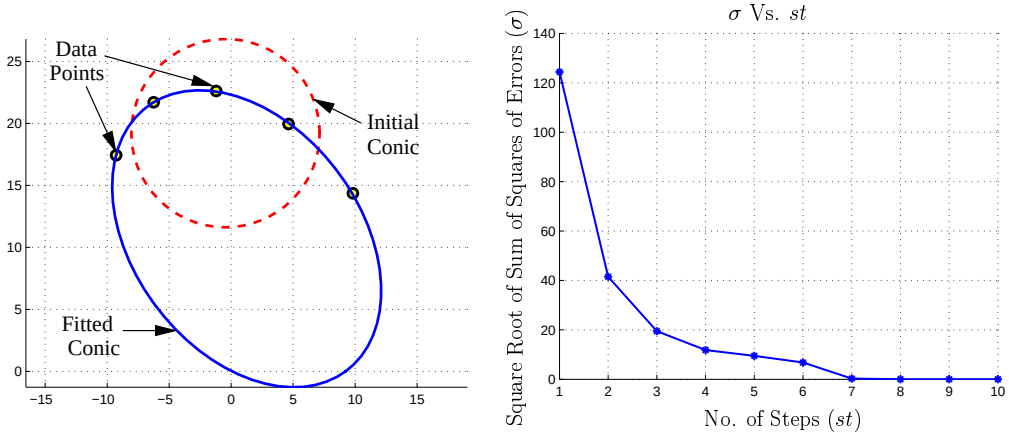
4 EXPERIMENTAL RESULTS

First, we evaluated the performance of the algorithm for points on randomly generated conics. For this experiment, 1000 random conics were generated, and 5, 10, 15, 20 and 25 points on these conics were selected randomly. Then, the proposed general conic fitting algorithm was run on the points. The type of conic was not restricted in any way, except that a check was done for imaginary conics, and in a situation where one was generated, it was discarded, and another random conic generated in its place. Table 1 shows the results of the fitting, where all the values are averaged over the 1000 random conics. σ is the square root of the mean squared orthogonal distance, while σ_n is the same error normalized over the number of points, calculated for the sake of comparison where different numbers of points are involved in the fitting.

Table 1 shows that the mean error per point σ_n on average (for all 5000 cases) is less than 0.3 pixels. Furthermore, the results indicate that the number of fits that have an error less than 0.001, 0.01, 0.1, and 1 are 65.72%, 80.88%, 92.82%, and 96.74% respectively. The average number of iterations to convergence is approximately 16 steps. For a general conic fitting algorithm, these results are very accurate, and the speed of convergence is also acceptable (see (Ahn et al., 2001)). Figure 5 shows how the proposed algorithm performed on some points on random conics of different types.

Next, we compared the performance of the proposed conic fitting algorithm with other orthogonal distance fitting methods. To this end, first, we compare the time complexities of existing algorithms, iterative and non-iterative in their method of determining the orthogonal distance. Then, we focus on the performance of the proposed algorithm with others that calculate the orthogonal distance directly (but using conic specific information).

The general orthogonal distance fitting algorithm



(a) Initial and Fitted Conics

(b) Error at each Step of the Optimization

Figure 4: Fitting of a Conic

Points	Avg. σ (pixels)	Avg. σ_n (pixels)	$\sigma_n < 0.001$ (%)	$\sigma_n < 0.01$ (%)	$\sigma_n < 0.1$ (%)	$\sigma_n < 1$ (%)	Avg. Steps
5	1.2505	0.2501	78.1	85.5	92.4	97.4	17.29
10	1.7653	0.1765	64.6	81.3	91.6	96.2	16.33
15	4.5601	0.3040	60.6	79.3	92.5	95.9	17.11
20	4.7040	0.2352	65.0	81.0	95.3	97.5	15.26
25	7.7385	0.3095	60.3	77.3	92.3	96.7	14.60
Avg.	N/A	0.2551	65.72	80.88	92.82	96.74	16.12

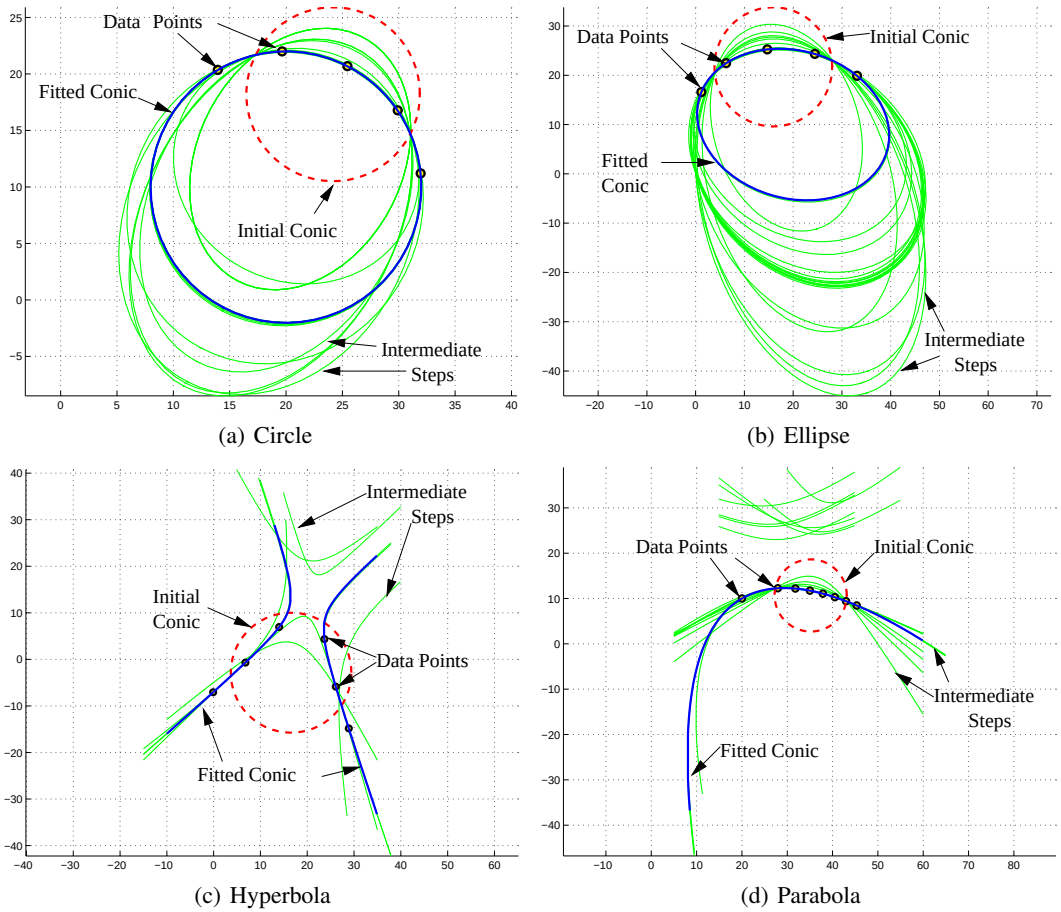
Table 1: Results for the Fitting of Random Conics

introduced in (Boggs et al., 1987) determines the model parameters and the orthogonal points simultaneously and has a time/space complexity of $O(n^2)$, while (Helfrich and Zwick, 1993) present a nested iteration scheme with a time complexity of $O(n)$. The general nested iterative method discussed in (Ahn, 2004) has similar time/space complexities. The proposed algorithm, also has time and memory usage proportional to the number of data points $O(n)$, but removes the nested iteration scheme of (Ahn, 2004) by calculating the orthogonal distance non-iteratively. The type specific direct fitting methods such as (Ahn et al., 2001), (Gander et al., 1994), and (Spath, 1995) also have the same time complexity and require a non-nested iteration for the fitting.

In light of the similarity in terms of the type of iteration (nested or not) and time/space complexity, the performance of the proposed algorithm can be evaluated against other direct fitting methods (Ahn et al., 2001; Gander et al., 1994; Spath, 1995). With respect to the others, the proposed method has a clear advantage in that, it has a time/space complexity of $O(n)$ and that it uses a non-nested iterative scheme. It should also be noted that, in most of these methods, accurate initial guesses are required for good performance. Therefore, algebraic (or in some cases, orthogonal) fitting is required to provide the initialization, whereas the proposed algorithm is more robust and requires only a loose initialization in the form of the RMS circle, irrespective of the type of conic the data resembles, as shown in the previous experiment.

Table 2 summarizes the results of the comparison of the direct conic fitting algorithms with respect to the data sets in (Gander et al., 1994) and (Spath, 1995). The data sets are for different conic types, and therefore, only the relevant algorithms were run on specific data sets. For example, the data sets provided for ellipse fitting were used only on ellipse specific algorithms. On the contrary, as the proposed algorithm is type independent, it was run on all data sets.

Further, to make the comparison more consistent, the results of the same algorithm, which uses different initializations, were averaged. For example, in the circle fitting of (Ahn et al., 2001), two initializations were used: the RMS circle, and an algebraically fitted circle. The results in table 2 shows the average performance of the two. This is done in an attempt to make the comparison more independent of the type



(a) Circle (b) Ellipse

(c) Hyperbola (d) Parabola

Figure 5: Fitting of Different Types of Conics using the Proposed Algorithm

of initialization used. For details on individual performance on various data sets with different initializations, refer to (Wijewickrema et al., 2006) and (Ahn et al., 2001).

Note that the error σ_{avg} is the square root of the mean squared orthogonal distance to the fitted conic, and $\| \Delta \mathbf{p} \|_{avg}$ is the average of the norm of the parameter update vector. As seen from the results, even dedicated fitting methods do not achieve the accuracy of the general algorithm proposed here, which scores the lowest value of $\sigma_{avg} = 0.8813$, over all others. The number of iteration steps, although higher than that of the others, is acceptable considering the rough initialization and type independent nature of the algorithm.

Therefore, from the results, it is seen that the proposed orthogonal distance fitting method performs exceptionally well on noisy as well as noiseless data.

However, it should be noted that, for relatively high levels of noise, the algorithm sometimes converges to local minima, as is the case with algorithms of its kind. To avoid this, a better initialization such as an algebraic fit can be used with the same algorithm.

5 CONCLUSION

In this paper, we have introduced a novel algorithm of orthogonal distance least squares fitting for general conics. We have then illustrated the accuracy and robustness of the algorithm by testing it on both noiseless and noisy data and comparing with other direct conic fitting methods. The main strength of the method is the simplicity of calculation and the fact that it can be used on any type of conic. The advantages of this proposed method as opposed to the existing methods are as follows:

1. Conic (type) specific information is not required
2. Achieves higher accuracy with a similar (or slightly higher) number of iterations
3. Simpler calculation of the orthogonal point and the Jacobian Matrix

Algorithm	Type	Avg. Steps	$\| \Delta \mathbf{p} \|_{avg}$	Error σ_{avg} (pixels)
Proposed	General	13.1	4.3×10^{-6}	0.8813
Ahn	Circle	8.5	4.5×10^{-6}	1.1080
Gander	Circle	11	2.1×10^{-6}	1.1080
Späth	Circle	116	1.0×10^{-4}	1.1080
Ahn	Ellipse	16.5	4.03×10^{-6}	1.2581
Gander	Ellipse	71	1.0×10^{-6}	1.1719
Späth	Ellipse	30	1.0×10^{-4}	1.4306
Ahn	Hyperbola	6.4	3×10^{-6}	0.8899
Späth	Hyperbola	100	1.0×10^{-4}	1.2532
Ahn	Parabola	11.7	5.9×10^{-6}	1.9263
Späth	Parabola	53.5	5.3×10^{-5}	2.1854

Table 2: Comparison of Direct Orthogonal Distance Fitting Algorithms

4. No prior fitting (algebraic or geometric) is required to obtain initial values (the same simple procedure can be used to obtain the initial values when fitting any conic)

Hence, we conclude that the above discussed algorithm is powerful and robust (in terms of requiring only a loose initialization in cases of relatively low noise) and in its simplicity, is quite suitable to be used in general conic fitting.

REFERENCES

Ahn, S. J. (2004). *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*. Lecture Notes in Computer Science, Vol.3151, Springer.

Ahn, S. J., Rauh, W., and Warnecke, H. J. (2001). Least squares orthogonal distance fitting of circle, sphere, hyperbola and parabola. *Pattern Recognition*, 34:2283–2303.

Boggs, P. T., Byrd, R. H., and Schnabel, R. B. (1987). A stable and efficient algorithm for nonlinear orthogonal distance regression. *SIAM Journal of Scientific and Statistical Computing*, 8(6):1052–1078.

Chaudhuri, B. B. and Samanta, G. P. (1991). Elliptic fit of objects in two and three dimensions by moment of inertia optimization. *Pattern Recognition Letters*, 12(1):1–7.

Faber, P. and Fisher, R. B. (2001). Euclidean fitting revisited. *Lecture Notes in Computer Science*, 2059:165–172.

Fitzgibbon, A. W. and Fisher, R. B. (1995). A buyer's guide to conic fitting. In *British Machine Vision Conference*, pages 513–522, Birmingham , UK.

Gander, W., Golub, G. H., and Strebel, R. (1994). Least-squares fitting of circles and ellipses. *BIT*, 34:558–578.

Gauss, C. F. (1963). Theory of the motion of heavenly bodies moving about the sun in conic sections (theoria motus corporum coelestium in sectionibus conicis solem ambientium). *First published in 1809, Translation by C. H. Davis. New York: Dover*.

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Helfrich, H.-P. and Zwick, D. (1993). A trust region method for implicit orthogonal distance regression. *Numerical Algorithms*, 5:535–545.

Hough, P. V. C. (1962). Methods and means for recognizing complex patterns. *US Patent 3 069 654*.

Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S. (2004). *An Invitation to 3D Vision*. Springer.

Miller, J. R. (1988). Analysis of Quadric Surface based Solid Models. *IEEE Computer Graphics and Applications*, 8(1):28–42.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1992). *Numerical Recipes in C*. Cambridge University Press, 2nd Edition.

Semple, J. G. and Kneebone, G. T. (1956). *Algebraic Projective Geometry*. Oxford University Press.

Spath, H. (1995). Orthogonal squared distance fitting with parabolas. In *G. Alefeld, J. Herzberger (Eds.), Proceedings of the IMACS-GAMM International Symposium on Numerical Methods and Error-Bounds, University of Oldenburg*, pages 261–269.

Taubin, G. (1991). Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138.

Wijewickrema, S. N. R., Papliński, A. P., and Esson, C. E. (2006). Orthogonal distance fitting revisited. Technical Report 2006/205, Clayton School of Information Technology, Monash University, Melbourne, Australia.

Young, J. W. (1930). *Projective Geometry*. The Mathematical Association of America.