Determining Ellipses from Low-Resolution Images with a Comprehensive Image Formation Model

Wojciech Chojnacki and Zygmunt L. Szpak

Australian Institute for Machine Learning, The University of Adelaide, SA 5005, Australia

Email: {wojciech.chojnacki, zygmunt.szpak}@adelaide.edu.au

Abstract—When determining the parameters of a parametric planar shape based on a single low-resolution image, common estimation paradigms lead to inaccurate parameter estimates. The reason behind poor estimation results is that standard estimation frameworks fail to model the image formation process at a sufficiently detailed level of analysis. We propose a new method for estimating the parameters of a planar elliptic shape based on a single photon-limited, low-resolution image. Our technique incorporates the effects of several elements—point-spread function, discretisation step, quantisation step, and photon noise—into a single cohesive and manageable statistical model. While we concentrate on the particular task of estimating the parameters of elliptic shapes, our ideas and methods have a much broader scope and can be used to address the problem of estimating the parameters of an arbitrary parametrically representable planar shape. Comprehensive experimental results on simulated and real imagery demonstrate that our approach yields parameter estimates with unprecedented accuracy. Furthermore, our method supplies a parameter covariance matrix as a measure of uncertainty for the estimated parameters, as well as a planar confidence region as a means for visualising the parameter uncertainty. The mathematical model developed in this paper may prove useful in a variety of disciplines which operate with imagery at the limits of resolution.

Index Terms—Ellipse fitting, Poisson noise, quantisation, discretisation, image formation model, photon-limited, low-resolution.

I. INTRODUCTION

WE present a method for recovering the parameters of a planar elliptic shape from a low-resolution, photon-limited digital image. Our procedure provides unparalleled parameter estimation accuracy. We develop a systematic but manageable statistical model of the image formation process that distinguishes our approach from contemporary methods. Our model accounts for the point-spread function (PSF), the inherent continuous-to-discrete mapping of the image formation process, as well as the uncertainty due to quantisation and photon noise. Figure 1 illustrates an example of the diverse images that our model accommodates. While our paper focuses on the particular task of estimating the parameters of elliptic shapes, the ideas and methods formulated in this article have a much broader scope and can be used to address the problem of estimating the parameters of an arbitrary parametrically representable planar shape. Determining the parameters of an ellipse from a low-resolution planar image has essential applications in camera calibration [1], but the core of our contribution lies in the details of the mathematical framework and conceptual methodology. We believe that our general approach is worthy of imitation and may lead to substantial progress in confocal microscopy, long-range surveillance, high-accuracy camera calibration, and astronomy.

II. RELATED WORK

The majority of ellipse estimation methods fit a curve to a planar set of points. One distinguishes between point-based ellipse fitting methods by considering the nature of the cost function that the algorithms minimise. Methods which explicitly decrease the distance between the points and the ellipse curve are considered geometric methods. The quintessential geometric method is orthogonal distance regression, which minimises the orthogonal distance from a point to the curve [2]–[8]. Algebraic methods, on the other hand, try to ensure that the data points satisfy an ellipse implicit equation as accurately as possible. One differentiates between algebraic methods by considering how they penalise the degree to which a data point fails to satisfy an implicit equation [9]–[12]. Algebraic methods, in particular, have been the focus of considerable study, and recent works have concentrated on improving their statistical efficacy and accuracy [13]–[16].

The fact that the most advanced ellipse fitting methods operate on data points is problematic when one wishes to fit an ellipse to a photon-limited low-resolution image of an elliptic region. The difficulty lies in extracting a set of data points that precisely lie on the contour of the ellipse. The standard approach involves gradient estimation, non-maxima suppression, and thresholding. These steps produce, at best, a set of data points that approximate the contour only at the level of resolution of the pixel grid. Moreover, each of these steps introduces substantial errors and biases which the noise models of prevailing ellipse fitting methods disregard.

It is possible to obtain data points with sub-pixel coordinates by using sub-pixel edge detection methods [17]. However, sub-pixel techniques usually do not characterise the uncertainty or bias of their estimates, and so one cannot attribute meaningful covariance matrices to the sub-pixel data points. The inability to characterise the uncertainty and bias of the sub-pixel points is a severe limitation and, effectively, violates the modelling assumptions associated with point-based ellipse fitting methods. Moreover, the sub-pixel estimation methods are themselves sensitive to noise and do not take quantisation into account.

As an alternative to point-based ellipse fitting, Ouellet and Hébert [18] proposed a region-based method. The region-based method exploits a duality relationship between a point and a line in projective geometry. The duality relationship states that the homogeneous representation of a point may simultaneously be interpreted as a description of a line. Hence, one can equivalently state the problem of fitting an ellipse to a set of points, as the task of fitting an ellipse to a set of lines. The particular set of lines that satisfy an ellipse equation are called the envelope of tangent lines. The region-based method takes advantage of the observation that lines perpendicular to the gradient of an ideal ellipse image are tangent to the ellipse. The method proceeds by computing the gradient of an image and discarding pixels where the magnitude of the gradient is below a specified threshold. For each remaining pixel, the method constructs a line that is perpendicular to the orientation of the gradient. The algorithm then minimises an algebraic error, which is similar to the well-known point-based direct ellipse fit (DEF) [19], [20]. An important difference is that each line contributes to the algebraic cost by a weight equal to the magnitude of the gradient from which the line was derived.

The limitations mentioned for the point-based ellipse fitting methods also hold for the aforementioned region-based method. This particular region-based method does not model the image formation process, and
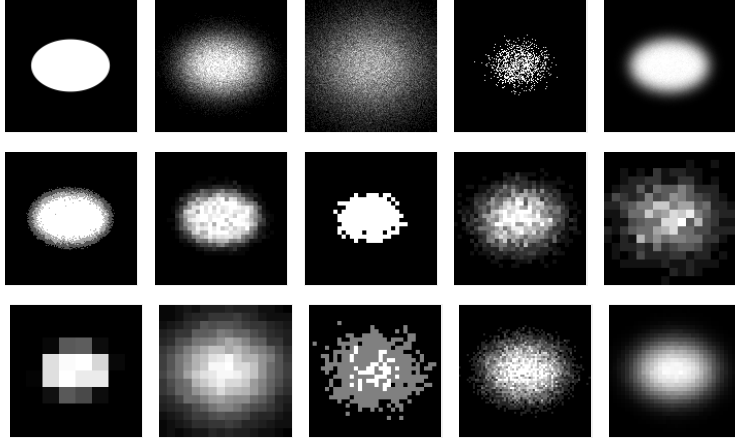
Fig. 1. Examples of different digital images of the same elliptic region. The top-left image denotes an ideal digital image of an elliptic region. All the other images are observations of this image with different point-spread functions at varying levels of resolution, noise level, and quantisation. Our aim is to recover the ellipse parameters associated with the ideal digital image (top-left) given its corrupted observation.

so cannot accommodate Poisson noise or quantisation in a principled manner. Furthermore, the method still operates at the level of pixels and so does not address the resolution problem.

We address the shortcomings of these established ellipse fitting methods by developing a new estimation framework which incorporates an intricate but tractable model of the image formation process.

III. IMAGE FORMATION PROCESS

One can conceptualise an imaging system as a continuous-to-discrete operator which maps a function of continuous variables (the elliptic shape in our 3D world) to a finite set of numbers (the discrete image) [21]. The information loss in the passage from the continuous domain to the digital one occurs in four stages (see Fig. 2). First, the 3D world is projected onto a 2D plane using one of several projection methods available (perspective, fish-eye, catadioptric, etc.). The projection process produces an analogue image with infinite resolution, or an ideal geometric image. The geometric image is an idealisation—any real optical device (e.g., a camera lens) imposes certain imprecisions such as geometric distortions and blurring. The second stage models the effect of the errors and leads to the real analogue image. This image still resides in the analogue domain and is not directly observable. The third and fourth stage, discretisation plus addition of noise (including photon counting noise, electronic noise and quantisation round-off noise), finally produces an observable digital image.

To keep the transition from the ideal to the real analogue image tractable, it is standard to assume that the amount of blurring does not vary within the field of view, and as such can be modelled by a convolution of the ideal image with a single function [22]. This function is known as the PSF of the image acquisition device. With the PSF at hand, the relation between the ideal and real analogue images can be written as

$$f_{\text{real}}(x, y) = (f_{\text{ideal}} * \text{psf})(x, y), \tag{1}$$

where $*$ denotes convolution in the plane [23]. As it turns out, the PSF can in practice be well approximated by a Gaussian kernel,

$$\text{psf}(x, y) = \frac{1}{2\pi\sigma_{\text{PSF}}^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_{\text{PSF}}^2}\right) \tag{2}$$

with some positive σ_{PSF}. We shall adopt this approximation in our discussion.

The discretisation stage is reflective of the fact that the image values are constant over each pixel from a grid of pixels, and that they are
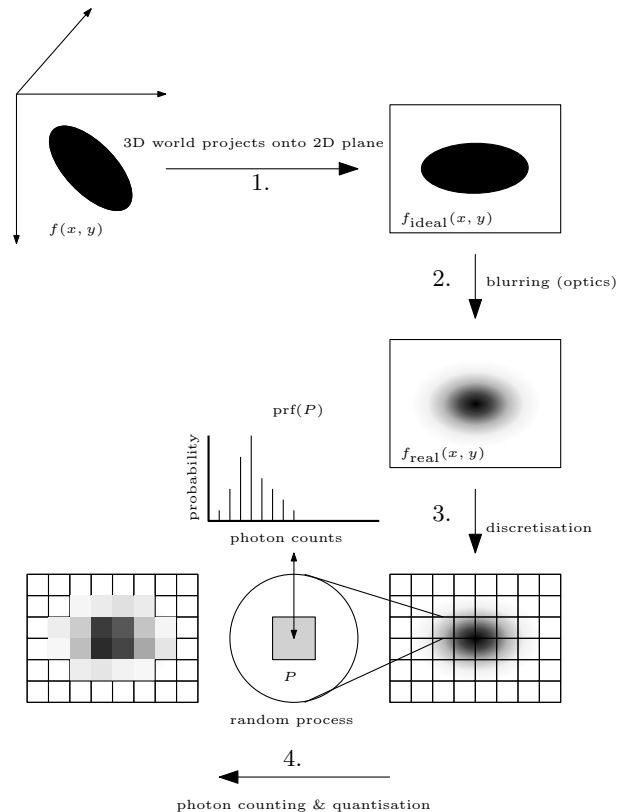

Fig. 2. An overview of an image formation model. In the first step, the 3D world is projected onto a 2D image plane resulting in an ideal geometric image. The second stage accounts for imprecisions such as geometric distortions and blurring. The third step imposes a grid of pixels, introduces a statistical model for the number of photons that hit each pixel, and accounts for noise. The final step models the loss of information due to quantisation of the photon counts.

multiples of a single numerical value. The latter effect is explained by the image intensity at a particular pixel being proportional to the number of photo-electrons recorded by the pixel's sensor. Instrumental in the passage from the real analogue to the digital image is a pixel response function:

$$\text{prf}(P) = \frac{1}{|P|} \int_P f_{\text{real}}(x, y) \, \mathrm{d}x \, \mathrm{d}y, \tag{3}$$

where, for a pixel P, $|P|$ denotes the area of P. We shall regard the image intensity at the pixel P as a random value fluctuating around $\mathrm{prf}(P)$.

IV. PROBABILISTIC MODEL

Let C be the conversion factor linking the image intensity with the photo-electron count. If the image intensity is a number between, say, 0 and 1, then the corresponding value of the photo-electron count is an integer between 0 and C. Neglecting—temporarily—the digitisation error, it is natural to model the photo-electron count N_P at P stochastically by applying a Poisson noise to $\mathrm{prf}(P)$, i.e., by letting

$$N_P = X_P, \tag{4}$$

where X_P is a Poisson-distributed random variable with mean $C\,\mathrm{prf}(P)$,

$$\Pr(X_P = k) = \frac{(C\,\mathrm{prf}(P))^k}{k!}\mathrm{e}^{-kC\,\mathrm{prf}(P)} \tag{5}$$

for $k = 0, 1, 2, \dots$.

To include the quantisation error, we modify the above definition and add to X_P an integer-valued random variable U_P uniformly distributed in the range

$$[-b, b] \cap \mathbb{Z} = \{-b, -b+1, \dots, -1, 0, 1, \dots, b-1, b\}, \tag{6}$$

where b is a non-negative integer. In other words, we let

$$N_P = X_P + U_P. \tag{7}$$

Our proposed image recovery method will be based on the statistical model of a pixel value embodied in the above formula. To proceed, it will be critical to identify the form of the pixel response function and the shape of the probability distribution of N_P.

V. PIXEL RESPONSE FUNCTION

We now provide a computationally convenient expression for the pixel response function for an image of a uniform white planar shape. We consider two scenarios whereby the shape appears against (1) a completely black backdrop and (2) a grey backdrop.

A. Black background

Let D be a uniform white region on a black planar background. Then the image associated with D can be described as

$$f_{\mathrm{ideal}}(x, y) = 1_D(x, y), \tag{8}$$

where 1_D stands for the characteristic function of D. In view of (1),

$$f_{\mathrm{real}}(x, y) = \int_{\mathbb{R}^2} 1_D(x-s, y-t)\,\mathrm{psf}(s, t)\,\mathrm{d}s\,\mathrm{d}t. \tag{9}$$

For $(s, t) \in \mathbb{R}^2$, let $T_{(s,t)}D$ denote the *translate* of D by (s, t),

$$T_{(s,t)}D = \{(x, y) \in \mathbb{R}^2 \mid (x-s, y-t) \in D\}. \tag{10}$$

It is readily checked that

$$1_D(x-s, y-t) = 1_{T_{(s,t)}D}(x, y). \tag{11}$$

With this in mind, for a given pixel P, we have

$$\int_P \left[\int_{\mathbb{R}^2} 1_D(x-s, y-t)\,\mathrm{psf}(s, t)\,\mathrm{d}s\,\mathrm{d}t \right]\mathrm{d}x\,\mathrm{d}y$$
$$= \int_P \left[\int_{\mathbb{R}^2} 1_{T_{(s,t)}D}(x, y)\,\mathrm{psf}(s, t)\,\mathrm{d}s\,\mathrm{d}t \right]\mathrm{d}x\,\mathrm{d}y$$
$$= \int_{\mathbb{R}^2} \left[\int_P 1_{T_{(s,t)}D}(x, y)\,\mathrm{d}x\,\mathrm{d}y \right]\mathrm{psf}(s, t)\,\mathrm{d}s\,\mathrm{d}t$$
$$= \int_{\mathbb{R}^2} \left| P \cap T_{(s,t)}D \right|\mathrm{psf}(s, t)\,\mathrm{d}s\,\mathrm{d}t$$
$$= \int_{\mathbb{R}^2} \left| T_{(-s,-t)}P \cap D \right|\mathrm{psf}(s, t)\,\mathrm{d}s\,\mathrm{d}t, \tag{12}$$

where the identity

$$\left| P \cap T_{(s,t)}D \right| = \left| T_{(-s,-t)}P \cap D \right| \tag{13}$$

results from

$$T_{(-s,-t)}(P \cap T_{(s,t)}D) = T_{(-s,-t)}P \cap D \tag{14}$$

and the fact that the area of $P \cap T_{(s,t)}D$ is the same as the area of the translate $T_{(-s,-t)}(P \cap T_{(s,t)}D)$ of $P \cap T_{(s,t)}D$ by $(-s, -t)$. Combining Eqs. (3), (9), and (12), we finally obtain

$$\mathrm{prf}(P) = \frac{1}{|P|}\int_{\mathbb{R}^2} \left| T_{(-s,-t)}P \cap D \right|\mathrm{psf}(s, t)\,\mathrm{d}s\,\mathrm{d}t. \tag{15}$$

B. Grey background

Let D be a uniform white region on a grey planar background. Suppose that the background has intensity c, where $0 \le c < 1$. Then the image associated with D can be described as

$$f_{\mathrm{ideal}}(x, y) = 1_D(x, y) + c1_{\mathbb{R}^2 \setminus D}(x, y) \tag{16}$$

or, equivalently, as

$$f_{\mathrm{ideal}}(x, y) = (1-c)1_D(x, y) + c. \tag{17}$$

It then immediately follows from (15) that the pixel response function in this case is given by

$$\mathrm{prf}(P) = c + \frac{1-c}{|P|}\int_{\mathbb{R}^2} \left| T_{(-s,-t)}P \cap D \right|\mathrm{psf}(s, t)\,\mathrm{d}s\,\mathrm{d}t. \tag{18}$$

The above formula will play a key role in the subsequent development. Note that (15) is a particular case of (18) with $c = 0$.

VI. PROBABILITY DISTRIBUTION

We now calculate the probability distribution of the random variable N_P defined in (7).

Suppose that X and Y are two independent random variables, with X being Poisson-distributed with parameter λ, and Y being discrete uniformly distributed over the set $[-b, b] \cap \mathbb{Z}$,

$$X \sim Poisson(\lambda), \quad Y \sim DU(-b, b). \tag{19}$$

Then, for each $n \in [-b, \infty) \cap \mathbb{Z} = \{-b, b+1, \dots\}$, the event $X + Y = n$ is the union of the pairwise disjoint events $\{X + Y = n\}$ and $\{Y = m\}$, where m runs over $[-b, b] \cap \mathbb{Z}$. It follows that

$$\Pr(X + Y = n) = \sum_{m=-b}^{b} \Pr(X + Y = n, Y = m). \tag{20}$$

Since

$$\Pr(X + Y = n, Y = m) = \Pr(X = n - Y, Y = m)$$
$$= \Pr(X = n - m, Y = m)$$
$$= \Pr(X = n - m)\Pr(Y = m), \tag{21}$$

where the last equality holds by the independence of X and Y, and

$$\Pr(X = n - m) = e^{-\lambda} \frac{\lambda^{n-m}}{(n-m)!} 1_{[0,\infty)}(n-m), \qquad (22)$$

$$\Pr(Y = m) = \frac{1}{2b+1}, \qquad (23)$$

we have

$$\Pr(X + Y = n) = \frac{1}{2b+1} \sum_{m=-b}^{b} e^{-\lambda} \frac{\lambda^{n-m}}{(n-m)!} 1_{[0,\infty)}(n-m)$$

$$= \frac{1}{2b+1} \sum_{m=-b}^{b} e^{-\lambda} \frac{\lambda^{n+m}}{(n+m)!} 1_{[0,\infty)}(n+m). \qquad (24)$$

Since $1_{[0,\infty)}(n+m) = 0$ whenever $m < -n$, the dummy integer-valued variable m in the last sum has to satisfy two conditions: $-b \le m \le b$ and $n + m \ge 0$ or, equivalently, $-n \le m$. These conditions may conveniently be combined into a single condition, namely, $\max(-b, -n) \le m \le b$ or, what is the same, $-\min(b, n) \le m \le b$, with the proviso that $-\min(b,n) \le b$; if $-\min(b,n) > b$, we necessarily have $\Pr(X+Y = n) = 0$. Thus

$$\Pr(X + Y = n)$$
$$= \begin{cases} \frac{1}{2b+1} \sum_{m=-\min(b,n)}^{b} e^{-\lambda} \frac{\lambda^{n+m}}{(n+m)!} & \text{if } -\min(b,n) \le b, \\ 0 & \text{otherwise.} \end{cases} \qquad (25)$$

We immediately infer from this formula that if $b = 0$, then

$$\Pr(X + Y = n) = e^{-\lambda} \frac{\lambda^{n}}{n!} = \Pr(X) \qquad (26)$$

for each $n \in \mathbb{Z}_{\ge 0} := \{0, 1, 2, \dots\}$.

Suppose that $b > 0$. Then if $n < -b$, then $\min(b, n) = n$ and so $-\min(b,n) > b$, implying, according to (25), that

$$\Pr(X + Y = n) = 0. \qquad (27)$$

If $-b \le n \le b$, then $\min(b, n) = n$, and we have

$$\Pr(X + Y = n) = \frac{1}{2b+1} \sum_{m=-n}^{b} e^{-\lambda} \frac{\lambda^{n+m}}{(n+m)!}$$

$$= \frac{1}{2b+1} \sum_{k=0}^{n+b} e^{-\lambda} \frac{\lambda^{k}}{k!}. \qquad (28)$$

If $n > b$, then $\min(b, n) = b$, and we have

$$\Pr(X + Y = n) = \frac{1}{2b+1} \sum_{m=-b}^{b} e^{-\lambda} \frac{\lambda^{n+m}}{(n+m)!}$$

$$= \frac{1}{2b+1} \sum_{k=n-b}^{n+b} e^{-\lambda} \frac{\lambda^{k}}{k!}$$

$$= \frac{1}{2b+1} \left(\sum_{k=0}^{n+b} e^{-\lambda} \frac{\lambda^{k}}{k!} - \sum_{k=0}^{n-b} e^{-\lambda} \frac{\lambda^{k}}{k!} \right). \qquad (29)$$

Using the upper incomplete gamma function

$$\Gamma(\alpha, x) = \int_{x}^{\infty} e^{-t} t^{\alpha-1} \, dt \qquad (30)$$

and the formula

$$\Gamma(n + 1, x) = n! \sum_{m=0}^{n} e^{-x} \frac{x^{m}}{m!} \qquad (31)$$

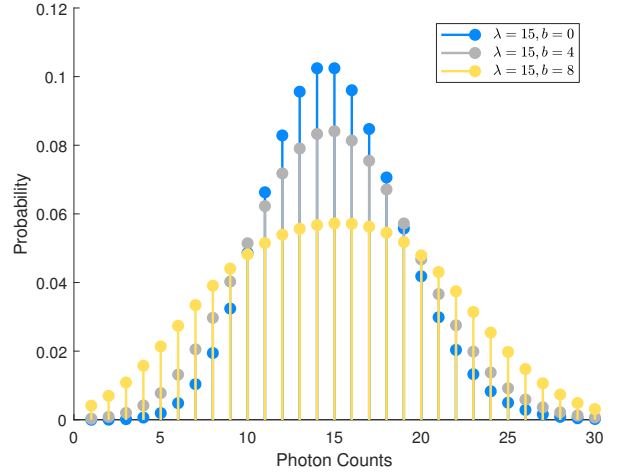for $n \in \mathbb{Z}_{\ge 0}$, we can rewrite Eqs. (27)–(29) as

Fig. 3. A visualisation of the probability mass function of quantised photon counts given by equations Eqs. (25) and (32).

$$\Pr(X + Y = n)$$
$$= \begin{cases} 0 & \text{if } n < -b, \\[2mm] \dfrac{\Gamma(n+b+1, \lambda)}{(2b+1)(n+b)!} & \text{if } -b \le n \le b, \\[4mm] \dfrac{1}{2b+1} \left(\dfrac{\Gamma(n+b+1, \lambda)}{(n+b)!} \right. & \\ \left. - \dfrac{\Gamma(n-b+1, \lambda)}{(n-b)!} \right) & \text{if } n > b. \end{cases} \qquad (32)$$

An illustration of this probability mass function is presented in Fig. 3.

VII. ESTIMATION METHOD

Suppose that our region of interest, say the interior of an ellipse, is a member of a family of candidate regions $\{D_{\boldsymbol{\xi}} \mid \boldsymbol{\xi} \in \boldsymbol{\Xi}\}$ indexed by a vector of parameters $\boldsymbol{\xi}$ running over a parameter space $\boldsymbol{\Xi}$. Let $\boldsymbol{\xi}_*$ be the vector of parameters determining the region of interest. For each $\boldsymbol{\xi} \in \boldsymbol{\Xi}$, suppose that $D_{\boldsymbol{\xi}}$ is transformed via a physical process into a digital image. Let $\mathcal{P}$ denote the set of corresponding pixels. For each pixel P in $\mathcal{P}$, we assume that the image intensity f_P of P is modelled as a random variable N_P given by (7). Moreover, we assume that the N_P, $P \in \mathcal{P}$, are stochastically independent. This means that with

$$N = (N_P)_{P \in \mathcal{P}} \quad \text{and} \quad f_{\text{digital}} = (f_P)_{P \in \mathcal{P}}, \qquad (33)$$

we have

$$\Pr(N = f_{\text{digital}}) = \prod_{P \in \mathcal{P}} \Pr(N_P = f_P). \qquad (34)$$

The expression on the right hand side depends on the parameters $\boldsymbol{\xi}$, b, c, C, and σ_{PSF}. More explicitly, we have, in accordance with (32),

$$\Pr(N_P = f_P)$$
$$= \begin{cases} 0 & \text{if } n < -b, \\[2mm] \dfrac{\Gamma(f_P + b + 1, \lambda_P)}{(2b+1)(f_P + b)!} & \text{if } -b \le f_P \le b, \\[4mm] \dfrac{1}{2b+1}\left(\dfrac{\Gamma(f_P + b + 1, \lambda_P)}{(f_P + b)!} \right. & \\ \left. \quad - \dfrac{\Gamma(f_P - b + 1, \lambda_P)}{(f_P - b)!} \right) & \text{if } f_P > b, \end{cases} \tag{35}$$

where, in line with Eqs. (5) and (18),

$$\lambda_P = C\left(c + \frac{1-c}{|P|} \int_{\mathbb{R}^2} \left| T_{(-s,-t)} P \cap D_{\boldsymbol{\xi}} \right| \operatorname{psf}(s,t)\, \mathrm{d}s\, \mathrm{d}t \right) \tag{36}$$

depends on $\boldsymbol{\xi}$, c, C, and, in view of (2), also on σ_{PSF}. We treat b, c and C as values known a priori and fixed. Writing, more emphatically, $\Pr(N = f_{\mathrm{digital}})$ as $\Pr(f_{\mathrm{digital}} \mid \boldsymbol{\xi}, \sigma_{\mathrm{PSF}})$, we may treat $(\boldsymbol{\xi}, \sigma_{\mathrm{PSF}}) \mapsto \Pr(f_{\mathrm{digital}} \mid \boldsymbol{\xi}, \sigma_{\mathrm{PSF}})$ as the likelihood function for $\boldsymbol{\xi}$ and σ_{PSF}. Using the maximum likelihood (ML) principle, we may next estimate $\boldsymbol{\xi}$ and σ_{PSF}, given an observed digital image $f_{\mathrm{observed}} = (f_P^{\mathrm{ob}})_{P \in \mathcal{P}}$, by minimising the corresponding negative log-likelihood function

$$\ell(\boldsymbol{\xi}, \sigma_{\mathrm{PSF}}) := - \sum_{P \in \mathcal{P}} \ln \Pr(N_P = f_P^{\mathrm{ob}}). \tag{37}$$

In other words,

$$\{\widehat{\boldsymbol{\xi}}, \widehat{\sigma}_{\mathrm{PSF}}\} = \arg\min_{\boldsymbol{\xi}, \sigma_{\mathrm{PSF}}} \ell(\boldsymbol{\xi}, \sigma_{\mathrm{PSF}}). \tag{38}$$

We are solely interested in obtaining an estimate of the region of concern, so once the minimisation is performed, we may discard $\widehat{\sigma}_{\mathrm{PSF}}$ and the remaining $\widehat{\boldsymbol{\xi}}$ is then an estimate of $\boldsymbol{\xi}_*$. We refer to $\widehat{\boldsymbol{\xi}}$ as the *ML estimate* of $\boldsymbol{\xi}_*$, and dub the method of generating $\widehat{\boldsymbol{\xi}}$ the *ML estimator* for region estimation.

The above-described estimation method requires, critically, a means for calculating the term $\left| T_{(-s,-t)} P \cap D_{\boldsymbol{\xi}} \right|$ in (36). As it turns out, the evaluation of $\left| T_{(-s,-t)} P \cap D_{\boldsymbol{\xi}} \right|$ can be performed effectively in the case where $D_{\boldsymbol{\xi}}$ is an ellipse (with $\boldsymbol{\xi}$ the vector of the ellipse's parameters). This is due to the fact that there exist explicit formulae for determining the area of intersection between an ellipse and a rectangle (a pixel) in the case that the sides of the rectangle are parallel to the semi-axes of the ellipse. We present these rather involved formulae in Appendix A and henceforth concentrate our discussion exclusively on ellipse estimation.

A. Characterising an ellipse region

An ellipse in general position can be expressed parametrically as

$$x = H + A \cos\alpha \cos\tau - B \sin\alpha \sin\tau,$$
$$y = K + A \cos\alpha \sin\tau + B \sin\alpha \cos\tau. \tag{39}$$

Here, A and B represent the length of the semi-major and semi-minor axis of the ellipse, H and K denote the x and y coordinates of the centre of the ellipse, τ is the angle formed by the major axis with the positive x axis, and α is the angular co-ordinate of the point (x, y) on the ellipse. The vector $\boldsymbol{\xi} = [A, B, H, K, \tau]^{\top}$ (excluding α) encompasses the geometric parameters of the ellipse and uniquely describes the ellipse as a set. Alongside the parametric form, the ellipse can be represented in Cartesian form as the locus of points (x, y) in the plane satisfying

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{40}$$

where a, b, c, d, e, f are real numbers such that $b^2 < 4ac$. The vector $\boldsymbol{\theta} = [a, b, c, d, e, f]^{\top}$ of the algebraic parameters of the ellipse determines the ellipse uniquely, however the reverse correspondence is not univocal—all non-zero multiples of $\boldsymbol{\theta}$ describe one and the same ellipse. Using the Cartesian form, the interior of the ellipse can be conveniently characterised as the locus of points (x, y) in the plane satisfying

$$ax^2 + bxy + cy^2 + dx + ey + f < 0. \tag{41}$$

The above two descriptions of an ellipse are fully equivalent, each being obtainable from the other by means of a conversion formula. The explicit formulas for conversion will be of relevance in what follows. The rule for the passage from the geometric parameters to the algebraic parameters is given by

$$a = \frac{\cos^2\tau}{A^2} + \frac{\sin^2\tau}{B^2},$$
$$b = \left(\frac{1}{A^2} - \frac{1}{B^2} \right) \sin 2\tau,$$
$$c = \frac{\cos^2\tau}{B^2} + \frac{\sin^2\tau}{A^2},$$
$$d = \frac{2\sin\tau\, (K\cos\tau - H\sin\tau)}{B^2} - \frac{2\cos^2\tau\, (H + K\tan\tau)}{A^2},$$
$$e = \frac{2\cos\tau\, (H\sin\tau - K\cos\tau)}{B^2} - \frac{2\sin\tau\, (H\cos\tau + K\sin\tau)}{A^2},$$
$$f = \frac{(H\cos\tau + K\sin\tau)^2}{A^2} + \frac{(K\cos\tau - H\sin\tau)^2}{B^2} - 1. \tag{42}$$

To present the rule for the passage from the algebraic parameters into geometric parameters, we first let

$$\Delta = b^2 - 4ac,$$
$$\lambda_{\pm} = \frac{1}{2}\left(a + c \mp \left(b^2 + (a - c)^2 \right)^{1/2} \right),$$
$$\psi = bde - ae^2 - b^2 f + c(4af - d^2),$$
$$V^{\pm} = \left(\frac{\psi}{\lambda_{\pm}\Delta} \right)^{1/2}, \tag{43}$$

where $\pm$ and $\mp$ are shorthand for $+$ *or* $-$, which allows presentation of two expressions in one formula, with the upper $-$ of $\mp$ associated
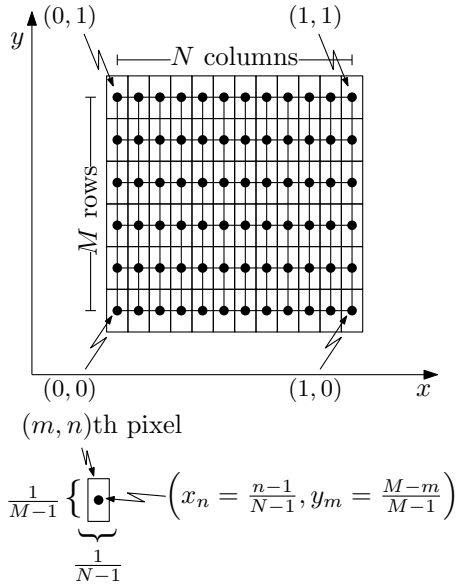
Fig. 4. $M \times N$ array of pixels with the corresponding grid of pixel centres spanning the unit box $[0,1] \times [0,1]$. The diagram includes a formula for converting the matrix indices of a pixel to the pixel's Cartesian coordinates.

with the $+$ of $\pm$. We can now state the rule in question as

$$A = \max(V^+, V^-), \quad B = \min(V^+, V^-),$$
$$H = \frac{2cd - be}{\Delta}, \quad K = \frac{2ae - bd}{\Delta},$$

$$\tau = \begin{cases}
\frac{1}{2}\operatorname{arccot}\left(\frac{a-c}{b}\right) & \text{if } b < 0,\ a < c \text{ and } V^+ \geq V^-, \\
\frac{\pi}{4} & \text{if } b < 0,\ a = c \text{ and } V^+ \geq V^-, \\
\frac{1}{2}\operatorname{arccot}\left(\frac{a-c}{b}\right) + \frac{\pi}{2} & \text{if } b < 0,\ a > c \text{ and } V^+ \geq V^-, \\
0 & \text{if } b = 0,\ a < c \text{ and } V^+ \geq V^-, \\
\frac{\pi}{2} & \text{if } b = 0,\ a \geq c \text{ and } V^+ \geq V^-, \\
\frac{1}{2}\operatorname{arccot}\left(\frac{a-c}{b}\right) + \pi & \text{if } b > 0,\ a < c \text{ and } V^+ \geq V^-, \\
\frac{3\pi}{4} & \text{if } b > 0,\ a = c \text{ and } V^+ \geq V^-, \\
\frac{1}{2}\operatorname{arccot}\left(\frac{a-c}{b}\right) + \frac{\pi}{2} & \text{if } b > 0,\ a > c \text{ and } V^+ \geq V^-, \\
\frac{1}{2}\operatorname{arccot}\left(\frac{a-c}{b}\right) + \frac{\pi}{2} & \text{if } b < 0,\ a < c \text{ and } V^+ < V^-, \\
\frac{3\pi}{4} & \text{if } b < 0,\ a = c \text{ and } V^+ < V^-, \\
\frac{1}{2}\operatorname{arccot}\left(\frac{a-c}{b}\right) + \pi & \text{if } b < 0,\ a > c \text{ and } V^+ < V^-, \\
\frac{\pi}{2} & \text{if } b = 0,\ a < c \text{ and } V^+ < V^-, \\
0 & \text{if } b = 0,\ a \geq c \text{ and } V^+ < V^-, \\
\frac{1}{2}\operatorname{arccot}\left(\frac{a-c}{b}\right) + \frac{\pi}{2} & \text{if } b > 0,\ a < c \text{ and } V^+ < V^-, \\
\frac{\pi}{4} & \text{if } b > 0,\ a = c \text{ and } V^+ < V^-, \\
\frac{1}{2}\operatorname{arccot}\left(\frac{a-c}{b}\right) & \text{if } b > 0,\ a > c \text{ and } V^+ < V^-
\end{cases}$$
$$(44)$$

(see [24, Section 4.10.2] for the starting point of the derivation of the formulas). We remark that the formula for τ is valid only under the assumption that the ellipse is not a circle, i.e., provided the inequality $(a - c)^2 + b^2 > 0$ holds.

B. Forming a digital image of an ellipse region

To be able to make use of the negative log-likelihood given in (37), one needs to have a way of constructing digital images of candidate ellipse regions that incorporate the effects of the PSF, discretisation step, quantisation step, and photon noise. In this section, we outline the procedure for constructing such images.

With a view to generating a single specific image, given a pair of integers M and N such that $M \geq 2$ and $N \geq 2$, we create a grid of

pixels in the form of an $M \times N$ array of rectangles aligned with the x and y axes, each of size $(M - 1)^{-1} \times (N - 1)^{-1}$, with the centre (x_n, y_m) of the (m, n)th rectangle specified by

$$x_n = \frac{n-1}{N-1} \quad \text{and} \quad y_m = \frac{M-m}{M-1} \tag{45}$$

for every $1 \leq m \leq M$ and every $1 \leq n \leq N$ (see Fig. 4). Given a particular ellipse specified by a parameter vector $\boldsymbol{\theta}$, we construct a digital image via the following steps. We first determine the geometric parameters of the ellipse. The relevant formulae are given in Subsection VII-A. Next, we take advantage of the fact that an application of the coordinate transformation

$$x' = (x - H)\cos\tau + (y - K)\sin\tau,$$
$$y' = -(x - H)\sin\tau + (y - K)\cos\tau \tag{46}$$

brings the ellipse to a standard form. More specifically, we apply the above transformation to each pixel centre, obtaining points (x'_n, y'_m) $(m = 1, \ldots, M,\ n = 1, \ldots, N)$. For each pair (m, n), we form a rectangle of size $(M - 1)^{-1} \times (N - 1)^{-1}$ centred at (x'_n, y'_m) and aligned with the x' and y' axes, and calculate the area of intersection between this rectangle and the ellipse transformed to the x'-y' coordinate system (this ellipse is uniquely determined by A and B). In our calculations we use formulae from Appendix A. Subsequently, we divide the intersection area by the area of the rectangle. The outcome yields the value of a pixel-averaged ideal digital image at the (n, m)th pixel, $f_{\text{averaged}}(x_n, y_m)$. To incorporate the effect of the PSF, we implement, for a value c of the background intensity, a discretised version of (18) in the form

$$\text{prf}(x_n, y_m) = c + \frac{1-c}{Z}\sum_{s=1}^{M}\sum_{t=1}^{N} f_{\text{averaged}}(x_s, y_t)$$
$$\times \exp\left(-\frac{(x_s - x_n)^2 + (y_t - y_m)^2}{2\sigma_{\text{PSF}}^2}\right), \tag{47}$$

where Z is a normalisation constant given by

$$Z = \sum_{s=1}^{M}\sum_{t=1}^{N} \exp\left(-\frac{(x_s - x_n)^2 + (y_t - y_m)^2}{2\sigma_{\text{PSF}}^2}\right). \tag{48}$$

The array $[\text{prf}(x_n, y_m)]_{1 \leq m \leq M,\ 1 \leq n \leq N}$ has entries between 0 and 1. Scaling each entry of this array by a conversion factor C and simulating, for each pair (m, n) independently, effects of Poisson noise with parameter $C\,\text{prf}(x_n, y_m)$ with the aid of Algorithm 1, we next obtain an array $[f_{\text{Poisson}}(x_n, y_m)]_{1 \leq m \leq M,\ 1 \leq n \leq N}$ of plausible photon counts, or a Poisson-corrupted image. Recall that the standard deviation of Poisson noise is equal to the square-root of the average number of events. Hence, when applying Poisson noise to an image, the signal-to-noise ratio is equal to

$$\text{SNR} = \frac{C\,\text{prf}(x_n, y_m)}{\sqrt{C\,\text{prf}(x_n, y_m)}} = \sqrt{C\,\text{prf}(x_n, y_m)}. \tag{49}$$

For a large choice of C, the signal-to-noise ratio will be significant, and the image will appear relatively noise free. Conversely, for small values of C, corresponding to low-light conditions, the noise will be much more prominent. To model the quantisation step of the digital image formation process, we partition the Poisson-corrupted image into G grey levels. The partitioning is achieved by grouping the intensities into discrete bins. Let b denote the half-width of a bin. For modelling convenience, we shall assume that both b and C are powers

Algorithm 1 RandomPoissonVariate(λ) [25]

Require: $\lambda > 0$
1: $L \leftarrow -\lambda$
2: $k \leftarrow 0$
3: $p \leftarrow 0$
4: condition $\leftarrow$ True
5: **while** condition **do**
6:     $k \leftarrow k + 1$
7:     $p \leftarrow p + \ln(\text{RAND})$ ▷ RAND generates a uniform number in the interval $[0, 1]$.
8:     condition $\leftarrow (p \geq L)$ ▷ Inequality is either True or False.
9: **return** $k - 1$ ▷ Result is a random sample from a Poisson distribution with mean λ.

---

of two, which ensures that the number of grey levels $G = C/(2b)$ is also a power of two. With the quantisation function

$$q(x) = \begin{cases} 1 & \text{if } -\infty < x < 2b \times 1, \\ 2 & \text{if } 2b \times 1 \leq x < 2b \times 2, \\ 3 & \text{if } 2b \times 2 \leq x < 2b \times 3, \\ \vdots \\ G-1 & \text{if } 2b \times (G-2) \leq x < 2b \times (G-1), \\ G & \text{if } 2b \times (G-1) \leq x < \infty, \end{cases} \quad (50)$$

the final digital image is given by the relation

$$f_{\text{digital}}(x_n, y_m) = 2b \times q\left(Cf_{\text{Poisson}}(x_n, y_m)\right) - b. \quad (51)$$

Our quantisation model can be interpreted as follows. The scale factor C denotes the number of photons that would yield a maximum amount of charge in a pixel and produce the brightest intensity. The interval from zero to C is partitioned into sub-intervals (bins), and generally the observed photon count is replaced by the centre value of the interval into which the photon count falls. The last interval extends into positive infinity to capture the notion of saturation. A pixel is said to be saturated if at least C photons have reached it. If the number of photons exceeds C, then any additional photons that reach the pixel will not be registered and hence effectively quantised to the same value as the maximum count C.

An illustration of the different stages of the digital image formation process is presented in Fig. 5.

C. Implementing the maximum likelihood estimator

A numerically stable implementation of formula (37) for the negative log-likelihood is presented in Algorithm 2 and Algorithm 3. We minimise the negative log-likelihood using the BFGS Quasi-Newton method, and evaluate the required gradient and approximate Hessian numerically. Among the variables that we choose to parametrise the log-likelihood with in Algorithm 2 are real-valued variables labelled $\sqrt{A}$, $\sqrt{B}$, and $\sqrt{\sigma_{\text{PSF}}}$. During the optimisation process we square $\sqrt{A}$, $\sqrt{B}$, and $\sqrt{\sigma_{\text{PSF}}}$, and ensure in that way that the values of A, B, and σ_{PSF} are non-negative. To prevent σ_{PSF} from attaining the value of zero, we add a small positive constant ϵ to $(\sqrt{\sigma_{\text{PSF}}})^2$. We assume that the scale factor C and the quantisation factor b are known or have been estimated from the data, and we do not optimise them further. Thus our overall parameter vector is $\boldsymbol{\eta} = [\sqrt{A}, \sqrt{B}, H, K, \tau, \sqrt{\sigma_{\text{PSF}}}]^\top$, with the derived vector of geometric parameters $\boldsymbol{\xi} = [(\sqrt{A})^2, (\sqrt{B})^2, H, K, \tau]^\top$.

D. Characterising the uncertainty of the estimate

To characterise the uncertainty or reliability of the ML estimate $\widehat{\boldsymbol{\xi}}$ of the geometric parameters of the region's bounding ellipse, we use

---

Algorithm 2 LikelihoodDiscreteUniformPoisson($\boldsymbol{\eta}$, f_{digital},C,b)

Require: $\boldsymbol{\eta} = [\sqrt{A}, \sqrt{B}, H, K, \tau, \sqrt{\sigma_{\text{PSF}}}]^\top$, $C \in \mathbb{N}$, $b \in \mathbb{N}$
1: $\boldsymbol{\xi} = [(\sqrt{A})^2, (\sqrt{B})^2, H, K, \tau]^\top$
2: $f_{\text{digital}} \leftarrow \text{DIGITALIMAGE}\left(\boldsymbol{\xi}, C, b, (\sqrt{\sigma_{\text{PSF}}})^2 + \epsilon\right)$ ▷ Form a candidate digital image as per Subsection VII-B.
3: $l \leftarrow 0$
4: **for** $m \leftarrow 1 \dots M$ **do**
5:     **for** $n \leftarrow 1 \dots N$ **do**
6:         $v \leftarrow f_{\text{observed}}(x_n, y_m)$ ▷ The variable $f_{\text{observed}}(x_n, y_m)$ is the observed digital image.
7:         $\lambda \leftarrow \max\left(f_{\text{digital}}(x_n, y_m), \epsilon\right)$ ▷ Precaution to avoid taking the log of zero.
8:         **for** $w \leftarrow -\min(b, v) \dots b$ **do**
9:             $z[w + \min(b, v)] \leftarrow$ ▷
            $-\lambda + (v + w)\ln\lambda - \text{GAMMALN}(w + v + 1)$
Use the natural logarithm of the gamma function.
10:         $l \leftarrow l - \ln(2b + 1) + \text{LOGSUMEXP}(z)$ ▷ Apply the log-sum-exp trick for numerically stable evaluation of the sum of exponentials.
11: **return** $-l$ ▷ Result is the negative log-likelihood.

Algorithm 3 LogSumExp(z)

Require: length-N array z
1: $a \leftarrow \max\{z[n]: n = 1 \dots N\}$ ▷ Take the largest entry.
2: $s \leftarrow 0$
3: **for** $n \leftarrow 1 \dots N$ **do**
4:     $s \leftarrow s + \exp(z[n] - a)$ ▷
$\ln(\sum_n^N e^{z_n}) = a + \ln(\sum_n^N e^{z_n - a})$.
5: **return** $a + \ln s$

---

the covariance matrix $\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\xi}}}$ of $\widehat{\boldsymbol{\xi}}$. We calculate the latter by exploiting the covariance matrix $\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\eta}}}$ of the maximum likelihood estimate $\widehat{\boldsymbol{\eta}}$. Taking into account that the covariance matrix of a ML estimate is approximately equal to the inverse Hessian of the negative log-likelihood at the estimate, we let

$$\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\eta}}} = \left(\left[\frac{\partial^2 \ell(\boldsymbol{\eta}, b, C)}{\partial \boldsymbol{\eta} \partial \boldsymbol{\eta}^\top}\right]_{\boldsymbol{\eta} = \widehat{\boldsymbol{\eta}}}\right)^{-1} \quad (52)$$

(see [26, Section 3.2]). Next, applying the rule of covariance propagation, we propagate $\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\eta}}}$ through the transformation $\boldsymbol{\eta} \mapsto \boldsymbol{\xi}$ to obtain

$$\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\xi}}} = [\partial_{\boldsymbol{\eta}} \boldsymbol{\xi}]_{\boldsymbol{\eta} = \widehat{\boldsymbol{\eta}}} \boldsymbol{\Lambda}_{\widehat{\boldsymbol{\eta}}} \left([\partial_{\boldsymbol{\eta}} \boldsymbol{\xi}]_{\boldsymbol{\eta} = \widehat{\boldsymbol{\eta}}}\right)^\top. \quad (53)$$

The Jacobian matrix of $\boldsymbol{\eta} \mapsto \boldsymbol{\xi}$ is explicitly given by

$$\partial_{\boldsymbol{\eta}} \boldsymbol{\xi} = \begin{bmatrix} 2\sqrt{A} & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\sqrt{B} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (54)$$

1) Visualising a planar confidence region: The reliability of $\widehat{\boldsymbol{\xi}}$ can alternatively be expressed in terms of a confidence region. One typically constructs a confidence region of a parameter vector estimate as a portion of the parameter space that contains the correct parameter vector with a given high probability. But since the parameter space for the totality of all ellipses is five-dimensional, a canonical confidence region is difficult to visualise and interpret. Hence, we formulate a more visually appealing form of an ellipse-specific confidence region, namely a confidence region in the plane. Such an area is meant to
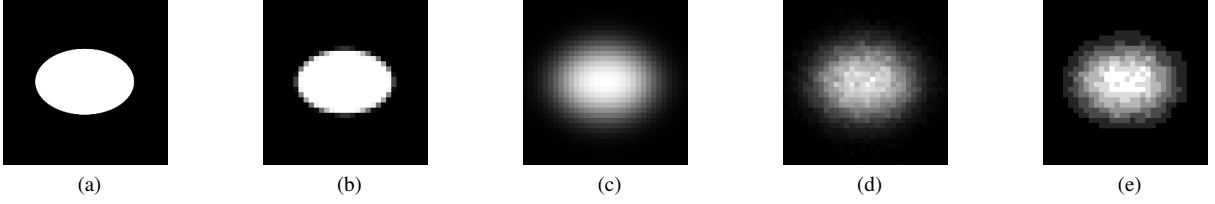
Fig. 5. Example images detailing the stages of the image formation process. (a) The ideal (unobservable) analogue image. (b) Consequence of discretisation. (c) Consequence of discretisation and the point spread function. (d) Consequence of discretisation, the point spread function and Poisson noise. (e) Consequence of discretisation, the point spread function, Poisson noise and quantisation.

cover the in-plane locus of the actual ellipse with a specified high probability.

The first to consider planar confidence regions for ellipse fits was Porrill [27]. Our approach is inspired by Scheffé's S-method for constructing simultaneous confidence bands for linear regression [28], [29, Sect. 9.4–5], and we have previously used it to establish planar confidence regions for a point-based ellipse fitting method [30]. The construction that we set forth exploits the algebraic parameters of the ellipse and in particular involves the covariance matrix of an algebraically expressed ML estimate of the ellipse. To obtain a meaningful expression for such a matrix, it is mandatory to eliminate a redundant indeterminate scale of algebraic parameters [31], [32]. We proceed with scale elimination by imposing the normalisation constraint $\|\boldsymbol{\theta}\| = 1$. Let $\boldsymbol{\kappa}$ denote the mapping $\boldsymbol{\xi} \mapsto \boldsymbol{\theta}$ defined in Eq. (42) and let $\boldsymbol{\pi}$ denote the normalisation transformation $\boldsymbol{\theta} \mapsto \|\boldsymbol{\theta}\|^{-1}\boldsymbol{\theta}$. We take $\widehat{\boldsymbol{\theta}} = \boldsymbol{\pi}(\boldsymbol{\kappa}(\widehat{\boldsymbol{\xi}}))$ for the algebraically expressed ML estimate, with $\boldsymbol{\pi}$ here guaranteeing that $\widehat{\boldsymbol{\theta}}$ is unit-normalised. Applying the rule of covariance propagation, we find that the concomitant algebraic covariance matrix $\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\theta}}}$ is given by

$$\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\theta}}} = [\partial_{\boldsymbol{\theta}}\boldsymbol{\pi}]_{\boldsymbol{\theta}=\boldsymbol{\kappa}(\widehat{\boldsymbol{\xi}})} \, [\partial_{\boldsymbol{\xi}}\boldsymbol{\kappa}]_{\boldsymbol{\xi}=\widehat{\boldsymbol{\xi}}} \, \boldsymbol{\Lambda}_{\widehat{\boldsymbol{\xi}}} \left([\partial_{\boldsymbol{\xi}}\boldsymbol{\kappa}]_{\boldsymbol{\xi}=\widehat{\boldsymbol{\xi}}}\right)^{\top} \left([\partial_{\boldsymbol{\theta}}\boldsymbol{\pi}]_{\boldsymbol{\theta}=\boldsymbol{\kappa}(\widehat{\boldsymbol{\xi}})}\right)^{\top}. \tag{55}$$

The Jacobian matrix of $\boldsymbol{\kappa}$,

$$\partial_{\boldsymbol{\xi}}\boldsymbol{\kappa} = [\partial_{\boldsymbol{\xi}}a^{\top}, \partial_{\boldsymbol{\xi}}b^{\top}, \partial_{\boldsymbol{\xi}}c^{\top}, \partial_{\boldsymbol{\xi}}d^{\top}, \partial_{\boldsymbol{\xi}}e^{\top}, \partial_{\boldsymbol{\xi}}f^{\top}]^{\top}, \tag{56}$$

is given explicitly by

$$\partial_{\boldsymbol{\xi}}a = \left[-\frac{2\cos^2\tau}{A^3}, \quad -\frac{2\sin^2\tau}{B^3}, \quad 0, \quad 0, \quad \left(\frac{1}{B^2}-\frac{1}{A^2}\right)\sin 2\tau\right],$$

$$\partial_{\boldsymbol{\xi}}b = \left[-\frac{2\sin 2\tau}{A^3}, \quad \frac{2\sin 2\tau}{B^3}, \quad 0, \quad 0, \quad 2\left(\frac{1}{A^2}-\frac{1}{B^2}\right)\cos 2\tau\right],$$

$$\partial_{\boldsymbol{\xi}}c = \left[-\frac{2\sin^2\tau}{A^3}, \quad -\frac{2\cos^2\tau}{B^3}, \quad 0, \quad 0, \quad \left(\frac{1}{A^2}-\frac{1}{B^2}\right)\sin 2\tau\right],$$

$$\partial_{\boldsymbol{\xi}}d = \left[\frac{4\cos^2\tau(H+K\tan\tau)}{A^3}, \quad \frac{4\sin\tau(H\sin\tau-K\cos\tau)}{B^3}, \right.$$
$$-\frac{2\cos^2\tau}{A^2}-\frac{2\sin^2\tau}{B^2}, \quad \left(\frac{1}{B^2}-\frac{1}{A^2}\right)\sin 2\tau,$$
$$\left.\frac{2(A-B)(A+B)(K\cos 2\tau-H\sin 2\tau)}{A^2B^2}\right],$$

$$\partial_{\boldsymbol{\xi}}e = \left[\frac{4\sin\tau(H\cos\tau+K\sin\tau)}{A^3}, \quad \frac{4\cos\tau(K\cos\tau-H\sin\tau)}{B^3}, \right.$$
$$\left(\frac{1}{B^2}-\frac{1}{A^2}\right)\sin 2\tau, \quad -\frac{2\cos^2\tau}{B^2}-\frac{2\sin^2\tau}{A^2},$$
$$\left.\frac{2(A-B)(A+B)(H\cos 2\tau+K\sin 2\tau)}{A^2B^2}\right],$$

$$\partial_{\boldsymbol{\xi}}f = \left[-\frac{2(H\cos\tau+K\sin\tau)^2}{A^3}, \quad -\frac{2(K\cos\tau-H\sin\tau)^2}{B^3}, \right.$$
$$\left.\frac{2(H+K\tan\tau)\cos^2\tau}{A^2}+\frac{2(H\sin\tau-K\cos\tau)\sin\tau}{B^2},\right.$$

$$\frac{2\sin\tau(H\cos\tau+K\sin\tau)}{A^2}+\frac{2\cos\tau(K\cos\tau-H\sin\tau)}{B^2},$$
$$\frac{2\left(B^2-A^2\right)(K\cos\tau-H\sin\tau)(H\cos\tau+K\sin\tau)}{A^2B^2}\right]. \tag{57}$$

The Jacobian of $\boldsymbol{\pi}$ is given by the more concise formula

$$\partial_{\boldsymbol{\theta}}\boldsymbol{\pi} = \|\boldsymbol{\theta}\|^{-1}\left(\mathbf{I}_6 - \|\boldsymbol{\theta}\|^{-2}\boldsymbol{\theta}\boldsymbol{\theta}^{\top}\right), \tag{58}$$

where $\mathbf{I}_6$ is the 6×6 identity matrix.

In what follows, we use the notation $\mathbf{x} = [x,y]^{\top}$ and $\mathbf{u}(\mathbf{x}) = [x^2, xy, y^2, x, y, 1]^{\top}$, with which the ellipse equation (40) can be succinctly written as $\boldsymbol{\theta}^{\top}\mathbf{u}(\mathbf{x}) = 0$. The starting point for the main construction is the observation that when $\widehat{\boldsymbol{\theta}}$ is viewed as a multivariate normally distributed random vector,

$$\widehat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}_*, \boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}), \tag{59}$$

where $\boldsymbol{\theta}_*$ is the unit-normalised parameter vector of the true ellipse and $\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}$ is a covariance matrix, the scalar random variable $\widehat{\boldsymbol{\theta}}^T\mathbf{u}(\mathbf{x})$ is normally distributed with variance $\mathbf{u}(\mathbf{x})^{\top}\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}\mathbf{u}(\mathbf{x})$ for every point $\mathbf{x}$ on the locus $\mathcal{E}_{\boldsymbol{\theta}_*} = \{\mathbf{x} \in \mathbb{R}^2 \mid \boldsymbol{\theta}_*^T\mathbf{u}(\mathbf{x}) = 0\}$ of the true ellipse. The observation is based on the fact that $\widehat{\boldsymbol{\theta}}^T\mathbf{u}(\mathbf{x}) = (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^T\mathbf{u}(\mathbf{x})$ whenever $\mathbf{x} \in \mathcal{E}_{\boldsymbol{\theta}_*}$ and the fact that by the rule of covariance propagation $(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^T\mathbf{u}(\mathbf{x})$ has variance $\mathbf{u}(\mathbf{x})^T\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}\mathbf{u}(\mathbf{x})$. Consequently, under the assumption that $\widehat{\boldsymbol{\theta}}$ is an unbiased estimate of $\boldsymbol{\theta}_*$,

$$z_{\mathbf{x}} = \frac{(\widehat{\boldsymbol{\theta}}^T\mathbf{u}(\mathbf{x}))^2}{\mathbf{u}(\mathbf{x})^{\top}\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}\mathbf{u}(\mathbf{x})} \tag{60}$$

is a squared normal random variable for every $\mathbf{x} \in \mathcal{E}_{\boldsymbol{\theta}_*}$. Each $z_{\mathbf{x}}$, insofar as $\mathbf{x}$ belongs to $\mathcal{E}_{\boldsymbol{\theta}_*}$, 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 $\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}$ 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

$$\widehat{z}_{\mathbf{x}} = \frac{(\widehat{\boldsymbol{\theta}}^T\mathbf{u}(\mathbf{x}))^2}{\mathbf{u}(\mathbf{x})^{\top}\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\theta}}}\mathbf{u}(\mathbf{x})}, \tag{61}$$

where the covariance estimate $\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\theta}}}$ serves as a natural replacement for $\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}$. Again, large observed values of $\widehat{z}_{\mathbf{x}}$ are less plausible than small observed values as long as $\mathbf{x} \in \mathcal{E}_{\boldsymbol{\theta}_*}$. It is thus natural to consider confidence regions for $\widehat{\boldsymbol{\theta}}$ in the form $\left\{\mathbf{x} \in \mathbb{R}^2 \mid \widehat{z}_{\mathbf{x}} \le c\right\}$, where c is a positive constant. Ideally, for a confidence region at (confidence) level $1 - \alpha$, we should choose c such that

$$\Pr\left(z_{\mathbf{x}} \le c \text{ for all } \mathbf{x} \in \mathcal{E}_{\boldsymbol{\theta}_*}\right) = \Pr\left(\sup_{\mathbf{x}\in\mathcal{E}_{\boldsymbol{\theta}_*}} z_{\mathbf{x}} \le c\right) = 1 - \alpha. \tag{62}$$

But the distribution of $\sup_{\mathbf{x}\in\mathcal{E}_{\boldsymbol{\theta}_*}} z_{\mathbf{x}}$ is not easy to determine, so as a second best choice we shall replace $\sup_{\mathbf{x}\in\mathcal{E}_{\boldsymbol{\theta}_*}} z_{\mathbf{x}}$ by a random upper

bound whose distribution can be readily calculated. Proceeding to the specifics, we first note that, since $\|\widehat{\boldsymbol{\theta}}\| = \|\boldsymbol{\theta}_*\| = 1$, we have $(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top \boldsymbol{\theta}_* = -\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*\|^2/2$. Consequently, resorting to the first order approximation around $\boldsymbol{\theta}_*$, we may next assume that

$$(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top \boldsymbol{\theta}_* = 0. \tag{63}$$

Given a length-n vector $\mathbf{a}$, let $\mathbf{P}_{\mathbf{a}}^{\perp}$ denote the $n \times n$ symmetric projection matrix given by

$$\mathbf{P}_{\mathbf{a}}^{\perp} = \mathbf{I}_n - \|\mathbf{a}\|^{-2}\mathbf{a}\mathbf{a}^\top. \tag{64}$$

It is readily seen that, for each length-n vector $\mathbf{x}$, $\mathbf{P}_{\mathbf{a}}^{\perp}\mathbf{x}$ represents the orthogonal projection of $\mathbf{x}$ onto the orthogonal complement of the space spanned by $\mathbf{a}$ in $\mathbb{R}^n$. Now, (63) can be restated as

$$\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_* = \mathbf{P}_{\boldsymbol{\theta}_*}^{\perp}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*). \tag{65}$$

We also note that, in view of (63),

$$\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}\boldsymbol{\theta}_* = \mathsf{E}((\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top)\boldsymbol{\theta}_* = \mathbf{0}, \tag{66}$$

where $\mathsf{E}(\mathbf{X})$ denotes the expectation of the random matrix $\mathbf{X}$. Thus the null space of $\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}$, $\mathcal{N}(\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*})$, contains $\boldsymbol{\theta}_*$. Typically, $\mathcal{N}(\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*})$ will be one-dimensional and will be spanned by $\boldsymbol{\theta}_*$. Given a matrix $\mathbf{A}$, let $\mathbf{A}^+$ denote the Moore–Penrose pseudo-inverse of $\mathbf{A}$, and, when $\mathbf{A}$ non-negative definite, let $\mathbf{A}^{1/2}$ denote the unique non-negative definite square root of $\mathbf{A}$. By a general rule, $(\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+)^{1/2}\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^{1/2} = (\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^{1/2})^+\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^{1/2}$ is a symmetric projection matrix representing the orthogonal projection onto the orthogonal complement of $\mathcal{N}(\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*})$. Assuming that $\mathcal{N}(\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*})$ is spanned by $\boldsymbol{\theta}_*$, $(\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+)^{1/2}\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^{1/2}$ is a symmetric projection matrix representing the orthogonal projection onto the orthogonal complement of the space spanned by $\boldsymbol{\theta}_*$. This is the same as saying that

$$(\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+)^{1/2}\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^{1/2} = \mathbf{P}_{\boldsymbol{\theta}_*}^{\perp}. \tag{67}$$

Now note that if $\mathbf{x} \in \mathcal{E}_{\boldsymbol{\theta}_*}$, then, by (65),

$$\widehat{\boldsymbol{\theta}}^\top \mathbf{u}(\mathbf{x}) = (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top \mathbf{u}(\mathbf{x}) = (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top \mathbf{P}_{\boldsymbol{\theta}_*}^{\perp}\mathbf{u}(\mathbf{x}) \tag{68}$$

and further, by (67),

$$\begin{aligned}\widehat{\boldsymbol{\theta}}^\top \mathbf{u}(\mathbf{x}) &= (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top (\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+)^{1/2}\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^{1/2}\mathbf{u}(\mathbf{x}) \\ &= ((\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+)^{1/2}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*))^\top \boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^{1/2}\mathbf{u}(\mathbf{x}).\end{aligned} \tag{69}$$

By the Cauchy–Bunyakovsky–Schwarz inequality,

$$(\widehat{\boldsymbol{\theta}}^\top \mathbf{u}(\mathbf{x}))^2 \leq \|(\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+)^{1/2}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)\|^2 \|\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^{1/2}\mathbf{u}(\mathbf{x})\|^2. \tag{70}$$

Also,

$$\|(\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+)^{1/2}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)\|^2 = (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top \boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*) \tag{71}$$

and

$$\|\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^{1/2}\mathbf{u}(\mathbf{x})\|^2 = \mathbf{u}(\mathbf{x})^\top \boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}\mathbf{u}(\mathbf{x}). \tag{72}$$

Hence,

$$z_{\mathbf{x}} \leq (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top \boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*). \tag{73}$$

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

$$\sup_{\mathbf{x} \in \mathcal{E}_{\boldsymbol{\theta}_*}} z_{\mathbf{x}} \leq (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top \boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*). \tag{74}$$

Now the random variable $(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)^\top \boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}^+(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*)$ has approximately a chi-squared distribution with 5 degree of freedom. Let $\chi_{5,\alpha}^2$ denote the $100(1 - \alpha)\%$ percentile of the χ^2 distribution with 5 degrees of freedom, characterised by the relation $\Pr\left(\chi^2 \leq \chi_{5,\alpha}^2\right) = 1 - \alpha$. Inequality (74) guarantees that

$$\Pr\left(\sup_{\mathbf{x} \in \mathcal{E}_{\boldsymbol{\theta}_*}} z_{\mathbf{x}} \leq \chi_{5,\alpha}^2\right) \geq 1 - \alpha. \tag{75}$$

Substituting $\boldsymbol{\Lambda}_{\widehat{\boldsymbol{\theta}}}$ for $\boldsymbol{\Lambda}_{\boldsymbol{\theta}_*}$, we also approximately have

$$\Pr\left(\sup_{\mathbf{x} \in \mathcal{E}_{\boldsymbol{\theta}_*}} \hat{z}_{\mathbf{x}} \leq \chi_{5,\alpha}^2\right) \geq 1 - \alpha. \tag{76}$$

This allows an approximate confidence region at level $1 - \alpha$ for $\widehat{\boldsymbol{\theta}}$ to be taken as $\left\{ \mathbf{x} \in \mathbb{R}^2 \mid \hat{z}_{\mathbf{x}} \leq \chi_{5,\alpha}^2 \right\}$. We finally point out that if α is set to the standard conventional value of 0.05, then $\chi_{5,\alpha}^2 = 11.07$.

VIII. Experimental results

We compared the ML estimator for ellipse fitting introduced in Section VII against the point-based direct ellipse fit (DEF points) [19] and its region-based gradient variant (DEF gradient) [18]. We obtained 2D points for DEF by applying Canny edge detection to the input image. The gradient for the gradient-based DEF was computed using the Sobel operator. We seeded the ML method with the point-based DEF. For some experimental conditions the point-based DEF was of such poor quality that the ML method converged to a sub-optimal solution. The sub-optimal solutions manifest as outliers in the boxplots of the ML results—had the ML method been seeded with a better initial value it would have converged to a superior solution. In all of our plots, we denote ML with blue (——), point-based DEF with grey (——), and gradient-based DEF with yellow (——) colours.

A. Synthetic images

In this section, we present comprehensive results on simulated data. The simulated data methodically imitates the image formation process and incorporates the effects of the point spread function, Poisson noise, discretisation error, and quantisation error. Without loss of generality, the actual ellipses which gave rise to digital images always lay within the unit box $[0, 1] \times [0, 1]$. We followed the steps outlined in Subsection VII-B to form the digital images.

The experimental design facilitates a cogent interpretation of the standard deviation of the PSF as percentage area. For example, a value of $\sigma_{\mathrm{PSF}} = 0.1$ corresponds to ten percent of the digital image area.

1) Varying signal-to-noise ratio: Based on (49), we characterised the signal-to-noise ratio as the square root of the number of photons in the brightest part of the *real* digital image f_{real}. This definition allowed us to distinguish between Poisson noise and the uncertainty introduced by quantisation. In our first experiment the true parameter vector was given by $\boldsymbol{\xi} = [0.25, 0.05, 0.5, 0.5, 0.785]^\top$. We sampled this ellipse with a square grid of 32 pixels and a Gaussian point spread function with a standard deviation of 5%. For the quantisation step, we set $b = 1$. We explored how the photon count affects the accuracy of the estimator by varying the conversion factor C in powers of two (16, 32, 64, 128, and 256). Each conversion factor induced a different signal-to-noise ratio. For each choice of C we conducted a hundred random trials. The results of these experiments are displayed in Fig. 6.

Our second experiment was identical to the first, except that we used a Gaussian PSF with a standard deviation of 15%. The results of the second experiment are displayed in Fig. 7.

Discussion: Both experiments demonstrate that prevailing ellipse estimation methods are remarkably inadequate when working at the limits of resolution. The boxplots for the DEF estimates evidence a substantial bias and demonstrate that these methods cannot recover the true ellipse parameters. The example ellipse fits communicate the deficiencies of DEF in a visually noticeable manner. In contrast the proposed ML method yields accurate estimates even with very low signal-to-noise ratios. The planar confidence regions rightly communicate the fact that there is more uncertainty in the estimate of the major axis, and that the uncertainty is greater for photon-limited images.
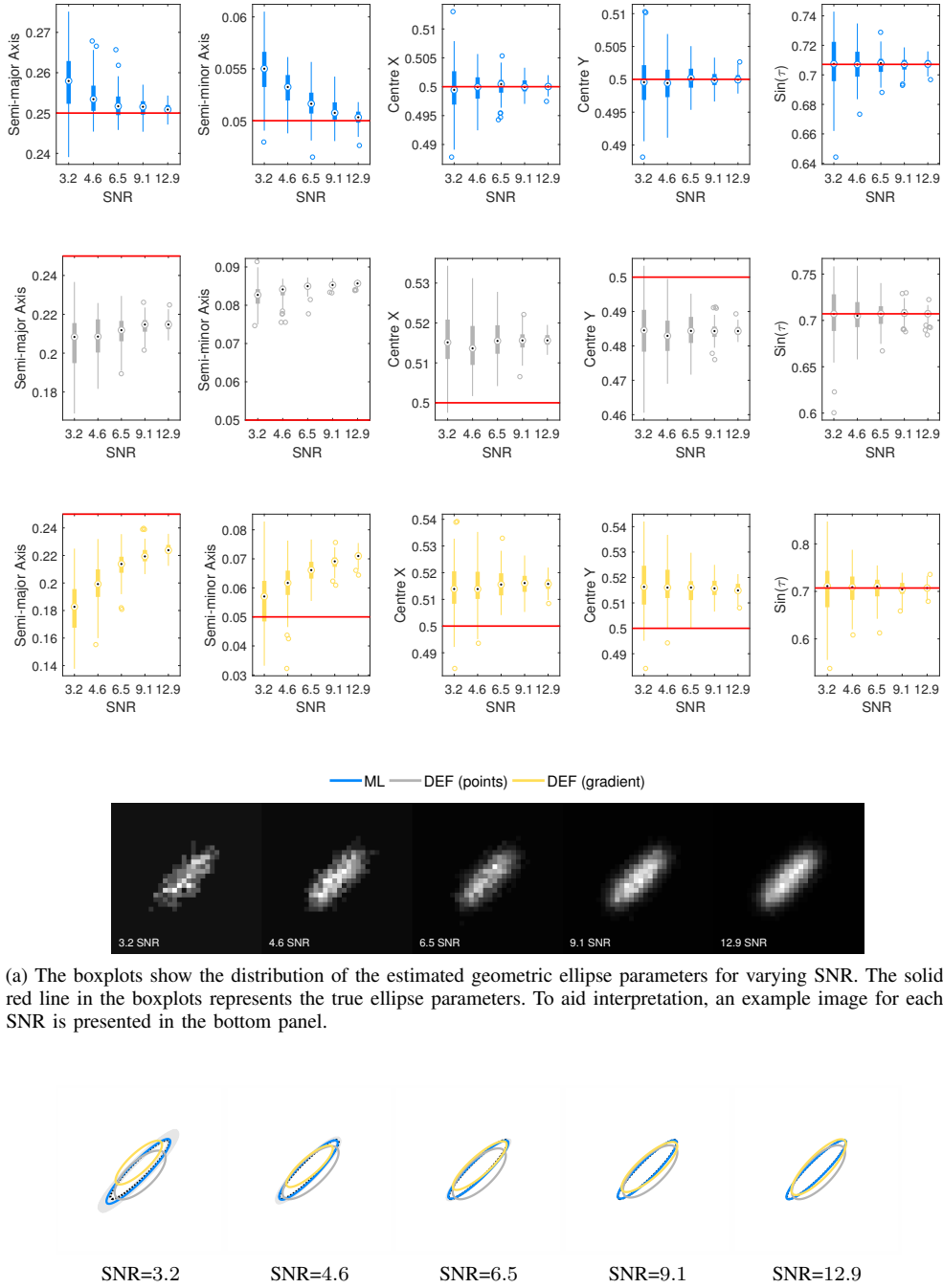
(a) The boxplots show the distribution of the estimated geometric ellipse parameters for varying SNR. The solid red line in the boxplots represents the true ellipse parameters. To aid interpretation, an example image for each SNR is presented in the bottom panel.

SNR=3.2   SNR=4.6   SNR=6.5   SNR=9.1   SNR=12.9

(b) Randomly chosen example result including a planar confidence region for each SNR level. The dotted black ellipse denotes the truth.

Fig. 6.   Estimation results on ellipse with varying signal-to-noise ratios (SNRs) sampled on a 32×32 pixel grid with $\sigma_{\mathrm{PSF}} = 0.05$ and $b = 1$.

2) Varying quantisation: For our third experiment we explored how different quantisation levels impact the precision of the estimates. The true parameter vector was given by $\boldsymbol{\xi} = [0.35, 0.15, 0.5, 0.5, 0]^{\top}$. We sampled this ellipse with a square grid of 32 pixels and a Gaussian point spread function with a standard deviation of 15%. We modified the quantisation half-width b in powers of two (2, 4, 8, 16, and 32) and conducted a hundred random trials for each value. Figure 8 summarises the outcome of the third experiment.

Discussion: The experiment validates our quantisation model. Even with extreme quantisation—a binary image—the ML method still yields estimates that are almost the correct parameters. The planar confidence regions also confirm that greater quantisation levels inflate

the uncertainty of the estimates. In comparison, both DEF estimates produce inadequate results for all quantisation levels.

3) Varying eccentricity: In general, parameter estimation of an ellipse is more challenging when the eccentricity is substantial. In the fourth series of experiments, we investigated how eccentricity affects the quality of the estimates. In particular, we generated ellipses with eccentricities ranging from 0.78 to 0.99 and sampled these ellipse with a square grid of 32 pixels and a Gaussian PSF with a standard deviation of 5%. For the quantisation step, we set $b = 1$. Figure 9 summarises our findings.

Discussion: The results indicate that both DEF methods significantly and systematically underestimate the length of the semi-major
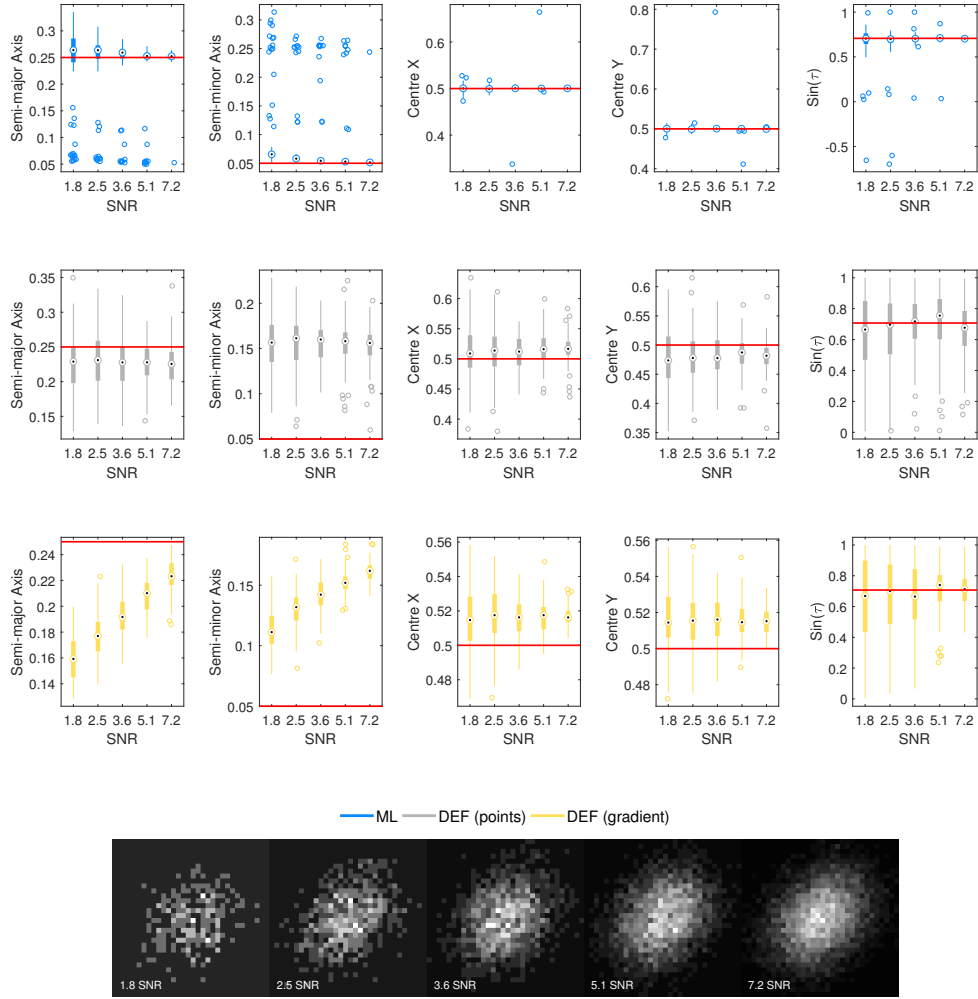
(a) The boxplots show the distribution of the estimated geometric ellipse parameters for varying SNRs. The solid red line in the boxplots represents the true ellipse parameters. To aid interpretation, an example image for each SNR is presented in the bottom panel.
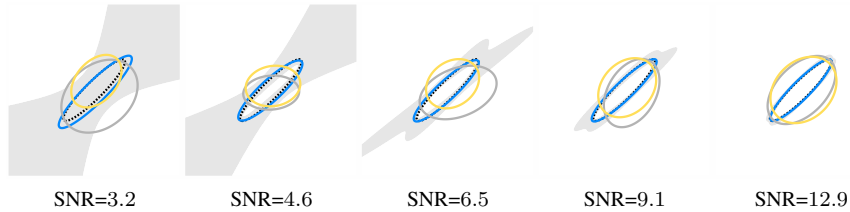
(b) Randomly chosen example result including a planar confidence region for each SNR level. The dotted black ellipse denotes the truth.

Fig. 7. Estimation results on ellipse with varying signal-to-noise ratios (SNR) sampled on a 32×32 pixel grid with $\sigma_{\mathrm{PSF}} = 0.15$ and $b = 1$.

axis. The bias is even more prominent for high-eccentricity ellipses. Contrastingly, the ML method produces accurate results for ellipses with high or low eccentricity. The planar confidence regions indicate that the semi-major axis is less certain for high-eccentricity ellipses.

4) Varying sampling grid: In our final set of experiments, we investigated how image resolution influences the precision of the estimates. Specifically, we varied the sampling grid in powers of two (8×8, 16×16, 32×32, 64×64, and 128×128). The true parameter vector was given by $\boldsymbol{\xi} = [0.35, 0.15, 0.5, 0.5, 0]^{\top}$. We used a Gaussian point spread function with a standard deviation of 15% which produced a SNR ranging from 4.1977 to 4.4527. For quantisation, we set $b = 1$. Figure 10 illustrates the results.

Discussion: The final set of simulations affirm the necessity and efficacy of our maximum likelihood model. Even at the limits of resolution, using an 8×8 pixel grid, the ML method produces plausible parameter estimates. In accordance with expectation, the planar confidence regions demonstrate greater parameter uncertainty for low-resolution pixel grids than for higher resolution grids. The performance of the DEF methods is poor. Evidently, the DEF methods are not applicable for these types of low-resolution images.

B. Real images

To corroborate the conclusions of the synthetic data experiments we conducted further laboratory experiments with real images. We used
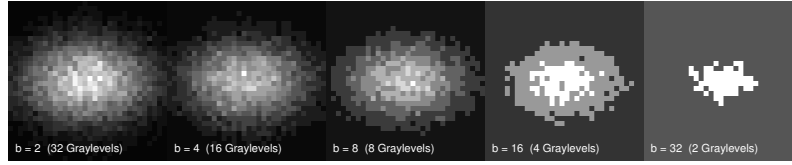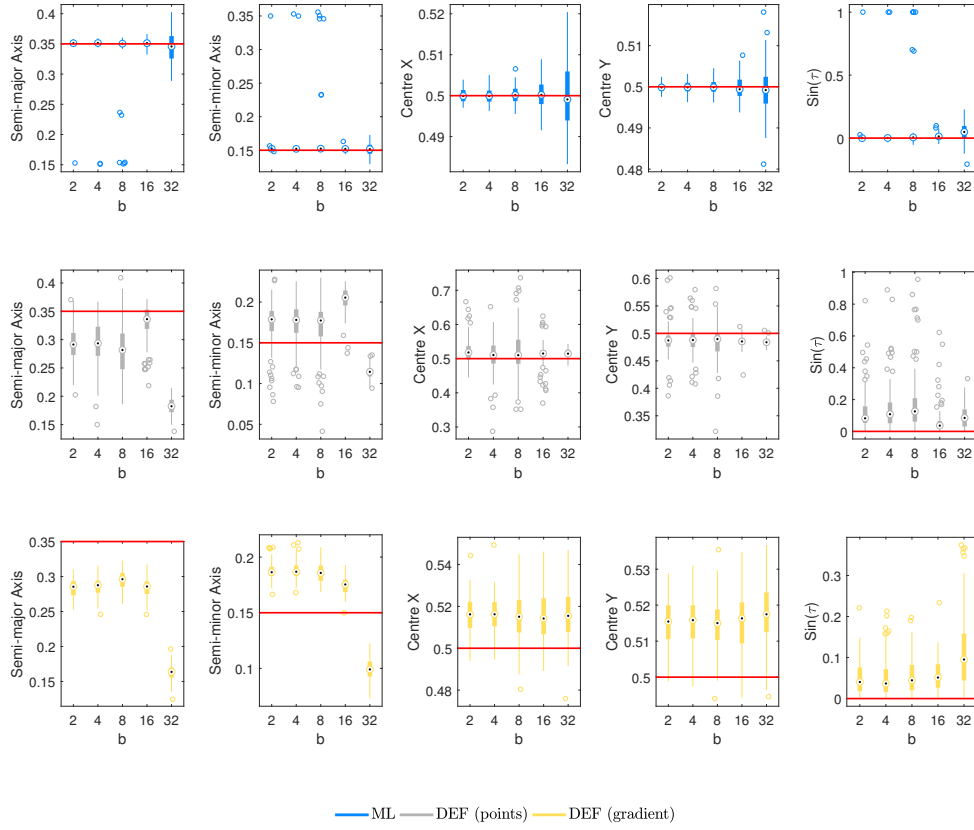
(a) The boxplots show the distribution of the estimated geometric ellipse parameters for varying quantisation levels. The solid red line in the boxplots represents the true ellipse parameters. To aid interpretation an example image for each quantisation level b is presented in the bottom panel together with a description of the final number of grey levels.
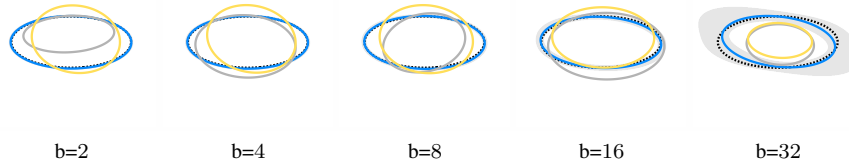
(b) Randomly chosen example result including a planar confidence region for each quantisation level. The dotted black ellipse denotes the truth.

Fig. 8. Estimation results on ellipses with varying quantisation levels sampled on a 32×32 pixel grid with $\sigma_{\mathrm{PSF}} = 0.15$ and SNR equal to 8.8683.

the UI-1220LE-M-GL camera (IDS Imaging Development Systems GmbH) and attached the MP0814-MP2 (IDS Imaging Development Systems GmbH) lens to the camera. The camera has a global shutter and an 8-bit monochrome CMOS sensor with a resolution of 752×480 pixels. We constructed a real ellipse region by glueing a white, elliptic sticker onto a piece of black cardboard.

1) Experimental design: The software for the camera permits the configuration of various low-level settings, such as exposure time, gain, black-level offset, and quantisation, to name but a few. We set the gamma correction factor to unity to guarantee a linear luminance output.

To ensure that we obtained a reasonable approximation of the exact ellipse parameters from the real images, we carefully adjusted the lens and configured the camera to produce sharp and clear images. We then recorded a sequence of 240 images and took the average geometric ellipse parameters estimated by our ML method as the best guess for the exact parameters. We substantiated our methodology by noting that the variance of the 240 estimated parameters was negligible and that the DEF methods produced similar estimates on the sharp and clear images.

The black cardboard was not absolutely black, and so, unlike our synthetic experiments, we did not estimate elliptic regions using a
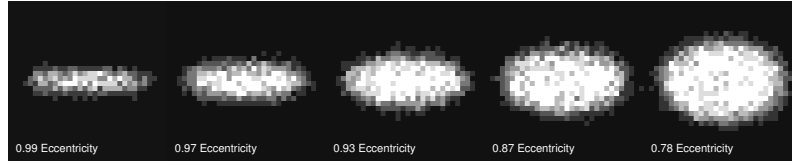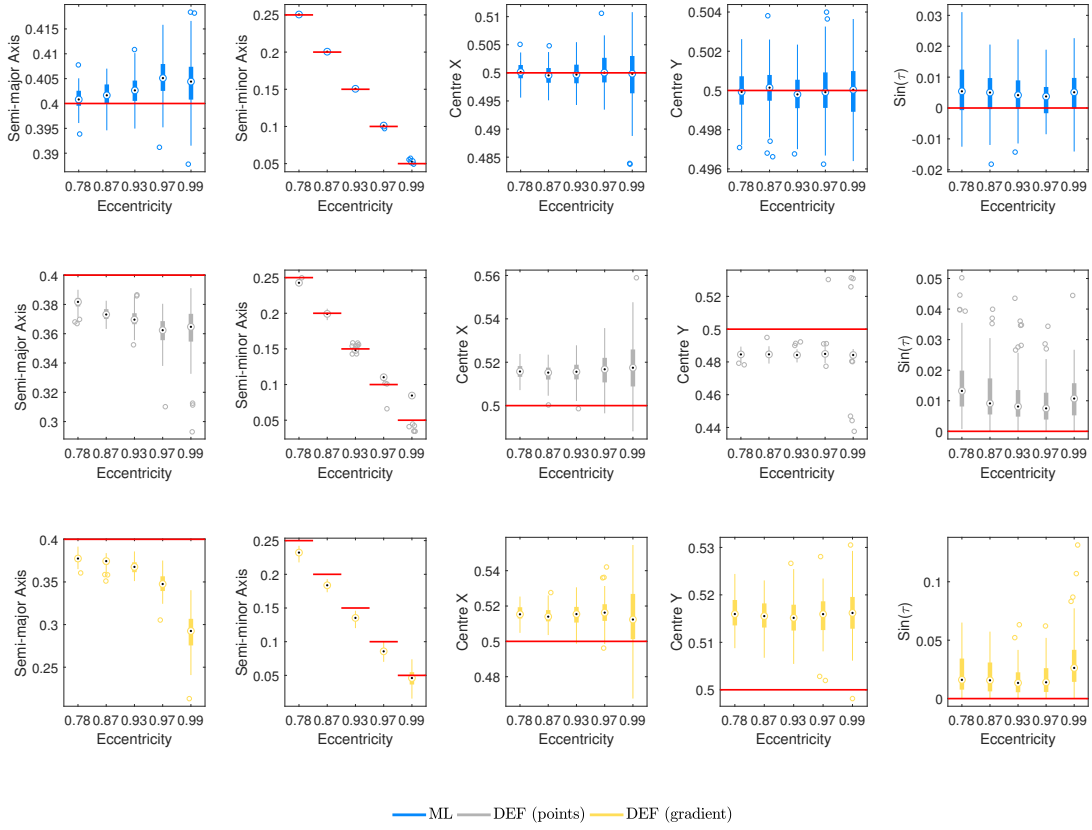
(a) The boxplots show the distribution of the estimated geometric ellipse parameters for varying eccentricities. The solid red line in the boxplots represents the true ellipse parameters. To aid interpretation, an example image for each eccentricity is presented in the bottom panel.
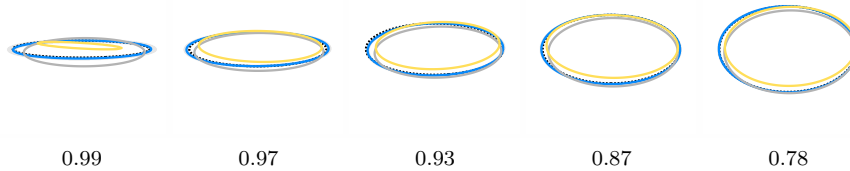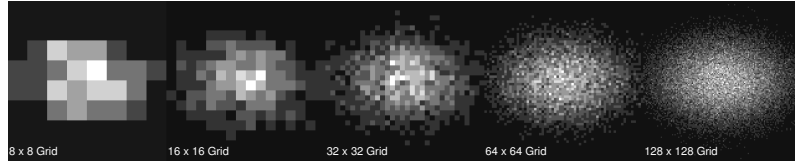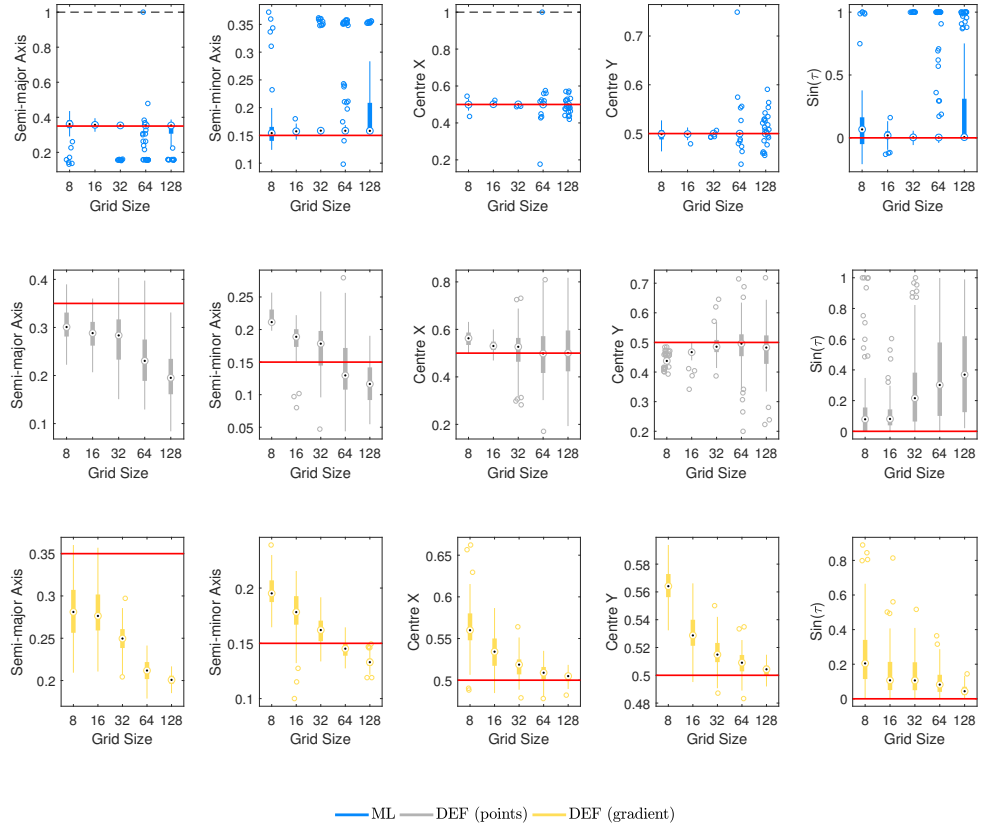

(b) Randomly chosen example result including a planar confidence region for each eccentricity. The dotted black ellipse denotes the truth. The planar confidence regions encompass the truth tightly reflecting that there is not much uncertainty. They are not visible without zooming in.

Fig. 9. Estimation results on ellipses of varying eccentricity sampled on a 32×32 pixel grid with $\sigma_{\mathrm{PSF}} = 0.05$ and SNR ranging from 4.5200 to 5.6568.

model of a black background. Instead, we used the model for a grey background described in Subsection V-B. Upon inspecting the histograms of a series of images, which revealed grey values in the range from 10 to 30 for the black cardboard, we set the background intensity value to $c = 0.15$.

Another important aspect when working with real images is finding an appropriate value for the conversion factor C which links the image intensity with the photo-electron count (see Section IV). The value of C determines the level of Poisson noise. In practice, we constrain C to be a multiple of the maximum intensity in the image. The intuition underpinning this constraint is that we first need to

adjust our model image which lies in the unit interval so that its brightest value (a value of 1) matches the brightest observed intensity in the actual image (a value between 0 and 255). Subsequently, we need to convert the intensities into plausible photon counts. If the image is dark and noisy, then we multiply by a small positive integer to model a photon-limited scenario. If it looks relatively noise free, then we can multiply by a more significant positive integer. Since our model can generate a synthetic image, finding a suitable value for C is not too complicated. A wrong choice of C will result in a synthetic image that either looks too noisy or not noisy enough. The correct choice of C will produce an image that resembles the observed

(a) The boxplots show the distribution of the estimated geometric ellipse parameters for varying sampling grids. The solid red line in the boxplots represents the true ellipse parameters. A circle against a dashed black line in a boxplots indicates that a particular value exceeded the plot limit of the y axis. To aid interpretation, an example image for each grid size is presented in the bottom panel.

8×8 16×16 32×32 64×64 128×128
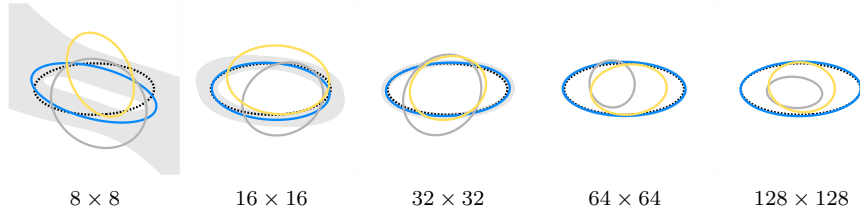
(b) Randomly chosen example result including a planar confidence region for each sampling grid size. The dotted black ellipse denotes the truth.

Fig. 10. Estimation results on ellipses sampled on varying pixel grids with $\sigma_{\mathrm{PSF}} = 0.15$ and SNR ranging from 4.1977 to 4.4527.

image. Apart from choosing an appropriate value of C by qualitatively comparing the synthetic images against the actual images, one could also quantify the root-mean-square error between the synthesised and actual image. Furthermore, one could develop a particular calibration step to identify the correct conversion factor. We opted to set C based on empirical observations and settled on a value of $C = 25 \times G$, where G is the maximum grey value in a given image.

By altering the configuration properties of the camera and adjusting the lens we were able to replicate many of the synthetic image experiments. For each experimental condition, we recorded a series of 240 images and used these to test the performance of the algorithms.

2) Experiments: We quantified the performance of the estimators by considering the algebraic ellipse parameters. The fidelity of the algebraic parameters was evaluated by using an *algebraic parameter error*, defined as $\left\| \mathbf{P}_{\boldsymbol{\theta}_*}^{\perp} \widehat{\boldsymbol{\theta}} \right\|$, where $\boldsymbol{\theta}_*$ denotes the true value, and both $\boldsymbol{\theta}_*$ and $\widehat{\boldsymbol{\theta}}$ are assumed to have unit norm.

Experiment 1: In our first set of experiments, we adjusted the camera lens so that the target image was out of focus and blurred. We cropped a 64×64 region of interest that contained the ellipse region and used it as input to our estimators. We initialised the Gaussian PSF with a standard deviation of $.5\%$ and set b equal to 0. The results are displayed in Fig. 11.
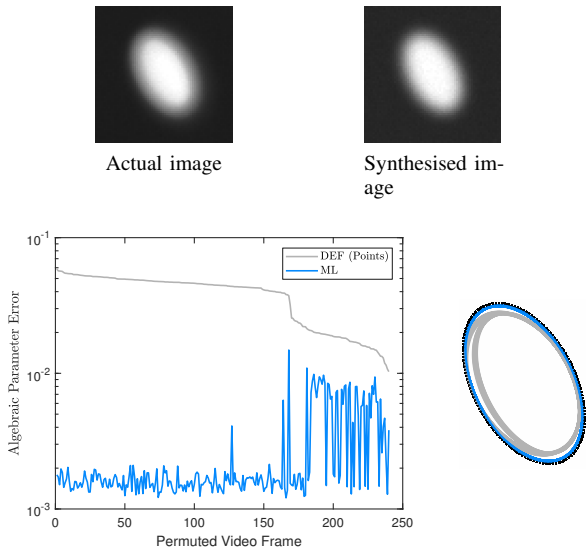
Actual image

Synthesised image

Fig. 11. Experiment 1. Estimation results on a sequence of real images (64×64 pixels) where the lens was defocused to induce a blurred image. We permuted the sequence of images in the bottom left panel so that the error of the point-based direct ellipse fit is sorted in descending order. The bottom right panel shows the estimates for the entire sequence overlaid on a single figure. The dotted black line demarcates the true ellipse.
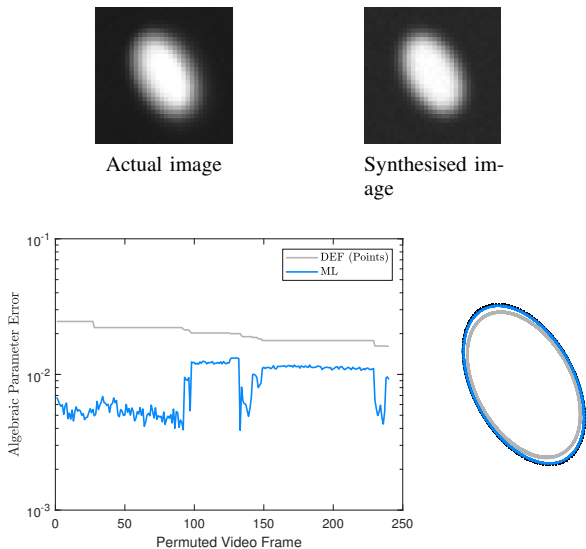

Actual image

Synthesised image

Fig. 12. Experiment 2. Estimation results on a sequence of real images (32×32 pixels) where the lens was defocused to induce a blurred image. We permuted the sequence of images in the bottom left panel so that the error of the point-based direct ellipse fit is sorted in descending order. The bottom right panel shows the estimates for the entire sequence overlaid on a single figure. The dotted black line demarcates the true ellipse.

Experiment 2: The second experiment was identical to the first, except that we configured the camera to downsample the resolution by a half. After cropping the downsampled image, we obtained a 32×32 square grid of pixels that encapsulated the ellipse region. We initialised the Gaussian PSF with a standard deviation of 1% and set b equal to 0. The results are displayed in Fig. 12.

Experiment 3: The third experiment was also identical to the first, except that we configured the camera to downsample the resolution by
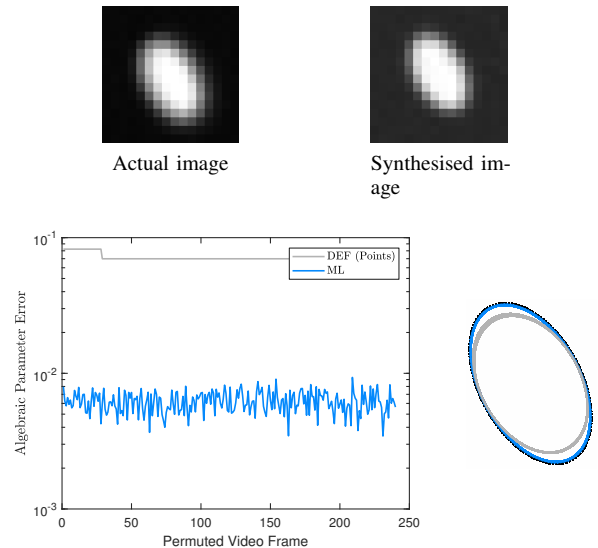

Actual image

Synthesised image

Fig. 13. Experiment 3. Estimation results on a sequence of real images (16×16 pixels) where the lens was defocused to induce a blurred image. We permuted the sequence of images in the bottom left panel so that the error of the point-based direct ellipse fit is sorted in descending order. The bottom right panel shows the estimates for the entire sequence overlaid on a single figure. The dotted black line demarcates the true ellipse.

a quarter. Downsampling and cropping produced a 16×16 pixel grid of the ellipse region. We initialised the Gaussian PSF with a standard deviation of 2% and set b equal to 0. The results are displayed in Fig. 13.

Experiment 4: In the fourth experiment, we repeated the first experiment but this time configured the camera to quantise the luminance to 5 bits (32 grey levels). We initialised the Gaussian PSF with a standard deviation of $.5\%$ and set b equal to 4. The results are displayed in Figure Fig. 14.

Experiment 5: The fifth experiment mirrored the second experiments, except that we also configured the camera to quantise the luminance to 5 bits (32 grey levels). We initialised the Gaussian with a standard deviation of 1% and set b equal to 4. The results are displayed in Fig. 15.

Experiment 6: The sixth experiment mirrored the third experiments, except that we also configured the camera to quantise the luminance to 5 bits (32 grey levels). We initialised the Gaussian PSF with a standard deviation of 2% and set b equal to 4. The results are displayed in Fig. 16.

IX. DISCUSSION

The experiments that we conducted on real imagery have further demonstrated the correctness and versatility of our statistical model. It is remarkable that for each experiment, the synthetic image associated with the maximum likelihood solution is visually almost indistinguishable from the real picture. Evidently, our mathematical development strikes the correct balance between tractability and authenticity.

For each experiment, our maximum likelihood method outperformed the point-based direct ellipse fit by several orders of magnitude. The variance of the ML estimator is also substantially less than the point-based estimates. The stability of the ML estimate is apparent in the overlayed ellipse plots. Substantially only a single blue ellipse (ML) is evident for each experiment in contrast to numerous grey curves (DEF).
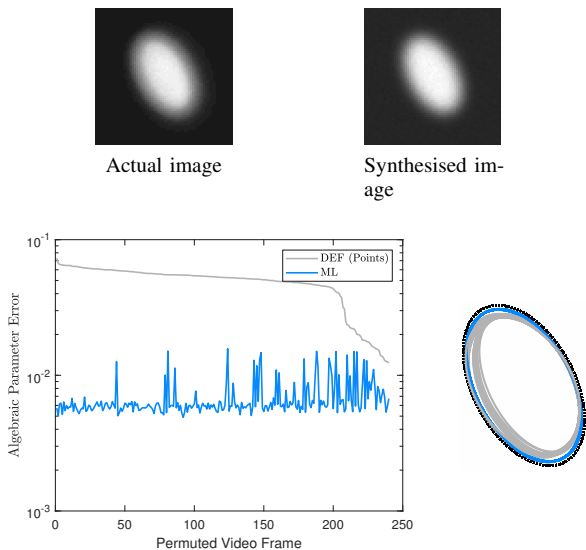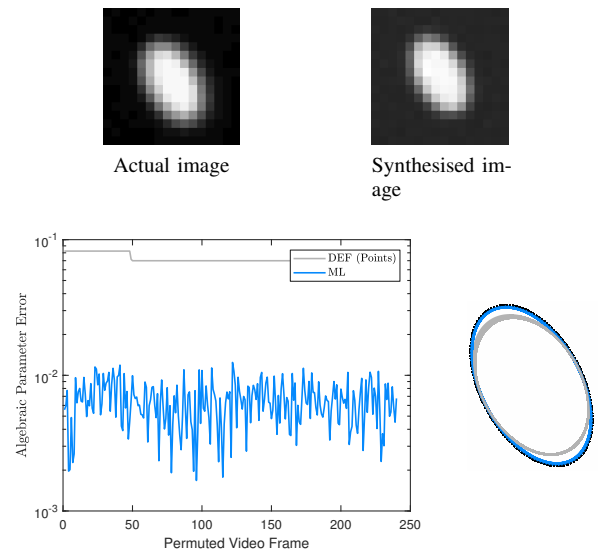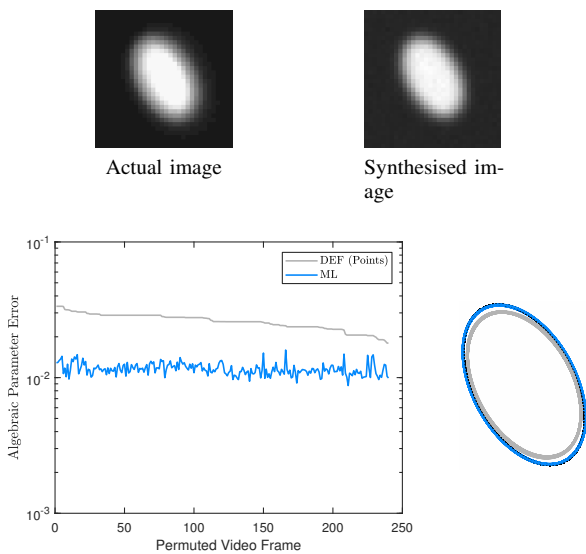
Fig. 14. Experiment 4. Estimation results on a sequence of real quantised images (64×64 pixels and 32 grey levels) where the lens was defocused to induce a blurred image. We permuted the sequence of images in the bottom left panel so that the error of the point-based direct ellipse fit is sorted in descending order. The bottom right panel shows the estimates for the entire sequence overlaid on a single figure. The dotted black line demarcates the true ellipse.

Fig. 16. Experiment 6. Estimation results on a sequence of real quantised images (16×16 pixels and 32 grey levels) where the lens was defocused to induce a blurred image. We permuted the sequence of images in the bottom left panel so that the error of the point-based direct ellipse fit is sorted in descending order. The bottom right panel shows the estimates for the entire sequence overlaid on a single figure. The dotted black line demarcates the true ellipse.

Fig. 15. Experiment 5. Estimation results on a sequence of real quantised images (32×32 pixels and 32 grey levels) where the lens was defocused to induce a blurred image. We permuted the sequence of images in the bottom left panel so that the error of the point-based direct ellipse fit is sorted in descending order. The bottom right panel shows the estimates for the entire sequence overlaid on a single figure. The dotted black line demarcates the true ellipse.

X. CONCLUSION

We have developed and tested a coherent mathematical framework for estimating the parameters of a planar shape from a single low-resolution, photon-limited digital image. Our work unifies the uncertainty due to discretisation, photon noise, and quantisation into a unique manageable statistical model. We have presented a careful and meticulous exposition of each component of the model. Comprehensive experiments on real and synthetic data have also demonstrated the groundbreaking accuracy of our approach. The ideas presented in this report provide new foundations for working on image processing problems at the limits of resolution. Our future work will focus on generalising the method to other more complicated shapes, with one possible approach being the use of level-set methods and dynamic implicit surfaces. The main problem to resolve is how to compute the area of intersection of a pixel with a particular shape.

APPENDIX A: AREA OF INTERSECTION OF AN ELLIPSE AND A RECTANGLE

The problem of determining the area of intersection between a rectangle and an ellipse in the case that the sides of the rectangle are parallel to the semi-axes of the ellipse was addressed in 1963 by Groves [33] in the military context of devising mathematical methods for the evaluation of small arms. Our exposition of the solution, including several diagrams, is based on Groves' systematic account.

We shall assume that the ellipse is described by the ellipse equation in standard form

$$\left(\frac{x}{A}\right)^2 + \left(\frac{y}{B}\right)^2 = 1, \tag{A1}$$

where A and B represent the ellipse's semi-major and semi-minor axis lengths, respectively. Furthermore, we shall assume that the rectangle is centred at a point $(\bar{x}, \bar{y})$ and has width w and height h (see Fig. 17). Groves' solution for computing the intersection area between an arbitrary rectangle and a standard ellipse involves partitioning the rectangle into sub-rectangles A_i ($i = 1 \ldots 4$) such that each sub-rectangle is entirely contained in one of the four quadrants of the Cartesian coordinate system. Then, for each part, one constructs an equivalent first-quadrant rectangle A_i^1 and calculates its area of intersection, S_i, with the ellipse (see Fig. 18). If S denotes the total area of intersection for the original rectangle, then $S = \sum_{i=1}^{4} S_i$. Each first quadrant rectangle A_i^1 will be specified by four non-negative
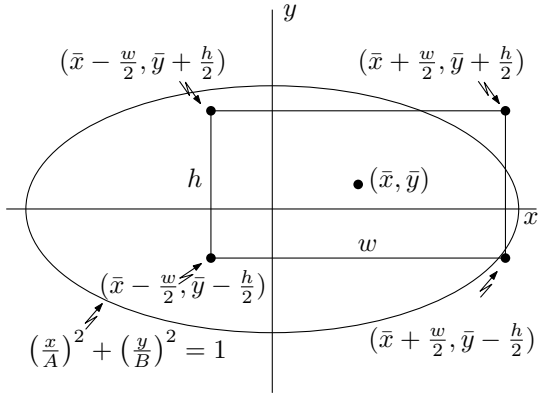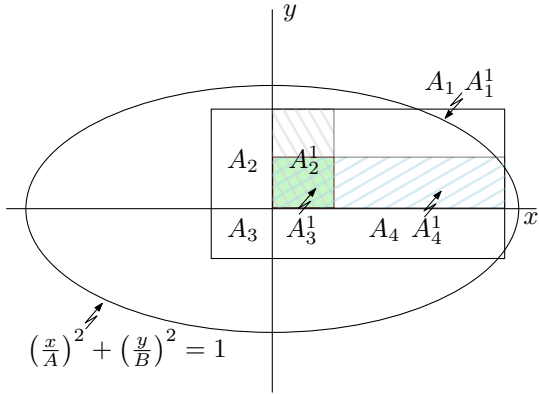
Fig. 17. General situation.

Fig. 18. A partitioning of the rectangle into four sub-rectangles A_1, A_2, A_3, and A_4 together with equivalent rectangles A_1^1, A_2^1, A_3^1, and A_4^1 all located in the first quadrant of the Cartesian coordinate system.

numbers (a_i, b_i, c_i, d_i), where (a_i, b_i) are the coordinates of the vertex of A_i^1 closest to the origin, and c_i is the width and d_i is the height of A_i^1 in x and y directions, respectively (see Fig. 19 for an example). Groves derived formulae for (a_i, b_i, c_i, d_i) in terms of the original rectangle by enumerating the different ways in which the rectangle can span the four quadrants. There are nine possible cases: (1–4) the rectangle is completely in one of the four quadrants; (5) partly in quadrant I and II; (6) partly in quadrant II and III; (7) partly in quadrant III and IV; (8) partly in quadrant IV and I; and (9) one vertex of the rectangle is in each of the four quadrants. These nine
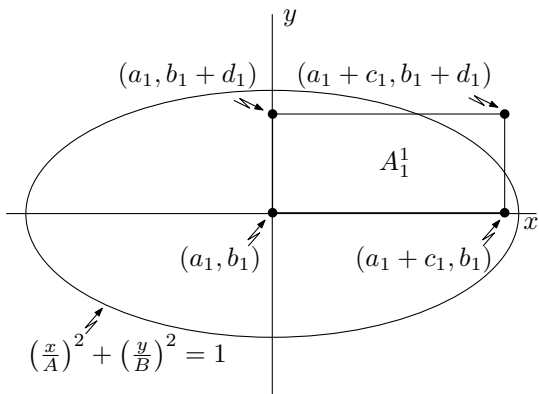
Fig. 19. Representation of A_1^1.

cases are all simultaneously handled by the following formulae:

$$A_1^1: \qquad a_1 = \max\left\{0, \bar{x} - \frac{w}{2}\right\},$$
$$b_1 = \max\left\{0, \bar{y} - \frac{h}{2}\right\},$$
$$c_1 = \max\left\{0, \bar{x} + \frac{w}{2} - a_1\right\},$$
$$d_1 = \max\left\{0, \bar{y} + \frac{h}{2} - b_1\right\}; \qquad \text{(A2)}$$

$$A_2^1: \qquad a_2 = \max\left\{0, -\bar{x} - \frac{w}{2}\right\},$$
$$b_2 = \max\left\{0, \bar{y} - \frac{h}{2}\right\},$$
$$c_2 = \max\left\{0, -\bar{x} + \frac{w}{2} - a_2\right\},$$
$$d_2 = \max\left\{0, \bar{y} + \frac{h}{2} - b_2\right\}; \qquad \text{(A3)}$$

$$A_3^1: \qquad a_3 = \max\left\{0, -\bar{x} - \frac{w}{2}\right\},$$
$$b_3 = \max\left\{0, -\bar{y} - \frac{h}{2}\right\},$$
$$c_3 = \max\left\{0, -\bar{x} + \frac{w}{2} - a_3\right\},$$
$$d_3 = \max\left\{0, -\bar{y} + \frac{h}{2} - b_3\right\}; \qquad \text{(A4)}$$

$$A_4^1: \qquad a_4 = \max\left\{0, \bar{x} - \frac{w}{2}\right\},$$
$$b_4 = \max\left\{0, -\bar{y} - \frac{h}{2}\right\},$$
$$c_4 = \max\left\{0, \bar{x} + \frac{w}{2} - a_4\right\},$$
$$d_4 = \max\left\{0, -\bar{y} + \frac{h}{2} - b_4\right\}. \qquad \text{(A5)}$$

which can also be expressed as a function of i,

$$A_i^1: \qquad a_i = \max\left\{0, (-1)^{\frac{1}{2}(i^2-1)}\bar{x} - \frac{w}{2}\right\},$$
$$b_i = \max\left\{0, (-1)^{\frac{1}{2}(i^2+i-2)}\bar{y} - \frac{h}{2}\right\},$$
$$c_i = \max\left\{0, (-1)^{\frac{1}{2}(i^2-i)}\bar{x} + \frac{w}{2} - a_i\right\},$$
$$d_i = \max\left\{0, (-1)^{\frac{1}{2}(i^2+i-2)}\bar{y} + \frac{h}{2} - b_i\right\}. \qquad \text{(A6)}$$

If the original rectangle does not overlap with the ith quadrant, then A_i^1 will reduce to a line segment with no area (either c_i or d_i will be zero) and the area of intersection S_i will be zero.

Dropping the subscripts, we now focus exclusively on deriving formulae for the intersection area of a rectangle in the first quadrant. Let the four vertices of the rectangle be indexed in the following manner according to their coordinates:

$$v_2 : (a, b+d), \quad v_3 : (a+c, b+d),$$
$$v_1 : (a, b), \qquad v_4 : (a+c, b). \qquad \text{(A7)}$$

There are six distinct intersection cases that need to be considered, depending on which vertices are inside the ellipse. These are:

I. no vertices inside the ellipse,
II. v_1 inside; v_2, v_3 and v_4 outside,
III. v_1 and v_4 inside; v_2 and v_3 outside,
IV. v_1 and v_2 inside; v_3 and v_4 outside,
V. v_1, v_2 and v_4 inside; v_3 outside
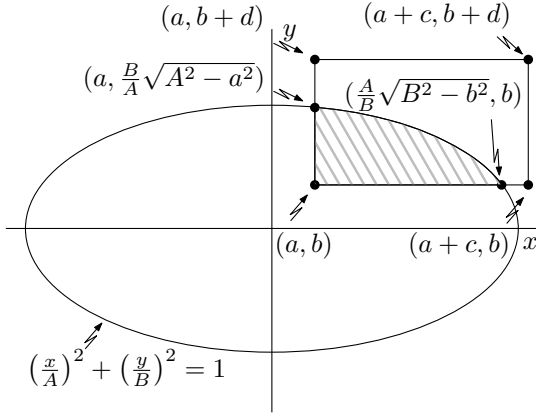**VI.** all vertices inside the ellipse.

Fig. 20. Case II: v_1 inside, and v_2, v_3, and v_4 outside.
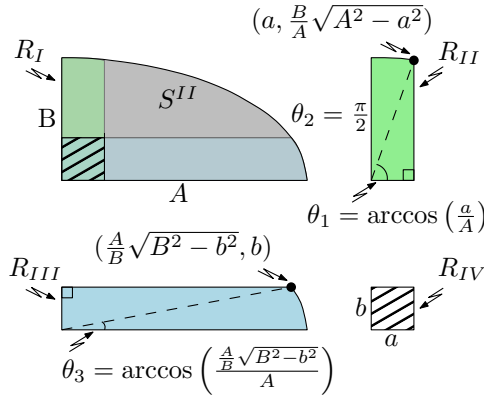

Fig. 21. Case II. The area of intersection $S^{II} = R_I - R_{II} - R_{III} + R_{IV}$.

a) *Case I:* This case is identified by the condition

$$\left(\frac{a}{A}\right)^2 + \left(\frac{b}{B}\right)^2 \geq 1 \tag{A8}$$

which indicates that the vertex closest to the origin, v_1, is outside the ellipse. Consequently, the area of intersection, denoted by S^I, is zero.

b) *Case II:* The conditions required to identify this case are

$$\left(\frac{a}{A}\right)^2 + \left(\frac{b}{B}\right)^2 < 1 \qquad (v_1 \text{ inside}), \tag{A9}$$

$$\left(\frac{a}{A}\right)^2 + \left(\frac{b+d}{B}\right)^2 \geq 1 \qquad (v_2 \text{ outside}) \tag{A10}$$

and

$$\left(\frac{a+c}{A}\right)^2 + \left(\frac{b}{B}\right)^2 \geq 1 \qquad (v_4 \text{ outside}); \tag{A11}$$

see Fig. 20. If we partition the first quadrant into four regions as illustrated in Fig. 21, then the area of intersection is given by

$$S^{II} = R_I - R_{II} - R_{III} + R_{IV}$$
$$= \underbrace{\left(\frac{\pi}{4}AB\right)}_{R_I} - \underbrace{\left(\frac{(\theta_2 - \theta_1)AB}{2} + \frac{aB}{2A}\sqrt{A^2 - a^2}\right)}_{R_{II}}$$
$$- \underbrace{\left(\frac{\theta_3 AB}{2} + \frac{bA}{2B}\sqrt{B^2 - b^2}\right)}_{R_{III}} + \underbrace{\left(ab\right)}_{R_{IV}} \tag{A12}$$

$$= \frac{\pi}{4}AB + ab$$
$$- \underbrace{\left(\frac{AB}{2}\arcsin\left(\frac{a}{A}\right) + \frac{aB}{2}\sqrt{1 - \left(\frac{a}{A}\right)^2}\right)}_{R_{II}}$$
$$- \underbrace{\left(\frac{\pi}{4}AB - \frac{AB}{2}\arcsin\left(\sqrt{1 - \left(\frac{b}{B}\right)^2}\right)\right)}_{R_{III} \cdots}$$
$$+ \underbrace{\frac{bA}{2}\sqrt{1 - \left(\frac{b}{B}\right)^2}}_{\cdots} \tag{A13}$$

$$= \frac{AB}{2}\left(\arcsin\left(\sqrt{1 - \left(\frac{b}{B}\right)^2}\right) - \arcsin\left(\frac{a}{A}\right)\right.$$
$$\left. - \frac{a}{A}\sqrt{1 - \left(\frac{a}{A}\right)^2} - \frac{b}{B}\sqrt{1 - \left(\frac{b}{B}\right)^2} + 2\left(\frac{a}{A}\right)\left(\frac{b}{B}\right)\right). \tag{A14}$$

In (A12) regions R_{II} and R_{III} are each partitioned into the sum of two terms: the area of an ellipse sector (first term) and the area of a right-angled triangle (second term). The angles θ_k ($k = 1, 3$) that are formed between the x axis and corresponding points (x_k, y_k) on the ellipse are found from the first of the following parametric ellipse equations:

$$x = A\cos\theta \implies \theta = \arccos\frac{x}{A},$$
$$y = B\sin\theta \implies \theta = \arcsin\frac{y}{B}. \tag{A15}$$

In (A13) we use the complementary angle relation, $\arccos x = \frac{\pi}{2} - \arcsin x$, to write the ellipse sector area in R_{II} and R_{III} in terms of the arcsine function. For further simplification, let

$$\theta = \arcsin\left(\sqrt{1 - \left(\frac{b}{B}\right)^2}\right) - \arcsin\left(\frac{a}{A}\right). \tag{A16}$$

Then

$$\sin\theta = \sqrt{1 - \left(\frac{b}{B}\right)^2}\sqrt{1 - \left(\frac{a}{A}\right)^2} - \left(\frac{a}{A}\right)\left(\frac{b}{B}\right) \tag{A17}$$

or

$$\theta = \arcsin\left(\sqrt{1 - \left(\frac{b}{B}\right)^2}\sqrt{1 - \left(\frac{a}{A}\right)^2} - \left(\frac{a}{A}\right)\left(\frac{b}{B}\right)\right). \tag{A18}$$

In (A17) we used the angle-difference identity for sine followed by the Pythagorean trigonometric identity. In conclusion,

$$S^{II} = \frac{AB}{2}F\left(\frac{a}{A}, \frac{b}{B}\right), \tag{A19}$$

where

$$F(U, V) = \arcsin\left(\sqrt{1 - U^2}\sqrt{1 - V^2} - UV\right)$$
$$- U\sqrt{1 - U^2} - V\sqrt{1 - V^2} + 2UV. \tag{A20}$$

c) *Case III:* The situation is identified by the conditions

$$\left(\frac{a+c}{A}\right)^2 + \left(\frac{b}{B}\right)^2 < 1 \qquad (v_4 \text{ inside}) \tag{A21}$$

and

$$\left(\frac{a}{A}\right)^2 + \left(\frac{b+d}{B}\right)^2 \geq 1 \qquad (v_2 \text{ outside}). \tag{A22}$$
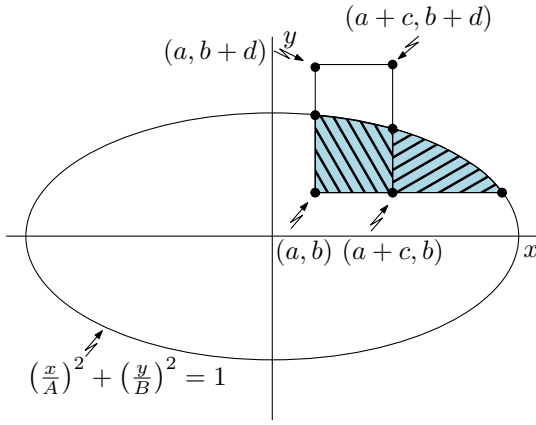
Fig. 22. Case III: v_1 and v_2 inside, and v_3 and v_4 outside. The area of intersection equals the joint area of the falling and rising striped region minus the rising striped region. Both these regions are captured by case II.

The illustration Fig. 22 suggests that the area S^{III} is simply the difference between two areas of the type considered in the second case. Thus

$$S^{III} = \frac{AB}{2} \left(F\left(\frac{a}{A}, \frac{b}{B}\right) - F\left(\frac{a+c}{A}, \frac{b}{B}\right) \right). \tag{A23}$$

d) Case IV: The conditions required to identify this case are

$$\left(\frac{a}{A}\right)^2 + \left(\frac{b+d}{B}\right)^2 < 1 \qquad (v_2 \text{ inside}) \tag{A24}$$

and

$$\left(\frac{a+c}{A}\right)^2 + \left(\frac{b}{B}\right)^2 \geq 1 \qquad (v_4 \text{ outside}). \tag{A25}$$

This area is also difference between two areas of the type considered in Case II:

$$S^{IV} = \frac{AB}{2} \left(F\left(\frac{a}{A}, \frac{b}{B}\right) - F\left(\frac{a}{A}, \frac{b+d}{B}\right) \right). \tag{A26}$$

e) Case V: The three conditions required to identify this case are

$$\left(\frac{a}{A}\right)^2 + \left(\frac{b+d}{B}\right)^2 < 1 \qquad (v_2 \text{ inside}) \tag{A27}$$

$$\left(\frac{a+c}{A}\right)^2 + \left(\frac{b}{B}\right)^2 < 1 \qquad (v_4 \text{ inside}). \tag{A28}$$

and

$$\left(\frac{a+c}{A}\right)^2 + \left(\frac{b+d}{B}\right)^2 \geq 1 \qquad (v_3 \text{ outside}). \tag{A29}$$

Applying the result given in Case II, the area is

$$S^V = \frac{AB}{2} \left(F\left(\frac{a}{A}, \frac{b}{B}\right) - F\left(\frac{a+c}{A}, \frac{b}{B}\right) - F\left(\frac{a}{A}, \frac{b+d}{B}\right) \right). \tag{A30}$$

f) Case VI: The sole condition required to identify this case is

$$\left(\frac{a+c}{A}\right)^2 + \left(\frac{b+d}{B}\right)^2 \leq 1 \qquad (v_3 \text{ inside}). \tag{A31}$$

Since all of the vertices are inside the ellipse, the intersection area is simply $S^{VI} = cd$.

REFERENCES

[1] D. Mariyanayagam, P. Gurdjos, S. Chambon, F. Brunet, and V. Charvillat, "Pose estimation of a single circle using default intrinsic calibration," arXiv:1804.04922, 2018.

[2] S. J. Ahn, *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, vol. 3151.

[3] Z. Zhang, "Parameter estimation techniques: a tutorial with application to conic fitting," *Image Vision Comput.*, vol. 15, no. 1, pp. 59–76, 1997.

[4] I. Kim, "Orthogonal distance fitting of ellipses," *Commun. Korean Math. Soc.*, vol. 17, no. 1, pp. 121–142, 2002.

[5] W. Gander, G. H. Golub, and R. Strebel, "Least-squares fitting of circles and ellipses," *BIT*, vol. 34, no. 4, pp. 558–578, 1994.

[6] P. Sturm and P. Gargallo, "Conic fitting using the geometric distance," in *Proc. 8th Asian Conf. Computer Vision*, ser. Lecture Notes in Computer Science, vol. 4844. Berlin, Heidelberg: Springer, 2007, pp. 784–795.

[7] N. Chernov and H. Ma, "Least squares fitting of quadratic curves and surfaces," in *Computer Vision*, S. R. Yoshida, Ed. New York: Nova Science Publishers, 2011, pp. 285–302.

[8] N. Chernov, Q. Huang, and H. Ma, "Fitting quadratic curves to data points," *Br. J. Math. Comput. Sci.*, vol. 4, no. 1, pp. 33–60, 2014.

[9] P. L. Rosin, "A note on the least squares fitting of ellipses," *Pattern Recognition Lett.*, vol. 14, no. 10, pp. 799–808, 1993.

[10] A. Al-Sharadqah and N. Chernov, "A doubly optimal ellipse fit," *Comput. Statist. Data Anal.*, vol. 56, no. 9, pp. 2771–2781, 2012.

[11] A. Kukush, I. Markovsky, and S. V. Huffel, "Consistent estimation in an implicit quadratic measurement error model," *Comput. Statist. Data Anal.*, vol. 47, no. 1, pp. 123–147, 2004.

[12] L. Hunyadi and I. Vajk, "Constrained quadratic errors-in-variables fitting," *Visual Comput.*, vol. 30, no. 12, pp. 1347–1358, 2013.

[13] K. Kanatani, "Ellipse fitting with hyperaccuracy," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 10, pp. 2653–2660, 2006.

[14] Z. L. Szpak, W. Chojnacki, and A. van den Hengel, "A comparison of ellipse fitting methods and implications for multiple-view geometry estimation," in *Proc. Int. Conf. Digital Image Computing: Techniques and Applications*, 2012, pp. 1–8.

[15] K. Kanatani, A. Al-Sharadqah, N. Chernov, and Y. Sugaya, "Renormalization returns: hyper-renormalization and its applications," in *Proc. 12th European Conf. Computer Vision*, ser. Lecture Notes in Computer Science, vol. 7574. Berlin, Heidelberg: Springer, 2012, pp. 384–397.

[16] M. J. Collett and G. J. Tee, "Ellipse fitting for interferometry. part 1: static methods," *J. Opt. Soc. Am. A*, vol. 31, no. 12, pp. 2573–2583, 2014.

[17] A. Fabijańska, "Subpixel edge detection in blurry and noisy images," *Int. J. Comput. Sci. Appl.*, vol. 12, no. 2, pp. 1–19, 2015.

[18] J.-N. Ouellet and P. Hébert, "Precise ellipse estimation without contour point extraction," *Mach. Vision Appl.*, vol. 21, no. 1, p. 59, 2008.

[19] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, 1999.

[20] R. Halíř and J. Flusser, "Numerically stable direct least squares fitting of ellipses," in *Proc. Sixth Int. Conf. in Central Europe on Computer Graphics and Visualization*, vol. 1, 1998, pp. 125–132.

[21] H. H. Barrett and K. J. Myers, *Foundations of Image Science*. New York: John Wiley & Sons, 2004.

[22] W. K. Pratt, *Introduction to Digital Image Processing*. Boca Raton, FL: CRC Press, 2013.

[23] U. Köthe, "What can we learn from discrete images about the continuous world?" in *Proc. 14th IAPR Int. Conf. Discrete Geometry for Computer Imagery*, ser. Lecture Notes in Computer Science, vol. 4992. Berlin, Heidelberg: Springer, 2008, pp. 4–19.

[24] D. Zwillinger, Ed., *CRC Standard Mathematical Tables and Formulae*, 32nd ed. Boca Raton, FL: CRC Press, 2012.

[25] D. E. Knuth, *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc, 1997.

[26] D. S. Sivia and J. Skilling, *Data Analysis: A Bayesian Tutorial*, 2nd ed. Oxford: Oxford University Press, 2006.

[27] J. Porrill, "Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter," *Image Vision Comput.*, vol. 8, no. 1, pp. 37–41, 1990.

[28] H. Scheffé, "A method for judging all contrasts in the analysis of variance," *Biometrika*, vol. 40, no. 1–2, pp. 87–104, 1953.

[29] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*, 3rd ed. New York: Springer, 2005.

[30] Z. L. Szpak, W. Chojnacki, and A. van den Hengel, "Guaranteed ellipse fitting with a confidence region and an uncertainty measure for centre, axes, and orientation," *J. Math. Imaging Vis.*, vol. 52, no. 2, pp. 173–199, 2015.

[31] K. Kanatani and D. D. Morris, "Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy," *IEEE Trans. Inform. Theory*, vol. 47, no. 5, pp. 2017–2028, 2001.

[32] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Proc. Int. Workshop Vision Algorithms*, ser. Lecture Notes in Computer Science, vol. 1883. Berlin, Heidelberg: Springer, 1999, pp. 298–372.

[33] A. D. Groves, "Area of intersection of an ellipse and a rectangle," Ballistic Research Laboratories, Aberdeen Proving Ground, Maryland, USA, Tech. Rep. 410103, 1963.

[34] P. Kovesi, "Good colour maps: how to design them," arXiv:1509.03700, 2015.