A fast robust geometric fitting method for parabolic curves

Ezequiel López-Rubio[a], Karl Thurnhofer-Hemsi[a,*], Elidia Beatriz Blázquez-Parra[b], Óscar David de Cózar-Macías[b], M. Carmen Ladrón de Guevara-Muñoz[b]

[a]*Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071, Málaga, Spain*
[b]*Department of Graphical Engineering, Design and Projects, University of Málaga, C/ Doctor Ortiz Ramos, 29017, Málaga, Spain*

Abstract

Fitting discrete data obtained by image acquisition devices to a curve is a common task in many fields of science and engineering. In particular, the parabola is some of the most employed shape features in electrical engineering and telecommunication applications. Standard curve fitting techniques to solve this problem involve the minimization of squared errors. However, most of these procedures are sensitive to noise. Here, we propose an algorithm based on the minimization of absolute errors accompanied by a normalization of the directrix vector that leads to an improved stability of the method. This way, our proposal is substantially resilient to noisy samples in the input dataset. Experimental results demonstrate the good performance of the algorithm in terms of speed and accuracy when compared to previous approaches, both for synthetic and real data.

Keywords: parabolic fitting, geometric curve fitting, noise, minimization of absolute errors, robust estimation

1. Introduction

Fitting of conic sections such as parabola, hyperbola and ellipse is a fundamental task in digital image analysis, pattern recognition, computer graphics, computer vision, reverse engineering and statistics. In these fields, fitting geometric primitive models is especially significant for reverse engineering, whose core is to reconstruct the curve from scattered data, and request the fitting curve reflecting the shapes and features of the original data points [1, 2, 3, 4, 5, 6, 7, 8]. Also, there are some records in the literature on the fitting of quadratic curves and surfaces to sampling points that occurs frequently in several metrology,

*Corresponding author
Email address: `karlkhader@lcc.uma.es` (Karl Thurnhofer-Hemsi)

astronomy, medical diagnosis, geology [9, 10, 11, 12, 13], industrial and architectural applications [14, 15]. For example, a fitting of conic sections was proposed for biological application, such as the chromosome shape analyzed by [16].

In recent years, there are some research works in medical field. To cite an instance, a study on teeth [17]. Another example in this field is related with the corneal shape. A direct mathematical approach for obtaining relevant parameters of corneal surface was proposed in [18]. Other study connected to eyelid location using image focus has been carried out by [19]. According to the anatomic research, parabola-like vascular geometry model is the most common way used to detect fovea [20, 21, 22]. Other medical applications that have been studied are presented by [23] who intended to solve the segmenting ribs problem in a chest radiography image as an halfway step for eliminating rib shadows for an effective Computer-Aided Diagnosis System (CAD). The proposed system facilitated a novel strategy to fit a parabolic curve to all rib seeds acquired by a log Gabor filtering approach. In the same problem as image diagnosis for prostate cancer recognition, [24] proposed a semi-quantitative model to represent perfusion behavior of 3-dimensional prostate voxels in DCE-MRI sequences based on parametric evaluation of parabolic polynomials. Perfusion data of each prostate voxel is modeled on to a best fit parabolic function using second order non-linear regression.

Moreover, telecommunications and industrial applications are presented by [25, 26, 27, 28, 29]. In [27], the problem of fitting a rotated paraboloid to given measured data in 3-space has been discussed. Furthermore, [28] claimed that one of the most important issues for engineers implicated in the structural design of a large antenna is compensation for degrading electromagnetic efficiency. Others industrial applications have been carried out by [29]. They integrated the analogy of gray value of power lines into particle filtering to track the points on power lines, and use those selected points to fit the power line as a parabola.

On the other hand, the detection of automated lane is an essential part of driver assistance systems in smart vehicles [30, 31]. For example, [32] proposed a new kind of lane boundary detection algorithm based on parabola model in order to improve lane detection accuracy under different road conditions for intelligent vehicles. Another research presented by [33] showed a multilane detection method based on omnidirectional images to address the difficulties from the limited view field of the rectilinear cameras. In physics field as detecting in [26] was presented an accurate measurement method for optics system with lunar imaging. The results show that the method is highly accuracy and frequency for focal length measurement.

An atmospheric application related to sound propagation can be found in [25]. Parabolic equations, approximations of the Helmholtz equation in cylindrical coordinates, are used widely in underwater acoustics context, see [34, 35]. They have been further employed to simulate the analog problem of sound propagation in an inhomogeneous atmosphere. Finite element [36, 37], finite difference [38, 39], and Fourier/Green's function methods [40, 41], have been used to discretize the parabolic equations in cylindrically symmetric domains.

In parabolic equation models, a waveguide with irregular boundaries can be encountered. In computational fluid dynamics [42], a common practice to transform the physical domain to a rectangular computational one, is the use of a boundary fitting curvilinear coordinate system. In [43] a similar analytical transformation of coordinates was used in a finite element discretization of the conventional parabolic equation in a sea environment with variable bottom. Irregular terrain is also encountered in electromagnetic parabolic equation models [44].

Finally, an example in the architectural field is showed in [15], which provides a method that expect to determine the best fits to a geometric shape of an arch of a heritage building. This method only involves standard geometric processes, computing, statistics, numerical processes and data acquisition.

Most of the current state of the art methods to fit parabolas are based on the minimization of squared errors. Such methodologies have little resilience to noisy observations, due to the excessive sensitivity of the squared error function to them. This problem is further exacerbated by the fact that for many input datasets there are spurious solutions that the fitting algorithms can easily fall into. Here we aim to develop a method to fit parabolas from a set of sample points which is more robust, based on the minimization of absolute errors. The main contribution of this work is the proposal of a fitting algorithm which is resilient both to outlying input points, by means of absolute error minimization, and to the presence of spurious solutions, and the introduction of random restarts in the search. Moreover, our proposal attains a good balance between the quality of the solution and the execution speed.

This paper is structured as follows. Section 2 presents previous works and the definition of earlier parabola fitting methods. Subsequently, Section 3 describes the formulation of our method and Section 4 reports the obtained experimental results using synthetic and real data. Finally, Section 5 summarizes the main conclusions of this research.

2. Previous work

Fitting quadratic curves occurs frequently in computer vision, pattern recognition, and image processing applications. Given a parametric function

$$P(x, g) = 0 \tag{1}$$

where P represents a quadratic function, x stands for the point coordinate vector, and g denotes curve parameters, the main purpose is to estimate g from noisy samples

$$x_i = \bar{x}_i + \tilde{x}_i \tag{2}$$

where $\bar{x}_i$ is noiseless data and $\tilde{x}_i$ is a noise contribution such that for all $\bar{x}_i$ we have

$$P(\bar{x}_i, g) = 0 \tag{3}$$

One of the most accurate approaches to estimate g in Eq. (1) is with maximum likelihood (ML) methods, where the geometric distance of data points to the curve P is minimized. These methods are typically more computationally expensive and usually converge slowly if it is formulated as an iterative process. Traditional least squares methods (LSM), in contrast, which minimize the squared equation error, also called as algebraic fit, are cheaper and involve no iterations but may be heavily biased and non-robust systems [45, 46].

The main objective of [47] is to (1) use the simplicity of the algebraic methods, avoiding (2) the statistical inaccuracy in least squares approaches, while (3) enforcing constraints that arise from a priori information at a low computational cost. They work with images where data points are polluted by a large amount of Gaussian noise.

As it has been mentioned before, for the conic fitting, there are two problems to be considered [48]. One is the mathematical form of the formula for the fitting conic, and the other is the objective function of the fitting conic. The selection of the objective function, during the fitting of the curve, can be divided into two categories: firstly, based on minimizing the geometric distance from the point to the curve and secondly, based on minimizing the algebraic distance.

The general form of a conic section is:

$$Q\left(x,y\right) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{4}$$

The roots of Eq. (4) evaluated at infinity allow to identify the type of conic [49].

Explicitly,

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{5}$$

Through (5), the type of conics can be characterized as follows:

$$B^2 - 4AC \begin{cases} > 0\, real\, asymptotes \Rightarrow hyperbola \\ = 0\, real\, parrallel\, asymptotes \Rightarrow parabola \\ < 0\, complex\, asymptotes \Rightarrow ellipse \end{cases} \tag{6}$$

The discriminant $B^2 - 4AC = 0$ depicts a second order surface in the (A,B,C) tridimensional parameter space, i.e., it is an elliptical cone with the origin as its vertex. This surface represents all parabolas. All ellipses are enclosed inside the cone, whereas all hyperbolas are external to it.

According to Bookstein [9] the invariance must be with respect to transformations of the Euclidean plane, such as rotations, translations and scale changes. If the dataset coordinates are transformed then the resulting best fitting conic should match the best fitting conic that would be obtained by the algorithm if the non transformed samples were provided to the algorithm. The Yu method [50] has a cost function with geometric interpretation where the parameters are intrinsic of the conic and are translation and rotation invariant. Our method is also invariant to these transformations.

Many proposals can be found about parabola fitting in the literature, but there are few state-of-art methods with available source code that can be used in order to obtain interesting experimental results.

Harker [49] considers that the Direct method is the first known direct solution for parabola fitting. This fitting is reached by establishing an orthogonal basis vector set in the Grassmannian space of the quadratic terms' coefficients. The linear combination of the basis vectors that satisfied the parabolic condition and has a minimum residual norm is resolved using Lagrange multipliers.

Cals Method fits a parabola using Consistent Algebraic Least Squares (CALS). That implies a pre-processed scatter matrix for a Direct least squares fitting. An improved matrix partitioning is employed, through an extension of Halíř and Flusser's work [51]. A generalization of the Eckart-Young-Mirsky matrix approximation theorem allows for an incremental orthogonal residualization of the partitioned scatter matrix [52].

García [3] has presented a method to fit parabolas to scattered data which is applicable in noisy images. They employ a trimming procedure and avoids trying to fit all the points in the data set, but only a proportion $1-\alpha$ of them. This approach has the disadvantage that it does not only need the samples as input parameters, but also the trimming level α and two step parameters (L updating steps and S random initializations) for the iterative process, which depend on the characteristics of the dataset.

2.1. Geometric distance

The geometric distance from a point $(x, y) \in \mathbb{R}^2$ to a conic is the smallest Euclidean distance from the point to all the points in the conic [53]:

$$d_G(x, y) = \min\left\{\|(x, y) - (r, s)\| \mid (r, s) \in \mathbb{R}^2,\ Q(r, s) = 0\right\}$$

where $\|\cdot\|$ stands for the Euclidean distance.

The objective function based on minimizing the geometric distance is a 4th order equation. This can be solved by mathematical methods, but it needs to use non-linear procedures and the results are not quite stable [54, 55]. So the applications are somehow limited.

2.2. Algebraic distance

The algebraic distance from a point $(x, y) \in \mathbb{R}^2$ to a conic is given by [55]:

$$d_A(x, y) = Q^2(x, y)$$

The minimizing algebraic distance method has been used by several researchers. For example, [56] used six different restrictions to obtain six basic conics, and produced the final fitting conic by adding certain weights to the coefficients of the six basic conics. Rosin discussed the objective functions based on minimizing algebraic distance from the aspects of curvature bias, singularities, and transformational invariance [57]. On their part, [9, 58, 59] introduce different constraints to the objective function, transformed it to the extreme problem, and obtained varied results depending on the restriction. Furthermore, [49] used the partition matrix to fitting a specific conic and introduced a bias correction method. Based on the aforecited researches, a conic fitting method is provided in Li's work [48], which minimizes the point-to-curve algebraic distance for the given data points. This proposed method can preserve the original outline of the conic data points, whereas the fitting effects are improved.

3. The model

Next our robust parabola fitting method is presented. The equation of a parabola in 2D is given by:

$$\frac{(ax+by+c)^2}{a^2+b^2} = (x-u)^2 + (y-v)^2 \tag{7}$$

where the directrix is $ax+by+c=0$ with a and b not both zero, and the focus point is $(u,v) \in \mathbb{R}^2$. Please note that at this point it is not advantageous to normalize the directrix vector, $a^2+b^2=1$, since such normalization would force to express either a or b in terms of the other, for example $b=\sqrt{a^2-1}$, which would clutter the subsequent derivations.

We may write the five parameters which define the parabola in vector form:

$$\mathbf{p} = (a,b,c,u,v)$$

Please note that the left hand side of (7) is the squared distance of point $(x,y) \in \mathbb{R}^2$ to the directrix, and the right hand side of (7) is the squared distance of point $(x,y) \in \mathbb{R}^2$ to the focus point. Therefore we can rewrite (7) as:

$$E_d(x,y) = E_f(x,y)$$

$$E_d(x,y) = \sqrt{\frac{(ax+by+c)^2}{a^2+b^2}} \tag{8}$$

$$E_f(x,y) = \sqrt{(x-u)^2 + (y-v)^2} \tag{9}$$

where $E_d(x,y)$ is the distance of the test point (x,y) to the directrix, and $E_f(x,y)$ is the distance of the test point of the focus point. Furthermore, the plane is divided into two regions:

$$R_d = \left\{(x,y) \in \mathbb{R}^2 \mid E_d(x,y) \le E_f(x,y)\right\} \tag{10}$$

$$R_f = \left\{(x,y) \in \mathbb{R}^2 \mid E_f(x,y) < E_d(x,y)\right\} \tag{11}$$

where R_d contains the points which are closer to the directrix than to the focus point, or at the same distance, and R_f contains the points which are closer to the focus point than to the directrix (see Figure 1).

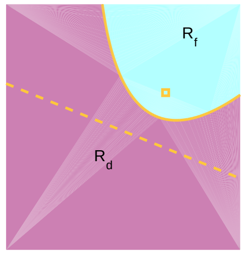


Figure 1: Depiction of the R_d and R_f regions. The directrix of the parabola is plotted with a dashed line, and the focus is drawn as a square.

Now we may consider the minimization of the following cost function:

$$\mathcal{E}\left(\mathbf{p}\right) = \frac{1}{N}\sum_{i\in R_d}\left(E_f\left(x_i,y_i\right) - E_d\left(x_i,y_i\right)\right) + \\ \frac{1}{N}\sum_{i\in R_f}\left(E_d\left(x_i,y_i\right) - E_f\left(x_i,y_i\right)\right) - \lambda E_d\left(u,v\right) \tag{12}$$

where N is the number of training points, the last term is introduced in order to avoid degenerate solutions with the focus on the directrix, and λ is an adjustable penalty parameter, $\lambda > 0$. Please note that the last term includes the distance from the focus point to the directrix, so that the degenerate solutions which must be avoided have $E_d\left(u,v\right) = 0$. Nondegenerate parabolas have $E_d(u,v) > 0$ which leads to $-\lambda E_d(u,v) < 0$. Degenerate parabolas with $E_d(u,v) = 0$ lead to $-\lambda E_d(u,v) = 0$ which is the maximum, i.e. worst since we are minimizing $\mathcal{E}$, possible value of the term $-\lambda E_d(u,v)$. In other words, the value of the term $-\lambda E_d(u,v)$ is always worse (higher) for degenerate parabolas than for nondegenerate parabolas.

The gradient of the error is computed as follows:

$$\frac{\partial\mathcal{E}}{\partial a} = \frac{1}{N}\sum_{i\in R_d}\frac{\left(ax_i+by_i+c\right)\left(ac+aby_i-b^2x_i\right)}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)} \\ -\frac{1}{N}\sum_{i\in R_f}\frac{\left(ax_i+by_i+c\right)\left(ac+aby_i-b^2x_i\right)}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)} \\ +\lambda\frac{\left(au+bv+c\right)\left(ac+abv-b^2u\right)}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{13}$$

$$\frac{\partial\mathcal{E}}{\partial b} = \frac{1}{N}\sum_{i\in R_d}\frac{\left(ax_i+by_i+c\right)\left(bc+abx_i-a^2y_i\right)}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)}$$

$$-\frac{1}{N}\sum_{i\in R_f}\frac{\left(ax_i+by_i+c\right)\left(bc+abx_i-a^2y_i\right)}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)}$$

$$+\lambda\frac{\left(au+bv+c\right)\left(bc+abu-a^2v\right)}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{14}$$

$$\frac{\partial\mathcal{E}}{\partial c}=-\frac{1}{N}\sum_{i\in R_d}\frac{ax_i+by_i+c}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)}$$

$$+\frac{1}{N}\sum_{i\in R_f}\frac{ax_i+by_i+c}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)}$$

$$-\lambda\frac{au+bv+c}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{15}$$

$$\frac{\partial\mathcal{E}}{\partial u}=\frac{1}{N}\sum_{i\in R_d}\frac{u-x_i}{E_f\left(x_i,y_i\right)}-\frac{1}{N}\sum_{i\in R_f}\frac{u-x_i}{E_f\left(x_i,y_i\right)}$$

$$-\lambda\frac{a\left(au+bv+c\right)}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{16}$$

$$\frac{\partial\mathcal{E}}{\partial v}=\frac{1}{N}\sum_{i\in R_d}\frac{v-y_i}{E_f\left(x_i,y_i\right)}-\frac{1}{N}\sum_{i\in R_f}\frac{v-y_i}{E_f\left(x_i,y_i\right)}$$

$$-\lambda\frac{b\left(au+bv+c\right)}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{17}$$

It has been found in experiments that keeping the normal vector of the directrix normalized improves the numerical stability of the method, so that the obtained solutions are better. It is also advantageous to restart the search from a random state close to the initial estimation each T steps, because this allows the method to escape from local minima of the cost function $\mathcal{E}$.

The algorithm is:

1. Initialization. The focus point is initialized to the mean of the dataset:

$$\left(p_4,p_5\right)=\left(\frac{1}{N}\sum_{i=1}^{N}x_i,\frac{1}{N}\sum_{i=1}^{N}y_i\right)$$

The directrix is initialized to the straight line which best fits the dataset in the least squares sense:

$$\left(p_1,p_2,p_3\right)=\arg\min_{(a,b,c)}\sum_{i=1}^{N}\left(ax_i+by_i+c\right)^2$$

In order to keep a scale in the parameter space for the gradient descent, the directrix is normalized:

$$(\bar{p}_1, \bar{p}_2, \bar{p}_3) = \left(\frac{p_1}{\sqrt{p_1^2 + p_2^2}}, \frac{p_2}{\sqrt{p_1^2 + p_2^2}}, \frac{p_3}{\sqrt{p_1^2 + p_2^2}} \right)$$

Then a random perturbation drawn from a Gaussian distribution with zero mean and half the standard deviation of the dataset is added to the initial focus point. These are rough estimations, as the method will refine them later. The step counter is initialized to $t = 0$.

2. Determine the errors $E_d(x_i, y_i)$ and $E_f(x_i, y_i)$ for all training points (x_i, y_i) by (8) and (9).
3. Compute which training points belong to R_d and R_f with (10) and (11), where points on the boundary between both regions are assigned to one of them uniformly at random.
4. Find a new solution by gradient descent:

$$\mathbf{p}_{new} = \mathbf{p}_{old} - \eta \frac{\partial \mathcal{E}}{\partial \mathbf{p}}$$

where $\frac{\partial \mathcal{E}}{\partial \mathbf{p}}$ is given by Eqs. (13)-(17).

5. Normalize the directrix (as in Step 1) of the new solution in order to improve the numerical stability of the method.
6. Increase the step counter t. If the current solution has a lower value of the cost function (12), then store it as the best solution so far.
7. If the step counter t is an integer multiple of T, then restart the method by setting the current solution to a randomly perturbed version of the initial non perturbed solution. The random perturbation is carried out by adding a normally distributed random number to each of the components of the initial non perturbed solution of step 1. These random numbers are generated with zero mean and half the standard deviation of the dataset.
8. If convergence has been achieved or a maximum number of steps M has been reached, then return the stored best solution and halt. Otherwise, go to step 2.

The design of step 1 is aimed to obtain a valid solution, i.e. a non degenerate parabola. The random perturbation ensures that the perturbed solution is not degenerate. Input data which are aligned or almost aligned can lead to a degenerate initial parabola. Furthermore, the initial focus and the directrix are chosen to be close to the input data, so that the initial parabola has enough curvature, i.e. it is not too open. On the other hand, the normalization of step 5 ensures that the values of the directrix parameters a, b and c do not grow too large or become too small, which would cause loss of precision in the calculations and would ultimately destabilize the algorithm from the numerical point of view.

The convergence monitoring of step 8 is carried out by analyzing the updates of the current best solution which are done at step 6. Convergence is declared when the current best solution has not been updated for $2T$ steps.

It must be noted that our proposed algorithm might have some resemblances to the Expectation-Maximization (EM) algorithm. In particular, in step 3 of our algorithm it is estimated which side of the parabola each training sample is. This would be similar to the Expectation step, if we interpret that the “missing value” or “latent variable” would be a binary variable for each training sample which says which side of the parabola the sample is. Then in step 4 of our algorithm the five model parameters a, b, c, u, v are updated by assuming that the estimated values of such binary “latent variables” are correct, which is somehow similar to the Maximization step. However, our algorithm also has fundamental differences with respect to Expectation Maximization. First of all, no probabilistic model of the input data is defined or assumed. Therefore, there is no likelihood function or log-likelihood function to be optimized. This means that the EM algorithm cannot be applied to our parabola fitting approach. Secondly, unlike the Maximization step of the EM algorithm, our step 4 does not maximize the cost function given the estimated values of the binary “latent variables”, but only moves the parameters in the direction of the gradient descent.

4. Experimental Results

In this section several experiments are carried out to show the performance of the proposed method. Firstly, Subsection 4.1 briefs the tuning of the parameters of the algorithm. Then, some computational experiments for synthetic datasets are reported in Subsection 4.2. Simulations with and without outliers are presented, but also were analyzed datasets with Gaussian noise, degenerate parabolas and the effect of different initializations. Finally, real examples with data from natural images were used in Subsection 4.3.

Two of the competing approaches mentioned in Section 2 which have been tested are considered in [49], Direct Method and Cals Method. In addition to this, a third method has been used for the comparison, which was called as García Method [3]. For the experiments, fixed values $L = 400$ and $S = 40$ have been adopted following the advice of the authors of this method, and α with a value 0.1 higher than the outlier level, i.e. if we add 10% of outliers, we employ $\alpha = 0.2$. If no anomalous samples are present in the dataset, $\alpha = 0$ was used. Ours, Direct and Cals methods are implemented as Matlab scripts (R2018a), with no use of the GPU. García method is implemented in R, but we linked it with Matlab by the RMatlab package [60] to perform all the comparisons under the Matlab environment.

To measure the quality of the fit for the synthetic datasets, we compute the residual errors, consisting on the mean squared differences between the directrix and focus errors along the sample points:

$$Residual = \frac{1}{N}\sum_{i=1}^{N}\left(E_d\left(x_i, y_i\right) - E_f\left(x_i, y_i\right)\right)^2 \tag{18}$$

If spurious points (outliers) are added in the fitted dataset, these are not used to compute the Residual measure. This is done because the goal of a parabola fitting algorithm is to fit the inliers and not the outliers. Also, we obtained the runtime of all methods on a desktop personal computer with a 64-bit PC, with an Intel Core i7-4790, 3.6GHz CPU, 32 GB RAM and standard hardware.

4.1. Parameter selection

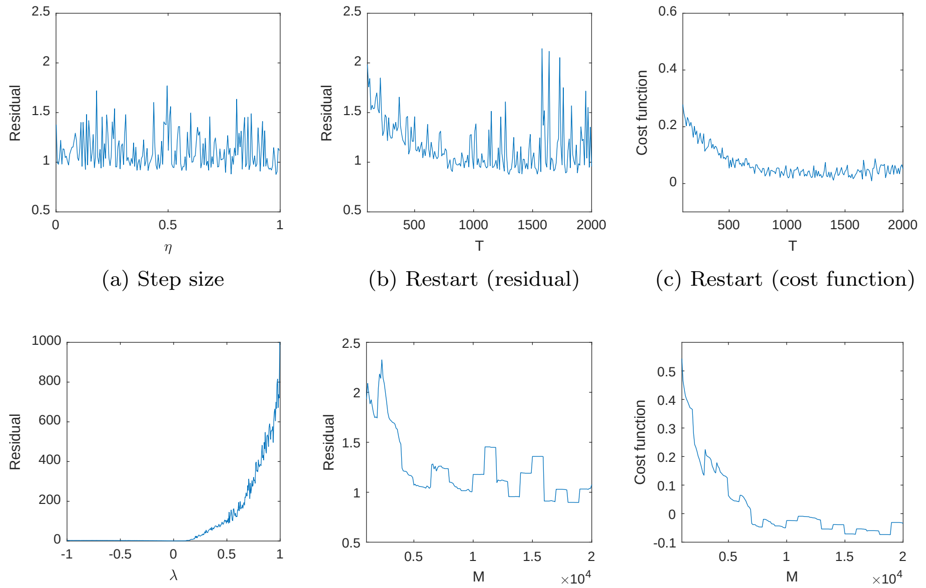
Our algorithm requires setting four parameters, the penalty parameter λ, the gradient descend step size η, the restart parameter T and the maximum number of steps M.

To determine the optimum values for the parameters, synthetic sets of points of a parabola were generated in the same manner as the first set of experiments detailed in Subsection 4.2. Then, we have run 100 executions of our algorithm and the Residual errors (18) were calculated for each execution varying the parameter of study and fixing the others. In addition to this, for the sake of completeness we have analyzed the cost function $\mathcal{E}$ (12) for the T and M parameters since there is no direct dependency on the parameters.

Figure 2 shows the results of the simulations. The average values of the measures along all the executions are depicted. In the case of the Step Size (Figure 2a), no significant differences can be appreciated between almost all possible values. The value 0.1 was chosen because is usually better to set a small value so that the gradient descent avoid abrupt jumps and the convergence is more reliable.

Regarding the restarting parameter T (Figures 2b-2c) and the maximum number of steps M (Figures 2e-2f), it can be seen that there is a moment from which the results stabilize for both parameters. This point was selected as our value for the tunable parameter.

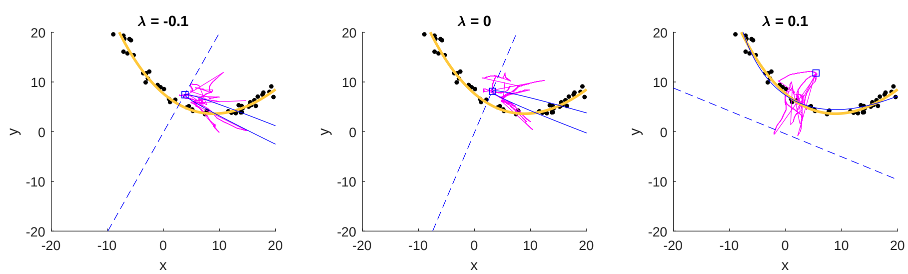
With respect to the penalty parameter (Figure 2d), the testing of negative values of λ has been done for the sake of completeness. The condition $\lambda < 0$ means that we are rewarding degenerate solutions, which is expected to give bad results, but we still want to experimentally show that this is the case. And when $\lambda > 0.15$, the error increases exorbitantly. The penalty parameter is designed to avoid finding a degenerate parabola when the true input parabola is not degenerate. To show the effect of λ we have carried out an experiment where we create an input dataset corresponding to a regular, i.e. non degenerate, parabola, and we have run our algorithm for $\lambda = -0.1$, $\lambda = 0$ and $\lambda = 0.1$. Figure 3 shows that the obtained solutions are degenerated for the first two cases, as it was expected, and the proper solution is reached with the positive value.



(a) Step size (b) Restart (residual) (c) Restart (cost function)

(d) Penalty parameter (e) Max. steps (residual) (f) Max. steps (cost function)

Figure 2: Study of the optimal parameters of our algorithm. Results of the residual errors and cost function for each parameter of our algorithm. Average results of 100 executions are displayed. The penalty parameter λ was varied from -1 to 1, and the gradient descend step size η from 0 to 1, both using a step of 0.005. The restart parameter T was varied from 100 to 2000, using a step of 10, and the maximum number of steps M from 1000 to 20000, using a step of 100. To analyze each parameter, the rest of them where fixed.



(a) Negative value (b) Zero value (c) Positive value

Figure 3: Outcomes of our algorithm for a non degenerate parabola when penalty parameter is $\lambda = -0.1$, $\lambda = 0$ and $\lambda = 0.1$, from left to right. The rest of the parameters where fixed.

Parameter	λ	η	T	M
Value	0.1	0.1	1000	10000

Table 1: Parameter selection of our method

The previous discussion is specific to a particular dataset. However, the resulting parameter selection has worked for all the other tested datasets, so the subsequent experiments validate the parameter values chosen in Subsection 4.1. All the selected parameter values for our method are summarized in Table 1.

4.2. Synthetic data

Our first set of experiments deals with artificially generated data. For each experiment, the focus and the directrix of a parabola are chosen at random:

$$u, v \sim U(0, 20) \tag{19}$$

$$a, b, c \sim U(0, 30) \tag{20}$$

where $U(\alpha, \beta)$ stands for the uniform distribution over the open interval of real numbers (α, β). Then 100,000 candidate points are uniformly generated on the square $[-20, 20] \times (-20, 20)$. Then, we select the points (x_i, y_i) whose which satisfy that:

$$(E_d(x_i, y_i) - E_f(x_i, y_i))^2 < 2 \tag{21}$$

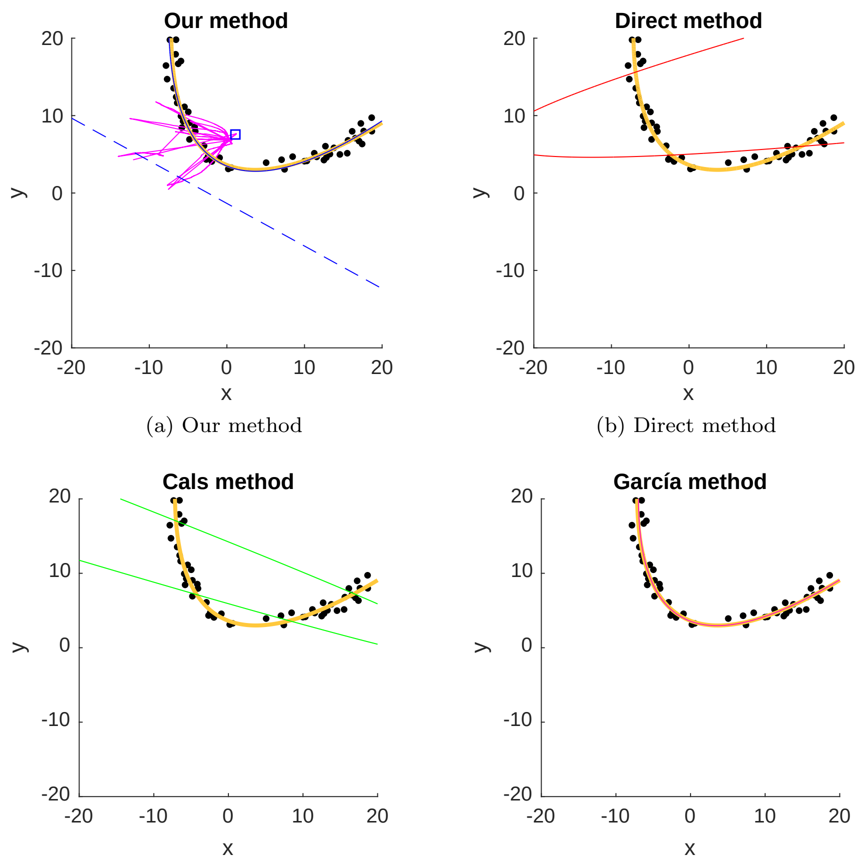
Note that (21) means that the input samples are close to the true parabola, which is given by:

$$(E_d(x_i, y_i) - E_f(x_i, y_i))^2 = 0 \tag{22}$$

but they are not exactly on the true curve, i.e. some noise is present. The value 2 was manually chosen so that the noise was significant but not so large that the parabolic shape was lost. Finally, from this candidates are randomly selected the input datasets with size $N = 50$.

If a percentage of outliers is added in the experiments, then the number of those is calculated as a fraction of N. That is, if we add $p_{out}\%$ of outlying points, then the final number of samples is $N + \frac{p_{out} \cdot N}{100}$. These extra points are randomly selected from those candidate points that do not satisfy (21), i.e., they are points where $(E_d(x_i, y_i) - E_f(x_i, y_i))^2 > 2$.

Next we present the results of the competing methods for some cases. First example is shown in Figure 4. A synthetic dataset was generated in absence of anomalous samples. Comparing all the fits, both ours and García methods yield the best results. As seen in Figure 4a, the trace of the estimated foci sometimes goes far away from the true solution. This is due to the existence of many spurious solutions which have good values of the cost function $\mathcal{E}$. And this is why Direct and Cals methods return incorrect estimations of the parabola, falling in a local minima. However, this issue has been addressed in our proposed algorithm by introducing restarts from randomly perturbed versions of the initial non perturbed solution (step 7 of the algorithm) at regular intervals. These restarts are seen in Figure 4a as line segments which come back to the vicinity of the true focus.



(a) Our method

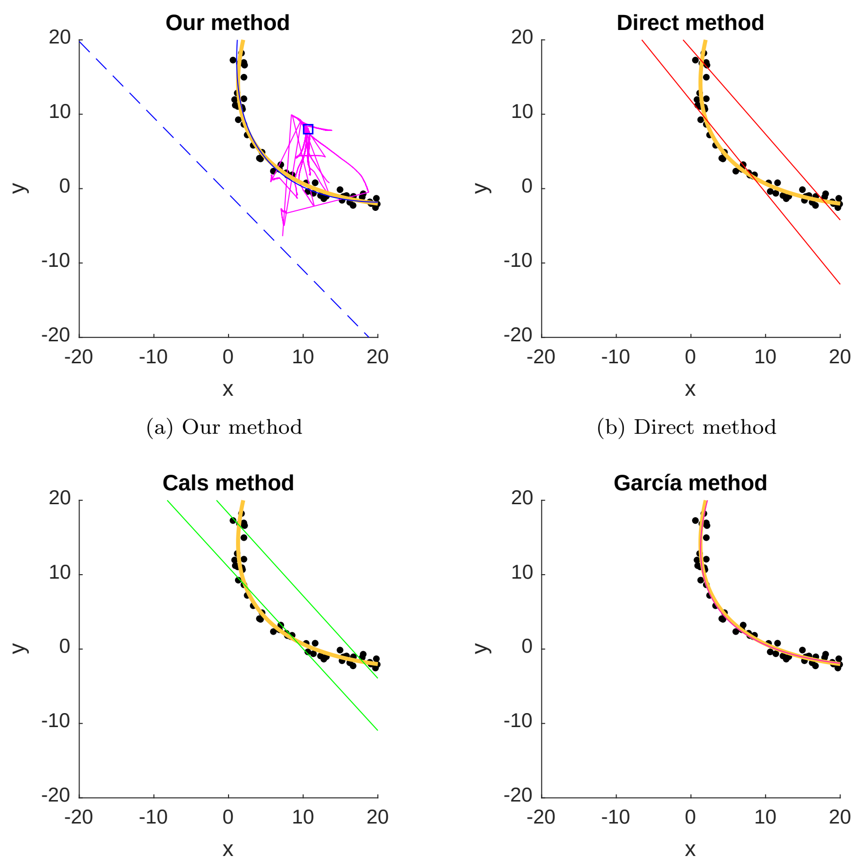
(b) Direct method

(c) Cals method

(d) García method

Figure 4: Solutions for the first synthetic dataset. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

In Figure 5 a second example dataset is presented. This dataset features a higher occlusion level, in the sense that the points corresponds to a small arc of the parabola, becoming difficult to determine which type of conic they are part from, so the fit is exposed to an incorrect fitting. As it is shown Direct and Cals methods yield similar results, neither of them converge to a good solution. The first and second synthetic datasets are similar, but it is relevant to depict their results because they are largely different for the unstable methods (Direct and Cals).

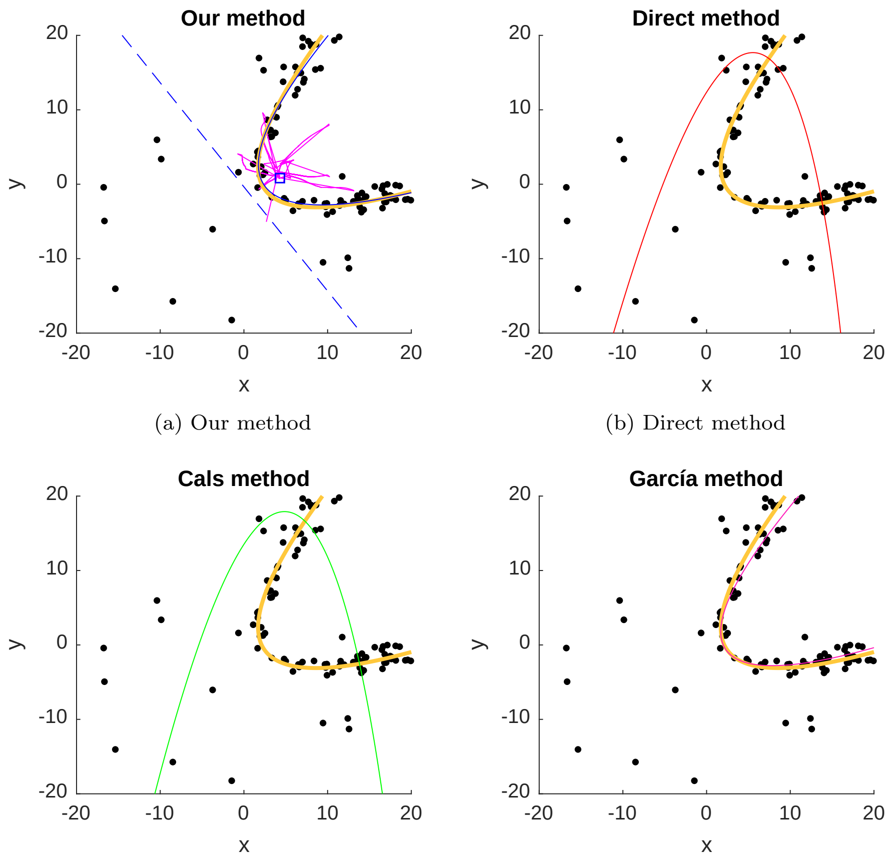


(a) Our method (b) Direct method

(c) Cals method (d) García method

Figure 5: Solutions for the second synthetic dataset. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

The last example dataset is shown in Figure 6, where a 20% of outliers is added to the training data. The results obtained by our method and García method are quite acceptable, while Direct and Cals methods show a clearly wrong outcome. It seems that in most of the examples, our approach have a similar or even better fit to the point cloud than García algorithm.

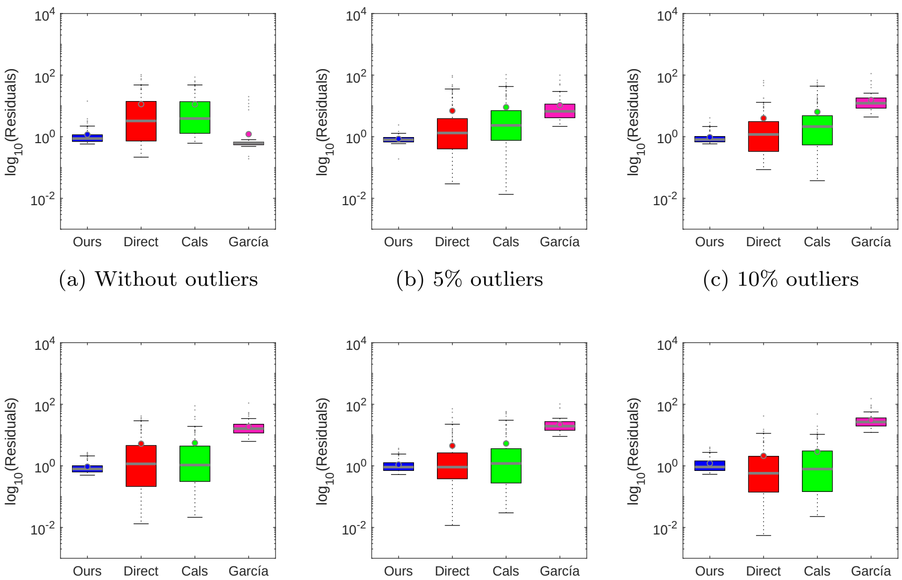


(a) Our method (b) Direct method

(c) Cals method (d) García method

Figure 6: Solutions for the third synthetic dataset. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

In general terms, it can be said that our method exhibits a higher resilience to noise than its competitors for synthetic data. To further confirm this, a set of 100 executions has carried out and the residual errors are shown in Figure 7. Simulations without outliers and with 5%, 10%, 15%, 20% and 25% of added outliers have been done. The outliers are not considered in the computation of the residuals. Boxplots were used to display the results. We have used the Residual measure defined in (18) to compare the results. As it can be seen, in almost all cases our method yields the best results, as the mean squared differences between the errors $E_d\left(x_i, y_i\right)$ and $E_f\left(x_i, y_i\right)$ along the training points are very close to zero, that is, the samples are very close to the fitted parabola. In the case of median values, the different approaches are more competitive but they have a higher deviations, as it can be seen in the higher size of the box.



(a) Without outliers (b) 5% outliers (c) 10% outliers

(d) 15% outliers (e) 20% outliers (f) 25% outliers

Figure 7: Comparative of the four methods along 100 executions with synthetic data. Grey circles and lines represent the mean and median value over all the simulations, respectively. Displayed whiskers are in the range of 5-95%. Results shown in a logarithmic scale..

The results are very similar when we have no outliers (Figure 7a). Clearly, Direct and Cals methods yield inconsistent results, while García approach obtains good performance. Figures 7c and 7b depict the results for adding 5% and 10% of anomalous points to the initial dataset, respectively, which implies the presence of few points that do no belong to the initial selection of points near to the parabola since Eq. (21). For this configuration, the deviation errors are quite similar to that one without outliers with the exception of García method, which suffers big mismatches as the number of outlying points increases. It is remarkable that the residuals, which are computed taking only into account the inliers, preserve the goodness of fit only in our case.

	Ours	Direct	Cals	García
0%	0.333 (0.036)	0.001 (0.005)	0.003 (0.009)	79.554 (1.818)
5%	0.334 (0.012)	0.001 (0.003)	0.003 (0.006)	84.087 (1.651)
10%	0.334 (0.006)	0.001 (0.000)	0.002 (0.000)	86.313 (1.390)
15%	0.3348 (0.006)	0.001 (0.000)	0.002 (0.000)	90.916 (1.584)
20%	0.342 (0.017)	0.001 (0.000)	0.002 (0.000)	93.654 (1.594)
25%	0.347 (0.040)	0.001 (0.000)	0.002 (0.000)	97.469 (1.972)

Table 2: Mean and standard deviation of the running time (in seconds) along the 100 executions for different noise levels.

The statistics when 15%, 20% and 25% of outliers is added to the training points are shown in Figures 7d, 7e and 7f. The outcomes of these three configurations are very similar. Although our method has the best results, Direct and Cals methods reduce the difference in terms of median values. However, these competing methods generate solutions with to much deviation. The mean and standard deviation of our proposal still remain clearly the best. There could be cases where Direct and Cals methods yield small values of the residuals, even if the number of outliers increases. That is because the residuals are computed setting aside outliers.

Table 2 summarizes the computational time required for the execution of the four methods. As we can see, our method is the only one that has an equilibrium between performance and computational efficiency. The mean running time employed by our approach is always less than 0.35 seconds. We can also see that Direct and Cals methods fit their parabolas faster than ours, but with the inconvenience of inconsistent results. García method has demonstrated good accuracy in some cases, but the required time to compute the parabola is excessive, it is around 90 sec., which is one hundred times slower than the other methods.

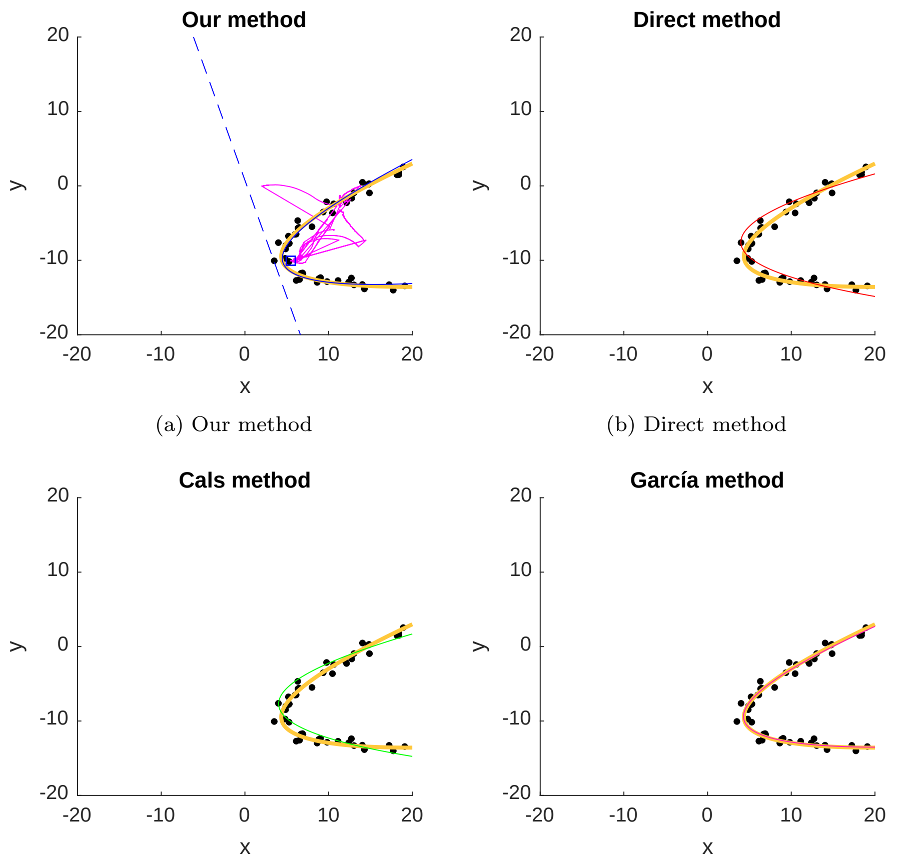
4.2.1. Gaussian distribution

A second set of experiments with synthetic data was carried out. Starting from the unit parabola with equation $y = x^2$, any other parabola can be inferred from this one using an affine transformation of the Euclidean plane in the form $\mathbf{x} \to \mathbf{f_0} + \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ is a regular matrix with column vectors $\mathbf{f_1}, \mathbf{f_2}$, and $\mathbf{f_0}$ is an arbitrary vector. Thus, the general parametric equation of a parabola is defined as

$$\mathbf{x} = \mathbf{f_0} + \mathbf{f_1} t + \mathbf{f_2} t^2,\ t \in \mathbb{R} \tag{23}$$

As before, 100,000 candidate points on the square $[-20, 20] \times (-20, 20)$ are generated following (23) with $\mathbf{A} = \mathbf{R}\Lambda$, where Λ is a diagonal matrix which is multiplied by a 2D rotation matrix $\mathbf{R}$. Then, Gaussian noise with standard deviation 0.5 was added to the points of the parabola. Finally, a subset with size $N = 50$ from this candidates are randomly selected.

In Figure 8 we present an example of an execution with this kind of dataset. Both Ours and García methods yields the best results in comparison with the true parabola. Direct and Cals methods also show a good fit, but not as good as expected since the LS approaches assume normally distributed residuals.



(a) Our method (b) Direct method

(c) Cals method (d) García method

Figure 8: Solutions for a synthetic dataset with Gaussian noise distribution. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

For an adecuate evaluation, we carried out a set of 100 execution with different parabolas generates as explaned before. The results are summarized in Figure 9. Our approach yields great results, specially in terms of small deviations. If we compare the outcomes obtained from the previous experiments (Figure 7), the Direct and Cals methods show lower errors but similar deviations, and García yields higher errors.

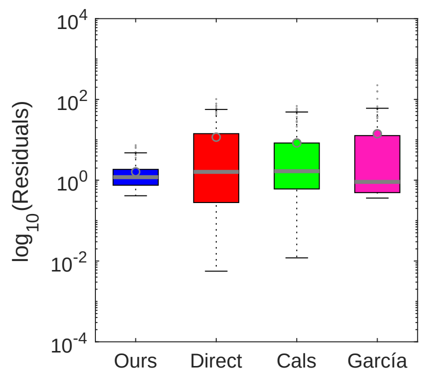
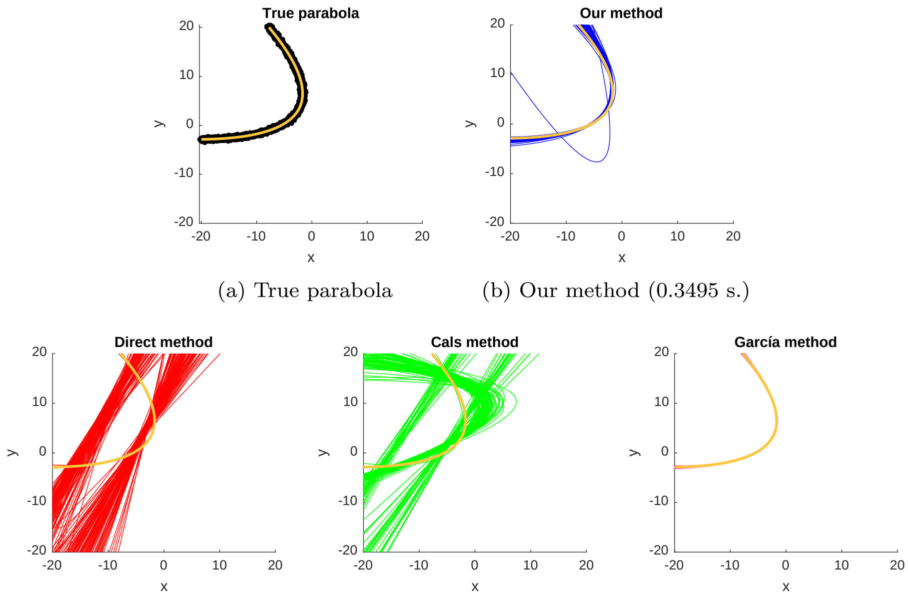


Figure 9: Comparative of the four methods along 100 executions with synthetic data and Gaussian noise distribution. Results shown in a logarithmic scale.

4.2.2. Variability of the fitting

In the next experiment, we see the differences in estimates for different initializations in order to study the variability of the algorithms. Thus, we construct a parabola as is explained in 4.2.1, and Gaussian noise with standard deviation 0.2 was added to the candidate points defined on the parabola. The generated cloud points is depicted in Figure 10a. Then, we carried out a total of 100 executions with different sample points with size $N = 50$, selected randomly from the set of candidate points.

In Figure 10 all the solutions for all the initializations are represented, so we can have a qualitative assessment of the four algorithms. Clearly, Direct and Cals methods fails too many times in the predictions, which is not a good symptom in terms of robustness. The most stable method is García. However, it is important to remark that it needs a lot of time to process the information and generate the solution, which could be an impediment for some real time applications. Our approach also achieves a good performance, since it only fails once, and its computing time is quite acceptable.



(a) True parabola (b) Our method (0.3495 s.)

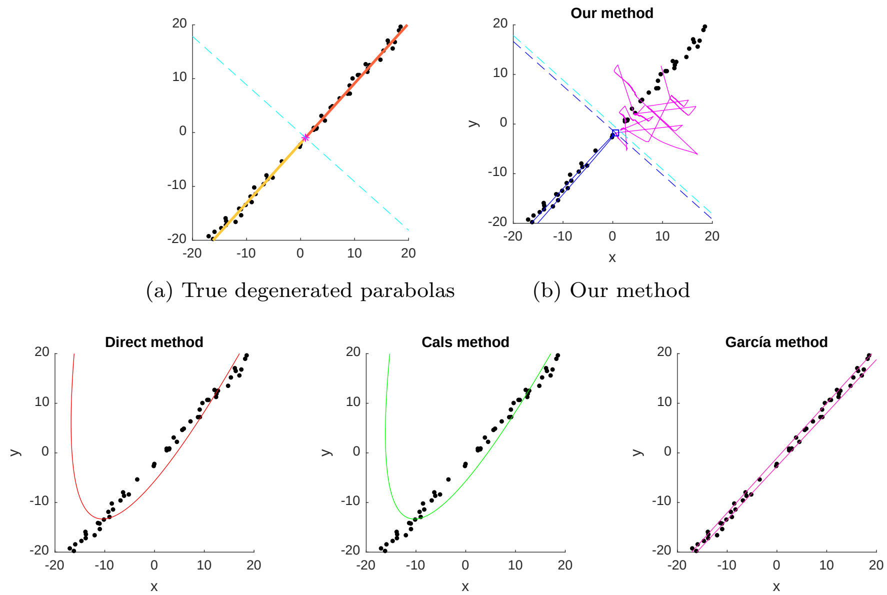
(c) Direct method (0.0011 s.) (d) Cals method (0.0025 s.) (e) García method (81.7085 s.)

Figure 10: Solutions for all the different initializations. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve. Each caption contains the mean execution time needed to obtain the fitted parabola.

4.2.3. Degenerated parabola

In this subsection we analyze the behavior for all methods in the case of a degenerated parabola. To generate it, a random directrix was computed following (20) and a point in this line has been chosen as the focus. Thus, two symmetric branches are the two solutions of this configuration, as is shown in Figure 11a in colors yellow and brown. These degenerate parabolas are orthogonal to the directrix, which is shown in a dotted light blue line. To generate the dataset, we followed the methodology explained at the beginning of the experimental section but varying the error margin (21) to a value very close to zero. In this example we used an error of 0.001. Residuals were also computed in order to compare the methods from a quantitative point of view.

The best approximation to the real degenerated parabola is achieved by our proposal, which is the only one that finds one of the two solutions. It is not a straight line, but the obtained parabola is the flattest one. The Residual error is 0.000502, which is very close to zero. In addition, the estimated directrix is almost the same as the true one. Both Direct and Cals methods fail in the prediction, with Residuals 3.7190 and 3.4084 resp., and García approach estimates a flattened parabola but the focus is wrongly estimated to lie at one



(a) True degenerated parabolas (b) Our method

(c) Direct method (d) Cals method (e) García method

Figure 11: Solutions for a synthetic dataset with Gaussian noise distribution. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

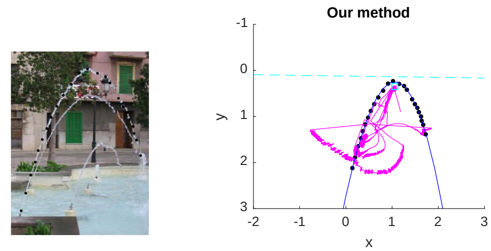
extreme of the data domain. However, the Residual is the best one, 0.0000207.

4.3. Real data

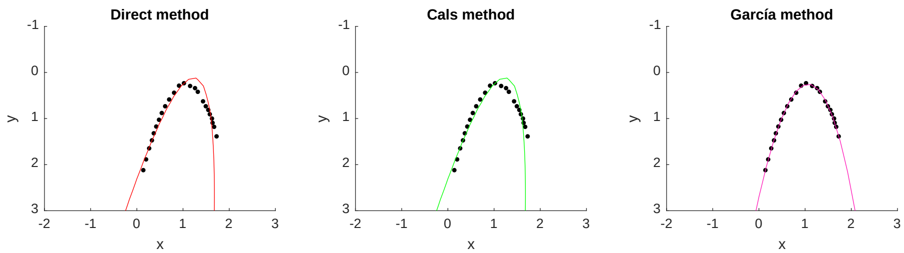
Next some real dataset examples are presented such as a fountain, a denture sample, a bridge and a reflector. We have overlayed the data points on each image for the sake of clarity.

The first two examples, the fountain and the denture sample, are based on manually selected points on the parabolic curves defined by them. As shown in Figure 12 our results are better than the competing methods. As known, the trajectory of the water drops or any other bodies under the sole influence of the gravitational field of the Earth is a parabola. Although all the algorithms return a fitted parabola very similar that one described by the fountain, Direct and Cals fail in the orientation. Both Ours and García generate good quality solutions but with very different CPU times.

In the case of the denture dataset (see Figure 13), the centers of the teeth should describe a parabola for an ideal denture. Having this in mind, this example can be considered as a dataset with a small level of noise. It is shown



(a) Fountain example: real image (b) Our method (0.4042 s.)



(c) Direct method (0.0390s.) (d) Cals method (0.0458 s.) (e) García method (40.2343 s.)

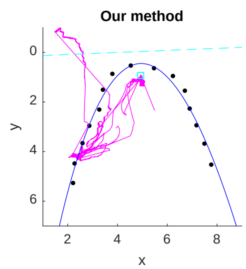
Figure 12: Solutions for the fountain dataset. Points were manually selected from one of the water streams. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to coincide with the image visualization format.

that both García and our methods do not fail and have a better performance than the other two methods.

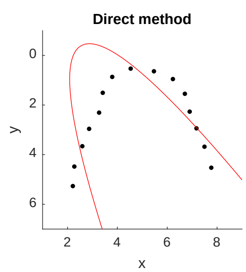
Next, we used the image intensities to extract the bridge sample dataset (see Figure 14a). A white color threshold was applied followed by the application of a region of interest (ROI) mask to remove all pixels that are below the bottom part of the bridge arc, in order to remove the crossing pathway. Then, a random selection of 25 points from the segmented image were carried out. As it can be seen in Figure 14a, there are two spurious points in the final data selection, i.e., 10% of outliers approx. Figure 14 demonstrate that our method is robust against anomalous data, achieving a great fit compared to Cals and Direct methods, which are unstable when the dataset does not describe a large arc of the parabola. García algorithm works very well but only if the trimming level is provided. Otherwise, i.e., $\alpha = 0$ the method fails in the fit. This parameter is hard to determine in real applications.

An industrial real data example is used in Figure 15. In this case, the dataset describes a parabola with a radius of curvature a little higher than the previous

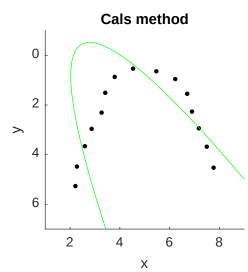
(a) Denture example: real image



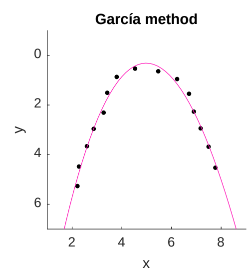
(b) Our method (0.4271 s.)



(c) Direct method (0.0396 s.)

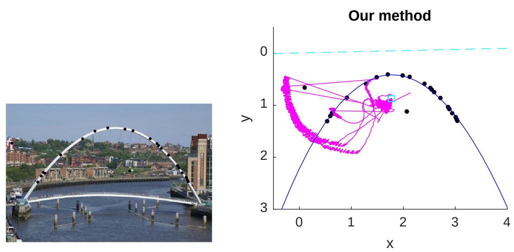


(d) Cals method (0.0471 s.)

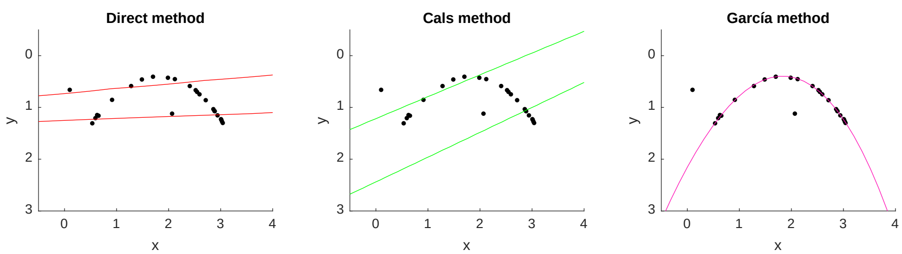


(e) García method (42.8631 s.)

Figure 13: Solutions for denture real data. Points were manually selected from the center of the teeth. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to coincide with the image visualization format.



(a) Bridge example: real image (b) Our method (0.4582 s.)



(c) Direct method (0.0403 s.) (d) Cals method (0.0507 s.) (e) García method (43.5763 s.)

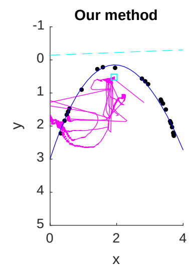
Figure 14: Solutions for the real data about a bridge. 25 points were randomly selected using a color threshold for pixels with values greater that 214 (in a scale [0-255]) for the three RGB channels, followed by a ROI with $0 < X < 215$ in the first dimension. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to concise with the image visualization format. We used $\alpha = 0.2$ as an input for García method.

examples. The Canny edge detection method has been employed to extract edge points on the parabolic curves from the image, and then the training samples have been randomly drawn from the set of edge points.

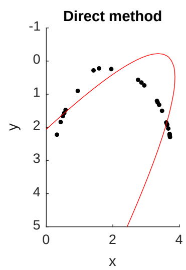
The dataset shown in Figure 15a exhibits three anomalous points near the vertex that clearly are not part of the parabolic shape. As seen, all the competing methods present worse results than ours (see Figure 15). Direct and Cals fail in the fit and García method generates a good fit, but still it is not as precise as ours and also the trimming level needed to be provided. García only works well if the true percentage of outliers is supplied, which would not be easy to get in a real application, while our method does not need that input.

Overall our proposal achieves a good fit to the above presented real datasets, and employing very little time to compute the fitted parabola. Therefore, its robustness against noise is further validated. It is also remarkable that the method do not need to be provided by input parameters based on the employed dataset.

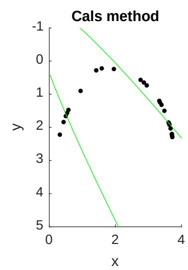
(a) Reflector example: real image



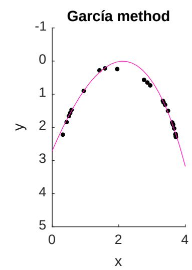
(b) Our method (0.4121 s.)



(c) Direct method (0.0312 s.)



(d) Cals method (0.0393 s.)

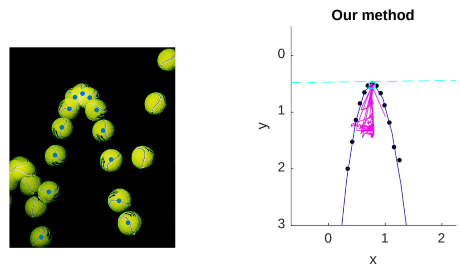


(e) García method (39.6732 s.)

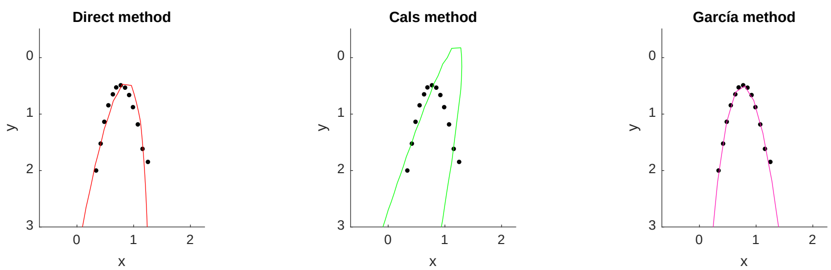
Figure 15: Solutions for the real data about a reflector. 25 points were randomly selected from the output of the Canny edge detection function of Matlab with threshold=0.95, followed by a ROI with $0 < X < 375$ in the first dimension. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to concide with the image visualization format. We used $\alpha = 0.2$ as an input for García method.

The following examples are based on stroboscopic images of spherical objects that describe a parabolic shot. In this case, the point were automatically extracted using the Circle Hough Transform, which is a feature extraction technique used to find imperfect instances of objects by a voting procedure. It has a great performance to detect object that have circular or elliptical shapes. MATLAB *imfindcircles* function was used to detect the objects in the image and determine its centroid.

Figures 16 and 17 show two examples of the execution of the algorithms with stroboscopic images. In Figure 16a we can see the movement of a tennis ball throwing and describing three parabolas. We select the middle one and extracted the points as specified in the figure caption. We can observe that the best fits are achieved again by our proposal and García method, but this one employs too much time although the number of sample points is very small.



(a) Tennis ball example: stroboscopic image (b) Our method (0.3988 s.)



(c) Direct method (0.0304 s.) (d) Cals method (0.0447 s.) (e) García method (38.4468 s.)

Figure 16: Solutions for tennis ball stroboscopic image. Points were selected using the Circle Hough Transform through MATLAB with Radius Range = [15 45] and Sensitivity = 0.9 as input parameters. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to coincide with the image visualization format. CPU time is displayed in subcaptions.

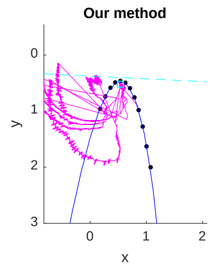
In Figure 17a a soccer ball is pictured. The first parabola is selected, which has an occluded part. However, this does not affect the estimations of the four methods, but Direct and Cals methods still have a bit worst performance.

If we compare all the processing times for all the experiments that we have carried out with real examples, we can see that the fastest ones are always Direct and Cals, with around 0.05 seconds. The third position is occupied by our algorithm, with a mean value of 0.42 seconds, but with the difference that the fitting performance is clearly better. At last, García method is very robust but needs more that 40 seconds to obtain the result.

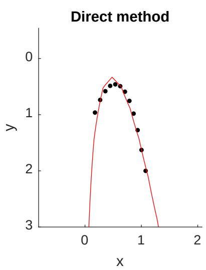
5. Conclusions

This paper presents a parabola fitting method based on minimization of the parabola geometric function to a set of points by minimizing the mean absolute deviation from the level set, which defines the parabola feature. The algorithm comprises of two stages: 1) determining the closed distance between data points to the directrix and to the focus point, and 2) solving an absolute

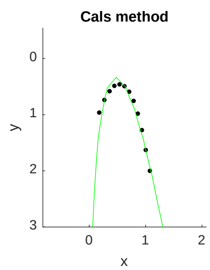
(a) Soccer ball example: stroboscopic image



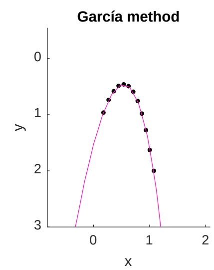
(b) Our method (0.4122 s.)



(c) Direct method (0.0318 s.)



(d) Cals method (0.0666 s.)



(e) García method (36.4591 s.)

Figure 17: Solutions for soccer ball stroboscopic image. Points were selected using the Circle Hough Transform through MATLAB with Radius Range = [15 45] and Sensitivity = 0.85 as input parameters. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to coincide with the image visualization format. CPU time is displayed in subcaptions.

geometric cost function for parameter estimates with a normalized directrix vector that improves the stability of the method. Furthermore, the search is restarted each certain number of steps in order to escape from local minima. This way the proposed algorithm converges to a finite solution irrespective of the initial parameters.

Experimental results demonstrate that our algorithm is robust since the solutions have a low sensitivity to noise data. The robustness of the proposed algorithm is due to the usage of absolute rather than squared errors, since the sum of absolute errors is more robust than the sum of squared errors. State of the art alternative methods have been tested, which are either much slower than ours, or clearly worse in their fitting accuracy. Therefore, it can be concluded that a fast and accurate approach for parabola fitting has been proposed, which can be used for datasets with significant amounts of noise and occlusion. This facilitates its application to real problems with low quality input data.

Directions for future work include further improving the accuracy of the estimation of the parabola parameters, while keeping the computational complexity low. As seen in the experiments, García method is quite robust, and it sometimes finds better estimations of the parabolas, but at the expense of being one or two orders of magnitude slower than our approach. Hence there is still room for enhancing our approach. This might be done by hybridizing the algorithm presented here with some optimization framework such as genetic algorithms, tabu search or particle swarm optimization. Another possible strategy is the combination of several high quality solutions, in the spirit of ensemble learning.

Acknowledgments

This work is partially supported by the Ministry of Economy and Competitiveness of Spain [grant number TIN2014-53465-R], project name Video surveillance by active search of anomalous events. It is also partially supported by the Autonomous Government of Andalusia (Spain) [grant number TIC-6213], project name Development of Self-Organizing Neural Networks for Information Technologies; and [grant number TIC-657], project name Self-organizing systems and robust estimators for video surveillance. All of them include funds from the European Regional Development Fund (ERDF). The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They have also been supported by the Biomedic Research Institute of Málaga (IBIMA). They also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU. Karl Thurnhofer-Hemsi is funded by a Ph.D. scholarship from the Spanish Ministry of Education, Culture and Sport under the FPU program [grant number FPU15/ 06512].

References

[1] P. Benko, G. Kos, T. Varady, L. Andor, R. Martin, Constrained fitting in reverse engineering, Computer Aided Geometric Design 19 (3) (2002) 173–206.

[2] C. Conti, L. Romani, D. Schenone, Semi-automatic spline fitting of planar curvilinear profiles in digital images using the Hough transform, Pattern Recognition 74 (2018) 64–76.

[3] L. A. García-Escudero, A. Mayo-Iscar, C. I. Sánchez-Gutiérrez, Fitting parabolas in noisy images, Computational Statistics and Data Analysis 112 (2017) 80–87.

[4] G. Hu, F. Yan, J. Kittler, W. Christmas, C. H. Chan, Z. Feng, P. Huber, Efficient 3D morphable face model fitting, Pattern Recognition 67 (2017) 366–379.

[5] M. L. Torrente, S. Biasotti, B. Falcidieno, Recognition of feature curves on 3D shapes using an algebraic approach to Hough transforms, Pattern Recognition 73 (2018) 111–130.

[6] T. Varady, R. Martin, J. Cox, Reverse engineering of geometric models - An introduction, Computer-Aided Design 29 (4) (1997) 255–268.

[7] F. de Vieillevile, J. O. Lachaud, Comparison and improvement of tangent estimators on digital curves, Pattern Recognition 42 (2009) 1693–1707.

[8] G. Xiao, H. Wang, T. Lai, D. Suter, Hypergraph modelling for geometric model fitting, Pattern Recognition 60 (2016) 748–760.

[9] F. Bookstein, Fitting conic sections to scattered data, Computer Graphics and Image Processing 9 (1979) 56–71.

[10] B. P. Butler, A. B. Forbes, P. M. Harris, Algorithms for geometric tolerance assessment, Teddington, UK, 1994.

[11] A. B. Forbes, Algorithms for Approximation II, Chapman and Hall, 1990, Ch. Least-squares best fit geometric elements, pp. 311–319.

[12] L. Krüger, C. Wölher, Accurate chequerboard corner localisation for camera calibration, Pattern Recognition Letters 32 (2011) 1428–1435.

[13] D. S. Zwick, Recent Advances in Total Least Squares and Errors-in-Variables Techniques, SIAM, Philadelphia, 1997, Ch. Applications of Orthogonal Distance Regression in Metrology, pp. 265–272.

[14] S. Chen, R. Xia, J. Zhao, Y. Chen, M. Hu, A hybrid method for ellipse detection in industrial images, Pattern Recognition 68 (2017) 82–98.

[15] A. Samper, G. González, B. Herrera, Determination of the geometric shape which best fits an architectural arch within each of the conical curve types and hyperbolic-cosine curve types: The case of Palau Güell by Antoni Gaudí, Journal of Cultural Heritage.

[16] K. Paton, Conic sections in chromosome analysis, Pattern Recognition 2 (1970) 39–40.

[17] A. Muhamad, W. Nezar, A. Azzaldeen, The curve of dental arch in normal occlusion, Open Science Journal of Clinical Medicine 3 (2) (2015) 47–54.

[18] E. Janunts, M. Kannengießer, A. Langenbucher, Parametric fitting of corneal height data to a biconic surface, Zeitschrift für Medizinische Physik 25 (2015) 25–35.

[19] Y. K. Jang, B. J. Kang, K. R. Park, A study on eyelid localization considering image focus for iris recognition, Pattern Recognition Letters 29 (2008) 1698–1704.

[20] K. S. Chin, E. Trucco, L. Tan, P. J. Wilson, Automatic fovea location in retinal images using anatomical priors and vessel density, Pattern Recognition Letters 34 (2013) 1152–1158.

[21] C. Y. Yu, C. C. Liu, S. S. Yu, A novel scheme for the fovea localization on retinal images, in: I. C. S. Washington (Ed.), Proceedings of the 2014 International Symposium on Computer, Consumer and Control, 2014, pp. 609–612.

[22] C. Y. Yu, C. C. Liu, S. S. Yu, A fovea localization scheme using vessel origin-based parabolic model, Algoritms 7 (2014) 456–470.

[23] B. B. Ogul, E. Sümer, H. Ogul, Unsupervised rib delineation in chest radiographs by an integrative approach, in: Proceedings of the 10th International Conference on Computer Vision Theory and Aplications, Vol. 1, VISAPP, Berlin, Germany, 2015, pp. 260–265.

[24] G. Samarasinghe, A. Sowmya, D. A. Moses, A semi-quantitative analysis model with parabolic modelling for DCE-MRI sequences of prostate, in: 2014 International Conference on Digital lmage Computing: Techniques and Applications (DlCTA), IEEE, 2014.

[25] N. Kampanis, A. Delis, D. Antonopoulou, G. Kozyrakis, A finite element discretization of the standard parabolic equation in generalized boundary fitting coordinates, Applied Numerical Mathematics 67 (2013) 152–166.

[26] Y. Man, H. Li, Focal length precise measurement method for optics system based on lunar imaging, Proceedings of SPIE 9301 (2014) 93010M–93010M–9.

[27] H. Späth, Least squares fitting with rotated paraboloids, Mathematical Communications 6 (2001) 173–179.

[28] W. Wang, C. Wang, B. Duan, G. Leng, X. Li, Compensation for gravity deformation via subreflector motion of 65 m shaped cassegrain antenna, IET Microwaves, Antennas & Propagation 8 (3) (2014) 158 – 164.

[29] F. Ye, F. Li, S. Hu, Automatic extraction of power lines by particle filtering from aerial images, in: Proc. SPIE 9069, Fifth International Conference on Graphic and Image Processing, 2014.

[30] S. P. Narote, P. N. Bhujbal, A. S. Narote, D. M. Dhane, A review of recent advances in lane detection and departure warning system, Pattern Recognition 73 (2018) 216–234.

[31] J. Niu, J. Lu, M. Xu, P. Lv, X. Zhao, Robust lane detection using two-stage feature extraction with curve fitting, Pattern Recognition 59 (2016) 225–233.

[32] J. Wang, F. Gu, C. Zhang, G. Zhang, Lane boundary detection based on parabola model, in: Proceedings of the 2010 IEEE International Conference on Information and Automation, 2010.

[33] C. Li, B. Dai, R. Wang, Y. Fang, X. Yuan, T. Wu, Multi-lane detection based on omnidirectional camera using anisotropic steerable filters, IET Intelligent Transport Systems 10 (5) (2016) 298–307.

[34] V. Dougalis, N. Kampanis, F. Sturm, G. Zouraris, Numerical solution of the parabolic equation in range-dependent waveguides, in: Numerical Insights Series, CRC Press/Taylor & Francis, 2008, pp. 175–208.

[35] D. Lee, S. McDaniel, Ocean acoustic propagation by finite difference methods, Journal of Computational and Applied Mathematics 14 (305-423).

[36] K. Gilbert, M. White, Application of the parabolic equation to sound propagation in a refracting atmosphere, Journal of the Accoustical Society of America 85 (1989) 630–637.

[37] K. Gilbert, R. Raspet, X. Di, Calculation of turbulence effects in an upward-refracting atmosphere, Journal of the Accoustical Society of America 87 (1990) 2428–2437.

[38] J. Robertson, W. Siegmann, M. Jacobson, Low-frequency sound propagation modeling over a locally reacting boundary with the parabolic approximation, Journal of the Accoustical Society of America 98 (1995) 1130–1137.

[39] J. Robertson, P. Schlatter, W. Siegmann, Sound propagation over impedance discontinuities with the parabolic approximation, Journal of the Accoustical Society of America. 99 (1996) 761–767.

[40] K. Gilbert, X. Di, A fast Green's function method for one-way sound propagation in the atmosphere, Journal of the Accoustical Society of America 94 (1993) 2343–2352.

[41] E. Salomons, Improved Green's function parabolic equation method for atmospheric sound propagation, Journal of the Accoustical Society of America 104 (1998) 100–111.

[42] J. Anderson, Computational Fluid Dynamics, McGraw-Hill, New York, 1995.

[43] V. Dougalis, N. Kampanis, Finite element methods for the parabolic equation with interfaces, Journal of Computational Acoustics 4 (1996) 55–88.

[44] M. Levy, Parabolic equation methods for electromagnetic wave propagation, in: IEE Electromagnetic Waves Series, Vol. 4, The Institution of Electrical Engineers, 2000.

[45] N. Chernov, C. Lesort, Statistical efficiency of curve fitting algorithms, Computational Statistics & Data Analysis 30 (2004) 1347–1358.

[46] K. Kanatani, Statistical optimization for geometric fitting: theoretical accuracy bound and high order error analysis, International Journal of Computer Vision 80 (2) (2008) 167–188.

[47] L. Hunyadi, I. Vajk, Constrained quadratic errors-in-variables fitting, Visual Computer 30 (2014) 1347–1358.

[48] P. Li, C. Zhang, S. Jia, A new method of fitting implicit conic to plane scattered data points, Computer-Aided Design and Applications 9 (2) (2012) 147–154.

[49] M. Harker, P. O'Leary, P. Zsombor-Murray, Direct type-specific conic fitting and eigenvalue bias correction, Image and Vision Computing 26 (3) (2008) 372–381.

[50] J. Yu, S. Kulkarni, H. Poor, Robust fitting of ellipses and spheroids, in: Record of the Forty-Third Asilomar conference on Signals, Systems and Computers, 2009.

[51] R. Halíř, J. Flusser, Numerically stable direct least squares fitting of ellipses, Technical report citeseer.nj.nec.com/350661.html, Department of Software Engineering, Charles University, Czech Republic (2000).

[52] G. Golub, A. Hoffman, G. Stewart, A generalization of the Eckart-Young-Mirsky matrix approximation theorem, Linear Algebra and its Applications 88–89 (1987) 317–327.

[53] S. J. Ahn, W. Rauh, H.-J. Warnecke, Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola, Pattern Recognition 34 (2001) 2289–2303.

[54] P. Sturm, P. Gargallo, Conic fitting using the geometric distance, Lecture Notes in Computer Science 4844 (2) (2007) 784–795.

[55] Z. Y. Zhang, Parameter estimation techniques: A tutorial with application to conic fitting, Image and Vision Computing 15 (1) (1997) 59–76.

[56] H. Liu, C. Zhang, X. Liang, Fitting conic to scattered data in a plane, Journal of Computer Aided Geometric Design and Computer Graphics 16 (11) (2004) 1594–1596.

[57] P. Rosin, A note on the least squares fitting of ellipse, Pattern Recognition Letters 14 (1993) 799–808.

[58] A. Fitzqibbon, M. Pilu, R. Fisher, Direct least square fitting of ellipse, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (5) (1999) 476–480.

[59] P. L. Rosin, Analysing error of fit functions for ellipse, Pattern Recognition Letters 17 (1996) 1461– 1470.

[60] H. Bengtsson, R.matlab: Read and write MAT files and call MATLAB from within R. R package version 3.6.0-9000.
URL `https://github.com/HenrikBengtsson/R.matlab`