

Guaranteed Ellipse Fitting with a Confidence Region and an Uncertainty Measure for Centre, Axes, and Orientation

Zygmunt L. Szpak · Wojciech Chojnacki · Anton van den Hengel

Received: 10 December 2013 / Accepted: 30 August 2014 / Published online: 25 September 2014
© Springer Science+Business Media New York 2014

Abstract A simple and fast ellipse estimation method is presented based on optimisation of the Sampson distance serving as a measure of the quality of fit between a candidate ellipse and data points. Generation of ellipses, not just conics, as estimates is ensured through the use of a parametrisation of the set of all ellipses. Optimisation of the Sampson distance is performed with the aid of a custom variant of the Levenberg–Marquardt algorithm. The method is supplemented with a measure of uncertainty of an ellipse fit in two closely related forms. One of these concerns the uncertainty in the algebraic parameters of the fit and the other pertains to the uncertainty in the geometrically meaningful parameters of the fit such as the centre, axes, and major axis orientation. In addition, a means is provided for visualising the uncertainty of an ellipse fit in the form of planar confidence regions. For moderate noise levels, the proposed estimator produces results that are fully comparable in accuracy to those produced by the much slower maximum likelihood estimator. Due to its speed and simplicity, the method may prove useful in numerous industrial applications where a measure of reliability for geometric ellipse parameters is required.

Keywords Ellipse fitting · Maximum likelihood · Uncertainty measure · Simultaneous confidence region · Centre · Semi-major and semi-minor axes · Orientation

1 Introduction

The task of fitting an ellipse to data is frequently encountered in numerous disciplines. This is partially explained by the fact that even though very few shapes and trajectories are perfectly elliptical, non-elliptical shapes and trajectories can often be modelled using one or more ellipses. Recently, Wong et al. [60] compiled a comprehensive list of ellipse fitting applications with items such as human gait analysis, grain sorting, steel coils quality assurance, and cell segmentation—to name a few. All of these applications require an ellipse fitting method that is (1) simple to implement, (2) computationally efficient, (3) accurate, (4) capable of delivering a measure of uncertainty associated with any estimate generated, and (5) always produces a fit in the form of an ellipse. Unfortunately, there is currently no ellipse fitting method available that simultaneously satisfies all of these requirements.

For example, the maximum likelihood ellipse estimation method in its classic form seeks to minimise the sum of the orthogonal distances between data points and a candidate ellipse. Minimising the orthogonal distance, however, is a complicated process in practice [1, 8, 10, 19, 32, 53, 61]. It necessitates projecting points onto conics—a task for which there is no closed form solution, and one that has been described by numerous authors as time consuming and numerically unstable [2, 11]. So even though the classic maximum likelihood estimation method, also called the orthogonal distance regression method, yields accurate results, it is often inadequate for many practical scenarios.

There are also algebraic ellipse fitting techniques based on error measures that determine how well a collection of data points satisfies the ellipse equation in a certain least squares sense [3, 25, 33, 48]. Least squares error measures are different from the orthogonal distance, but are much

Z. L. Szpak (✉) · W. Chojnacki · A. van den Hengel
School of Computer Science, The University of Adelaide,
Adelaide, SA 5005, Australia
e-mail: zygmunt.szpak@adelaide.edu.au

W. Chojnacki
e-mail: wojciech.chojnacki@adelaide.edu.au

A. van den Hengel
e-mail: anton.vandenhengel@adelaide.edu.au

easier to compute and much easier to minimise. Therefore, algebraic ellipse fitting methods are typically fast and simple, and some can even always generate ellipses, not just conics, as estimates [17, 21, 42, 59]. However, in comparison to the orthogonal distance regression method, algebraic ellipse fitting techniques often produce estimates of inferior accuracy.

Another class of ellipse fitting methods comprises techniques tagged with labels such as *gradient-weighted*, *approximate maximum likelihood (AML)*, and *hyper-accurate* [28, 29, 55]. The methods in this class seek a balance between the accuracy of the orthogonal distance regression method and the simplicity of algebraic fitting methods. What differentiates these techniques from pure algebraic schemes is that they have credible statistical foundations [9, 13] and are capable of accommodating the uncertainty associated with a data point. One particular estimation method that has stood the test of time was proposed independently by Turner [58] in 1974 and by Sampson [50] in 1982. The cost function underlying this method falls into the category of AML cost functions and is often called the *Sampson distance* [24, Sect. 4.2.6]. For moderate noise levels, the Sampson distance is an excellent approximation of the orthogonal distance. Whilst ellipse fitting methods based on the Sampson distance are typically simple, fast, and accurate at moderate noise levels, most do not guarantee generation of an estimate in the form of an ellipse.

An ellipse fitting method that is fast, simple, utilises the Sampson distance, and guarantees a genuine ellipse fit was recently proposed by Szpak et al. [54]. To enforce the ellipticity constraint, the method uses a barrier term added to the AML cost function. In this paper, we show that an explicit barrier term may be dispensed with in favour of a parametrisation of the set of all ellipses that leads to the *implicit* enforcement of the ellipticity constraint. The estimator that we develop based on this parametrisation is conceptually and practically simpler than its immediate predecessor involving a barrier term. As a further extension of the barrier method, we develop a measure of uncertainty of a Sampson distance based ellipse estimate in two closely related forms. One of these concerns the uncertainty in the algebraic parameters of the estimate and the other pertains to the uncertainty in the geometrically meaningful parameters of the estimate such as the centre, axes, and major axis orientation. We also provide a means for visualising the uncertainty of an estimate in the form of planar confidence regions. By providing a measure of uncertainty of an ellipse estimate, we facilitate principled statistical decision making for users of our algorithm. Thus, although the algorithm uses a parametrisation of the ellipses that has no immediate geometric significance, practitioners can apply our algorithm and still have a measure of reliability of the geometrically meaningful parameters of an ellipse estimate.

2 Background

A *conic section*, or simply a *conic*, is the locus of solutions $\mathbf{x} = [m_1, m_2]^T$ in the Euclidean plane \mathbb{R}^2 of a quadratic equation

$$am_1^2 + bm_1m_2 + cm_2^2 + dm_1 + em_2 + f = 0, \quad (2.1)$$

where a, b, c, d, e, f are real numbers such that $a^2 + b^2 + c^2 > 0$. With $\boldsymbol{\theta} = [a, b, c, d, e, f]^T$ and $\mathbf{u}(\mathbf{x}) = [m_1^2, m_1m_2, m_2^2, m_1, m_2, 1]^T$, Eq. (2.1) can equivalently be written as

$$\boldsymbol{\theta}^T \mathbf{u}(\mathbf{x}) = 0. \quad (2.2)$$

Any multiple of $\boldsymbol{\theta}$ by a non-zero number corresponds to the same conic. A conic is *non-degenerate* if the *determinant* of the conic

$$D = \begin{vmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{vmatrix}$$

is non-zero. When $D = 0$, the conic is *degenerate*. A non-degenerate conic is either an *ellipse* (possibly with no graph, that is, an ellipse reduced to an empty set), a *parabola*, or a *hyperbola* depending on whether the *discriminant* $\Delta = b^2 - 4ac$ is negative, zero, or positive, respectively. A degenerate conic can be either: a single point ($\Delta < 0$); an empty set, a straight line, or two parallel lines ($\Delta = 0$); or a pair of intersecting lines ($\Delta > 0$). By convention, degenerate conics with $\Delta < 0$, $\Delta = 0$, and $\Delta > 0$ are referred to as degenerate ellipses, degenerate parabolas, and degenerate hyperbolas, respectively.

The condition $\Delta < 0$ characterising the ellipses (non-degenerate or otherwise, including the ellipses with no graph or reduced to a point) can alternatively be written as

$$\boldsymbol{\theta}^T \mathbf{F} \boldsymbol{\theta} > 0, \quad (2.3)$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{bmatrix}$$

and \otimes denotes Kronecker product [35]. Hereafter, of all conics, we shall specifically be concerned with ellipses.

The task of fitting an ellipse to a set of points $\mathbf{x}_1, \dots, \mathbf{x}_N$ requires a meaningful *cost function* that characterises the extent to which any particular $\boldsymbol{\theta}$ fails to satisfy the system of N copies of Eq. (2.2) associated with $\mathbf{x} = \mathbf{x}_n, n = 1, \dots, N$. Once a cost function is selected, the corresponding ellipse fit is generated by minimising the cost function subject to the constraint (2.3).

Effectively, though not explicitly, Varah [59] and Fitzgibbon et al. [17] proposed to use for ellipse fitting the *direct ellipse fitting* cost function defined by

$$J_{\text{DIR}}(\theta) = \begin{cases} \frac{\theta^T \mathbf{A} \theta}{\theta^T \mathbf{F} \theta} & \text{if } \theta^T \mathbf{F} \theta > 0, \\ \infty & \text{otherwise,} \end{cases}$$

where $\mathbf{A} = \sum_{n=1}^N \mathbf{u}(\mathbf{x}_n) \mathbf{u}(\mathbf{x}_n)^T$. The minimiser $\hat{\theta}_{\text{DIR}}$ of J_{DIR} is the same as the solution of the problem

$$\begin{aligned} &\text{minimise } \theta^T \mathbf{A} \theta \\ &\text{subject to } \theta^T \mathbf{F} \theta = 1, \end{aligned} \tag{2.4}$$

and it is in this form that $\hat{\theta}_{\text{DIR}}$ was originally introduced. The representation of $\hat{\theta}_{\text{DIR}}$ as a solution of the problem (2.4) makes it clear that $\hat{\theta}_{\text{DIR}}$ is always an ellipse. Extending the work of Varah and Fitzgibbon et al., Halff and Flusser [21] introduced a numerically stable algorithm for calculating $\hat{\theta}_{\text{DIR}}$.

Another cost function for ellipse fitting, and in fact more generally for conic fitting, was first proposed independently by Turner [58] and Sampson [50] and then popularised, in a broader context, by Taubin [56] and Kanatani [27]. It is associated with many names such as the *Sampson*, *gradient-weighted*, and *approximate maximum likelihood (AML)* distance or cost function, and takes the form

$$J_{\text{AML}}(\theta) = \sum_{n=1}^N \frac{\theta^T \mathbf{A}_n \theta}{\theta^T \mathbf{B}_n \theta}$$

with

$$\mathbf{A}_n = \mathbf{u}(\mathbf{x}_n) \mathbf{u}(\mathbf{x}_n)^T \quad \text{and} \quad \mathbf{B}_n = \partial_{\mathbf{x}} \mathbf{u}(\mathbf{x}_n) \mathbf{A}_{\mathbf{x}_n} \partial_{\mathbf{x}} \mathbf{u}(\mathbf{x}_n)^T \tag{2.5}$$

for each $n = 1, \dots, N$. Here, for any length-2 vector \mathbf{y} , $\partial_{\mathbf{x}} \mathbf{u}(\mathbf{y})$ denotes the 6×2 matrix of the partial derivatives of the function $\mathbf{x} \mapsto \mathbf{u}(\mathbf{x})$ evaluated at \mathbf{y} , and, for each $n = 1, \dots, N$, $\mathbf{A}_{\mathbf{x}_n}$ is a 2×2 symmetric *covariance matrix* describing the uncertainty of the data point \mathbf{x}_n [6, 13, 27]. The function J_{AML} is a first-order approximation of a genuine maximum likelihood cost function J_{ML} which can be evolved based on the *Gaussian model of errors* in conjunction with the *principle of maximum likelihood*. In the case when identical independent homogeneous Gaussian noise corrupts the data points, J_{ML} reduces—up to a numeric constant depending on the noise level—to the sum of orthogonal distances of the data points and an ellipse. For the specific task of ellipse fitting, the search space for minimising J_{AML} is reduced from the set of all length-6 vectors to the set $E = \{\theta \mid \Delta < 0\}$ of all ellipses, each member of the latter set being conventionally referred to as a *feasible* point. The approximated maximum

likelihood estimate $\hat{\theta}_{\text{AML}}$ is, by definition, the minimiser of J_{AML} selected from among all feasible points.

The existence of a minimiser of J_{AML} in the set of all ellipses is not always guaranteed. In contrast, there always exists a minimiser of J_{AML} in the joint set $EP = \{\theta \mid \Delta \leq 0\}$ of ellipses and parabolas. Indeed, J_{AML} is homogeneous in θ (meaning that $J_{\text{AML}}(\lambda\theta) = J_{\text{AML}}(\theta)$ for any scalar factor λ) and this implies that the range of J_{AML} —the set of values that J_{AML} can take—is the same as the range of the restriction of J_{AML} to the five-dimensional unit sphere $\mathbb{S}_5 = \{\theta \mid \|\theta\| = 1\}$, where $\|\cdot\|$ denotes the Euclidean norm. Consequently, the search for a minimiser of J_{AML} within EP may be restricted to the search within the intersection of EP and \mathbb{S}_5 , $EP \cap \mathbb{S}_5$, characterised by the conjunction of the conditions $\Delta \leq 0$ and $\|\theta\| = 1$. This intersection is a closed subset of \mathbb{S}_5 and as such is compact in a topological sense. Now, the function J_{AML} is continuous, and, given that any continuous function attains a minimum on a compact set, the function J_{AML} , considered over $EP \cap \mathbb{S}_5$, attains its minimum on $EP \cap \mathbb{S}_5$.

Typically, when the data set adheres sufficiently closely to a generative model producing data points based on a *bona fide* ellipse, the minimiser of J_{AML} within EP will represent a genuine ellipse. In some scenarios, however, the minimiser will be a member of $P = \{\theta \mid \Delta = 0\}$, the “parabolic” boundary of the set E of all ellipses. For example, when all data points lie on a line [7], the minimiser interpreted geometrically coincides with the line on which the points lie, this line being an instance of a degenerate parabola. In other words, the problem of minimising J_{AML} within the set of ellipses is ill posed, and solving it requires some form of regularisation. We shall regularise the minimisation problem by adaptively restricting the search domain for a minimiser to a subset of E . The restriction procedure will take the form of a stopping criterion for an optimisation algorithm, preventing iteratively generated estimates from going too close to the set of parabolas P . While this approach involves a good deal of arbitrariness, one has to immediately point out that such arbitrariness is unavoidable: no ill-posed problem admits a unique, or canonical, regularisation.

3 Optimisation Using a Barrier Term

In our previous work [54] we developed a technique for near-optimising J_{AML} subject to the ellipticity constraint. The method uses the *merit function*

$$P(\theta, \alpha) = J_{\text{AML}}(\theta) + \alpha g(\theta),$$

where g is a *barrier function* of the form

$$g(\theta) = \frac{\|\theta\|^4}{(\theta^T \mathbf{F} \theta)^2} = \|\theta\|^4 \Delta^{-2}$$

and α is a positive number. A solution representing an ellipse is sought in the form of a local minimiser of P . The critical property of the barrier function is that it tends to infinity as θ approaches the parabolic boundary $P = \{\theta \mid \theta^T \mathbf{F} \theta = 0\}$ between the feasible elliptic region $E = \{\theta \mid \theta^T \mathbf{F} \theta > 0\}$ and the infeasible hyperbolic region $H = \{\theta \mid \theta^T \mathbf{F} \theta < 0\}$ of the search space. This property ensures that if P is optimised in sufficiently short steps starting from a feasible point, then any local minimiser reached on the way is feasible; and if the value of α is small enough, this local minimiser is a good approximation of a local minimiser of J_{AML} . By design, the parameter α is set to a very small number (of the order of 10^{-15}). The search for an optimal solution is carried out by a fast converging iterative procedure that takes the direct ellipse fit $\hat{\theta}_{\text{DIR}}$ for a seed. The technique turns out to be statistically accurate and computationally efficient. However, an issue that might be of concern with regard to the method is that it produces ellipse estimates that are not exact minimisers of J_{AML} .

4 Optimisation Using a Parametrisation

Below we present a method capable of isolating ellipse estimates that are exact minimisers of J_{AML} within the feasible part of the search space, provided that J_{AML} admits a local minimum on that part. When J_{AML} has no local minimum amongst the ellipses, the method delivers an ellipse estimate which is an approximation of a (possibly generalised) parabola constituting the minimiser of J_{AML} within the joint set of ellipses and parabolas. The barrier term from our previous method plays no role here—this term is simply discarded. The method searches for a solution by employing a parametrisation of the set of all ellipses. The search is conducted with the aid of a specially crafted Levenberg–Marquardt (LM) algorithm, which ensures a fast convergence towards the optimal solution starting from an initial guess based on the direct ellipse fit. The specific design of our proposed algorithm is a substantial contribution of the present paper.

4.1 A Parametrisation of the Set of All Ellipses

The set of all ellipses admits a natural parametrisation. To reveal it, consider an ellipse described by θ . Then $a \neq 0$ and $c \neq 0$, for otherwise—should $a = 0$ or $c = 0$ hold—the discriminant $\Delta = b^2 - 4ac = b^2$ would be non-negative, and this would contradict the fact that for ellipses the discriminant is always negative. Given this and the fact that the scale of θ is irrelevant as far as the determination of the underlying ellipse is involved, it is safe to assume that $a = 1$. Under this assumption the condition $\Delta < 0$ becomes $4c > b^2$, and letting $b = 2p$, we see that $c > p^2$, or, equivalently,

$c = p^2 + q^{-2}$ for some q . It is now clear that the set of ellipses can be parametrised as

$$\theta = \kappa(\eta), \quad \kappa(\eta) = [1, 2p, p^2 + q^{-2}, r, s, t]^T, \tag{4.1}$$

with $\eta = [p, q, r, s, t]^T$ running over all possible length-5 vectors. A variant of the above parametrisation that will be particularly useful in what follows is given by

$$\theta = \pi(\eta), \quad \pi(\eta) = \|\kappa(\eta)\|^{-1} \kappa(\eta). \tag{4.2}$$

The significance of the unit-normalisation step will become apparent when we proceed to develop our customised version of the LM algorithm.

4.2 Optimisation Algorithm

With the parametrisation (4.2) in place, the effective function to optimise is

$$J'_{\text{AML}}(\eta) = J_{\text{AML}}(\pi(\eta)).$$

We shall next evolve a modified version of the LM algorithm for efficient optimisation of this function. It will be convenient to precede the description of our version of the LM scheme with a presentation of the standard form of the method.

4.2.1 Initialisation

Because the LM algorithm is an iterative technique, both its standard and modified form will require a proper initialisation. An initial value of η , η_0 , can be extracted from any estimate $\hat{\theta}$ representing a genuine ellipse by setting $\eta_0 = \mathbf{c}(\hat{\theta})$, where \mathbf{c} is the composition of the mapping $\theta \mapsto \theta/\theta_1 = [1, \theta_2/\theta_1, \dots, \theta_6/\theta_1]^T$ and the inverse of the mapping $\eta \mapsto \kappa(\eta)$. Specifically, $\mathbf{c}(\theta) = [p, q, r, s, t]^T$ is given by

$$p = \frac{\theta_2}{2\theta_1}, \quad q = \left(\frac{\theta_3}{\theta_1} - \left(\frac{\theta_2}{2\theta_1} \right)^2 \right)^{-1/2},$$

$$r = \frac{\theta_4}{\theta_1}, \quad s = \frac{\theta_5}{\theta_1}, \quad t = \frac{\theta_6}{\theta_1}. \tag{4.3}$$

As the direct ellipse fit always represents a genuine ellipse, a natural choice for η_0 is $\mathbf{c}(\hat{\theta}_{\text{DIR}})$.

4.2.2 Standard Form of LM

Both the standard and our version of the LM scheme will rely on a least squares expression for J'_{AML} . Let $\mathbf{r}(\theta) = [r_1(\theta), \dots, r_N(\theta)]^T$, where

$$r_n(\theta) = \left(\frac{\theta^T \mathbf{A}_n \theta}{\theta^T \mathbf{B}_n \theta} \right)^{1/2}$$

for each $n = 1, \dots, N$. Then the function J_{AML} can be written in least squares form as

$$J_{\text{AML}}(\boldsymbol{\theta}) = \mathbf{r}(\boldsymbol{\theta})^\top \mathbf{r}(\boldsymbol{\theta}) = \|\mathbf{r}(\boldsymbol{\theta})\|^2.$$

Similarly, the companion function J'_{AML} can be expressed as

$$J'_{\text{AML}}(\boldsymbol{\eta}) = \|\mathbf{r}'(\boldsymbol{\eta})\|^2,$$

where $\mathbf{r}'(\boldsymbol{\eta}) = [r'_1(\boldsymbol{\eta}), \dots, r'_N(\boldsymbol{\eta})]^\top$ with $r'_n(\boldsymbol{\eta}) = r_n(\boldsymbol{\pi}(\boldsymbol{\eta}))$ for each $n = 1, \dots, N$.

A full description of any variant of the LM scheme requires an explicit form of the Jacobian matrix of $\mathbf{r}'(\boldsymbol{\eta})$. A straightforward computation shows that the Jacobian matrix of $\mathbf{r}(\boldsymbol{\theta})$,

$$\partial_{\boldsymbol{\theta}} \mathbf{r}(\boldsymbol{\theta}) = [(\partial_{\boldsymbol{\theta}} r_1(\boldsymbol{\theta}))^\top, \dots, (\partial_{\boldsymbol{\theta}} r_N(\boldsymbol{\theta}))^\top]^\top,$$

is fully determined by the relations

$$\begin{aligned} (\partial_{\boldsymbol{\theta}} r_n(\boldsymbol{\theta}))^\top &= r_n^{-1}(\boldsymbol{\theta}) \mathbf{X}_n(\boldsymbol{\theta}) \boldsymbol{\theta}, \\ \mathbf{X}_n(\boldsymbol{\theta}) &= \frac{\mathbf{A}_n}{\boldsymbol{\theta}^\top \mathbf{B}_n \boldsymbol{\theta}} - \frac{\boldsymbol{\theta}^\top \mathbf{A}_n \boldsymbol{\theta}}{(\boldsymbol{\theta}^\top \mathbf{B}_n \boldsymbol{\theta})^2} \mathbf{B}_n \end{aligned}$$

for all $n = 1, \dots, N$. For each $m = 1, 2, \dots$, denote by \mathbf{I}_m the $m \times m$ identity matrix. Let $\mathbf{P}_{\boldsymbol{\theta}}^\perp$ be the symmetric matrix representing the projection along $\boldsymbol{\theta}$ onto the orthogonal complement of $\boldsymbol{\theta}$, given by

$$\mathbf{P}_{\boldsymbol{\theta}}^\perp = \mathbf{I}_6 - \|\boldsymbol{\theta}\|^{-2} \boldsymbol{\theta} \boldsymbol{\theta}^\top. \tag{4.4}$$

Then the Jacobian matrix of $\mathbf{r}'(\boldsymbol{\eta})$ can be expressed as

$$\partial_{\boldsymbol{\eta}} \mathbf{r}'(\boldsymbol{\eta}) = \partial_{\boldsymbol{\theta}} \mathbf{r}(\boldsymbol{\pi}(\boldsymbol{\eta})) \partial_{\boldsymbol{\eta}} \boldsymbol{\pi}(\boldsymbol{\eta}),$$

where

$$\partial_{\boldsymbol{\eta}} \boldsymbol{\pi}(\boldsymbol{\eta}) = \|\boldsymbol{\kappa}(\boldsymbol{\eta})\|^{-1} \mathbf{P}_{\boldsymbol{\kappa}(\boldsymbol{\eta})}^\perp \partial_{\boldsymbol{\eta}} \boldsymbol{\kappa}(\boldsymbol{\eta})$$

and

$$\partial_{\boldsymbol{\eta}} \boldsymbol{\kappa}(\boldsymbol{\eta}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 2p & -2q^{-3} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

We can now summarise the standard LM algorithm for optimising J'_{AML} as follows. Starting with an initial estimate $\boldsymbol{\eta}_0$, the scheme iteratively replaces a current estimate $\boldsymbol{\eta}_k$ with a new estimate $\boldsymbol{\eta}_{k+1}$ based on the update rule

$$\boldsymbol{\eta}_{k+1} = \boldsymbol{\eta}_k + \delta \boldsymbol{\eta}_k, \tag{4.5}$$

$$\delta \boldsymbol{\eta}_k = -((\partial_{\boldsymbol{\eta}} \mathbf{r}'(\boldsymbol{\eta}_k))^\top \partial_{\boldsymbol{\eta}} \mathbf{r}'(\boldsymbol{\eta}_k) + \lambda_k \mathbf{I}_5)^{-1} (\partial_{\boldsymbol{\eta}} \mathbf{r}'(\boldsymbol{\eta}_k))^\top \mathbf{r}'(\boldsymbol{\eta}_k). \tag{4.6}$$

Here λ_k is a non-negative scalar that dynamically changes from step to step. Details concerning the choice of λ_k can be found in [47]; see also Sect. 9.

4.2.3 Modified Form of LM

Experiments show that the standard form of the LM scheme applied to J'_{AML} can sometimes take relatively many iterations before reaching an optimal solution. To achieve a faster convergence, we shall modify the LM algorithm, replacing the standard weight matrix $\lambda \mathbf{I}_5$ in (4.6) by a non-standard, step-dependent, positive-definite weight matrix $\lambda \mathbf{W}_{\boldsymbol{\eta}}$. The customised scheme retains the logic of the standard scheme in that for large values of the scale factor λ , the LM step will align with a descent direction for J'_{AML} , which is the direction of $-(\lambda \mathbf{W}_{\boldsymbol{\eta}})^{-1} (\partial_{\boldsymbol{\eta}} \mathbf{r}'(\boldsymbol{\eta}_k))^\top \mathbf{r}'(\boldsymbol{\eta}_k)$. This direction is no longer the direction of the steepest descent in the ordinary Euclidean norm on the parameter space, but rather in the norm $\|\boldsymbol{\zeta}\| = (\boldsymbol{\zeta}^\top \mathbf{W}_{\boldsymbol{\eta}} \boldsymbol{\zeta})^{1/2}$ [43, Sect. 8.2.2]. For precedents on the use of weight matrices in LM different from multiples of the identity matrix, see [5, 44]. Passing to the specifics of our scheme, we shall adopt

$$\mathbf{W}_{\boldsymbol{\eta}} = (\partial_{\boldsymbol{\eta}} \boldsymbol{\pi}(\boldsymbol{\eta}))^\top \partial_{\boldsymbol{\eta}} \boldsymbol{\pi}(\boldsymbol{\eta}), \tag{4.7}$$

with the main ingredient of update rule then becoming

$$\delta \boldsymbol{\eta}_k = -((\partial_{\boldsymbol{\eta}} \mathbf{r}'(\boldsymbol{\eta}_k))^\top \partial_{\boldsymbol{\eta}} \mathbf{r}'(\boldsymbol{\eta}_k) + \lambda_k \mathbf{W}_{\boldsymbol{\eta}_k})^{-1} (\partial_{\boldsymbol{\eta}} \mathbf{r}'(\boldsymbol{\eta}_k))^\top \mathbf{r}'(\boldsymbol{\eta}_k). \tag{4.8}$$

The motivation behind this choice of the weight matrix has to do with the experimentally observed fact that the LM algorithm for optimising the non-parametrised function J_{AML} (which produces conics, but not necessarily ellipses, as optimal solutions), based on the rule

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \delta \boldsymbol{\theta}_k, \tag{4.9}$$

$$\delta \boldsymbol{\theta}_k = -((\partial_{\boldsymbol{\theta}} \mathbf{r}(\boldsymbol{\theta}_k))^\top \partial_{\boldsymbol{\theta}} \mathbf{r}(\boldsymbol{\theta}_k) + \lambda_k \mathbf{I}_6)^{-1} (\partial_{\boldsymbol{\theta}} \mathbf{r}(\boldsymbol{\theta}_k))^\top \mathbf{r}(\boldsymbol{\theta}_k), \tag{4.10}$$

runs faster than the standard LM algorithm for optimising J'_{AML} (as per (4.5) and (4.6)). One might hope then that if $\mathbf{W}_{\boldsymbol{\eta}}$ is chosen in such a way that the increment

$$\delta \boldsymbol{\eta} = -((\partial_{\boldsymbol{\eta}} \mathbf{r}')^\top \partial_{\boldsymbol{\eta}} \mathbf{r}' + \lambda \mathbf{W}_{\boldsymbol{\eta}})^{-1} (\partial_{\boldsymbol{\eta}} \mathbf{r}')^\top \mathbf{r}'$$

transformed into an increment of $\boldsymbol{\theta}$ by means of the relation

$$\delta \boldsymbol{\theta} = \partial_{\boldsymbol{\eta}} \boldsymbol{\pi} \delta \boldsymbol{\eta}$$

becomes

$$\delta \boldsymbol{\theta} = -((\partial_{\boldsymbol{\theta}} \mathbf{r})^\top \partial_{\boldsymbol{\theta}} \mathbf{r} + \lambda \mathbf{I}_6)^{-1} (\partial_{\boldsymbol{\theta}} \mathbf{r})^\top \mathbf{r}$$

and in so doing mimics the rule given in (4.10), then the LM algorithm employing such $\delta \boldsymbol{\eta}$ will run faster than the standard version of LM. Of course, the above is just a guiding principle whose merit has to be scrutinised experimentally. Fortunately, relevant results confirm the efficacy of the proposed approach—see Sect. 10.

We now discuss how the above condition on \mathbf{W}_η leads to formula (4.7). The defining property of \mathbf{W}_η is

$$\begin{aligned} & \partial_\eta \boldsymbol{\pi} ((\partial_\eta \mathbf{r}')^\top \partial_\eta \mathbf{r}' + \lambda \mathbf{W}_\eta)^{-1} (\partial_\eta \mathbf{r}')^\top \mathbf{r}' \\ &= ((\partial_\theta \mathbf{r})^\top \partial_\theta \mathbf{r} + \lambda \mathbf{I}_6)^{-1} (\partial_\theta \mathbf{r})^\top \mathbf{r}. \end{aligned}$$

Here, of course, \mathbf{r}' and $\partial_\eta \mathbf{r}'$ are evaluated at η , and \mathbf{r} and $\partial_\theta \mathbf{r}$ are evaluated at $\boldsymbol{\pi}(\eta)$; in particular, $\mathbf{r}' = \mathbf{r}'(\eta) = \mathbf{r}(\boldsymbol{\pi}(\eta)) = \mathbf{r}$. Taking into account that

$$\begin{aligned} & \partial_\eta \boldsymbol{\pi} ((\partial_\eta \mathbf{r}')^\top \partial_\eta \mathbf{r}' + \lambda \mathbf{W}_\eta)^{-1} (\partial_\eta \mathbf{r}')^\top \mathbf{r}' \\ &= \partial_\eta \boldsymbol{\pi} ((\partial_\eta \mathbf{r}')^\top \partial_\eta \mathbf{r}' + \lambda \mathbf{W}_\eta)^{-1} (\partial_\eta \boldsymbol{\pi})^\top (\partial_\theta \mathbf{r})^\top \mathbf{r}', \end{aligned}$$

we see that \mathbf{W}_η has to satisfy

$$\partial_\eta \boldsymbol{\pi} ((\partial_\eta \mathbf{r}')^\top \partial_\eta \mathbf{r}' + \lambda \mathbf{W}_\eta)^{-1} (\partial_\eta \boldsymbol{\pi})^\top = ((\partial_\theta \mathbf{r})^\top \partial_\theta \mathbf{r} + \lambda \mathbf{I}_6)^{-1}. \tag{4.11}$$

For a given matrix \mathbf{A} , denote by \mathbf{A}^+ the Moore–Penrose pseudo-inverse of \mathbf{A} [35]. It is easily seen that

$$(\partial_\eta \boldsymbol{\pi})^+ = ((\partial_\eta \boldsymbol{\pi})^\top \partial_\eta \boldsymbol{\pi})^{-1} (\partial_\eta \boldsymbol{\pi})^\top, \tag{4.12a}$$

$$((\partial_\eta \boldsymbol{\pi})^\top)^+ = \partial_\eta \boldsymbol{\pi} ((\partial_\eta \boldsymbol{\pi})^\top \partial_\eta \boldsymbol{\pi})^{-1}. \tag{4.12b}$$

Hence, in particular,

$$(\partial_\eta \boldsymbol{\pi})^+ \partial_\eta \boldsymbol{\pi} = \mathbf{I}_5, \tag{4.13a}$$

$$(\partial_\eta \boldsymbol{\pi})^\top ((\partial_\eta \boldsymbol{\pi})^\top)^+ = \mathbf{I}_5. \tag{4.13b}$$

Pre-multiplying and post-multiplying both sides of (4.11) by $(\partial_\eta \boldsymbol{\pi})^+$ and $((\partial_\eta \boldsymbol{\pi})^\top)^+$, respectively, and invoking (4.13a) and (4.13b), we get

$$\begin{aligned} ((\partial_\eta \mathbf{r}')^\top \partial_\eta \mathbf{r}' + \lambda \mathbf{W}_\eta)^{-1} &= (\partial_\eta \boldsymbol{\pi})^+ ((\partial_\theta \mathbf{r})^\top \partial_\theta \mathbf{r} + \lambda \mathbf{I}_6)^{-1} \\ &\quad \times ((\partial_\eta \boldsymbol{\pi})^\top)^+. \end{aligned}$$

Now the critical ingredient of our argument is the equality

$$\begin{aligned} & (\partial_\eta \boldsymbol{\pi})^+ ((\partial_\theta \mathbf{r})^\top \partial_\theta \mathbf{r} + \lambda \mathbf{I}_6)^{-1} ((\partial_\eta \boldsymbol{\pi})^\top)^+ \\ &= ((\partial_\eta \boldsymbol{\pi})^\top ((\partial_\theta \mathbf{r})^\top \partial_\theta \mathbf{r} + \lambda \mathbf{I}_6) \partial_\eta \boldsymbol{\pi})^{-1}. \end{aligned} \tag{4.14}$$

We defer the rather lengthy proof of this equality to Appendix 1. Assuming (4.14) for now, we find that

$$((\partial_\eta \mathbf{r}')^\top \partial_\eta \mathbf{r}' + \lambda \mathbf{W}_\eta)^{-1} = ((\partial_\eta \boldsymbol{\pi})^\top ((\partial_\theta \mathbf{r})^\top \partial_\theta \mathbf{r} + \lambda \mathbf{I}_6) \partial_\eta \boldsymbol{\pi})^{-1}$$

and further

$$\begin{aligned} (\partial_\eta \mathbf{r}')^\top \partial_\eta \mathbf{r}' + \lambda \mathbf{W}_\eta &= (\partial_\eta \boldsymbol{\pi})^\top ((\partial_\theta \mathbf{r})^\top \partial_\theta \mathbf{r} + \lambda \mathbf{I}_6) \partial_\eta \boldsymbol{\pi} \\ &= (\partial_\eta \mathbf{r}')^\top \partial_\eta \mathbf{r}' + \lambda (\partial_\eta \boldsymbol{\pi})^\top \partial_\eta \boldsymbol{\pi}. \end{aligned}$$

Hence, immediately, $\mathbf{W}_\eta = (\partial_\eta \boldsymbol{\pi})^\top \partial_\eta \boldsymbol{\pi}$, as desired.

5 Two Forms of Estimate Covariances

To be truly useful, an estimate of an ellipse must be accompanied by a reliable measure of its uncertainty characterising the dispersion of values that the estimate could reasonably attain. A fundamental advantage of the AML cost function over the ML cost function is that the former allows one to obtain, in a relatively straightforward way, a quantitative measure of uncertainty for any estimate related to that function. The measure takes the form of a covariance matrix. Here we present two closely related expressions for the covariance matrix of an AML estimate of an ellipse. The first of these concerns the case that the estimate is expressed in terms of the familiar algebraic parameters $\boldsymbol{\theta}$. The second concerns the case that the estimate is represented via the natural geometric parameters of an ellipse, namely centre co-ordinates, semi-major and semi-minor axes, and orientation.

5.1 Covariance of the Algebraically Parametrised Estimate

Any effort to assess the uncertainty of an estimate like $\widehat{\boldsymbol{\theta}}_{\text{AML}}$ has to involve assumptions as to the nature of noise in the data. The assumption behind the formula for the covariance matrix of $\widehat{\boldsymbol{\theta}}_{\text{AML}}$, $\mathbf{A}_{\widehat{\boldsymbol{\theta}}_{\text{AML}}}$, is the same as that underlying the development of J_{AML} , namely that the observed data points $\mathbf{x}_1, \dots, \mathbf{x}_N$ are noisy versions of noise-free data points, with the noise in each \mathbf{x}_n independent and Gaussian, of zero mean and covariance matrix $\mathbf{A}_{\mathbf{x}_n}$.

Given an $m \times m$ matrix \mathbf{A} and a positive integer r no greater than m , we denote by \mathbf{A}_r the r -truncated SVD of \mathbf{A} and by \mathbf{A}_r^+ the r -truncated pseudo-inverse of \mathbf{A} . These are defined as follows. If $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ is the singular value decomposition (SVD) of \mathbf{A} , with $\mathbf{D} = \text{diag}(d_1, \dots, d_m)$, then $\mathbf{A}_r = \mathbf{U}\mathbf{D}_r\mathbf{V}^\top$ with $\mathbf{D}_r = \text{diag}(d_1, \dots, d_r, 0, \dots, 0)$, and $\mathbf{A}_r^+ = \mathbf{V}\mathbf{D}_r^+\mathbf{U}^\top$ with $\mathbf{D}_r^+ = \text{diag}(d_1^+, \dots, d_r^+, 0, \dots, 0)$, where $d_i^+ = d_i^{-1}$ when $d_i \neq 0$ and $d_i^+ = 0$ otherwise. Continuing with the preparations, we introduce the matrix

$$\mathbf{M}_\theta = \sum_{n=1}^N \frac{\mathbf{A}_n}{\boldsymbol{\theta}^\top \mathbf{B}_n \boldsymbol{\theta}}. \tag{5.1}$$

We are now ready to present our first covariance matrix formula. Under the assumption that $\widehat{\boldsymbol{\theta}}_{\text{AML}}$ is normalised, $\|\widehat{\boldsymbol{\theta}}_{\text{AML}}\| = 1$, the covariance matrix of $\widehat{\boldsymbol{\theta}}_{\text{AML}}$ is given by

$$\mathbf{A}_{\widehat{\boldsymbol{\theta}}_{\text{AML}}} = \mathbf{P}_{\widehat{\boldsymbol{\theta}}_{\text{AML}}}^\perp (\mathbf{M}_{\widehat{\boldsymbol{\theta}}_{\text{AML}}})_5^+ \mathbf{P}_{\widehat{\boldsymbol{\theta}}_{\text{AML}}}^\perp. \tag{5.2}$$

We remark that the presence of $\mathbf{P}_{\widehat{\boldsymbol{\theta}}_{\text{AML}}}^\perp$ here makes the matrix $\mathbf{A}_{\widehat{\boldsymbol{\theta}}_{\text{AML}}}$ singular, with $\widehat{\boldsymbol{\theta}}_{\text{AML}}$ in the null space of $\mathbf{A}_{\widehat{\boldsymbol{\theta}}_{\text{AML}}}$. This reflects the particular way in which the scale ambiguity of the estimates has been eliminated, namely by scaling the estimates to unit length. The details of the derivation of (5.2)

are somewhat technical and are deferred to Appendix 2. For additional results related to the content of Appendix 2, the interested reader is referred to [18] and [31].

5.2 Covariance of the Geometrically Parametrised Estimate

Alongside a θ -vector of algebraic parameters, an ellipse can be represented by a ξ -vector of *geometric parameters*, $\xi = [A, B, H, K, \tau]^T$. Here A and B denote the *semi-major axis* and the *semi-minor axis* of the ellipse, H and K denote the x and y coordinates of the *centre* of the ellipse, and τ is the *angle* formed by the major axis with the positive x -axis. Recalling that Δ denotes the discriminant ($\Delta = b^2 - 4ac$), we let

$$\lambda_{\pm} = \frac{1}{2} \left(a + c \mp (b^2 + (a - c)^2)^{1/2} \right),$$

$$\psi = bde - ae^2 - b^2f + c(4af - d^2),$$

$$V^{\pm} = \left(\frac{\psi}{\lambda_{\pm}\Delta} \right)^{1/2},$$

where \pm and \mp are shorthand for *+* or *-* that allow presentation of two expressions in one formula, with the upper $-$ of \mp associated with the $+$ of \pm . It is a matter of a straightforward but tedious analysis to establish the following conversion rules for passing from the θ -based to the ξ -based description of the ellipse (see [62, Sect. 4.10.2] for the starting point of the derivation of the formulae):

$$A = \max(V^+, V^-), \quad B = \min(V^+, V^-), \tag{5.3}$$

$$H = \frac{2cd - be}{\Delta}, \quad K = \frac{2ae - bd}{\Delta}, \tag{5.4}$$

$$\tau = \begin{cases} \frac{1}{2} \operatorname{arccot} \left(\frac{a-c}{b} \right) & \text{if } b < 0, a < c \text{ and } V^+ \geq V^-, \\ \frac{\pi}{4} & \text{if } b < 0, a = c \text{ and } V^+ \geq V^-, \\ \frac{1}{2} \operatorname{arccot} \left(\frac{a-c}{b} \right) + \frac{\pi}{2} & \text{if } b < 0, a > c \text{ and } V^+ \geq V^-, \\ 0 & \text{if } b = 0, a < c \text{ and } V^+ \geq V^-, \\ \frac{\pi}{2} & \text{if } b = 0, a \geq c \text{ and } V^+ \geq V^-, \\ \frac{1}{2} \operatorname{arccot} \left(\frac{a-c}{b} \right) + \pi & \text{if } b > 0, a < c \text{ and } V^+ \geq V^-, \\ \frac{3\pi}{4} & \text{if } b > 0, a = c \text{ and } V^+ \geq V^-, \\ \frac{1}{2} \operatorname{arccot} \left(\frac{a-c}{b} \right) + \frac{\pi}{2} & \text{if } b > 0, a > c \text{ and } V^+ \geq V^-, \\ \frac{1}{2} \operatorname{arccot} \left(\frac{a-c}{b} \right) + \frac{\pi}{2} & \text{if } b < 0, a < c \text{ and } V^+ < V^-, \\ \frac{3\pi}{4} & \text{if } b < 0, a = c \text{ and } V^+ < V^-, \\ \frac{1}{2} \operatorname{arccot} \left(\frac{a-c}{b} \right) + \pi & \text{if } b < 0, a > c \text{ and } V^+ < V^-, \\ \frac{\pi}{2} & \text{if } b = 0, a < c \text{ and } V^+ < V^-, \\ 0 & \text{if } b = 0, a \geq c \text{ and } V^+ < V^-, \\ \frac{1}{2} \operatorname{arccot} \left(\frac{a-c}{b} \right) + \frac{\pi}{2} & \text{if } b > 0, a < c \text{ and } V^+ < V^-, \\ \frac{\pi}{4} & \text{if } b > 0, a = c \text{ and } V^+ < V^-, \\ \frac{1}{2} \operatorname{arccot} \left(\frac{a-c}{b} \right) & \text{if } b > 0, a > c \text{ and } V^+ < V^-. \end{cases} \tag{5.5}$$

We remark that the formula for τ is valid only under the assumption that the ellipse is not a circle; that is, provided the inequality $(a - c)^2 + b^2 > 0$ holds.

With the aid of (5.3)–(5.5), the geometric content of $\widehat{\theta}_{\text{AML}}$ can be revealed as $\xi(\widehat{\theta}_{\text{AML}})$. We term $\xi(\widehat{\theta}_{\text{AML}})$ the AML estimate of the geometric parameters of an ellipse and denote it by $\widehat{\xi}_{\text{AML}}$. For consistency, we shall refer to $\widehat{\theta}_{\text{AML}}$ as the AML estimate of the algebraic parameters of an ellipse. The rule of covariance propagation readily permits finding the covariance matrix of the AML estimate of the geometric parameters of an ellipse from the covariance matrix of the AML estimate of the algebraic parameters. Assuming that $\widehat{\theta}_{\text{AML}}$ is normed to unity, we have

$$A_{\widehat{\xi}_{\text{AML}}} = [\partial_{\theta} \xi]_{\theta=\widehat{\theta}_{\text{AML}}} A_{\widehat{\theta}_{\text{AML}}} [(\partial_{\theta} \xi)]_{\theta=\widehat{\theta}_{\text{AML}}}^T, \tag{5.6}$$

The 5×6 Jacobian matrix $\partial_{\theta} \xi = [\partial_{\theta} A^T, \partial_{\theta} B^T, \partial_{\theta} H^T, \partial_{\theta} K^T, \partial_{\theta} \tau^T]^T$ can be specified explicitly as follows. The first two rows are given by

$$\partial_{\theta} A = \begin{cases} \partial_{\theta} V^+ & \text{if } A = V^+, \\ \partial_{\theta} V^- & \text{if } A = V^-, \end{cases} \tag{5.7}$$

$$\partial_{\theta} B = \begin{cases} \partial_{\theta} V^+ & \text{if } B = V^+, \\ \partial_{\theta} V^- & \text{if } B = V^-, \end{cases} \tag{5.8}$$

where $\partial_{\theta} V^{\pm} = [\partial_a V^{\pm}, \partial_b V^{\pm}, \partial_c V^{\pm}, \partial_d V^{\pm}, \partial_e V^{\pm}, \partial_f V^{\pm}]$ is given by

$$\partial_a V^{\pm} = \frac{1}{2\lambda_{\pm}\Delta} \left(\frac{\psi}{\lambda_{\pm}\Delta} \right)^{-1/2} \left[4cf - e^2 + 4\Delta^{-1}c\psi - \frac{\psi}{2\lambda_{\pm}} \left(1 \pm \frac{c-a}{((a-c)^2 + b^2)^{1/2}} \right) \right], \tag{5.9}$$

$$\partial_b V^{\pm} = \frac{1}{2\lambda_{\pm}\Delta} \left(\frac{\psi}{\lambda_{\pm}\Delta} \right)^{-1/2} \left[de - 2bf - 2\Delta^{-1}b\psi \pm \frac{b\psi}{2\lambda_{\pm}((a-c)^2 + b^2)^{1/2}} \right], \tag{5.10}$$

$$\partial_c V^{\pm} = \frac{1}{2\lambda_{\pm}\Delta} \left(\frac{\psi}{\lambda_{\pm}\Delta} \right)^{-1/2} \left[4af - d^2 + 4\Delta^{-1}a\psi - \frac{\psi}{2\lambda_{\pm}} \left(1 \pm \frac{a-c}{((a-c)^2 + b^2)^{1/2}} \right) \right], \tag{5.11}$$

$$\partial_d V^{\pm} = \left(\frac{\psi}{\lambda_{\pm}\Delta} \right)^{1/2} \frac{be - 2cd}{2\psi}, \tag{5.12}$$

$$\partial_e V^{\pm} = \left(\frac{\psi}{\lambda_{\pm}\Delta} \right)^{1/2} \frac{bd - 2ae}{2\psi}, \tag{5.13}$$

$$\partial_f V^{\pm} = -\frac{1}{2\lambda_{\pm}} \left(\frac{\psi}{\lambda_{\pm}\Delta} \right)^{-1/2}. \tag{5.14}$$

The remaining three rows are given by

$$\partial_{\theta} H = \left[\frac{4c(2cd - be)}{\Delta^2}, \frac{b^2e + 4ace - 4bcd}{\Delta^2}, \frac{2b(bd - 2ae)}{\Delta^2}, \frac{2c}{\Delta}, -\frac{b}{\Delta}, 0 \right], \tag{5.15}$$

$$\partial_{\theta} K = \left[\frac{2b(be - 2cd)}{\Delta^2}, \frac{b^2d + 4acd - 4abe}{\Delta^2}, \frac{4a(2ae - bd)}{\Delta^2}, -\frac{b}{\Delta}, \frac{2a}{\Delta}, 0 \right], \tag{5.16}$$

$$\partial_{\theta} \tau = \left[-\frac{b}{2(b^2 + (a - c)^2)}, \frac{a - c}{2(b^2 + (a - c)^2)}, \frac{b}{2(b^2 + (a - c)^2)}, 0, 0, 0 \right]. \tag{5.17}$$

6 Confidence Regions

The reliability of an AML estimate of an ellipse can be suggestively expressed using the concept of a confidence region. It is standard to build a confidence region of a parameter vector estimate as a portion of the parameter space that contains the true parameter vector with a given high probability. The parameter space for all ellipses is five dimensional and as such is difficult to grasp. Here we propose a more visually appealing form of ellipse-specific confidence region, namely a confidence region in the plane. Such a region is meant to cover the in-plane locus of the true ellipse with a given high probability.

The probability statement associated with a confidence region requires some explanation if it is not to be misinterpreted. The point is that the statement is not meant to imply that *every* confidence region contains the true parameters with a specified high probability. Rather, the probability expresses how *frequently* a confidence region can be expected to contain the true parameters in the long-run process of taking random data points and obtaining regions.

The first to consider planar confidence regions for ellipse fits was Porcill [46]. Our approach is inspired by Scheffé’s S-method for constructing simultaneous confidence bands for linear regression [51], [34, Sect. 9.4–5]. The starting point for the construction is the observation that when $\hat{\theta}_{AML}$ is viewed as a multivariate normally distributed random vector,

$$\hat{\theta}_{AML} \sim N(\theta_*, \Lambda_{\theta_*}),$$

where θ_* is the parameter vector of the true ellipse and Λ_{θ_*} is a covariance matrix, the scalar random variable $\hat{\theta}_{AML}^T \mathbf{u}(\mathbf{x})$ is normally distributed with variance $\mathbf{u}(\mathbf{x})^T \Lambda_{\theta_*} \mathbf{u}(\mathbf{x})$ for every point \mathbf{x} on the locus $\mathcal{E}_{\theta_*} = \{\mathbf{x} \in \mathbb{R}^2 \mid \theta_*^T \mathbf{u}(\mathbf{x}) = 0\}$ of the true ellipse. The observation is based on the fact that $\hat{\theta}_{AML}^T$

$\mathbf{u}(\mathbf{x}) = (\hat{\theta}_{AML} - \theta_*)^T \mathbf{u}(\mathbf{x})$ whenever $\mathbf{x} \in \mathcal{E}_{\theta_*}$ and the fact that by the rule of covariance propagation $(\hat{\theta}_{AML} - \theta_*)^T \mathbf{u}(\mathbf{x})$ has variance $\mathbf{u}(\mathbf{x})^T \Lambda_{\theta_*} \mathbf{u}(\mathbf{x})$. Consequently, under the assumption that $\hat{\theta}_{AML}$ is an unbiased estimate of θ_* ,¹

$$z_{\mathbf{x}} = \frac{(\hat{\theta}_{AML}^T \mathbf{u}(\mathbf{x}))^2}{\mathbf{u}(\mathbf{x})^T \Lambda_{\theta_*} \mathbf{u}(\mathbf{x})}$$

is a squared normal random variable for every $\mathbf{x} \in \mathcal{E}_{\theta_*}$. Each $z_{\mathbf{x}}$, insofar as \mathbf{x} belongs to \mathcal{E}_{θ_*} , attains large values with less probability than small values, with the probability of any particular set of values regarded as large or small being independent of \mathbf{x} . This suggests using the $z_{\mathbf{x}}$ as building blocks in the construction of a confidence region in the plane. Since the covariance Λ_{θ_*} is unknown, the $z_{\mathbf{x}}$ do not have observable realisations and, for the sake of construction, have to be replaced with these variables’ *observable* variants

$$\hat{z}_{\mathbf{x}} = \frac{(\hat{\theta}_{AML}^T \mathbf{u}(\mathbf{x}))^2}{\mathbf{u}(\mathbf{x})^T \Lambda_{\hat{\theta}_{AML}} \mathbf{u}(\mathbf{x})}, \tag{6.1}$$

where the covariance estimate $\Lambda_{\hat{\theta}_{AML}}$ serves as a natural replacement for Λ_{θ_*} . Again, large observed values of $\hat{z}_{\mathbf{x}}$ are less plausible than small observed values as long as $\mathbf{x} \in \mathcal{E}_{\theta_*}$. It is thus natural to consider confidence regions for $\hat{\theta}_{AML}$ in the form

$$\{\mathbf{x} \in \mathbb{R}^2 \mid \hat{z}_{\mathbf{x}} \leq c\},$$

where c is a positive constant. Ideally, for a confidence region at (confidence) level $1 - \alpha$, we should choose c such that

$$P(z_{\mathbf{x}} \leq c \text{ for all } \mathbf{x} \in \mathcal{E}_{\theta_*}) = P\left(\sup_{\mathbf{x} \in \mathcal{E}_{\theta_*}} z_{\mathbf{x}} \leq c\right) = 1 - \alpha,$$

where $P(A)$ denotes the probability of the event A . But the distribution of $\sup_{\mathbf{x} \in \mathcal{E}_{\theta_*}} z_{\mathbf{x}}$ is not easy to determine, so as a second best choice we shall replace $\sup_{\mathbf{x} \in \mathcal{E}_{\theta_*}} z_{\mathbf{x}}$ by a random upper bound whose distribution can be readily calculated. Proceeding to the specifics, we may assume, in line with fact that the parameter space of all ellipses is five-dimensional, that $(\hat{\theta}_{AML} - \theta_*)^T \theta_* = 0$, or equivalently, $\hat{\theta}_{AML} - \theta_* = \mathbf{P}_{\theta_*}^{\perp} (\hat{\theta}_{AML} - \theta_*)$. This then leads to

$$\mathbf{P}_{\theta_*}^{\perp} \Lambda_{\theta_*} = \Lambda_{\theta_*} \mathbf{P}_{\theta_*}^{\perp} = \Lambda_{\theta_*}. \tag{6.2}$$

Now, if $\mathbf{x} \in \mathcal{E}_{\theta_*}$, then

$$\hat{\theta}_{AML}^T \mathbf{u}(\mathbf{x}) = (\hat{\theta}_{AML} - \theta_*)^T \mathbf{u}(\mathbf{x}) = (\hat{\theta}_{AML} - \theta_*)^T \mathbf{P}_{\theta_*}^{\perp} \mathbf{u}(\mathbf{x}).$$

¹ This is a realistic assumption, as $\hat{\theta}_{AML}$ is known to be an unbiased estimate of θ_* up to the first order. See [26], [27, Sect. 10.2.2], and [37] for more details.

But it follows from (6.2) that $(\mathbf{A}_{\theta_*}^+)^{1/2} \mathbf{A}_{\theta_*}^{1/2} = \mathbf{P}_{\theta_*}^\perp$, so

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_{\text{AML}}^\top \mathbf{u}(\mathbf{x}) &= (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*)^\top (\mathbf{A}_{\theta_*}^+)^{1/2} \mathbf{A}_{\theta_*}^{1/2} \mathbf{u}(\mathbf{x}) \\ &= \left((\mathbf{A}_{\theta_*}^+)^{1/2} (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*) \right)^\top \mathbf{A}_{\theta_*}^{1/2} \mathbf{u}(\mathbf{x}). \end{aligned}$$

By the Cauchy–Bunyakovsky–Schwarz inequality,

$$(\widehat{\boldsymbol{\theta}}_{\text{AML}}^\top \mathbf{u}(\mathbf{x}))^2 \leq \|(\mathbf{A}_{\theta_*}^+)^{1/2} (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*)\|^2 \|\mathbf{A}_{\theta_*}^{1/2} \mathbf{u}(\mathbf{x})\|^2.$$

Also

$$\begin{aligned} &\|(\mathbf{A}_{\theta_*}^+)^{1/2} (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*)\|^2 \\ &= (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*)^\top \mathbf{A}_{\theta_*}^+ (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*) \end{aligned}$$

and

$$\|\mathbf{A}_{\theta_*}^{1/2} \mathbf{u}(\mathbf{x})\|^2 = \mathbf{u}(\mathbf{x})^\top \mathbf{A}_{\theta_*} \mathbf{u}(\mathbf{x}).$$

Hence

$$z_{\mathbf{x}} \leq (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*)^\top \mathbf{A}_{\theta_*}^+ (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*).$$

Since \mathbf{x} is an arbitrary member of \mathcal{E}_{θ_*} , we have

$$\sup_{\mathbf{x} \in \mathcal{E}_{\theta_*}} z_{\mathbf{x}} \leq (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*)^\top \mathbf{A}_{\theta_*}^+ (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*). \tag{6.3}$$

Now the random variable $(\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*)^\top \mathbf{A}_{\theta_*}^+ (\widehat{\boldsymbol{\theta}}_{\text{AML}} - \boldsymbol{\theta}_*)$ has approximately a chi-squared distribution with 5 degree of freedom. Let $\chi_{5,\alpha}^2$ denote the 100(1 - α)% percentile of the χ^2 distribution with 5 degrees of freedom, characterised by the relation $\mathbf{P}(\chi^2 \leq \chi_{5,\alpha}^2) = 1 - \alpha$. Inequality (6.3) guarantees that

$$\mathbf{P}\left(\sup_{\mathbf{x} \in \mathcal{E}_{\theta_*}} z_{\mathbf{x}} \leq \chi_{5,\alpha}^2\right) \geq 1 - \alpha$$

Substituting $\mathbf{A}_{\widehat{\boldsymbol{\theta}}_{\text{AML}}}$ for \mathbf{A}_{θ_*} , we also approximately have

$$\mathbf{P}\left(\sup_{\mathbf{x} \in \mathcal{E}_{\theta_*}} \hat{z}_{\mathbf{x}} \leq \chi_{5,\alpha}^2\right) \geq 1 - \alpha.$$

This allows an approximate confidence region at level 1 - α for $\widehat{\boldsymbol{\theta}}_{\text{AML}}$ to be taken as

$$\Omega_\alpha = \left\{ \mathbf{x} \in \mathbb{R}^2 \mid \hat{z}_{\mathbf{x}} \leq \chi_{5,\alpha}^2 \right\}. \tag{6.4}$$

We finally point out that if α is set to the standard conventional value of 0.05, then $\chi_{5,\alpha}^2 = 11.07$.

7 Data Normalisation

For reason of numerical stability, it is important to incorporate *data normalisation*, or *pre-conditioning*, in the process of calculating AML estimates and their covariances. One popular form of data normalisation is due to Hartley [12, 14, 23] and can be formulated with the aid of a data-dependent 3×3 matrix \mathbf{T} given by

$$\mathbf{T} = \begin{bmatrix} s^{-1} & 0 & -s^{-1}\bar{m}_1 \\ 0 & s^{-1} & -s^{-1}\bar{m}_2 \\ 0 & 0 & 1 \end{bmatrix},$$

where

$$\bar{m}_i = \frac{1}{N} \sum_{n=1}^N m_{n,i} \quad (i = 1, 2)$$

and

$$s = \left(\frac{1}{2N} \sum_{n=1}^N (m_{n,1} - \bar{m}_1)^2 + (m_{n,2} - \bar{m}_2)^2 \right)^{1/2}.$$

If $\mathbf{m} = [m_1, m_2, 1]^\top$ represents a data point in the original, un-normalised coordinates, then $\tilde{\mathbf{m}} = [\tilde{m}_1, \tilde{m}_2, 1]^\top$ defined by $\tilde{\mathbf{m}} = \mathbf{T}\mathbf{m}$ represents the same point in the normalised coordinates. The form of \mathbf{T} reflects the fact that the transformation $\mathbf{m} \mapsto \tilde{\mathbf{m}}$ scales down the data set to a unit box with centre at the origin. In the normalised coordinates, Eq. (2.2) can be equivalently written as $\tilde{\boldsymbol{\theta}}^\top \mathbf{u}(\tilde{\mathbf{x}}) = 0$ with $\tilde{\mathbf{x}} = [\tilde{m}_1, \tilde{m}_2]^\top$ and $\tilde{\boldsymbol{\theta}}$ related to $\boldsymbol{\theta}$ via the relation

$$\tilde{\boldsymbol{\theta}} = \mathbf{E}^{-1} \mathbf{P}_{(34)} \mathbf{D}_3 (\mathbf{T} \otimes \mathbf{T})^{-\top} \mathbf{D}_3 \mathbf{P}_{(34)} \mathbf{E} \boldsymbol{\theta}. \tag{7.1}$$

Here, \mathbf{D}_3 is the 9×6 duplication matrix given by

$$\mathbf{D}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$\mathbf{P}_{(34)}$ is the permutation matrix for interchanging the 3rd and 4th entries of length-6 vectors, given by

$$\mathbf{P}_{(34)} = \text{diag}(0, 1, 0) \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \text{diag}(1, 0, 1) \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and $\mathbf{E} = \text{diag}(1, 2^{-1}, 1, 2^{-1}, 2^{-1}, 1)$. The computation of $\widehat{\boldsymbol{\theta}}_{\text{AML}}$ involving data normalisation proceeds in two steps. First, an AML estimate of $\tilde{\boldsymbol{\theta}}$ relative to the normalised coordinates, $\tilde{\boldsymbol{\theta}}_{\text{AML}}$, is extracted by minimising the AML cost function

$$\tilde{J}_{\text{AML}}(\tilde{\boldsymbol{\theta}}) = \sum_{n=1}^N \frac{\tilde{\boldsymbol{\theta}}^\top \tilde{\mathbf{A}}_n \tilde{\boldsymbol{\theta}}}{\tilde{\boldsymbol{\theta}}^\top \tilde{\mathbf{B}}_n \tilde{\boldsymbol{\theta}}}$$

where, for each $n = 1, \dots, N$, $\tilde{\mathbf{A}}_n = \mathbf{u}(\tilde{\mathbf{x}}_n) \mathbf{u}(\tilde{\mathbf{x}}_n)^\top$, $\tilde{\mathbf{B}}_n = \partial_{\tilde{\mathbf{x}}} \mathbf{u}(\tilde{\mathbf{x}}_n) \partial_{\tilde{\mathbf{x}}} \mathbf{u}(\tilde{\mathbf{x}}_n)^\top$, and $\mathbf{A}_{\tilde{\mathbf{x}}_n}$ is the result of the propagation of $\mathbf{A}_{\mathbf{x}_n}$ to the normalised coordinates, namely

$$\begin{bmatrix} \mathbf{A}_{\tilde{\mathbf{x}}_n} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{A}_{\mathbf{x}_n} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} \mathbf{T}^\top.$$

Next, $\hat{\boldsymbol{\theta}}_{\text{AML}}$ is obtained by the un-normalisation procedure captured in the formula

$$\hat{\boldsymbol{\theta}}_{\text{AML}} = \mathbf{E}^{-1} \mathbf{P}_{(34)} \mathbf{D}_3^+ (\mathbf{T} \otimes \mathbf{T})^\top \mathbf{D}_3 \mathbf{P}_{(34)} \mathbf{E} \hat{\boldsymbol{\theta}}_{\text{AML}}. \tag{7.2}$$

This formula is an instance of the inverse equation to (7.1) and is an immediate consequence of the identity $J_{\text{AML}}(\boldsymbol{\theta}) = \tilde{J}_{\text{AML}}(\hat{\boldsymbol{\theta}})$. Notice that $\hat{\boldsymbol{\theta}}_{\text{AML}}$ given in (7.2) does not necessarily have unit length.

The computation of $\mathbf{A}_{\hat{\boldsymbol{\theta}}_{\text{AML}}}$ involving data normalisation starts with the computation of $\mathbf{A}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}}$ which is nothing else but an application of (5.2) with $\mathbf{P}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}}^\perp$ substituted for $\mathbf{P}_{\boldsymbol{\theta}_{\text{AML}}}^\perp$ and $\mathbf{M}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}}$ substituted for $\mathbf{M}_{\boldsymbol{\theta}_{\text{AML}}}$, where, of course,

$$\mathbf{M}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}} = \sum_{n=1}^N \frac{\tilde{\mathbf{A}}_n}{\tilde{\boldsymbol{\theta}}_{\text{AML}}^\top \tilde{\mathbf{B}}_n \tilde{\boldsymbol{\theta}}_{\text{AML}}}.$$

Here $\tilde{\boldsymbol{\theta}}_{\text{AML}}$ is assumed to be of unit length. Once $\mathbf{A}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}}$ is found, $\mathbf{A}_{\hat{\boldsymbol{\theta}}_{\text{AML}}}$ is given by

$$\mathbf{A}_{\hat{\boldsymbol{\theta}}_{\text{AML}}} = \|\hat{\boldsymbol{\theta}}_{\text{AML}}\|^{-2} \mathbf{P}_{\hat{\boldsymbol{\theta}}_{\text{AML}}}^\perp \mathbf{F} \mathbf{A}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}} \mathbf{F}^\top \mathbf{P}_{\hat{\boldsymbol{\theta}}_{\text{AML}}}^\perp, \tag{7.3}$$

where \mathbf{F} is short for $\mathbf{E}^{-1} \mathbf{P}_{(34)} \mathbf{D}_3^+ (\mathbf{T} \otimes \mathbf{T})^\top \mathbf{D}_3 \mathbf{P}_{(34)} \mathbf{E}$ and $\hat{\boldsymbol{\theta}}_{\text{AML}}$ on the right-hand side is given by (7.2)—that is, $\hat{\boldsymbol{\theta}}_{\text{AML}} = \mathbf{F} \tilde{\boldsymbol{\theta}}_{\text{AML}}$. The above expression for $\mathbf{A}_{\hat{\boldsymbol{\theta}}_{\text{AML}}}$ is the result of an application of the rule of covariance propagation to $\mathbf{A}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}}$ and the mapping $\tilde{\boldsymbol{\theta}} \mapsto \mathbf{F} \tilde{\boldsymbol{\theta}} / \|\mathbf{F} \tilde{\boldsymbol{\theta}}\|$.

Similarly, the data normalisation based computation of $\mathbf{A}_{\hat{\boldsymbol{\xi}}_{\text{AML}}}$ starts with the computation of $\mathbf{A}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}}$. The matrix $\mathbf{A}_{\tilde{\boldsymbol{\xi}}_{\text{AML}}}$ is next retrieved by applying the formula

$$\mathbf{A}_{\tilde{\boldsymbol{\xi}}_{\text{AML}}} = [\partial_{\tilde{\boldsymbol{\theta}}} \tilde{\boldsymbol{\xi}}]_{\tilde{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}_{\text{AML}}} \mathbf{A}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}} [(\partial_{\tilde{\boldsymbol{\theta}}} \tilde{\boldsymbol{\xi}})]_{\tilde{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}_{\text{AML}}}^\top,$$

which is a variant of (5.2) relative to the normalised coordinates. Finally, $\mathbf{A}_{\hat{\boldsymbol{\xi}}_{\text{AML}}}$ is given by the expression

$$\mathbf{A}_{\hat{\boldsymbol{\xi}}_{\text{AML}}} = \mathbf{S} \mathbf{A}_{\tilde{\boldsymbol{\xi}}_{\text{AML}}} \mathbf{S}^\top, \quad \mathbf{S} = \text{diag}(s, s, s, s, 1),$$

which encodes the covariance propagation for geometric parameters from the normalised to the un-normalised coordinates.

The procedures described above involve an arbitrary set of covariance matrices $\mathbf{A}_{\mathbf{x}_n}$. A common assumption in practice is that the noise in the data set is homogeneous isotropic Gaussian, so that $\mathbf{A}_{\mathbf{x}_n} = \sigma^2 \mathbf{I}_2$ for each $n = 1, \dots, N$, where σ is a common standard deviation. The value of σ is generally not known *a priori*. This value can, however, be learned from

the data. The key is the fact that two variants of J_{AML} , one based on the covariances $\mathbf{A}_{\mathbf{x}_n} = \sigma^2 \mathbf{I}_2$ and one based on the covariances $\mathbf{A}_{\mathbf{x}_n} = \mathbf{I}_2$, differ only by a multiplicative constant and, as a result, lead to a common value of $\hat{\boldsymbol{\theta}}_{\text{AML}}$. In other words, $\hat{\boldsymbol{\theta}}_{\text{AML}}$ can be obtained without knowing the actual value of σ , by minimising J_{AML} under the default assumption that $\mathbf{A}_{\mathbf{x}_n} = \mathbf{I}_2$. Once this is done, one can take

$$\hat{\sigma} = \sqrt{\frac{J_{\text{AML}}(\hat{\boldsymbol{\theta}}_{\text{AML}})}{N - 5}}$$

for an estimate of σ (cf. [27, Sect. 7.1.4]). For all practical purposes, σ may safely be identified with $\hat{\sigma}$. With this in effect, the $\mathbf{A}_{\mathbf{x}_n}$ may be assumed to be fully known.

8 Normalised Confidence Regions

Data normalisation can also be used in forming confidence regions. To generate a confidence region based on normalised data, we first define

$$\hat{z}_{\tilde{\mathbf{x}}} = \frac{(\tilde{\boldsymbol{\theta}}_{\text{AML}}^\top \tilde{\mathbf{u}}(\tilde{\mathbf{x}}))^2}{\tilde{\mathbf{u}}(\tilde{\mathbf{x}})^\top \mathbf{A}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}} \tilde{\mathbf{u}}(\tilde{\mathbf{x}})},$$

where $\tilde{\boldsymbol{\theta}}_{\text{AML}}$ is assumed to have unit norm. We then let

$$\tilde{\Delta}_\alpha = \{ \tilde{\mathbf{x}} \in \mathbb{R}^2 \mid \hat{z}_{\tilde{\mathbf{x}}} \leq \chi_{5,\alpha}^2 \}$$

and finally take

$$\Delta_\alpha = \{ \mathbf{x} \in \mathbb{R}^2 \mid \tilde{\mathbf{x}} \in \tilde{\Delta}_\alpha \}$$

for an approximate confidence region at level $1 - \alpha$ for $\hat{\boldsymbol{\theta}}_{\text{AML}}$. Equivalently, Δ_α can be defined as

$$\Delta_\alpha = \{ \mathbf{x} \in \mathbb{R}^2 \mid \hat{z}_{\mathbf{x}} \leq \chi_{5,\alpha}^2 \}$$

provided that $\hat{z}_{\mathbf{x}}$ is taken with $\hat{\boldsymbol{\theta}}_{\text{AML}} = \mathbf{F} \tilde{\boldsymbol{\theta}}_{\text{AML}}$ and $\mathbf{A}_{\hat{\boldsymbol{\theta}}_{\text{AML}}} = \mathbf{F} \mathbf{A}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}} \mathbf{F}^\top$,² with $\tilde{\boldsymbol{\theta}}_{\text{AML}}$ remaining normalised; this follows from the fact that with $\tilde{\boldsymbol{\theta}}_{\text{AML}}$ and $\mathbf{A}_{\tilde{\boldsymbol{\theta}}_{\text{AML}}}$ as above, we have $\hat{z}_{\mathbf{x}} = \hat{z}_{\tilde{\mathbf{x}}}$. Now, as a rule, the set Δ_α will be different from the set Ω_α , specified in (6.4), representing an approximate confidence region at level $1 - \alpha$ for $\hat{\boldsymbol{\theta}}_{\text{AML}}$ based on raw data. While typically the difference between confidence regions constructed using normalised versus raw data is rather small, normalised confidence regions tend to be more visually pleasing.

² This is a genuine formula for the covariance matrix of $\hat{\boldsymbol{\theta}}_{\text{AML}}$ corresponding a different *gauge constraint* than the one underlying formula (7.3), which is the constraint that $\tilde{\boldsymbol{\theta}}_{\text{AML}}$ be scaled to unit norm. Gauge constraints serve to eliminate redundant degrees of freedom in the parametrisation and lead to *gauged covariances*. See [57, Sect. 9] for more details.

9 Algorithms

Algorithm 1: GUARANTEED ELLIPSEFIT

Input: $\widehat{\theta}_{\text{DIR}}$, **data_points**
Output: $\Omega_k.\eta$

/ a data structure $\Omega = \{\mathbf{r}'(\eta), \partial_\eta \pi(\eta), \partial_\eta \mathbf{r}'(\eta), \mathbf{H}(\eta), \text{cost}_\eta, \eta, \Delta_\eta, \lambda, \nu, \gamma, \eta_wasUpdated\}$ is used to pass parameters between functions */*

- 1 **keep_going** \leftarrow **true**; **use_pseudo-inverse** \leftarrow **false**; $\lambda \leftarrow 0.01$; $k \leftarrow 1$; $\nu \leftarrow 1.2$; $\gamma \leftarrow 15$; $\Omega_1.\Delta_\eta \leftarrow [1, 1, 1, 1, 1]$; $\eta \leftarrow \mathbf{c}(\widehat{\theta}_{\text{DIR}})$ (see equation (4.3))
- 2 **while** **keep_going** and $k < \text{maxIter}$ **do**
- 3 $\Omega_k.\mathbf{r}'(\eta) \leftarrow [r'_1(\eta), \dots, r'_{N+1}(\eta)]^\top$ */* residuals computed on data_points */*
- 4 $\Omega_k.\partial_\eta \pi(\eta) \leftarrow \partial_\eta \pi(\eta)$ */* Jacobian of parameter transformation */*
- 5 $\Omega_k.\partial_\eta \mathbf{r}'(\eta) \leftarrow \partial_\theta \mathbf{r}(\pi(\eta)) \partial_\eta \pi(\eta)$ */* Jacobian matrix */*
- 6 $\Omega_k.\mathbf{H}(\eta) \leftarrow (\partial_\eta \mathbf{r}'(\eta))^\top \partial_\eta \mathbf{r}'(\eta)$ */* approximate halved Hessian matrix */*
- 7 $\Omega_k.\text{cost}_\eta \leftarrow \mathbf{r}'(\eta)^\top \mathbf{r}'(\eta)$ */* current cost */*
- 8 $\Omega_k.\eta \leftarrow \eta$ */* current parameter estimate */*
- 9 $\Omega_k.\lambda \leftarrow \lambda$ */* damping parameter */*
- 10 $\Omega_k.\nu \leftarrow \nu$ */* damping parameter decrement */*
- 11 $\Omega_k.\gamma \leftarrow \gamma$ */* damping parameter increment */*
- 12 $\Omega_k.\eta_wasUpdated \leftarrow \text{false}$ */* indicates if η was modified or not */*
- 13 $\Omega_{k+1} \leftarrow \text{LEVENBERGMARQUARDTSTEP}(\Omega_k)$
- 14 **if** **HASCONVERGED**(Ω_{k+1}) **then**
- 15 **keep_going** \leftarrow **false**
- 16 $k \leftarrow k + 1$
- 17 **end while**
- 18 **return** $\Omega_k.\eta$

19 **Procedure** **HASCONVERGED**(Ω)

- 20 $\theta = \kappa(\Omega.\eta)$ */* convert from the η parametrisation to the θ parametrisation (see equation (4.1)) */*
- 21 $D = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}$ */* compute determinant of the ellipse (note that $\theta = [a, b, c, d, e, f]^\top$) */*
- 22 $\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{bmatrix}$
- 23 **if** $\|\Omega_{k+1}.\eta - \Omega_k.\eta\| < \text{tol}_\eta$ **and** $\Omega_{k+1}.\eta_wasUpdated$ **then**
- 24 **return true**
- 25 **else if** $|\Omega_{k+1}.\text{cost}_\eta - \Omega_k.\text{cost}_\eta| < \text{tol}_{\text{cost}}$ **and** $\Omega_{k+1}.\eta_wasUpdated$ **then**
- 26 **return true**
- 27 **else if** $\|\Omega_{k+1}.\Delta_\eta\| < \text{tol}_\Delta$ **and** $\Omega_{k+1}.\eta_wasUpdated$ **then**
- 28 **return true**
- 29 **else if** $\|\Omega_{k+1}.\partial_\eta \mathbf{r}'(\eta)\| < \text{tol}_{\|\partial_\eta \mathbf{r}'(\eta)\|}$ **then**
- 30 **return true**
- 31 **else if** $\log(\|\theta\|^2 / \theta^\top \mathbf{F} \theta) < \text{tol}_{\log(\|\theta\|^2 / \theta^\top \mathbf{F} \theta)}$ **or** $|D| < \text{tol}_D$ **then**
- 32 **return true**
- 33 **else**
- 34 **return false**

Algorithm 2: LEVENBERGMARQUARDTSTEP

Input: Ω
Output: Ω

/* compute two potential updates based on different weightings of the damping matrix */

- 1 $\mathbf{W}_\eta \leftarrow (\Omega.\partial_\eta \pi(\eta))^\top \Omega.\partial_\eta \pi(\eta)$
- 2 $\Delta_a \leftarrow [\Omega.\mathbf{H}(\eta) + \Omega.\lambda \mathbf{W}_\eta]^{-1} (\Omega.\partial_\eta \mathbf{r}(\eta))^\top \Omega.\mathbf{r}(\eta)$
- 3 $\eta_a \leftarrow \Omega.\eta - \Delta_a$
- 4 $\Delta_b \leftarrow [\Omega.\mathbf{H}(\eta) + (\Omega.\nu)^{-1} \Omega.\lambda \mathbf{W}_\eta]^{-1} (\Omega.\partial_\eta \mathbf{r}(\eta))^\top \Omega.\mathbf{r}(\eta)$
- 5 $\eta_b \leftarrow \Omega.\eta - \Delta_b$

/* compute new residuals and costs based on these updates */

- 6 $\text{cost}_{\eta_a} \leftarrow \mathbf{r}'(\eta_a)^\top \mathbf{r}'(\eta_a)$
- 7 $\text{cost}_{\eta_b} \leftarrow \mathbf{r}'(\eta_b)^\top \mathbf{r}'(\eta_b)$

/* determine appropriate damping and if possible select an update */

- 8 **if** $\text{cost}_{\eta_a} \geq \Omega.\text{cost}_\eta$ **and** $\text{cost}_{\eta_b} \geq \Omega.\text{cost}_\eta$ **then**
- 9 $\Omega.\eta_wasUpdated \leftarrow \text{false}$ /* neither potential update reduced the cost */
- 10 $\Omega.\eta \leftarrow \Omega.\eta$ /* no change in parameters */
- 11 $\Omega.\Delta_\eta \leftarrow \Omega.\Delta_\eta$ /* no change in step direction */
- 12 $\Omega.\text{cost}_\eta \leftarrow \Omega.\text{cost}_\eta$ /* no change in cost */
- 13 $\Omega.\lambda \leftarrow (\Omega.\lambda)(\Omega.\gamma)$ /* next iteration add more of the damping matrix */
- 14 **else if** $\text{cost}_{\eta_b} < \Omega.\text{cost}_\eta$ **then**
- 15 $\Omega.\eta_wasUpdated \leftarrow \text{true}$ /* update 'b' reduced the cost function */
- 16 $\Omega.\eta \leftarrow \eta_b$ /* choose update 'b' */
- 17 $\Omega.\Delta_\eta \leftarrow \Delta_b$ /* store the step direction */
- 18 $\Omega.\text{cost}_\eta \leftarrow \text{cost}_{\eta_b}$ /* store the current cost */
- 19 $\Omega.\lambda \leftarrow (\Omega.\lambda)(\Omega.\nu)^{-1}$ /* next iteration reduce the influence of the damping matrix */
- 20 **else**
- 21 $\Omega.\eta_wasUpdated \leftarrow \text{true}$ /* update 'a' reduced the cost function */
- 22 $\Omega.\eta \leftarrow \eta_a$ /* choose update 'a' */
- 23 $\Omega.\Delta_\eta \leftarrow \Delta_a$ /* store the step direction */
- 24 $\Omega.\text{cost}_\eta \leftarrow \text{cost}_{\eta_a}$ /* store the current cost */
- 25 $\Omega.\lambda \leftarrow \Omega.\lambda$ /* keep the same damping for the next iteration */

/* return a data structure containing all the updates */

26 **return** Ω

Algorithm 3: ALGEBRAICPARAMETERCOVARIANCE

Input: $\widehat{\theta}_{\text{AML}}, \{\mathbf{x}_1, \dots, \mathbf{x}_N\}, \{\mathbf{A}_{\mathbf{x}_1}, \dots, \mathbf{A}_{\mathbf{x}_N}\}$
Output: $\mathbf{A}_{\widehat{\theta}_{\text{AML}}}$

- 1 $\mathbf{P}_{\widehat{\theta}_{\text{AML}}}^\perp \leftarrow \mathbf{I}_6 - \|\widehat{\theta}_{\text{AML}}\|^{-2} \widehat{\theta}_{\text{AML}} \widehat{\theta}_{\text{AML}}^\top$ /* represents the projection along $\widehat{\theta}_{\text{AML}}$ onto the orthogonal complement of $\widehat{\theta}_{\text{AML}}$ */
- 2 $\mathbf{M}_{\widehat{\theta}_{\text{AML}}} \leftarrow \sum_{n=1}^N \frac{\Lambda_n}{\widehat{\theta}_{\text{AML}}^\top \mathbf{B}_n \widehat{\theta}_{\text{AML}}} (\mathbf{A}_n = \mathbf{u}(\mathbf{x}_n) \mathbf{u}(\mathbf{x}_n)^\top, \mathbf{B}_n = \partial_{\mathbf{x}} \mathbf{u}(\mathbf{x}_n) \mathbf{A}_{\mathbf{x}_n} \partial_{\mathbf{x}} \mathbf{u}(\mathbf{x}_n)^\top)$
- 3 $\mathbf{A}_{\widehat{\theta}_{\text{AML}}} \leftarrow \mathbf{P}_{\widehat{\theta}_{\text{AML}}}^\perp (\mathbf{M}_{\widehat{\theta}_{\text{AML}}})_5^\dagger \mathbf{P}_{\widehat{\theta}_{\text{AML}}}^\perp$ /* refer to Section 5.1 for an explanation of how to compute $(\mathbf{M}_{\widehat{\theta}_{\text{AML}}})_5^\dagger$ */
- 4 **return** $\mathbf{A}_{\widehat{\theta}_{\text{AML}}}$

Algorithm 4: GEOMETRICPARAMETERCOVARIANCE

Input: $\widehat{\theta}_{AML}, \{x_1, \dots, x_N\}$ and optionally $\{A_{x_1}, \dots, A_{x_N}\}$
Output: $A_{\widehat{\xi}_{AML}}$

/ if the data point covariance matrices $\{A_{x_1}, \dots, A_{x_N}\}$ are specified then skip steps 1 to 4 */*

- 1 $A_{x_n} \leftarrow I_2 \quad \forall n = 1, \dots, N$ */* assume that covariance in data points is proportional to the identity matrix */*
- 2 $\text{cost}_{\widehat{\theta}_{AML}} \leftarrow \sum_{n=1}^N \frac{\widehat{\theta}_{AML}^T A_n \widehat{\theta}_{AML}}{\widehat{\theta}_{AML}^T B_n \widehat{\theta}_{AML}} \quad (A_n = \mathbf{u}(x_n)\mathbf{u}(x_n)^T, B_n = \partial_x \mathbf{u}(x_n) A_{x_n} \partial_x \mathbf{u}(x_n)^T)$ */* AML cost associated with $\widehat{\theta}_{AML}$ */*
- 3 $\hat{\sigma}^2 \leftarrow (N - 5)^{-1} \text{cost}_{\widehat{\theta}_{AML}}$ */* estimate of noise level */*
- 4 $A_{x_n} \leftarrow \hat{\sigma}^2 A_{x_n} \quad \forall n = 1, \dots, N$ */* represents covariance of data points in original coordinate system */*

/ transform $\widehat{\theta}_{AML}$, data points x_n , and covariances A_{x_n} into a Hartley–normalised coordinate system for improved numerical stability */*

- 5 $\widehat{\theta}_{AML} = \mathbf{E}^{-1} \mathbf{P}_{(34)} \mathbf{D}_3^+ (\mathbf{T} \otimes \mathbf{T})^{-T} \mathbf{D}_3 \mathbf{P}_{(34)} \mathbf{E} \widehat{\theta}_{AML}$ */* refer to Section 7 for the definition of matrices $\mathbf{E}, \mathbf{P}_{(34)}, \mathbf{D}_3$, and \mathbf{T} */*
- 6 $\tilde{x}_n = \mathbf{T} x_n \quad \forall n = 1, \dots, N$
- 7 $\begin{bmatrix} A_{\tilde{x}_n} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \leftarrow \mathbf{T} \begin{bmatrix} A_{x_n} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \mathbf{T}^T \quad \forall n = 1, \dots, N$

/ estimate covariance for the algebraic parametrisation in the Hartley–normalised coordinate system */*

- 8 $A_{\widehat{\theta}_{AML}} \leftarrow \text{ALGEBRAICPARAMETERCOVARIANCE}(\widehat{\theta}_{AML}, \{\tilde{x}_1, \dots, \tilde{x}_N\}, \{A_{\tilde{x}_1}, \dots, A_{\tilde{x}_N}\})$
/ refer to eqs. (5.7) to (5.17) for the definition of $\partial_{\widehat{\theta}} \tilde{\xi}$, with $\widehat{\theta}$ substituted for θ and $\tilde{\xi}$ substituted for ξ */*
- 9 $A_{\widehat{\xi}_{AML}} \leftarrow [\partial_{\widehat{\theta}} \tilde{\xi}]_{\widehat{\theta}=\widehat{\theta}_{AML}} A_{\widehat{\theta}_{AML}} [(\partial_{\widehat{\theta}} \tilde{\xi})]_{\widehat{\theta}=\widehat{\theta}_{AML}}^T$
- 10 $\mathbf{S} \leftarrow \text{diag}(s, s, s, s, 1)$ */* refer to Section 7 for the definition of the scalar s */*
- 11 $A_{\widehat{\xi}_{AML}} \leftarrow \mathbf{S} A_{\widehat{\xi}_{AML}} \mathbf{S}^T$ */* transform geometric covariance matrix back to the original coordinate system */*
- 12 **return** $A_{\widehat{\xi}_{AML}}$

Our overall optimisation scheme is summarised in Algorithms 1 and 2. Algorithms 3 and 4 give standalone procedures for computing the covariances of the algebraic and geometric parameters of an ellipse estimate obtained by applying our scheme. It should be noted that an integral part of Algorithm 1 are conditions for operation termination. We shall elaborate on these conditions in Sect. 10.2.3 while discussing details of our implementation of Algorithm 1.

10 Performance Analysis

In order to characterise the accuracy and computational efficiency of our method, we conducted numerous simulations under diverse experimental conditions. We manipulated the eccentricity of ellipses, the number of data points sampled from ellipses, the level of noise that was applied to data points, as well as the length of ellipse segments from which data points were generated. Taking into account the length of ellipse segments allowed us to determine how an ellipse estimate is influenced by the proportion of the length of an ellipse segment to the length of the entire underlying ellipse. Based on experiments with changing this proportion, we learned to distinguish between ill-posed and well-posed ellipse fitting problems. We now consider an ellipse fitting problem as ill-posed when the length of a segment of an ellipse is less than half of the ellipse’s perimeter, and as well-posed when the length of a segment of an ellipse is greater than half of the ellipse’s perimeter.

In evaluating the quality of ellipse estimates, we distinguished between training data and testing data. Training data consists of noise perturbed points sampled from an ellipse possibly reduced to a small fragment. In contrast, testing data consists of noiseless data points uniformly sampled from an entire ellipse. Any particular ellipse in our experiments was estimated using training data, and the ellipse’s estimates were evaluated using corresponding testing data. By distinguishing between training data and testing data, we were explicitly interpreting ellipse fitting as a prediction problem. This means that we were not interested in finding the best ellipse that fits a given training data *per se*, but expected the estimated ellipse to be representative of the true ellipse in all regions, not just in the region from which the training data was sampled.

10.1 Procedure for Generating Synthetic Data and Evaluating Ellipse Estimates

We now summarise the procedure for generating training data and testing data, and explain in more detail how the quality of ellipse estimates was characterised.

10.1.1 Data Generation

The first step of the data generation procedure is to determine an ellipse in standard form $x^2/a^2 + y^2/b^2 = 1$ by choosing first a random value of a in the interval (101, 200) and next

a random value of b in the interval $(100, a)$. Such a choice ensures that the inequality $a > b$ holds and, consequently, that a is the length of the semi-major axis and b is the length of the semi-minor axis of the ellipse in question.

Once an initial ellipse has been selected, training and testing data-sets are constructed using the following steps:

1. Sample K points equidistantly along a segment of the ellipse with a specified length. The length of the segment is a fraction of the ellipse perimeter, and is measured clockwise starting from the positive y -axis. For example, the segment length can be half of the ellipse perimeter (see Fig. 1a). The K points will serve as a basis for the training data.
2. Sample N points equidistantly along the entire perimeter of the ellipse. These N points will serve as a basis for the testing data.

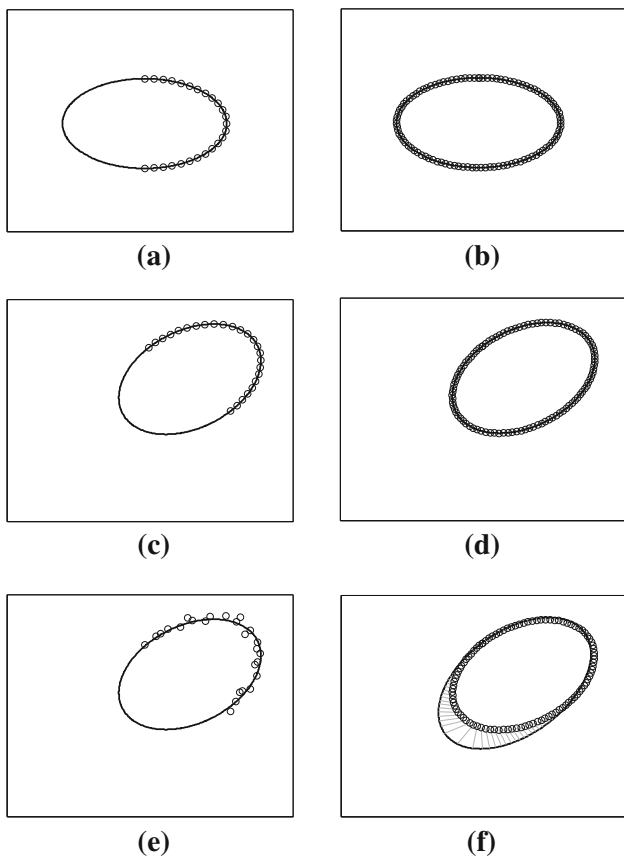


Fig. 1 Summary of the simulation procedure. (a) Training data points are sampled from a segment of an ellipse. In this example, the length of the segment is half of the ellipse perimeter. (b) Testing data points are sampled equidistantly from the entire ellipse. (c) Training data is randomly rotated and translated. (d) The same rotation and translation is applied to testing data. (e) Noise perturbed training data serves as input to ellipse estimation methods. (f) The quality of an estimated ellipse is characterised in terms of the root-mean-square orthogonal distance between the testing data set and the estimated ellipse. The orthogonal distance is depicted by straight lines joining testing data points and the corresponding closest points on the ellipse

3. Rotate and translate the ellipse together with the points generated in the first two steps by a random angle and random offset (see Fig. 1d).
4. Add zero-mean Gaussian noise with desired standard deviation to the rotated K points (see Fig. 1e). Take these noise-perturbed data points as the training data (serving as input to the ellipse estimation methods).
5. Take the rotated set of N points as the testing data.

10.1.2 Estimates Evaluation

After an ellipse has been estimated on the training data with any particular estimation method, the quality of the respective estimate is evaluated on the testing data using the *root-mean-square (RMS) orthogonal distance*

$$\sqrt{\frac{1}{2N} \sum_{n=1}^N d_n^2},$$

where d_n denotes the orthogonal distance between the n th data point and the ellipse constituting the estimate (see Fig. 1f). The RMS orthogonal distance measures the *geometric error* of the estimate with respect to the testing data points. The process of computing the orthogonal distances d_n is rather involved—detailed formulae can be found in [11] and [61].

10.2 Results

We compared our estimation technique with the orthogonal distance regression method and the direct ellipse fitting method, which represent the gold standard and the baseline technique for ellipse fitting, respectively. Both the orthogonal distance regression method and our proposed method were initialised with the result of the direct ellipse fitting technique. All estimation schemes operated on Hartley-normalised data points.

10.2.1 Synthetic Data

In the first set of simulations we held the noise level fixed at $\sigma = 5$ pixels, and varied the number of training data points and the length of ellipse segments from which the data points were sampled. For each combination of the number of data points and the length of an ellipse segment, we conducted 250 simulation trials and recorded the mean root-mean-square error. The mean-root-mean-square errors are displayed using two-dimensional contour plots in Fig. 2.

Two important conclusions can be drawn from the results of the first experiment:

1. When the length of an ellipse segment is less than half of the length of the entire ellipse, the direct ellipse

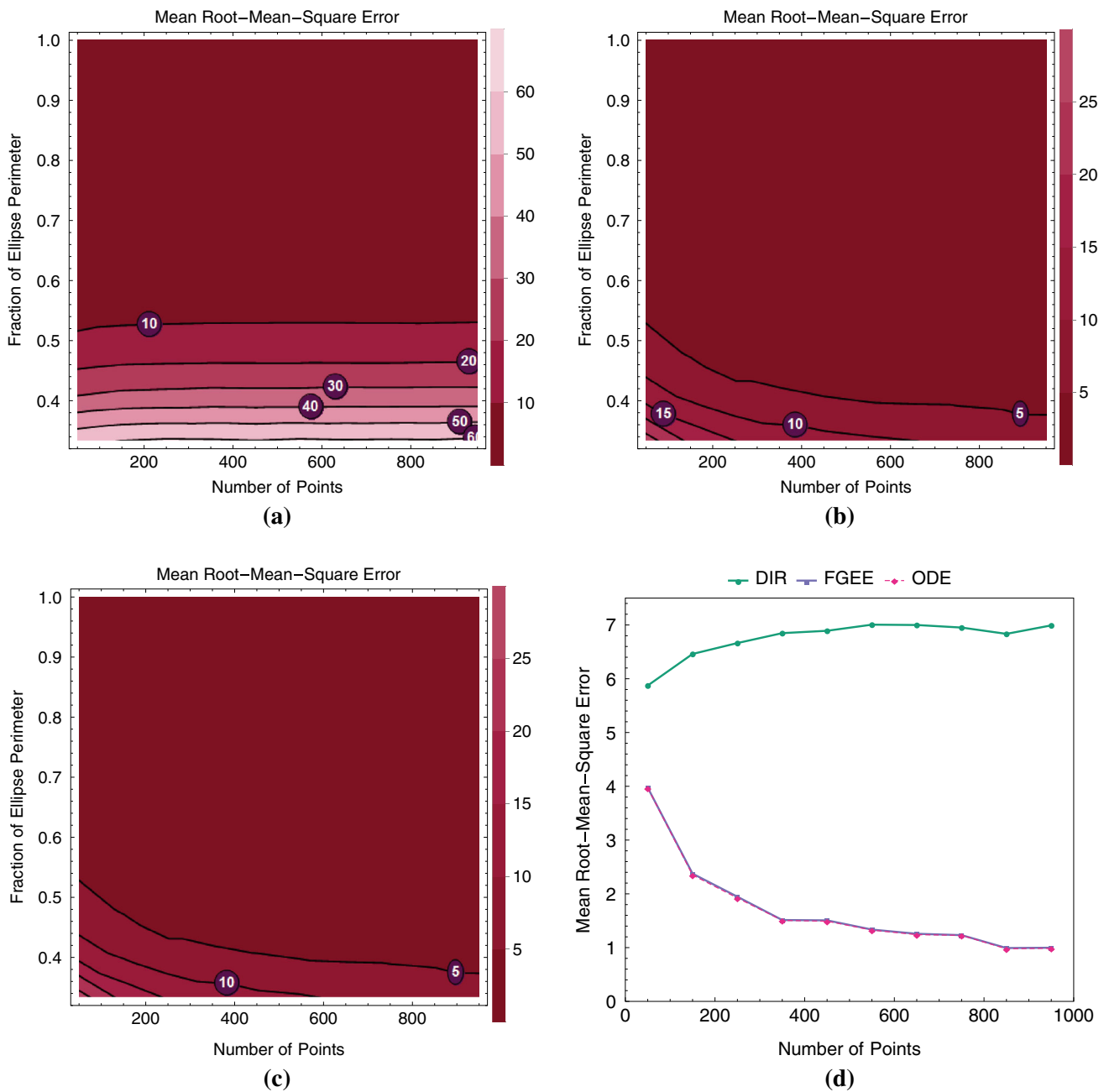


Fig. 2 Comparison of mean root-mean-square orthogonal distance error for a fixed noise level of $\sigma = 5$ pixels, as both the number of data points and the portion from which the data points are sampled are varied. The colour intensity of the plot is correlated with the mean root-mean-square orthogonal distance error—the lighter the colour, the larger the error. Contour lines represent level sets with constant error. When the fraction of the ellipse from which data points are sampled is less than half of the ellipse perimeter, the error of the direct ellipse estimation (DIR) method fails to improve as the number of data points

is increased. In contrast, both our fast guaranteed ellipse estimation method (FGEE) and the gold standard orthogonal distance estimation (ODE) method exhibit a consistent reduction in error as the number of data points is increased. (d) may be interpreted as comparing a sample of values on the “number of points” axis in (a–c), while holding the corresponding “fraction of ellipse perimeter” axis fixed at 0.55. (a) Direct ellipse estimation. (b) Fast guaranteed ellipse estimation. (c) Orthogonal distance estimation. (d) Comparing all three methods (Color figure online)

estimate does not improve as the number of data points is increased.

2. The Sampson distance based ellipse fitting method and the orthogonal distance regression method yield almost indistinguishable results.

In the second set of simulations we utilised only 25 data points, but still varied the noise level and the length of ellipse segments from which data points were sampled. The results of the second simulation are summarised using two-dimensional contour plots in Fig. 3. The second experi-

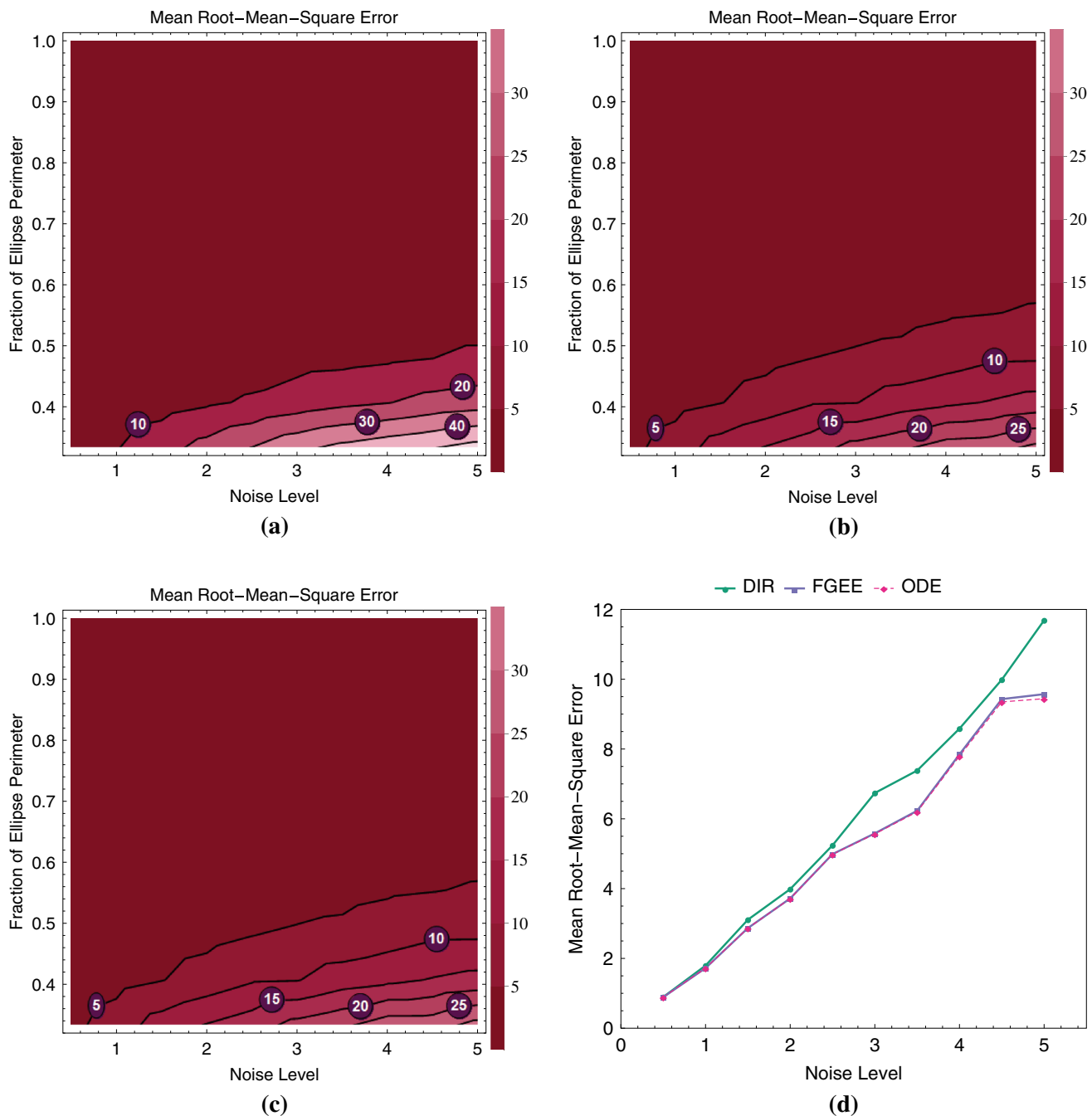


Fig. 3 Comparison of mean root-mean-square orthogonal distance error for 25 data points, as both the portion from which the data points are sampled and the standard deviation of the noise level (measured in pixels) are varied. The colour intensity of the plot is correlated with the mean root-mean-square orthogonal distance error—the lighter the colour, the larger the error. Contour lines represent level-sets with constant error. When the fraction of the ellipse from which data points are sampled is less than half of the ellipse perimeter, the error of the direct

ellipse estimation method is considerably higher than that of the other methods. In contrast, our fast guaranteed ellipse estimation method and the gold standard orthogonal distance estimation method exhibit similar error levels. (d) may be interpreted as comparing a sample of values on the “number of points” axis in (a–c), while holding the corresponding “fraction of ellipse perimeter” axis fixed at 0.45. (a) Direct ellipse estimation. (b) Fast guaranteed ellipse estimation. (c) Orthogonal distance estimation. (d) Comparing all three methods (Color figure online)

ment confirms that our Sampson distance based ellipse fitting method produces similar results to those of the gold standard orthogonal distance regression method for a variety of noise

levels. The second experiment also shows that our Sampson distance based ellipse fitting imitates the orthogonal distance regression even when both the noise level and the ellipse

segment length are varied. The direct ellipse fitting method does not share this desirable property, and instead yields inferior results.

10.2.2 Stability and Efficiency

For every experiment, we verified that our algorithm was indeed producing an ellipse fit by confirming that the discriminant of the estimated parameters was less than zero. We also monitored the average running time of a MATLAB implementation of our algorithm,³ while we varied the number of data points as well as the fraction of the ellipse perimeter from which data points were sampled. A summary of our findings together with a comparison against the running time of the orthogonal distance regression method is presented using two-dimensional density plots in Fig. 4.

Based on the outcome of the experiment, we make the following two observations:

1. The running time of our fast guaranteed ellipse fitting algorithm increases gradually and smoothly as the number of data points is increased, and as the problem becomes more ill-posed.
2. The running time of the orthogonal distance regression method follows a similar trend, but does not increase gradually nor smoothly.

Our experiments indicate that the running time of the orthogonal distance regression method is much more unpredictable than the running time of our technique.

In Sect. 4.2 we mentioned that utilising the LM method in standard form to optimise our cost function may occasionally take many iterations to converge, and proposed a modification to the LM scheme. A comparison of the running time of our fast guaranteed ellipse fitting algorithm against the basic guaranteed ellipse fitting algorithm that uses the standard LM method is presented in Fig. 5. The results show that for ill-posed ellipse fitting problems, our fast guaranteed ellipse fitting implementation is twice as fast as the basic guaranteed ellipse fitting algorithm.

10.2.3 Attributes of the Estimate

While our ellipse specific parametrisation explicitly excludes hyperbola fits, it does permit degenerate ellipses or ellipses that approach parabolas in the limit. As discussed in Sect. 2, degenerate ellipses are those for which the corresponding determinant D is zero. In turn, ellipses close to parabolas are those for which the corresponding discriminant Δ is close to zero. To avoid producing degenerate ellipses, our algorithm terminates when the absolute value of the determinant of an

ellipse falls below a user-specified threshold. To prevent the solution from coming too close to the parabolas, our algorithm stops when the discriminant approaches zero.

In our experiments the threshold for the determinant was set to 10^{-5} , and the discriminant was ensured to be large enough in modulus by forcing the algorithm to terminate when $\log(\|\theta\|^2/\theta^T \mathbf{F} \theta)$ rose above a threshold of 15.5.

We label close to degenerate ellipses and ellipses that approximate parabolas as *depreciated ellipses*. In contrast to *bona fide* ellipses, depreciated ellipses are characterised by the property that only a subset of geometric ellipse parameters are estimated with reasonable certainty. Although not as informative as genuine ellipses, depreciated ellipses still provide useful knowledge. For example, an ellipse that approximates a parabola will have tremendous uncertainty associated with its semi-major axis, but the semi-minor axis and orientation of the axes may still be estimated with great precision, and this will be reflected in the covariance matrix.

In order to determine how frequently depreciated ellipses occur in typical scenarios, we conducted numerous simulations which are summarised in Table 1.

The results of our simulations are based on 10,000 trials. For each trial an ellipse was generated by randomly selecting a length for the semi-major and semi-minor axes, while keeping the axes aligned with the Cartesian axes. Points were then sampled from the upper half, right half, and upper right quarter of the ellipse. Our experiments revealed that with small noise levels and with at least ten data points, depreciated ellipses occur on rare occasions. The most challenging situation arises when data points are sampled from the upper right quarter of an ellipse and the noise level is large. This constitutes a very ill-posed problem—neither orthogonal distance regression nor AML estimation are capable of producing an ellipse that is close to a true ellipse without recourse to some kind of regularisation. We believe that ill-posed estimation problems are best solved within a Bayesian setting, where regularisation is achieved through a suitable choice of a prior distribution over the parameter space. The design of an appropriate prior over the space of ellipses that might serve to guide the estimate away from degenerate ellipses and parabolas is a challenging problem which we intend to pursue in future work.

10.2.4 Real Data

To validate the conclusions drawn from the synthetic experiments, we also compared the ellipse fitting methods on real images. In Fig. 6, we utilised two images of a Martian moon eclipse captured by the Opportunity rover, and in Fig. 7, we used images of two of Saturn's moons captured by the Cassini spacecraft. Our experiments confirm that the estimates produced by our fast guaranteed ellipse fitting method agree with the estimates obtained via the orthogonal distance regression.

³ <http://sites.google.com/site/szpakz/>.

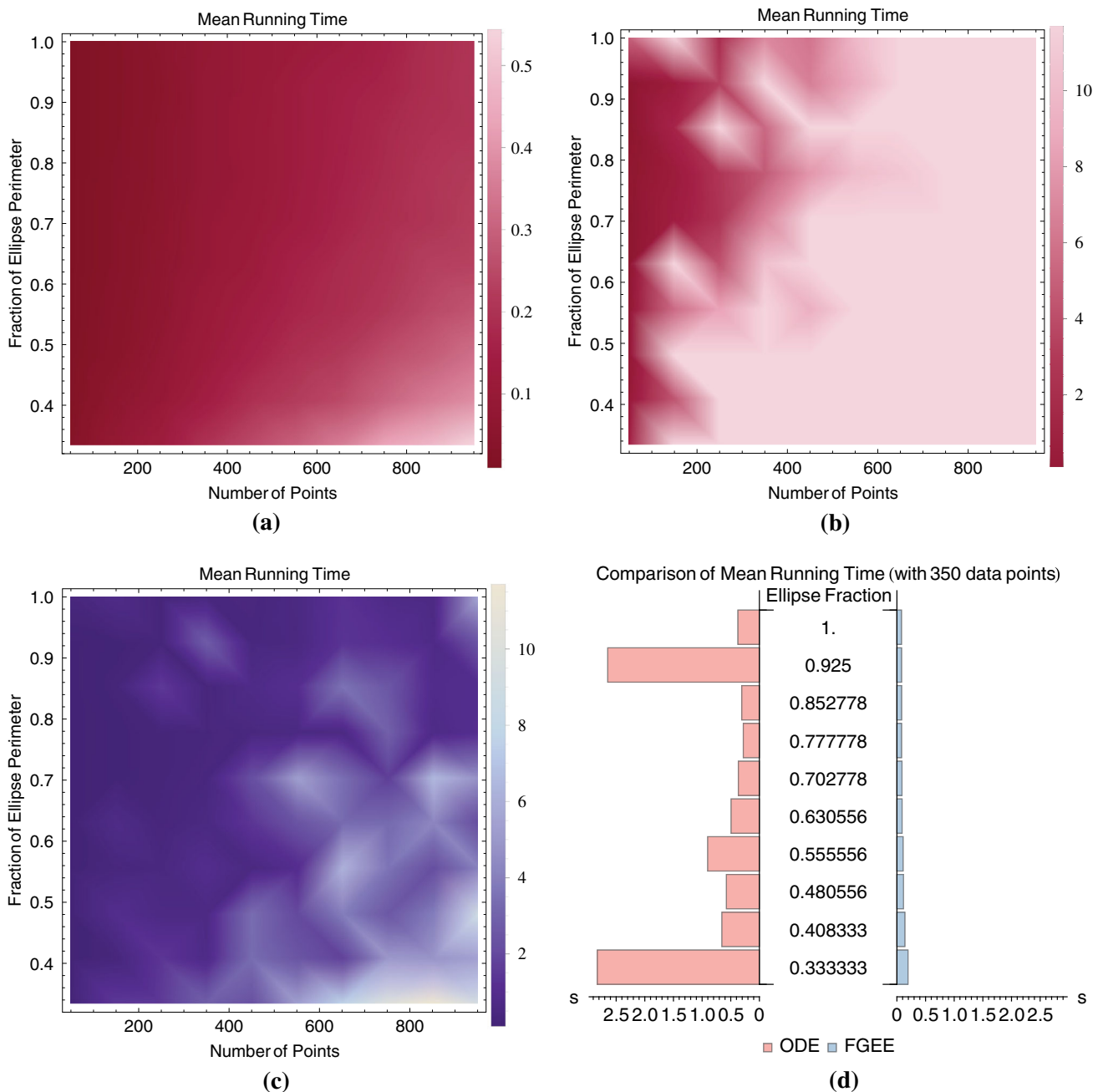


Fig. 4 Comparison of mean running time for fixed noise level of $\sigma = 5$ pixels, as both the portion from which the data points are sampled and the number of data points are varied. The running time is measured in seconds. The colour intensity of the plot is correlated with the average running time—the lighter the colour, the greater the running time. In **(a)** and **(b)** the same colour scaling function was used to map running time to colour, thereby ensuring that the two plots are directly comparable. According to **(a)**, the average running time of the fast guaranteed ellipse estimation method increases smoothly as the number of points is increased and as the fraction of the ellipse perimeter from which data points are sampled is decreased. Whilst the running time of the orthogonal distance estimation method in **(b)** follows a similar trend,

the trade-off is not as smooth, indicating that the running time of the orthogonal distance estimation method for a sample of data points is much more unpredictable. In **(c)** the irregularity of the running time of the orthogonal distance estimation method is made more prominent by a different choice of the colour scaling function. **(d)** may be interpreted as comparing a sample of values on the “fraction of ellipse perimeter” axis in **(a)** and **(b)**, while holding the corresponding “number of points” axis fixed at 350. **(a)** Fast guaranteed ellipse estimation. **(b)** Orthogonal distance estimation. **(c)** Orthogonal distance estimation. **(d)** Fast guaranteed ellipse estimation vs orthogonal distance estimation (Color figure online)

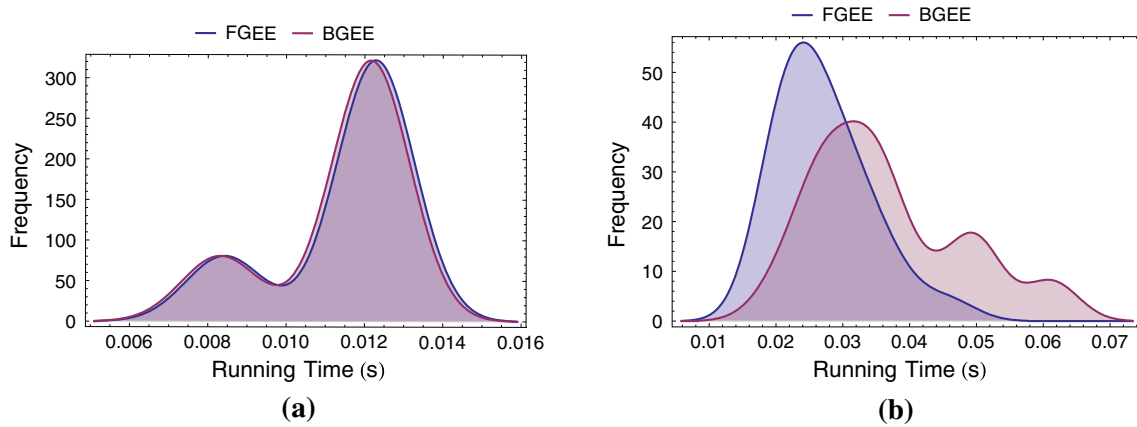


Fig. 5 Comparison of smoothed histograms of the number of seconds that elapsed before our algorithm that uses a modified damping matrix in the LM optimisation scheme converged for (a) well-posed and (b) ill-posed problems, and of similar histograms for the basic guaranteed ellipse fitting method that uses an identity damping matrix. The results indicate that the use of the modified damping matrix in the LM scheme

can considerably decrease the running time for ill-posed problems. On average, the modified damping matrix reduces the running time by half for ill-posed problems. For well-posed problems, there is no notable difference in running time as both algorithms converge within a handful of iterations. (a) 25 data points sampled from the entire ellipse. (b) 25 data points sampled from one third of the ellipse

Table 1 Prevalence of ellipses, depreciated ellipses, and hyperbolas when minimising AML using the conic parametrisation (θ) or the ellipse-specific parametrisation (η)

| Parametrisation | Conic type | Upper half of ellipse | | Right half of ellipse | | Quarter of ellipse | |
|--|---------------------|-----------------------|------------------|-----------------------|------------------|--------------------|------------------|
| | | $\sigma = 1$ (%) | $\sigma = 5$ (%) | $\sigma = 1$ (%) | $\sigma = 5$ (%) | $\sigma = 1$ (%) | $\sigma = 5$ (%) |
| <i>(a) Results based on 10,000 simulations with 10 data points</i> | | | | | | | |
| η | Ellipse | 100.0 | 100.0 | 100.0 | 100.0 | 96.61 | 61.43 |
| | Depreciated Ellipse | 0.0 | 0.0 | 0.0 | 0.0 | 3.39 | 38.57 |
| θ | Ellipse | 100.0 | 99.9 | 100.0 | 100.0 | 95.92 | 61.89 |
| | Hyperbola | 0.0 | 0.1 | 0.0 | 0.0 | 4.08 | 38.11 |
| <i>(b) Results based on 10,000 simulations with 50 data points</i> | | | | | | | |
| η | Ellipse | 100.0 | 100.0 | 100.0 | 100.0 | 99.86 | 74.44 |
| | Depreciated Ellipse | 0.0 | 0.0 | 0.0 | 0.0 | 0.14 | 25.56 |
| θ | Ellipse | 100.0 | 100.0 | 100.0 | 100.0 | 99.83 | 72.23 |
| | Hyperbola | 0.0 | 0.0 | 0.0 | 0.0 | 0.17 | 27.77 |

Results are based on 10,000 simulations with 10 data points and 10,000 simulations with 50 data points

10.2.5 Accuracy of Geometric Parameter Covariance Estimation

We performed additional simulations to test the validity of our geometric parameter covariance formulae. Our validation was based on comparing the covariance matrix given by the propagation formula (5.6) to a covariance matrix resulting from a Monte Carlo simulation. In particular, we sampled 250 points equidistantly between 0° and 225° on the ellipse parametrised by $\xi = [100, 50, 250, 250, 0.7854]^T$. We then added zero-mean Gaussian noise at a pre-set noise level to the data points, and produced 10,000 simulation trials. The Monte Carlo covariance matrix was computed with the aid of the formula

$$A_{\hat{\xi}_{\text{MONTE}}} = \frac{1}{9995} \sum_{n=1}^{10000} (\hat{\xi}_{\text{AML},n} - \bar{\xi}_{\text{AML}}) (\hat{\xi}_{\text{AML},n} - \bar{\xi}_{\text{AML}})^T,$$

where $\hat{\xi}_{\text{AML},n}$ is the geometric parameter estimate corresponding to the n th trial and $\bar{\xi}_{\text{AML}}$ represents the corresponding mean geometric parameter vector. We then compared the relative error between our propagated covariance matrix and the Monte Carlo covariance matrix for each simulation trial. The relative error is defined as the absolute error between our propagated covariance matrix and the Monte Carlo covariance matrix, divided by the Frobenius norm of the Monte Carlo covariance matrix. The relative error is sensitive to very small differences. We also computed the angu-

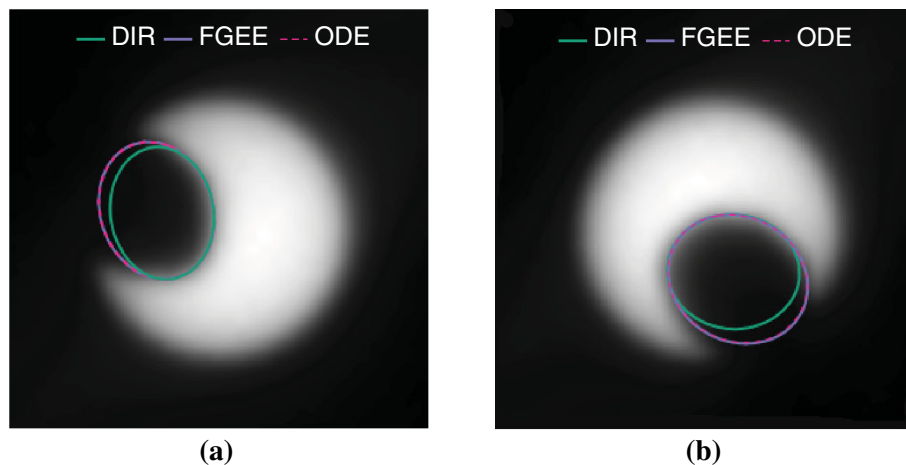


Fig. 6 Comparison of ellipse fitting methods on two images of a Martian moon eclipse. The two images capture the Martian moon Phobos as it occludes the sun. A Canny edge detector was applied to the two images, and the maximum edge response corresponding to the common border of the moon and the sun was used as input to the ellipse fitting

methods. In both cases the smallest ellipse corresponds to the direct ellipse fit, whilst our fast guaranteed ellipse estimation method and the orthogonal distance estimation method yield indistinguishable results (the ellipse contours overlap). Image Credit: NASA/JPL/Cornell. (a) Martian moon eclipse stage 1. (b) Martian moon eclipse stage 2

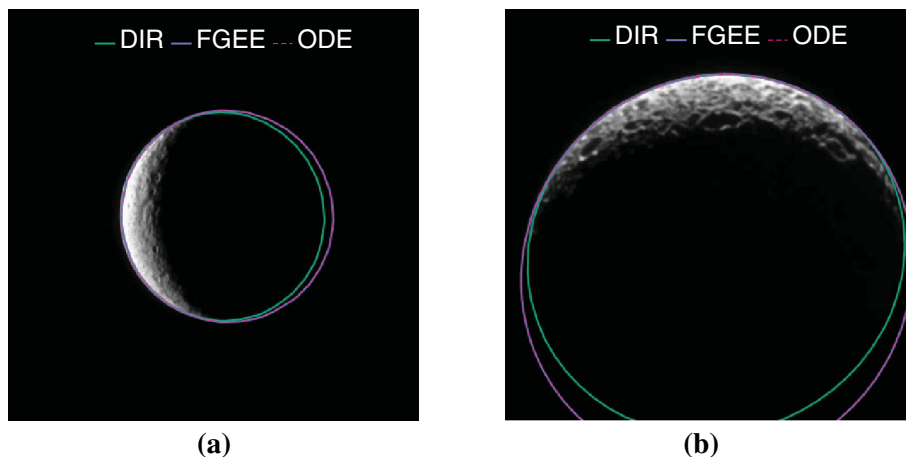


Fig. 7 Comparison of ellipse fitting methods on Saturn's crescent moons. A Canny edge detector was applied to the two images, and the maximum edge response corresponding to the outer border of the crescents was used as input to the ellipse fitting methods. In both cases the smallest ellipse corresponds to the direct ellipse fit, whilst our fast

guaranteed ellipse estimation method and the orthogonal distance estimation method yield indistinguishable results (the ellipse contours overlap) Image Credit: NASA/JPL/Space Science Institute. (a) Rhea. (b) Mimas

lar error defined as the angle between vectorised and unit-normalised variants of the propagated covariance and Monte Carlo covariance matrices. The angular error is invariant to differences in scale between the two covariance matrices. The median relative and median angular errors are presented in Table 2 for various noise levels. The results show that our covariance matrix estimates are very accurate, achieving a relative error of less than 15%, provided that the noise level is less than $\sigma = 3$ pixels.

To further illustrate the practical insight that can be gleaned from the covariance matrices, we plot a 95% confidence region and report the estimated geometric param-

eters and corresponding parameter standard errors on four different data-sets (see Fig. 8). In accordance with theoretical expectations, the true ellipses happened to fall inside the confidence regions and the confidence regions became narrower as the number of data points was increased. While the results presented in Fig. 8 are representative of the kinds of confidence regions one would typically observe in many practical scenarios, they may not be representative of confidence regions arising when data points are sampled from a very short fragment of an ellipse (an ill-posed problem)—in that case the assumptions underlying the generation of confidence regions may not be

Table 2 Median errors between our propagated geometric parameter covariance matrix and a Monte Carlo estimate of the geometric parameter covariance matrix

| σ (pixels) | Relative error (%) | Angular error (°) |
|-------------------|--------------------|-------------------|
| 1 | 6.8232 | 0.8348 |
| 2 | 9.3241 | 1.4280 |
| 3 | 15.3709 | 2.6416 |
| 4 | 21.7922 | 3.5146 |
| 5 | 34.0976 | 5.4763 |

Results are based on 10,000 trials. The relative error is sensitive to very small differences between the propagated covariance estimate and the Monte Carlo covariance estimate. The angular error is invariant to differences in scale between the two covariance matrices. The results show that our covariance matrix estimates are very accurate, achieving a relative error of less than 15% provided that the noise level is less than $\sigma = 3$ pixels

satisfied and the confidence regions may thus be too narrow.

11 Discussion

Our experiments show that the Sampson distance is an excellent approximation to the orthogonal distance for the purpose of fitting an ellipse to data. Our results are in agreement with the findings of Kanatani and Rangarajan who report that “[it] has also been observed that the solution that minimizes the Sampson error agrees with the ML [orthogonal distance] solution up to several significant digits.” [30, p. 2204].

The utility of the Sampson distance was also acknowledged by Rosin. This author conducted a comprehensive comparison of various ellipse fitting cost functions and concluded that “...for small amounts of noise EOF₂ [the Sampson distance] is the overall best approximation to use ...” [49, p. 501].

Some researchers, however, including Chernov and Ma [10, p. 299], have cautioned against replacing the orthogonal distance estimate with the Sampson distance. Their concerns are based on a list of disadvantages associated with algebraic fitting methods,⁴ which was compiled by Anh [1]. The list includes the following items:

1. Error definition does not comply with the measurement guidelines.
2. Conversion of the algebraic parameters to the physical parameters (shape, size, position, and rotation parameters) is highly difficult.
3. Fitting errors are weighted.

⁴ The Sampson distance based estimation technique is sometimes classified as a geometric fitting method and sometimes as an algebraic fitting method. Chernov and Ma as well as Anh regard the Sampson distance optimisation as an algebraic fitting method.

4. The estimated model parameters are biased.
5. It is very difficult to test the reliability of the estimated model parameters (particularly in terms of physical parameters).
6. The fitting procedure sometimes ends with an unintended model feature (e.g. a hyperbola instead of an ellipse).
7. The model parameters are not invariant to coordinate transformation (e.g. a parallel shift of the set of given points causes changes not only in position, but also in the form and rotation of the estimated model feature).

In the context of ellipse fitting, the disadvantages are not so severe. We offer the following rejoinder to some of the above criticisms (the numbers in brackets will refer to the original item numbers):

- [2]. The formulae for the conversion of algebraic ellipse parameters to physical ellipse parameters are straightforward, albeit tedious, to derive—see Sect. 5.2.
- [3]. The fact that fitting errors are weighted is not necessarily a disadvantage. It is through the introduction of gradient-cum-covariance weights into the algebraic distance that the Sampson distance is able to produce accurate estimates.
- [4]. Bias in the estimated model parameters is not limited to algebraic fitting methods. The orthogonal distance estimate is also biased even in the case of ellipse fitting [16, 38–41].
- [5]. Our fitting procedure includes a reliability measure for both algebraic and geometric ellipse parameters.
- [6]. Our fitting procedure always produces an ellipse estimate.
- [7]. If data points and data point covariance matrices are both jointly and appropriately modified in accordance with a change of coordinate system, then the Sampson distance is theoretically unchanged. However, certain coordinate systems (e.g. scaling all data points to lie within a unit box [14]) present favourable numerical advantages.

Our response above reduces the list of disadvantages to item (1). Now, passing to this remaining item, given that the Sampson distance yields accurate estimates, does it really matter that the error definition does not comply with measurement guidelines? In our view, it does not, but we concede that the answer may be domain dependent.

An astute reader would have probably noticed that the Sampson distance was devised more than forty years ago, and now may wonder whether the accuracy of the Sampson distance based estimation method has been surpassed by more recent techniques. To answer this question, a comprehensive evaluation of several new schemes was recently conducted [55]. These schemes are not based on optimisation of any particular cost function but rather exploit the idea

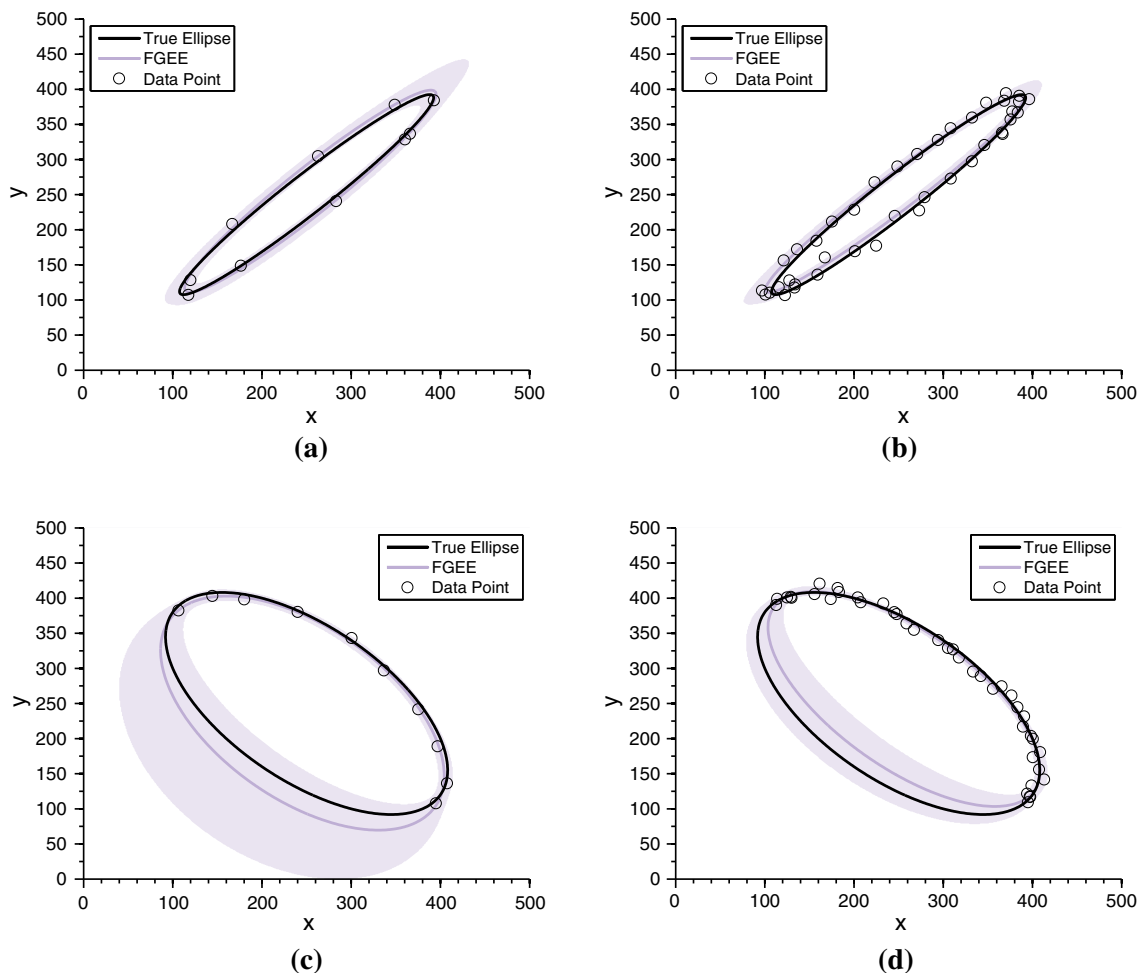


Fig. 8 Fast guaranteed ellipse estimation with corresponding 95 % confidence band (shaded region). **(a)** The estimated ellipse is centred at $(254.218 \pm 3.176, 254.866 \pm 3.580)$, with semi-major axis length of 199.347 ± 4.965 pixels, semi-minor axis length of 29.447 ± 1.683 pixels, and orientation of $45.612 \pm 0.549^\circ$. **(b)** The estimated ellipse is centred at $(245.330 \pm 2.612, 250.868 \pm 2.403)$, with semi-major axis length of 200.105 ± 3.613 pixels, semi-minor axis length of 23.596 ± 1.327 pixels, and orientation of $44.035 \pm 0.419^\circ$. **(c)** The estimated ellipse is centred at $(245.316 \pm 5.478, 236.282 \pm 7.851)$, with semi-major axis length of 201.939 ± 3.629 pixels, semi-minor axis length of 110.769 ± 9.859 pixels, and orientation of $132.503 \pm 1.772^\circ$. **(d)** The estimated ellipse is centred at $(256.101 \pm 3.549, 256.386 \pm 3.490)$, with semi-major axis length of 197.207 ± 1.719 pixels, semi-minor axis length of 88.211 ± 5.447 pixels, and orientation of $134.868 \pm 0.782^\circ$. **(a)** 10 data points sampled from an ellipse centred at $(250, 250)$, with semi-

major axis length of 200 pixels, semi-minor axis length of 25 pixels, and orientation of 45° . The data points were perturbed with zero-mean homogeneous Gaussian noise of $\sigma = 5$ pixels. **(b)** 40 data points sampled from an ellipse centred at $(250, 250)$, with semi-major axis length of 200 pixels, semi-minor axis length of 25 pixels, and orientation of 45° . The data points were perturbed with zero-mean homogeneous Gaussian noise of $\sigma = 5$ pixels. **(c)** 10 data points sampled from an ellipse centred at $(250, 250)$, with semi-major axis length of 200 pixels, semi-minor axis length of 100 pixels, and orientation of 135° . The data points were perturbed with zero-mean homogeneous Gaussian noise of $\sigma = 5$ pixels. **(d)** 40 data points sampled from an ellipse centred at $(250, 250)$, with semi-major axis length of 200 pixels, semi-minor axis length of 100 pixels, and orientation of 135° . The data points were perturbed with zero-mean homogeneous Gaussian noise of $\sigma = 5$ pixels

of unbiased existing estimators. The results of the comparison showed that whilst in some instances the newer techniques may lead to more accurate estimates, the improvements they yield are very small and can usually only be measured using several decimal places. Hence, in many situations, the improvement in accuracy that may be gleaned by using one of the more recently proposed techniques may not be practically useful.

12 Conclusion and Future Work

We have presented a straightforward and efficient algorithm for fitting an ellipse to data. The method exploits a parametrisation of the set of all ellipses to implicitly enforce the ellipse constraint. Computational efficiency is accomplished with a custom variant of the Levenberg–Marquardt algorithm. The

technique yields estimates that are almost indistinguishable from the estimates produced by the gold-standard orthogonal distance regression method, even at moderate noise levels. The proposed method is easy to implement, thanks to the reliance on the Sampson distance. Additionally, we have provided a means for plotting confidence regions and a measure of uncertainty for both algebraic and geometric parameters of an ellipse estimate. This measure of uncertainty may prove useful in various industrial applications and may also help in low-level vision tasks where deciding whether a group of pixels belong to a line, circle, or ellipse is often a prerequisite for higher-level image analysis. In future work we plan to investigate robust variants of our cost function to neutralise the impact of outliers on the estimate, and plan to explore suitable priors over the space of ellipses to improve the quality of the estimate for ill-posed problems. We also intend to characterise the empirical and theoretical accuracy of our measure of uncertainty more comprehensively by extending the scope of the Monte Carlo simulations and experiments.

Acknowledgments This work was partially supported by the Australian Research Council.

Appendices

Appendix 1: Proof of Equation (4.14)

In this appendix, we establish Eq. (4.14). The proof will *not* rely on general identities for pseudo-inverse matrices but rather will involve the specifics of $\mathbf{r}(\boldsymbol{\theta})$ and $\boldsymbol{\pi}(\boldsymbol{\eta})$.

Differentiating $\mathbf{r}(t\boldsymbol{\theta}) = \mathbf{r}(\boldsymbol{\theta})$ with respect to t and evaluating at $t = 1$, we find that $\partial_{\boldsymbol{\theta}}\mathbf{r}(\boldsymbol{\theta})\boldsymbol{\theta} = \mathbf{0}$. Hence, in particular, $\partial_{\boldsymbol{\theta}}\mathbf{r}(\boldsymbol{\pi}(\boldsymbol{\eta}))\boldsymbol{\pi}(\boldsymbol{\eta}) = \mathbf{0}$. Consequently, recalling (4.4), we have

$$\begin{aligned} \partial_{\boldsymbol{\theta}}\mathbf{r}(\boldsymbol{\pi}(\boldsymbol{\eta}))\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp} &= \partial_{\boldsymbol{\theta}}\mathbf{r}(\boldsymbol{\pi}(\boldsymbol{\eta})) \\ &\quad - \|\boldsymbol{\pi}(\boldsymbol{\eta})\|^{-2}\partial_{\boldsymbol{\theta}}\mathbf{r}(\boldsymbol{\pi}(\boldsymbol{\eta}))\boldsymbol{\pi}(\boldsymbol{\eta})\boldsymbol{\pi}(\boldsymbol{\eta})^{\top} \\ &= \partial_{\boldsymbol{\theta}}\mathbf{r}(\boldsymbol{\pi}(\boldsymbol{\eta})). \end{aligned}$$

We summarise this simply as

$$\partial_{\boldsymbol{\theta}}\mathbf{r}\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp} = \partial_{\boldsymbol{\theta}}\mathbf{r}$$

in line with our earlier convention that $\partial_{\boldsymbol{\theta}}\mathbf{r}$ be evaluated at $\boldsymbol{\pi}(\boldsymbol{\eta})$. As an immediate consequence, we obtain

$$((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp} = (\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp}.$$

This together with the observation that $(\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp}$ is symmetric (being the sum of the symmetric matrices

$(\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r}$ and $\lambda\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp}$) yields

$$\begin{aligned} ((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp} &= (((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp})^{\top} \\ &= (\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp})^{\top}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)^{\top} \\ &= \mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6). \end{aligned} \tag{13.1}$$

Now note that if \mathbf{A} and \mathbf{B} are square matrices of the same sizes, \mathbf{A} is invertible, and $\mathbf{AB} = \mathbf{BA}$, then $\mathbf{A}^{-1}\mathbf{B} = \mathbf{BA}^{-1}$, as is easily seen by pre- and post-multiplying the both sides of $\mathbf{AB} = \mathbf{BA}$ by \mathbf{A}^{-1} . This in conjunction with (13.1) implies

$$((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)^{-1}\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp} = \mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)^{-1}. \tag{13.2}$$

Differentiating the identity $\|\boldsymbol{\pi}(\boldsymbol{\eta})\|^2 = \boldsymbol{\pi}(\boldsymbol{\eta})^{\top}\boldsymbol{\pi}(\boldsymbol{\eta}) = 1$ with respect to $\boldsymbol{\eta}$, we get

$$(\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top}\boldsymbol{\pi}(\boldsymbol{\eta}) = 0. \tag{13.3}$$

Hence

$$\begin{aligned} (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top}\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp} &= (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top} - \|\boldsymbol{\pi}(\boldsymbol{\eta})\|^{-2}(\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top}\boldsymbol{\pi}(\boldsymbol{\eta})\boldsymbol{\pi}(\boldsymbol{\eta})^{\top} \\ &= (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top}. \end{aligned}$$

Pre-multiplying both sides of this equality by $((\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top}\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{-1}$ and invoking (4.12a), we obtain

$$(\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\perp}\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp} = (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\perp}. \tag{13.4}$$

Now recall that for any matrix \mathbf{A} , the matrix $\mathbf{A}^{\perp}\mathbf{A}$ represents the orthogonal projection onto the range (column space) of \mathbf{A}^{\top} , or equivalently, the orthogonal projection onto the orthogonal complement of the null space of \mathbf{A} . Since, on account of (13.3), the null space of $(\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top}$ is spanned by $\boldsymbol{\pi}(\boldsymbol{\eta})$, it follows that

$$((\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top})^{\perp}(\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top} = \mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp}. \tag{13.5}$$

We now have all ingredients needed to establish (4.14). We calculate as follows:

$$\begin{aligned} (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\perp}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)^{-1}((\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\top})^{\perp} &= (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\perp}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)\partial_{\boldsymbol{\eta}}\boldsymbol{\pi} \\ &= (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\perp}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)^{-1}\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)\partial_{\boldsymbol{\eta}}\boldsymbol{\pi} \\ &= (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\perp}\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)^{-1}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)\partial_{\boldsymbol{\eta}}\boldsymbol{\pi} \\ &= (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\perp}\mathbf{P}_{\boldsymbol{\pi}(\boldsymbol{\eta})}^{\perp}\partial_{\boldsymbol{\eta}}\boldsymbol{\pi} \\ &= (\partial_{\boldsymbol{\eta}}\boldsymbol{\pi})^{\perp}\partial_{\boldsymbol{\eta}}\boldsymbol{\pi} \\ &= \mathbf{I}_5. \end{aligned}$$

In the above the second line comes from the first by (13.5); the third line comes from the second by (13.2); the fourth line comes from the third by the tautological identity $((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6)^{-1}((\partial_{\boldsymbol{\theta}}\mathbf{r})^{\top}\partial_{\boldsymbol{\theta}}\mathbf{r} + \lambda\mathbf{I}_6) = \mathbf{I}_6$; the fifth line comes from the fourth by (13.4); and the sixth line comes

from the fifth by (4.13a). The end result of our calculation is what is exactly needed to establish (4.14).

Appendix 2: Proof of Equation (5.2)

In this appendix, we establish the covariance formula (5.2). The derivation is based on two ingredients: an equation characterising $\hat{\theta}_{AML}$ and a covariance propagation formula. The first ingredient, embodied in Eq. (13.7) below, comes from the optimality condition that $\hat{\theta}_{AML}$ satisfies as the minimiser of J_{AML} :

$$[\partial_{\theta} J_{AML}(\theta; \mathbf{x}_1, \dots, \mathbf{x}_N)]_{\theta=\hat{\theta}_{AML}} = \mathbf{0}^T. \tag{13.6}$$

Direct computation shows that

$$[\partial_{\theta} J_{AML}(\theta; \mathbf{x}_1, \dots, \mathbf{x}_N)]^T = 2\mathbf{X}_{\theta}\theta,$$

where

$$\mathbf{X}_{\theta} = \sum_{n=1}^N \frac{\mathbf{A}_n}{\theta^T \mathbf{B}_n \theta} - \sum_{n=1}^N \frac{\theta^T \mathbf{A}_n \theta}{(\theta^T \mathbf{B}_n \theta)^2} \mathbf{B}_n.$$

Accordingly, (13.6) can be reformulated as

$$\mathbf{X}_{\hat{\theta}} \hat{\theta} = \mathbf{0}, \tag{13.7}$$

where $\hat{\theta}_{AML}$ is abbreviated to $\hat{\theta}$ for clarity.

The second ingredient, which will be used in combination with the first, is the covariance propagation formula

$$\mathbf{A}_{\hat{\theta}} = \sum_{n=1}^N \partial_{\mathbf{x}_n} \hat{\theta} \mathbf{A}_{\mathbf{x}_n} (\partial_{\mathbf{x}_n} \hat{\theta})^T \tag{13.8}$$

(cf. [15, 22]). Here it is tacitly assumed that, corresponding to varying sets of data points $\mathbf{x}_1, \dots, \mathbf{x}_N$, the normalised vectors $\hat{\theta} = \hat{\theta}(\mathbf{x}_1, \dots, \mathbf{x}_N)$ have been chosen in a coordinated way, so that $\hat{\theta}$ varies smoothly, without sign flipping, as a function of $\mathbf{x}_1, \dots, \mathbf{x}_N$ and in particular may be differentiated.

As a first step towards the derivation of formula (5.2), we differentiate $\|\hat{\theta}\|^2 = 1$ with respect to \mathbf{x}_n to get $(\partial_{\mathbf{x}_n} \hat{\theta})^T \hat{\theta} = \mathbf{0}$. This jointly with (4.4) and (13.8) then implies

$$\mathbf{P}_{\hat{\theta}}^{\perp} \mathbf{A}_{\hat{\theta}} = \mathbf{A}_{\hat{\theta}} \mathbf{P}_{\hat{\theta}}^{\perp} = \mathbf{A}_{\hat{\theta}}. \tag{13.9}$$

Next, letting $\mathbf{x}_n = [m_{n,1}, m_{n,2}]^T$ and $\hat{\theta} = [\hat{\theta}_1, \dots, \hat{\theta}_6]^T$, and differentiating (13.7) with respect to $m_{n,i}$, we obtain

$$\left[[\partial_{m_{n,i}} \mathbf{X}_{\theta}]_{\theta=\hat{\theta}} + \sum_{j=1}^6 [\partial_{\theta_j} \mathbf{X}_{\theta}]_{\theta=\hat{\theta}} \partial_{m_{n,i}} \hat{\theta}_j \right] \hat{\theta} + \mathbf{X}_{\hat{\theta}} \partial_{m_{n,i}} \hat{\theta} = \mathbf{0}.$$

Introducing the Gauss-Newton approximation, i.e., neglecting the terms that contain $\hat{\theta}^T \mathbf{u}(\mathbf{x}_n)$, we arrive (after some

calculations) at

$$\frac{\mathbf{u}(\mathbf{x}_n)[\partial_{m_{n,i}} \mathbf{u}(\mathbf{x}_n)]^T \hat{\theta}}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} + \left[\sum_{n=1}^N \frac{\mathbf{A}_n}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} \right] \partial_{m_{n,i}} \hat{\theta} = \mathbf{0}.$$

This together with the observation that the scalar $[\partial_{m_{n,i}} \mathbf{u}(\mathbf{x}_n)]^T \hat{\theta}$ can also be written as $\hat{\theta}^T \partial_{m_{n,i}} \mathbf{u}(\mathbf{x}_n)$ leads to

$$\begin{aligned} \left[\sum_{n=1}^N \frac{\mathbf{A}_n}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} \right] \partial_{m_{n,i}} \hat{\theta} &= - \frac{\mathbf{u}(\mathbf{x}_n)[\partial_{m_{n,i}} \mathbf{u}(\mathbf{x}_n)]^T \hat{\theta}}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} \\ &= - \frac{\mathbf{u}(\mathbf{x}_n) \hat{\theta}^T \partial_{m_{n,i}} \mathbf{u}(\mathbf{x}_n)}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}}. \end{aligned}$$

Consequently,

$$\left[\sum_{n=1}^N \frac{\mathbf{A}_n}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} \right] \partial_{\mathbf{x}_n} \hat{\theta} = - \frac{\mathbf{u}(\mathbf{x}_n) \hat{\theta}^T \partial_{\mathbf{x}_n} \mathbf{u}(\mathbf{x}_n)}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}}$$

and further, recalling the definitions of \mathbf{A}_n and \mathbf{B}_n given in (2.5),

$$\begin{aligned} \left[\sum_{n=1}^N \frac{\mathbf{A}_n}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} \right] \partial_{\mathbf{x}_n} \hat{\theta} \mathbf{A}_{\mathbf{x}_n} (\partial_{\mathbf{x}_n} \hat{\theta})^T \left[\sum_{n=1}^N \frac{\mathbf{A}_n}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} \right] \\ = \frac{\mathbf{u}(\mathbf{x}_n) \hat{\theta}^T \partial_{\mathbf{x}_n} \mathbf{u}(\mathbf{x}_n) \mathbf{A}_{\mathbf{x}_n} [\partial_{\mathbf{x}_n} \mathbf{u}(\mathbf{x}_n)]^T \hat{\theta} \mathbf{u}(\mathbf{x}_n)^T}{(\hat{\theta}^T \mathbf{B}_n \hat{\theta})^2} \\ = \frac{\mathbf{u}(\mathbf{x}_n) \mathbf{u}(\mathbf{x}_n)^T}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} = \frac{\mathbf{A}_n}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}}. \end{aligned}$$

Now

$$\begin{aligned} \left[\sum_{n=1}^N \frac{\mathbf{A}_n}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} \right] \left[\sum_{n=1}^N \partial_{\mathbf{x}_n} \hat{\theta} \mathbf{A}_{\mathbf{x}_n} (\partial_{\mathbf{x}_n} \hat{\theta})^T \right] \left[\sum_{n=1}^N \frac{\mathbf{A}_n}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}} \right] \\ = \sum_{n=1}^N \frac{\mathbf{A}_n}{\hat{\theta}^T \mathbf{B}_n \hat{\theta}}. \end{aligned}$$

By (5.1) and (13.8), the last equality becomes

$$\mathbf{M}_{\hat{\theta}} \mathbf{A}_{\hat{\theta}} \mathbf{M}_{\hat{\theta}} = \mathbf{M}_{\hat{\theta}}. \tag{13.10}$$

At this stage, one might be tempted to conclude that $\mathbf{A}_{\hat{\theta}} = \mathbf{M}_{\hat{\theta}}^{-1}$, but this would contravene the fact that $\mathbf{A}_{\hat{\theta}}$ is singular. In order to exploit (13.10) properly as an approximate equality, we first note that, in view of (13.9) and the fact that $\mathbf{P}_{\hat{\theta}}^{\perp}$ is idempotent, $\mathbf{P}_{\hat{\theta}}^{\perp} = (\mathbf{P}_{\hat{\theta}}^{\perp})^2$, we have

$$\mathbf{P}_{\hat{\theta}}^{\perp} \mathbf{A}_{\hat{\theta}} \mathbf{P}_{\hat{\theta}}^{\perp} = \mathbf{A}_{\hat{\theta}}, \tag{13.11}$$

so (13.10) can be rewritten as

$$\mathbf{M}_{\hat{\theta}} \mathbf{P}_{\hat{\theta}}^{\perp} \mathbf{A}_{\hat{\theta}} \mathbf{P}_{\hat{\theta}}^{\perp} \mathbf{M}_{\hat{\theta}} = \mathbf{M}_{\hat{\theta}}.$$

Pre- and post-multiplying the last equation by \mathbf{P}_θ^\perp and letting

$$\mathbf{M}_\theta^\perp = \mathbf{P}_\theta^\perp \mathbf{M}_\theta \mathbf{P}_\theta^\perp$$

now leads to

$$\mathbf{M}_\theta^\perp \mathbf{A}_\theta \mathbf{M}_\theta^\perp = \mathbf{M}_\theta^\perp. \tag{13.12}$$

In turn, pre- and post-multiplying (13.12) by $(\mathbf{M}_\theta^\perp)^+$ yields

$$(\mathbf{M}_\theta^\perp)^+ \mathbf{M}_\theta^\perp \mathbf{A}_\theta \mathbf{M}_\theta^\perp (\mathbf{M}_\theta^\perp)^+ = (\mathbf{M}_\theta^\perp)^+ \mathbf{M}_\theta^\perp (\mathbf{M}_\theta^\perp)^+. \tag{13.13}$$

The matrix \mathbf{M}_θ^\perp is symmetric and its null space is spanned by $\hat{\theta}$, so

$$\mathbf{M}_\theta^\perp (\mathbf{M}_\theta^\perp)^+ = (\mathbf{M}_\theta^\perp)^+ \mathbf{M}_\theta^\perp = \mathbf{P}_\theta^\perp$$

(cf. [4, Cor. 3.5]). We also have $(\mathbf{M}_\theta^\perp)^+ \mathbf{M}_\theta^\perp (\mathbf{M}_\theta^\perp)^+ = (\mathbf{M}_\theta^\perp)^+$ by virtue of one of the four defining properties of the pseudo-inverse [4, Thm. 3.9]. Therefore (13.13) can be restated as

$$\mathbf{P}_\theta^\perp \mathbf{A}_\theta \mathbf{P}_\theta^\perp = (\mathbf{M}_\theta^\perp)^+,$$

which, on account of (13.11), implies

$$\mathbf{A}_\theta = (\mathbf{M}_\theta^\perp)^+. \tag{13.14}$$

We now deduce our final formula for \mathbf{A}_θ , namely

$$\mathbf{A}_\theta = \mathbf{P}_\theta^\perp (\mathbf{M}_\theta)_5^+ \mathbf{P}_\theta^\perp, \tag{13.15}$$

which is nothing else but Eq. (5.2) transcribed to the present notation. First we note that as, by (13.7), $\hat{\theta}$ spans the null space of \mathbf{X}_θ , \mathbf{X}_θ has rank 5. Next we observe that in the Gauss–Newton approximation \mathbf{X}_θ is equal to \mathbf{M}_θ , so, having rank 5, \mathbf{X}_θ is also approximately equal to $(\mathbf{M}_\theta)_5$. This in turn implies that, approximately,

$$(\mathbf{M}_\theta)_5^+ = \mathbf{X}_\theta^+,$$

given that the function $\mathbf{A} \mapsto \mathbf{A}^+$ is continuous when considered on sets of matrices of equal rank [20, 36, 45, 52], The last equality together with $\mathbf{P}_\theta^\perp \mathbf{X}_\theta^+ \mathbf{P}_\theta^\perp = \mathbf{X}_\theta^+$, which immediately follows from the facts that \mathbf{X}_θ is symmetric and that $\hat{\theta}$ spans the null space of \mathbf{X}_θ , implies

$$\mathbf{P}_\theta^\perp (\mathbf{M}_\theta)_5^+ \mathbf{P}_\theta^\perp = \mathbf{X}_\theta^+. \tag{13.16}$$

As \mathbf{M}_θ is approximately equal to \mathbf{X}_θ , it is clear that $\mathbf{M}_\theta^\perp (= \mathbf{P}_\theta^\perp \mathbf{M}_\theta \mathbf{P}_\theta^\perp)$ is approximately equal to $\mathbf{P}_\theta^\perp \mathbf{X}_\theta \mathbf{P}_\theta^\perp = \mathbf{X}_\theta^\perp$. Both \mathbf{M}_θ^\perp and \mathbf{X}_θ^\perp have rank 5, so, as they are approximately equal, their pseudo-inverses are also approximately equal,

$$(\mathbf{M}_\theta^\perp)^+ = \mathbf{X}_\theta^+,$$

by the aforementioned continuity property of the pseudo-inverse. Combining this last equation with (13.16) yields

$$(\mathbf{M}_\theta^\perp)^+ = \mathbf{P}_\theta^\perp (\mathbf{M}_\theta)_5^+ \mathbf{P}_\theta^\perp,$$

and this together (13.14) establishes (13.15).

References

- Ahn, S.J.: Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space. Lecture Notes in Computer Science, vol. 3151. Springer, Heidelberg (2004)
- Ahn, S.J., Rauh, W., Warnecke, H.: Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recog.* **34**(12), 2283–2303 (2001)
- Al-Sharadqah, A., Chernov, N.: A doubly optimal ellipse fit. *Comput. Stat. Data Anal.* **56**(9), 2771–2781 (2012)
- Albert, A.: Regression and the Moore–Penrose Pseudoinverse. Academic Press, New York (1972)
- Bartoli, A.: Towards gauge invariant bundle adjustment: a solution based on gauge dependent damping. In: Proceedings of 9th International Conference on Computer Vision, pp. 760–765 (2003)
- Brooks, M.J., Chojnacki, W., Gawley, D., van den Hengel, A.: What value covariance information in estimating vision parameters? In: Proceedings of 8th International Conference on Computer Vision, vol. 1, pp. 302–308 (2001)
- Chernov, N., Huang, Q., Ma, H.: Does the best-fitting curve always exist? *ISRN Prob. Stat.* (2012). doi:10.5402/2012/895178
- Chernov, N., Huang, Q., Ma, H.: Fitting quadratic curves to data points. *Br. J. Math. Comput. Sci.* **4**(1), 33–60 (2014)
- Chernov, N., Lesort, C.: Statistical efficiency of curve fitting algorithms. *Comput. Stat. Data Anal.* **47**(4), 713–728 (2004)
- Chernov, N., Ma, H.: Least squares fitting of quadratic curves and surfaces. In: Yoshida, S.R. (ed.) *Computer Vision*, pp. 285–302. Nova Science Publishers, New York (2011)
- Chernov, N., Wijewickrema, S.: Algorithms for projecting points onto conics. *J. Comput. Appl. Math.* **251**, 8–21 (2013)
- Chojnacki, W., Brooks, M.J.: On the consistency of the normalized eight-point algorithm. *J. Math. Imaging Vis.* **28**, 19–27 (2007)
- Chojnacki, W., Brooks, M.J., van den Hengel, A., Gawley, D.: On the fitting of surfaces to data with covariances. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(11), 1294–1303 (2000)
- Chojnacki, W., Brooks, M.J., van den Hengel, A., Gawley, D.: Revisiting Hartley’s normalized eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(9), 1172–1177 (2003)
- Csurka, G., Zeller, C., Zhang, Z., Faugeras, O.D.: Characterizing the uncertainty of the fundamental matrix. *Comput. Vis. Image Underst.* **68**(1), 18–36 (1997)
- Fazekas, I., Kukush, A., Zwanzig, S.: Correction of nonlinear orthogonal regression estimator. *Ukr. Math. J.* **56**(8), 1308–1330 (2004)
- Fitzgibbon, A., Pilu, M., Fisher, R.B.: Direct least square fitting of ellipses. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5), 476–480 (1999)
- Förstner, W.: Uncertainty and projective geometry. In: Corrochano, E.B., Förstner, W. (eds.) *Handbook of Geometric Computing*, pp. 493–534. Springer, Berlin Heidelberg (2005)
- Gander, W., Golub, G.H., Strelbel, R.: Least-squares fitting of circles and ellipses. *BIT* **34**(4), 558–578 (1994)
- Golub, G.H., Pereyra, V.: The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J. Numer. Anal.* **10**(2), 413–432 (1973)
- Halíf, R., Flusser, J.: Numerically stable direct least squares fitting of ellipses. In: Proceedings of 6th International Conference in

- Central Europe on Computer Graphics and Visualization, vol. 1, pp. 125–132 (1998)
22. Haralick, R.M.: Propagating covariance in computer vision. *Int. J. Pattern Recogn. Artif. Intell.* **10**(5), 561–572 (1996)
 23. Hartley, R.: In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(6), 580–593 (1997)
 24. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004)
 25. Hunyadi, L., Vajk, I.: Constrained quadratic errors-in-variables fitting. *Visual Comput.* (2013). doi:10.1007/s00371-013-0885-2
 26. Kanatani, K.: Statistical bias of conic fitting and renormalization. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(3), 320–326 (1994)
 27. Kanatani, K.: *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier, Amsterdam (1996)
 28. Kanatani, K.: Ellipse fitting with hyperaccuracy. *IEICE Trans. Inf. Syst.* **E89-D**(10), 2653–2660 (2006)
 29. Kanatani, K., Al-Sharadqah, A., Chernov, N., Sugaya, Y.: Renormalization returns: hyper-renormalization and its applications. In: *Proceedings of 12th European Conference on Computer Vision, Lecture Notes on Computer Science*, vol. 7574, pp. 384–397 (2012)
 30. Kanatani, K., Rangarajan, P.: Hyper least squares fitting of circles and ellipses. *Comput. Stat. Data Anal.* **55**(6), 2197–2208 (2011)
 31. Kanazawa, Y., Kanatani, K.: Optimal conic fitting and reliability evaluation. *IEICE Trans. Inf. Syst.* **E79-D**(9), 1323–1328 (1996)
 32. Kim, I.: Orthogonal distance fitting of ellipses. *Commun. Korean Math. Soc.* **17**(1), 121–142 (2002)
 33. Kukush, A., Markovsky, I., Huffel, S.V.: Consistent estimation in an implicit quadratic measurement error model. *Comput. Stat. Data Anal.* **47**(1), 123–147 (2004)
 34. Lehmann, E.L., Romano, J.P.: *Testing Statistical Hypotheses*, 3rd edn. Springer, New York (2005)
 35. Lütkepohl, H.: *Handbook of Matrices*. Wiley, Chichester (1996)
 36. Magnus, J.R., Neudecker, H.: *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, Chichester (1988)
 37. Matei, B., Meer, P.: Reduction of bias in maximum likelihood ellipse fitting. In: *Proceedings of 15th International Conference on Pattern Recognition*, vol. 3, pp. 794–798 (2000)
 38. Newsam, G.N., Redding, N.J.: Fitting the most probable curve to noisy observations. In: *Proceedings of International Conference on Image Processing* **2**, 752–755 (1997)
 39. Newsam, G.N., Redding, N.J.: Fitting the most likely curve through noisy data. Technical Report DSTO-RR-0242, Electronics and Surveillance Research Laboratory, Defence Science and Technology Organisation, Edinburgh, South Australia (2002)
 40. Okatani, T., Deguchi, K.: Improving accuracy of geometric parameter estimation using projected score method. In: *Proceedings of 12th International Conference on Computer Vision*, pp. 1733–1740 (2009)
 41. Okatani, T., Deguchi, K.: On bias correction for geometric parameter estimation in computer vision. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 959–966 (2009)
 42. O’Leary, P., Zsombor-Murray, P.: Direct and specific least-square fitting of hyperbolæ and ellipses. *J. Electron. Imaging* **13**(3), 492–503 (2004)
 43. Ortega, J.M., Rheinboldt, W.C.: *Iterative Solution of Nonlinear Equations in Several Variables*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2000)
 44. Osborne, M.R.: Nonlinear least squares—the Levenberg algorithm revisited. *J. Aust. Math. Soc. Ser. B* **19**(3), 343–357 (1976)
 45. Penrose, R.: A generalized inverse for matrices. *Math. Proc. Cambridge Philos. Soc.* **51**, 406–413 (1955)
 46. Porrill, J.: Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter. *Image Vis. Comput.* **8**(1), 37–41 (1990)
 47. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes in C*. Cambridge University Press, Cambridge (1995)
 48. Rosin, P.L.: A note on the least squares fitting of ellipses. *Pattern Recogn. Lett.* **14**(10), 799–808 (1993)
 49. Rosin, P.L.: Analysing error of fit functions for ellipses. *Pattern Recogn. Lett.* **17**(14), 1461–1470 (1996)
 50. Sampson, P.D.: Fitting conic sections to ‘very scattered’ data: an iterative refinement of the Bookstein algorithm. *Comput. Graph. Image Process.* **18**(1), 97–108 (1982)
 51. Scheffé, H.: A method for judging all contrasts in the analysis of variance. *Biometrika* **40**, 87–104 (1953)
 52. Stewart, G.W.: On the continuity of the generalized inverse. *SIAM J. Appl. Math.* **17**(1), 33–45 (1969)
 53. Sturm, P., Gargallo, P.: Conic fitting using the geometric distance. In: *Proceedings of 8th Asian Conference on Computer Vision, Lecture Notes on Computer Science*, vol. 4844, pp. 784–795 (2007)
 54. Szpak, Z., Chojnacki, W., van den Hengel, A.: Guaranteed ellipse fitting with the Sampson distance. In: *Proceedings of 12th European Conference on Computer Vision, Lecture Notes on Computer Science*, vol. 7576, pp. 87–100 (2012)
 55. Szpak, Z.L., Chojnacki, W., van den Hengel, A.: A comparison of ellipse fitting methods and implications for multiple-view geometry estimation. In: *Proceedings of International Conference on Digital Image Computing Techniques and Applications*, pp. 1–8 (2012)
 56. Taubin, G.: Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(11), 1115–1138 (1991)
 57. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: *Proceedings of International Workshop on Vision Algorithms, Lecture Notes in Computer Science*, vol. 1883, pp. 298–372 (1999)
 58. Turner, K.: *Computer perception of curved objects using a television camera*. Ph.D. thesis, Department of Machine Intelligence, University of Edinburgh (1974)
 59. Varah, J.M.: Least squares data fitting with implicit functions. *BIT* **36**(4), 842–854 (1996)
 60. Wong, C.Y., Lin, C.F., Ren, T.R., Kwok, N.M.: A survey on ellipse detection methods. In: *IEEE International Symposium on Industrial Electronics*, pp. 1105–1110 (2012)
 61. Zhang, Z.: Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing* **15**(1), 59–76 (1997)
 62. Zwillinger, D.: *CRC Standard Mathematical Tables and Formulae*, 32nd edn. CRC Press, Boca Raton (2012)



Zygmunt L. Szpak received his Ph.D. degree in Computer Science from the University of Adelaide, Australia, in 2013, and his M.Sc. degree in Computer Science from the University of KwaZulu-Natal, South Africa, in 2009. He is a senior research associate in the School of Computer Science at the University of Adelaide working on a broad range of practical computer vision problems. The overarching theme of his research is constrained parameter estimation, with an emphasis on applications in multiple view geometry.



Wojciech Chojnacki received his M.Sc., Ph.D., and D.Sc. degrees, all in Mathematics, in 1976 from Warsaw University, in 1980 from the Polish Academy of Sciences, and in 1988 from Warsaw University, respectively. He is a professor of mathematics in the Faculty of Mathematics and Natural Sciences, the College of Sciences, at Cardinal Stefan Wyszyński University in Warsaw. He is concurrently a senior research fellow in the School of Computer Science at

the University of Adelaide working on a range of problems in computer vision. His research interests include differential equations, mathematical foundations of computer vision, functional analysis, and harmonic analysis.



Anton van den Hengel received a Ph.D. degree in Computer Science in 2000, a Master in Computer Science degree in 1994, a Bachelor of Laws degree in 1993, and a Bachelor of Mathematical Science degree in 1991, all from the University of Adelaide. He is the founding Director of The Australian Centre for Visual Technologies (ACVT), an interdisciplinary research centre focusing on innovation and education in the production and analysis of visual digital media.

At more than 40 members, the ACVT is Australia's largest Computer Vision research group.