

Abstract

Extracting elliptic edges from images and fitting ellipse equations to them is one of the most fundamental tasks of computer vision. This is because circular objects, which are very common in daily scenes, are projected as ellipses in images. We can even compute their 3D positions from the fitted ellipse equations, which along with other applications are discussed in Chap. 9. In this chapter, we present computational procedures for accurately fitting an ellipse to extracted edge points by considering the statistical properties of image noise. The approach is classified into algebraic and geometric. The algebraic approach includes least squares, iterative reweight, the Taubin method, renormalization, HyperLS, and hyper-renormalization; the geometric approach includes FNS, geometric distance minimization, and hyper-accurate correction. We then describe the ellipse-specific method of Fitzgibbon et al. and the random sampling technique for avoiding hyperbolas, which may occur when the input information is insufficient. The RANSAC procedure is also discussed for removing nonelliptic arcs from the extracted edge point sequence.

2.1 Representation of Ellipses

An ellipse observed in an image is described in terms of a quadratic polynomial equation in the form

$$Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Ey) + f_0^2F = 0, \quad (2.1)$$

where f_0 is a constant for adjusting the scale. Theoretically, we can let it be 1, but for finite-length numerical computation it should be chosen that x/f_0 and y/f_0 have approximately the order of 1; this increases the numerical accuracy, avoiding the loss of significant digits. In view of this, we take the origin of the image xy coordinate

system at the center of the image, rather than the upper-left corner as is customarily done, and take f_0 to be the length of the side of a square that we assume to contain the ellipse to be extracted. For example, if we assume there is an ellipse in a 600×600 pixel region, we let $f_0 = 600$. Because Eq. (2.1) has scale indeterminacy (i.e., the same ellipse is represented if A, B, C, D, E , and F were multiplied by a common nonzero constant), we normalize them to

$$A^2 + B^2 + C^2 + D^2 + E^2 + F^2 = 1. \quad (2.2)$$

If we define the 6D vectors

$$\xi = \begin{pmatrix} x^2 \\ 2xy \\ y^2 \\ 2f_0x \\ 2f_0y \\ f_0^2 \end{pmatrix}, \quad \theta = \begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \end{pmatrix}, \quad (2.3)$$

Equation (2.2) can be written as

$$(\xi, \theta) = 0, \quad (2.4)$$

where and hereafter we denote the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$ by $(\mathbf{a}, \mathbf{b})$. The vector θ in Eq. (2.4) has scale indeterminacy, and the normalization of Eq. (2.2) is equivalent to vector normalization $\|\theta\| = 1$ to the unit norm.

2.2 Least-Squares Approach

Fitting an ellipse in the form of Eq. (2.1) to a sequence of points $(x_1, y_1), \dots, (x_N, y_N)$ in the presence of noise (Fig. 2.1) is to find A, B, C, D, E , and F such that

$$Ax_\alpha^2 + 2Bx_\alpha y_\alpha + Cy_\alpha^2 + 2f_0(Dx_\alpha + Ey_\alpha) + f_0^2 F \approx 0, \quad \alpha = 1, \dots, N. \quad (2.5)$$

If we write ξ_α for the value of ξ of Eq. (2.3) for $x = x_\alpha$ and $y = y_\alpha$, Eq. (2.5) can be equivalently written as

$$(\xi_\alpha, \theta) \approx 0, \quad \alpha = 1, \dots, N. \quad (2.6)$$

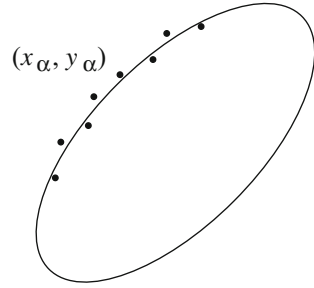
Our task is to compute such a unit vector θ . The simplest and most naive method is the following *least squares (LS)*.

Procedure 2.1 (Least squares)

1. Compute the 6×6 matrix

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N \xi_\alpha \xi_\alpha^\top. \quad (2.7)$$

Fig. 2.1 Fitting an ellipse to a noisy point sequence



2. Solve the eigenvalue problem

$$\mathbf{M}\boldsymbol{\theta} = \lambda\boldsymbol{\theta}, \quad (2.8)$$

and return the unit eigenvector $\boldsymbol{\theta}$ for the smallest eigenvalue λ .

Comments This is a straightforward generalization of line fitting to a point sequence ($\leftrightarrow$ Problem 2.1); we minimize the sum of squares

$$J = \frac{1}{N} \sum_{\alpha=1}^N (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2 = \frac{1}{N} \sum_{\alpha=1}^N \boldsymbol{\theta}^\top \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top \boldsymbol{\theta} = (\boldsymbol{\theta}, \mathbf{M}\boldsymbol{\theta}) \quad (2.9)$$

subject to $\|\boldsymbol{\theta}\| = 1$. As is well known in linear algebra, the minimum of this quadratic form in $\boldsymbol{\theta}$ is given by the unit eigenvector $\boldsymbol{\theta}$ of $\mathbf{M}$ for the smallest eigenvalue. Equation (2.9) is often called the *algebraic distance* (although it does not have the dimension of square length), and Procedure 2.1 is also known as *algebraic distance minimization*. It is sometimes called *DLT* (*direct linear transformation*). Inasmuch as the computation is very easy and the solution is immediately obtained, this method has been widely used. However, when the input point sequence covers only a small part of the ellipse circumference, it often produces a small and flat ellipse very different from the true shape (we show such examples in Sect. 2.8). Still, this is a prototype of all existing ellipse-fitting algorithms. How this can be improved has been a major motivation of many researchers.

2.3 Noise and Covariance Matrices

The reason for the poor accuracy of Procedure 2.1 is that the properties of image noise are not considered; for accurate fitting, we need to take the statistical properties of noise into consideration. Suppose the data x_α and y_α are disturbed from their true values $\bar{x}_\alpha$ and $\bar{y}_\alpha$ by Δx_α and Δy_α , and write

$$x_\alpha = \bar{x}_\alpha + \Delta x_\alpha, \quad y_\alpha = \bar{y}_\alpha + \Delta y_\alpha. \quad (2.10)$$

Substituting this into $\boldsymbol{\xi}_\alpha$, we obtain

$$\boldsymbol{\xi}_\alpha = \bar{\boldsymbol{\xi}}_\alpha + \Delta_1 \boldsymbol{\xi}_\alpha + \Delta_2 \boldsymbol{\xi}_\alpha, \quad (2.11)$$

where $\bar{\xi}_\alpha$ is the value of ξ_α for $x_\alpha = \bar{x}_\alpha$ and $y_\alpha = \bar{y}_\alpha$ and $\Delta_1 \xi_\alpha$, and $\Delta_2 \xi_\alpha$ are, respectively, the first-order noise terms (i.e., linear expressions in Δx_α and Δy_α) and the second-order noise terms (i.e., quadratic expressions in Δx_α and Δy_α). Specifically, they are

$$\Delta_1 \xi_\alpha = \begin{pmatrix} 2\bar{x}_\alpha \Delta x_\alpha \\ 2\Delta x_\alpha \bar{y}_\alpha + 2\bar{x}_\alpha \Delta y_\alpha \\ 2\bar{y}_\alpha \Delta y_\alpha \\ 2f_0 \Delta x_\alpha \\ 2f_0 \Delta y_\alpha \\ 0 \end{pmatrix}, \quad \Delta_2 \xi_\alpha = \begin{pmatrix} \Delta x_\alpha^2 \\ 2\Delta x_\alpha \Delta y_\alpha \\ \Delta y_\alpha^2 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (2.12)$$

Regarding the noise terms Δx_α and Δy_α as random variables, we define the *covariance matrix* of ξ_α by

$$V[\xi_\alpha] = E[\Delta_1 \xi_\alpha \Delta_1 \xi_\alpha^\top], \quad (2.13)$$

where $E[\cdot]$ denotes the expectation over the noise distribution, and $\top$ denotes the vector transpose. We assume that Δx_α and Δy_α are subject to an independent Gaussian distribution of mean 0 and standard deviation σ . Thus

$$E[\Delta x_\alpha] = E[\Delta y_\alpha] = 0, \quad E[\Delta x_\alpha^2] = E[\Delta y_\alpha^2] = \sigma^2, \quad E[\Delta x_\alpha \Delta y_\alpha] = 0. \quad (2.14)$$

Substituting Eq. (2.12) and using this relationship, we obtain the covariance matrix of Eq. (2.13) in the form

$$V[\xi_\alpha] = \sigma^2 V_0[\xi_\alpha], \quad V_0[\xi_\alpha] = 4 \begin{pmatrix} \bar{x}_\alpha^2 & \bar{x}_\alpha \bar{y}_\alpha & 0 & f_0 \bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha \bar{y}_\alpha & \bar{x}_\alpha^2 + \bar{y}_\alpha^2 & \bar{x}_\alpha \bar{y}_\alpha & f_0 \bar{y}_\alpha & f_0 \bar{x}_\alpha & 0 \\ 0 & \bar{x}_\alpha \bar{y}_\alpha & \bar{y}_\alpha^2 & 0 & f_0 \bar{y}_\alpha & 0 \\ f_0 \bar{x}_\alpha & f_0 \bar{y}_\alpha & 0 & f_0^2 & 0 & 0 \\ 0 & f_0 \bar{x}_\alpha & f_0 \bar{y}_\alpha & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (2.15)$$

All the elements of $V[\xi_\alpha]$ have the multiple σ^2 , therefore we factor it out and call $V_0[\xi_\alpha]$ the *normalized covariance matrix*. We also call the standard deviation σ the *noise level*. The diagonal elements of the covariance matrix $V[\xi_\alpha]$ indicate the noise susceptibility of each component of the vector ξ_α and the off-diagonal elements measure the correlation between its components.

The covariance matrix of Eq. (2.13) is defined in terms of the first-order noise term $\Delta_1 \xi_\alpha$ alone. It is known that incorporation of the second-order term $\Delta_2 \xi_\alpha$ will have little influence over the final results. This is because $\Delta_2 \xi_\alpha$ is very small as compared with $\Delta_1 \xi_\alpha$. Note that the elements of $V_0[\xi_\alpha]$ in Eq. (2.15) contain true values $\bar{x}_\alpha$ and $\bar{y}_\alpha$. They are replaced by their observed values x_α and y_α in actual computation. It is known that this replacement has practically no effect on the final results.

2.4 Algebraic Methods

2.4.1 Iterative Reweight

The following *iterative reweight* is an old and well-known method.

Procedure 2.2 (Iterative reweight)

1. Let $\theta_0 = \mathbf{0}$ and $W_\alpha = 1, \alpha = 1, \dots, N$.
2. Compute the 6×6 matrix

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \xi_\alpha \xi_\alpha^\top. \quad (2.16)$$

3. Solve the eigenvalue problem

$$\mathbf{M}\theta = \lambda\theta, \quad (2.17)$$

and compute the unit eigenvector θ for the smallest eigenvalue λ .

4. If $\theta \approx \theta_0$ up to sign, return θ and stop. Else, update

$$W_\alpha \leftarrow \frac{1}{(\theta, V_0[\xi_\alpha]\theta)}, \quad \theta_0 \leftarrow \theta, \quad (2.18)$$

and go back to Step 2.

Comments As is well known in linear algebra, computing the unit eigenvector for the smallest eigenvalue of a symmetric matrix $\mathbf{M}$ is equivalent to computing the unit vector θ that minimizes the quadratic form $(\theta, \mathbf{M}\theta)$. Because

$$(\theta, \mathbf{M}\theta) = (\theta, \left(\frac{1}{N} \sum_{\alpha=1}^N W_\alpha \xi_\alpha \xi_\alpha^\top\right)\theta) = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha (\theta, \xi_\alpha \xi_\alpha^\top \theta) = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha (\xi_\alpha, \theta)^2, \quad (2.19)$$

we minimize the sum of squares $(\xi_\alpha, \theta)^2$ weighted by W_α . This is commonly known as *weighted least squares*. According to statistics, the weights W_α are optimal if they are inversely proportional to the variance of each term, being small for uncertain terms and large for certain terms. Inasmuch as $(\xi_\alpha, \theta) = (\xi_\alpha, \theta) + (\Delta_1 \xi_\alpha, \theta) + (\Delta_2 \xi_\alpha, \theta)$ and $(\xi_\alpha, \theta) = 0$, the variance is given from Eqs. (2.13) and (2.15) by

$$E[(\xi_\alpha, \theta)^2] = E[(\theta, \Delta_1 \xi_\alpha \Delta_1^\top \xi_\alpha \theta)] = (\theta, E[\Delta_1 \xi_\alpha \Delta_1^\top \xi_\alpha] \theta) = \sigma^2(\theta, V_0[\xi_\alpha] \theta), \quad (2.20)$$

omitting higher-order noise terms. Thus we should let $W_\alpha = 1/(\theta, V_0[\xi_\alpha] \theta)$, but θ is not known yet. Therefore we instead use the weights W_α determined in the preceding iteration to compute θ and update the weights as in Eq. (2.18). Let us call the solution θ computed in the initial iteration the *initial solution*. Initially, we set $W_\alpha = 1$, therefore Eq. (2.19) implies that we are starting from the least-squares solution. The phrase “up to sign” in Step 4 reflects the fact that eigenvectors have sign indeterminacy; we align θ and θ_0 by reversing the sign $\theta \leftarrow -\theta$ when $(\theta, \theta_0) < 0$.

2.4.2 Renormalization and the Taubin Method

It has been well known that the accuracy of least squares and iterative reweight is rather low with large bias when the input elliptic arc is short; they tend to fit a smaller ellipse than expected. The following *renormalization* was introduced to reduce the bias.

Procedure 2.3 (Renormalization)

1. Let $\theta_0 = \mathbf{0}$ and $W_\alpha = 1$, $\alpha = 1, \dots, N$.
2. Compute the 6×6 matrices

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \xi_\alpha \xi_\alpha^\top, \quad \mathbf{N} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha V_0[\xi_\alpha]. \quad (2.21)$$

3. Solve the generalized eigenvalue problem

$$\mathbf{M}\theta = \lambda \mathbf{N}\theta, \quad (2.22)$$

and compute the unit generalized eigenvector θ for the generalized eigenvalue λ of the smallest absolute value.

4. If $\theta \approx \theta_0$ up to sign, return θ and stop. Else, update

$$W_\alpha \leftarrow \frac{1}{(\theta, V_0[\xi_\alpha]\theta)}, \quad \theta_0 \leftarrow \theta, \quad (2.23)$$

and go back to Step 2.

Comments As is well known in linear algebra, solving the generalized eigenvalue problem of Eq. (2.22) for symmetric matrices $\mathbf{M}$ and $\mathbf{N}$ is equivalent to computing the unit vector θ that minimizes the quadratic form $(\theta, \mathbf{M}\theta)$ subject to the constraint $(\theta, \mathbf{N}\theta) = \text{constant}$. Initially $W_\alpha = 1$, therefore the first iteration minimizes the sum of squares $\sum_{\alpha=1}^N (\xi_\alpha, \theta)^2$ subject to $(\theta, (\sum_{\alpha=1}^N V_0[\xi_\alpha])\theta) = \text{constant}$. This is known as the *Taubin method* for ellipse fitting ($\hookrightarrow$ Problem 2.2). Standard numerical tools for solving the generalized eigenvalue problem in the form of Eq. (2.22) assume that $\mathbf{N}$ is positive definite, but Eq. (2.15) implies that the sixth column and the sixth row of the matrix $V_0[\xi_\alpha]$ all consist of zero, therefore $\mathbf{N}$ is not positive definite. However, Eq. (2.22) is equivalently written as

$$\mathbf{N}\theta = \frac{1}{\lambda} \mathbf{M}\theta. \quad (2.24)$$

If the data contain noise, the matrix $\mathbf{M}$ is positive definite, therefore we can apply a standard numerical tool to compute the unit generalized eigenvector θ for the generalized eigenvalue $1/\lambda$ of the largest absolute value. The matrix $\mathbf{M}$ is not positive definite only when there is no noise. We need not consider that case in practice, but if $\mathbf{M}$ happens to have eigenvalue 0, which implies that the data are exact, the corresponding unit eigenvector θ gives the true solution.

2.4.3 Hyper-Renormalization and HyperLS

According to experiments, the accuracy of the Taubin method is higher than least squares and iterative reweight, and renormalization has even higher accuracy. The accuracy can be further improved by the following *hyper-renormalization*.

Procedure 2.4 (Hyper-renormalization)

1. Let $\theta_0 = \mathbf{0}$ and $W_\alpha = 1$, $\alpha = 1, \dots, N$.
2. Compute the 6×6 matrices

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \xi_\alpha \xi_\alpha^\top, \quad (2.25)$$

$$\begin{aligned} \mathbf{N} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \left(V_0[\xi_\alpha] + 2\mathcal{S}[\xi_\alpha \mathbf{e}^\top] \right) \\ - \frac{1}{N^2} \sum_{\alpha=1}^N W_\alpha^2 \left((\xi_\alpha, \mathbf{M}_5^- \xi_\alpha) V_0[\xi_\alpha] + 2\mathcal{S}[V_0[\xi_\alpha] \mathbf{M}_5^- \xi_\alpha \xi_\alpha^\top] \right), \end{aligned} \quad (2.26)$$

where $\mathcal{S}[\cdot]$ is the symmetrization ($\mathcal{S}[\mathbf{A}] = (\mathbf{A} + \mathbf{A}^\top)/2$), and $\mathbf{e}$ is the vector

$$\mathbf{e} = (1, 0, 1, 0, 0, 0)^\top. \quad (2.27)$$

The matrix $\mathbf{M}_5^-$ is the pseudoinverse of $\mathbf{M}$ of truncated rank 5.

3. Solve the generalized eigenvalue problem

$$\mathbf{M}\theta = \lambda \mathbf{N}\theta, \quad (2.28)$$

and compute the unit generalized eigenvector θ for the generalized eigenvalue λ of the smallest absolute value.

4. If $\theta \approx \theta_0$ up to sign, return θ and stop. Else, update

$$W_\alpha \leftarrow \frac{1}{(\theta, V_0[\xi_\alpha] \theta)}, \quad \theta_0 \leftarrow \theta, \quad (2.29)$$

and go back to Step 2.

Comments The vector $\mathbf{e}$ is defined in such a way that $E[\Delta_2 \xi_\alpha] = \sigma^2 \mathbf{e}$ for the second-order noise term $\Delta_2 \xi_\alpha$ in Eq.(2.12). The pseudoinverse $\mathbf{M}_5^-$ of truncated rank is computed by

$$\mathbf{M}_5^- = \frac{1}{\mu_1} \theta_1 \theta_1^\top + \dots + \frac{1}{\mu_5} \theta_5 \theta_5^\top, \quad (2.30)$$

where $\mu_1 \geq \dots \geq \mu_6$ are the eigenvalues of $\mathbf{M}$, and $\theta_1, \dots, \theta_6$ are the corresponding eigenvectors (note that $\mathbf{m}_6$ and θ_6 are not used). The derivation of Eq.(2.26) is given in Chap. 14.

The matrix $\mathbf{N}$ in Eq.(2.26) is not positive definite, but we can use a standard numerical tool by rewriting Eq.(2.28) in the form of Eq.(2.24) and computing the unit generalized eigenvector for the generalized eigenvalue $1/\lambda$ of the largest absolute value. The initial solution minimizes the sum of squares $\sum_{\alpha=1}^N (\xi_\alpha, \theta)^2$ subject to the constraint $(\theta, \mathbf{N}\theta) = \text{constant}$ for the matrix $\mathbf{N}$ obtained by letting $W_\alpha = 1$ in Eq.(2.26). This corresponds to the method called *HyperLS* ($\Leftrightarrow$ Problem 2.3).

2.4.4 Summary of Algebraic Methods

We have seen that all the above methods compute the θ that satisfies

$$\mathbf{M}\theta = \lambda\mathbf{N}\theta, \quad (2.31)$$

where the matrices $\mathbf{M}$ and $\mathbf{N}$ are defined from the data and contain the unknown θ . Different choices of them lead to different methods:

$$\mathbf{M} = \begin{cases} \frac{1}{N} \sum_{\alpha=1}^N \xi_{\alpha} \xi_{\alpha}^{\top}, & \text{(least squares, Taubin, HyperLS)} \\ \frac{1}{N} \sum_{\alpha=1}^N \frac{\xi_{\alpha} \xi_{\alpha}^{\top}}{(\theta, V_0[\xi_{\alpha}]\theta)}, & \text{(iterative reweight, renormalization, hyper-renormalization)} \end{cases} \quad (2.32)$$

$$\mathbf{N} = \begin{cases} \mathbf{I} & \text{(identity), (least squares, iterative reweight)} \\ \frac{1}{N} \sum_{\alpha=1}^N V_0[\xi_{\alpha}], & \text{(Taubin)} \\ \frac{1}{N} \sum_{\alpha=1}^N \frac{V_0[\xi_{\alpha}]}{(\theta, V_0[\xi_{\alpha}]\theta)}, & \text{(renormalization)} \\ \frac{1}{N} \sum_{\alpha=1}^N \left(V_0[\xi_{\alpha}] + 2\mathcal{S}[\xi_{\alpha}\mathbf{e}^{\top}] \right) \\ \quad - \frac{1}{N^2} \sum_{\alpha=1}^N \left((\xi_{\alpha}, \mathbf{M}_5^{-} \xi_{\alpha}) V_0[\xi_{\alpha}] + 2\mathcal{S}[V_0[\xi_{\alpha}]\mathbf{M}_5^{-} \xi_{\alpha} \xi_{\alpha}^{\top}] \right), & \text{(HyperLS)} \\ \frac{1}{N} \sum_{\alpha=1}^N \frac{1}{(\theta, V_0[\xi_{\alpha}]\theta)} \left(V_0[\xi_{\alpha}] + 2\mathcal{S}[\xi_{\alpha}\mathbf{e}^{\top}] \right) \\ \quad - \frac{1}{N^2} \sum_{\alpha=1}^N \frac{1}{(\theta, V_0[\xi_{\alpha}]\theta)^2} \left((\xi_{\alpha}, \mathbf{M}_5^{-} \xi_{\alpha}) V_0[\xi_{\alpha}] + 2\mathcal{S}[V_0[\xi_{\alpha}]\mathbf{M}_5^{-} \xi_{\alpha} \xi_{\alpha}^{\top}] \right). & \text{(hyper-renormalization)} \end{cases} \quad (2.33)$$

For least squares, Taubin, and HyperLS, the matrices $\mathbf{M}$ and $\mathbf{N}$ do not contain the unknown θ , thus Eq. (2.31) is a generalized eigenvalue problem that can be directly solved without iterations. For other methods (iterative reweight, renormalization, and hyper-renormalization), the unknown θ is contained in the denominators in the expressions of $\mathbf{M}$ and $\mathbf{N}$. We let the part that contains θ be $1/W_{\alpha}$, compute W_{α} using the value of θ obtained in the preceding iteration, and solve the generalized eigenvalue problem in the form of Eq. (2.31). We then use the resulting θ to update W_{α} and repeat this process.

According to experiments, HyperLS has accuracy comparable to renormalization, and hyper-renormalization has even higher accuracy. Because the iteration starts from the HyperLS solution, the convergence is very fast; usually, three to four iterations are sufficient.

2.5 Geometric Methods

2.5.1 Geometric Distance and Sampson Error

Geometric fitting refers to computing an ellipse such that the sum of squares of the distance of observed points (x_α, y_α) to the ellipse is minimized (Fig. 2.2). Let $(\bar{x}_\alpha, \bar{y}_\alpha)$ be the point on the ellipse closest to (x_α, y_α) . Let d_α be the distance between them. Geometric fitting computes the ellipse that minimizes

$$S = \frac{1}{N} \sum_{\alpha=1}^N \left((x_\alpha - \bar{x}_\alpha)^2 + (y_\alpha - \bar{y}_\alpha)^2 \right) = \frac{1}{N} \sum_{\alpha=1}^N d_\alpha^2. \quad (2.34)$$

This is called the *geometric distance* (although this has the dimension of square length, this term is widely used). It is also sometimes called the *reprojection error*. If we identify $(\bar{x}_\alpha, \bar{y}_\alpha)$ with the true position of the datapoint (x_α, y_α) , we see from Eq. (2.10) that Eq. (2.34) can also be written as $S = (1/N) \sum_{\alpha=1}^N (\Delta x_\alpha^2 + \Delta y_\alpha^2)$, where we regard Δx_α and Δy_α as unknown variables rather than stochastic quantities. When point (x_α, y_α) is close to the ellipse, the square distance d_α^2 can be written (except for high-order terms in Δx_α and Δy_α) as follows ($\leftrightarrow$ Problem 2.4).

$$d_\alpha^2 = (x_\alpha - \bar{x}_\alpha)^2 + (y_\alpha - \bar{y}_\alpha)^2 \approx \frac{(\xi_\alpha, \theta)^2}{(\theta, V_0[\xi_\alpha]\theta)}. \quad (2.35)$$

Hence, the geometric distance in Eq. (2.34) can be approximated by

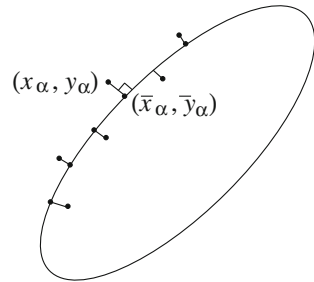
$$J = \frac{1}{N} \sum_{\alpha=1}^N \frac{(\xi_\alpha, \theta)^2}{(\theta, V_0[\xi_\alpha]\theta)}. \quad (2.36)$$

This is known as the *Sampson error*.

2.5.2 FNS

A well-known method to minimize the Sampson error of Eq. (2.36) is the *FNS* (*Fundamental Numerical Scheme*):

Fig. 2.2 We fit an ellipse such that the geometric distance (i.e., the average of the square distance of each point to the ellipse) is minimized



Procedure 2.5 (FNS)

1. Let $\boldsymbol{\theta} = \boldsymbol{\theta}_0 = \mathbf{0}$ and $W_\alpha = 1, \alpha = 1, \dots, N$.
2. Compute the 6×6 matrices

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, \quad \mathbf{L} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha^2 (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2 V_0[\boldsymbol{\xi}_\alpha]. \quad (2.37)$$

3. Compute the 6×6 matrix

$$\mathbf{X} = \mathbf{M} - \mathbf{L}. \quad (2.38)$$

4. Solve the eigenvalue problem

$$\mathbf{X}\boldsymbol{\theta} = \lambda\boldsymbol{\theta}, \quad (2.39)$$

and compute the unit eigenvector $\boldsymbol{\theta}$ for the smallest eigenvalue λ .

5. If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, update

$$W_\alpha \leftarrow \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \quad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}, \quad (2.40)$$

and go back to Step 2.

Comments This scheme solves $\nabla_{\boldsymbol{\theta}} J = \mathbf{0}$ for the Sampson error J , where $\nabla_{\boldsymbol{\theta}} J$ is the gradient of J , that is, the vector whose i th component is $\partial J / \partial \theta_i$. Differentiating Eq. (2.36), we obtain the following expression ($\hookrightarrow$ Problem 2.5(1)).

$$\nabla_{\boldsymbol{\theta}} J = 2(\mathbf{M} - \mathbf{L})\boldsymbol{\theta} = 2\mathbf{X}\boldsymbol{\theta}. \quad (2.41)$$

Here, $\mathbf{M}$, $\mathbf{L}$, and $\mathbf{X}$ are the matrices given by Eqs. (2.37) and (2.38). It can be shown that $\lambda = 0$ after the iterations have converged ($\hookrightarrow$ Problem 2.5(2)). Thus from Eq. (2.41) we obtain the value $\boldsymbol{\theta}$ that satisfies $\nabla_{\boldsymbol{\theta}} J = \mathbf{0}$. Because we start with $\boldsymbol{\theta} = \mathbf{0}$, the matrix $\mathbf{L}$ in Eq. (2.37) is initially $\mathbf{O}$, therefore Eq. (2.39) reduces to $\mathbf{M}\boldsymbol{\theta} = \lambda\boldsymbol{\theta}$. This means that the FNS iterations start from the least squares solution.

2.5.3 Geometric Distance Minimization

Once we have computed $\boldsymbol{\theta}$ by the above FNS, we can iteratively modify it such that it strictly minimizes the geometric distance of Eq. (2.34). To be specific, we modify the data $\boldsymbol{\xi}_\alpha$, using the current solution $\boldsymbol{\theta}$, and minimize a modified Sampson error to obtain a new solution $\boldsymbol{\theta}$. After a few iterations, the Sampson error coincides with the geometric distance. The procedure is as follows.

Procedure 2.6 (Geometric distance minimization)

1. Let $J_0 = \infty$ (a sufficiently large number), $\hat{x}_\alpha = x_\alpha$, $\hat{y}_\alpha = y_\alpha$, and $\tilde{x}_\alpha = \tilde{y}_\alpha = 0$, $\alpha = 1, \dots, N$.
2. Compute the normalized covariance matrix $V_0[\hat{\boldsymbol{\xi}}_\alpha]$ obtained by replacing $\bar{x}_\alpha$ and $\bar{y}_\alpha$ in the definition of $V_0[\boldsymbol{\xi}_\alpha]$ in Eq. (2.15) by $\hat{x}_\alpha$ and $\hat{y}_\alpha$, respectively.
3. Compute the following modified data $\hat{\boldsymbol{\xi}}_\alpha^*$.

$$\xi_{\alpha}^* = \begin{pmatrix} \hat{x}_{\alpha}^2 + 2\hat{x}_{\alpha}\tilde{x}_{\alpha} \\ 2(\hat{x}_{\alpha}\hat{y}_{\alpha} + \hat{y}_{\alpha}\tilde{x}_{\alpha} + \hat{x}_{\alpha}\tilde{y}_{\alpha}) \\ \hat{y}_{\alpha}^2 + 2\hat{y}_{\alpha}\tilde{y}_{\alpha} \\ 2f_0(\hat{x}_{\alpha} + \tilde{x}_{\alpha}) \\ 2f_0(\hat{y}_{\alpha} + \tilde{y}_{\alpha}) \\ f_0^2 \end{pmatrix}. \quad (2.42)$$

4. Compute the value θ that minimizes the following *modified Sampson error*.

$$J^* = \frac{1}{N} \sum_{\alpha=1}^N \frac{(\xi_{\alpha}^*, \theta)^2}{(\theta, V_0[\hat{\xi}_{\alpha}]\theta)}. \quad (2.43)$$

5. Update $\tilde{x}_{\alpha}$ and $\tilde{y}_{\alpha}$ as follows.

$$\begin{pmatrix} \tilde{x}_{\alpha} \\ \tilde{y}_{\alpha} \end{pmatrix} \leftarrow \frac{2(\xi_{\alpha}^*, \theta)}{(\theta, V_0[\hat{\xi}_{\alpha}]\theta)} \begin{pmatrix} \theta_1 & \theta_2 & \theta_4 \\ \theta_2 & \theta_3 & \theta_5 \end{pmatrix} \begin{pmatrix} \hat{x}_{\alpha} \\ \hat{y}_{\alpha} \\ f_0 \end{pmatrix}. \quad (2.44)$$

6. Update $\hat{x}_{\alpha}$ and $\hat{y}_{\alpha}$ as follows.

$$\hat{x}_{\alpha} \leftarrow x_{\alpha} - \tilde{x}_{\alpha}, \quad \hat{y}_{\alpha} \leftarrow y_{\alpha} - \tilde{y}_{\alpha}. \quad (2.45)$$

7. Compute

$$J^* = \frac{1}{N} \sum_{\alpha=1}^N (\tilde{x}_{\alpha}^2 + \tilde{y}_{\alpha}^2). \quad (2.46)$$

If $J^* \approx J_0$, return θ and stop. Else, let $J_0 \leftarrow J^*$, and go back to Step 2.

Comments Because $\hat{x}_{\alpha} = x_{\alpha}$, $\hat{y}_{\alpha} = y_{\alpha}$, and $\tilde{x}_{\alpha} = \tilde{y}_{\alpha} = 0$ in the initial iteration, the value ξ_{α}^* of Eq. (2.42) is the same as ξ_{α} . Therefore the modified Simpson error J^* in Eq. (2.43) is the same as the function J in Eq. (2.36), and the value θ that minimizes J is computed initially. Now, we can rewrite Eq. (2.34) in the form

$$\begin{aligned} S &= \frac{1}{N} \sum_{\alpha=1}^N \left((\hat{x}_{\alpha} + (x_{\alpha} - \hat{x}_{\alpha}) - \bar{x}_{\alpha})^2 + (\hat{y}_{\alpha} + (y_{\alpha} - \hat{y}_{\alpha}) - \bar{y}_{\alpha})^2 \right) \\ &= \frac{1}{N} \sum_{\alpha=1}^N \left((\hat{x}_{\alpha} + \tilde{x}_{\alpha} - \bar{x}_{\alpha})^2 + (\hat{y}_{\alpha} + \tilde{y}_{\alpha} - \bar{y}_{\alpha})^2 \right), \end{aligned} \quad (2.47)$$

where

$$\tilde{x}_{\alpha} = x_{\alpha} - \hat{x}_{\alpha}, \quad \tilde{y}_{\alpha} = y_{\alpha} - \hat{y}_{\alpha}. \quad (2.48)$$

In the next iteration step, we regard the corrected values $(\hat{x}_{\alpha}, \hat{y}_{\alpha})$ as the input data and minimize Eq. (2.47). Let $(\hat{\hat{x}}_{\alpha}, \hat{\hat{y}}_{\alpha})$ be the solution. If we ignore high-order small terms in $\hat{x}_{\alpha} - \bar{x}_{\alpha}$ and $\hat{y}_{\alpha} - \bar{y}_{\alpha}$ and rewrite Eq. (2.47), we obtain the modified Simpson error J^* in Eq. (2.43) ($\hookrightarrow$ Problem 2.6). We minimize this, regard $(\hat{\hat{x}}_{\alpha}, \hat{\hat{y}}_{\alpha})$ as $(\hat{x}_{\alpha}, \hat{y}_{\alpha})$, and repeat the same process. Because the current $(\hat{x}_{\alpha}, \hat{y}_{\alpha})$ are the best approximation to $(\bar{x}_{\alpha}, \bar{y}_{\alpha})$, Eq. (2.48) implies that $\tilde{x}_{\alpha}^2 + \tilde{y}_{\alpha}^2$ is the corresponding approximation of $(\bar{x}_{\alpha} - x_{\alpha})^2 + (\bar{y}_{\alpha} - y_{\alpha})^2$. Therefore we use Eq. (2.46) to evaluate the geometric distance S . The ignored higher-order terms decrease after each iteration, therefore we

obtain in the end the value θ that minimizes the geometric distance S , and Eq. (2.46) coincides with S . According to experiments, however, the correction of θ by this procedure is very small: the three or four significant digits are unchanged in typical problems. Therefore the corrected ellipse is indistinguishable from the original one when displayed or plotted. Thus, we can practically identify FNS with a method to minimize the geometric distance.

2.5.4 Hyper-Accurate Correction

The geometric method, whether it is geometric distance minimization or Sampson error minimization, is known to have small statistical bias, which means that the expectation $E[\hat{\theta}]$ of the computed solution $\hat{\theta}$ does not completely agree with the true value $\bar{\theta}$. We express the computed solution $\hat{\theta}$ in the form

$$\hat{\theta} = \bar{\theta} + \Delta_1\theta + \Delta_2\theta + \dots, \quad (2.49)$$

where $\Delta_k\theta$ is the k th-order term in the noise components Δx_α and Δy_α . Inasmuch as $\Delta_1\theta$ is a linear expression in Δx_α and Δy_α , we have $E[\Delta_1\theta] = \mathbf{0}$. However, $E[\Delta_2\theta] \neq \mathbf{0}$ in general. If we are able to evaluate $E[\Delta_2\theta]$ in an explicit form by doing error analysis, it is expected that the subtraction

$$\tilde{\theta} = \hat{\theta} - E[\Delta_2\theta] \quad (2.50)$$

will yield higher accuracy than $\hat{\theta}$, and its expectation is $E[\tilde{\theta}] = \bar{\theta} + O(\sigma^4)$, where σ is the noise level. Note that $\Delta_3\theta$ is a third-order expression in Δx_α and Δy_α , and hence $E[\Delta_3\theta] = \mathbf{0}$. The operation of increasing the accuracy by subtracting $E[\Delta_2\theta]$ is called *hyper-accurate correction*. The actual procedure is as follows.

Procedure 2.7 (Hyper-accurate correction)

1. Compute θ by FNS (Procedure 2.5).
2. Estimate σ^2 in the form

$$\hat{\sigma}^2 = \frac{(\theta, M\theta)}{1 - 5/N}, \quad (2.51)$$

where M is the value of the matrix M in Eq. (2.37) after the FNS iterations have converged.

3. Compute the correction term

$$\Delta_c\theta = -\frac{\hat{\sigma}^2}{N}M_5^- \sum_{\alpha=1}^N W_\alpha(\mathbf{e}, \theta)\xi_\alpha + \frac{\hat{\sigma}^2}{N^2}M_5^- \sum_{\alpha=1}^N W_\alpha^2(\xi_\alpha, M_5^- V_0[\xi_\alpha]\theta)\xi_\alpha, \quad (2.52)$$

where W_α is the value of W_α in Eq. (2.40) after the FNS iterations have converged, and $\mathbf{e}$ is the vector in Eq. (2.27). The matrix M_5^- is the pseudoinverse of truncated rank 5 given by Eq. (2.30).

4. Correct θ into

$$\theta \leftarrow \mathcal{N}[\theta - \Delta_c\theta], \quad (2.53)$$

where $\mathcal{N}[\cdot]$ denotes normalization to unit norm ($\mathcal{N}[\mathbf{a}] = \mathbf{a}/\|\mathbf{a}\|$).

Comments The derivation of Eqs. (2.51) and (2.52) is given in Chap. 15. According to experiments, the accuracy of FNS and geometric distance minimization is higher than renormalization but lower than hyper-renormalization. It is known, however, that after this hyper-accurate correction the accuracy improves and is comparable to hyper-renormalization.

2.6 Ellipse-Specific Methods

2.6.1 Ellipse Condition

Equation (2.1) does not necessarily represent an ellipse; depending on the coefficients, it can represent a hyperbola or a parabola. In special cases, it may degenerate into two lines, or (x, y) may not satisfy the equation. It is easy to show that Eq. (2.1) represents an ellipse when

$$AC - B^2 > 0. \quad (2.54)$$

We discuss this in Chap. 9. Usually, an ellipse is obtained when Eq. (2.1) is fit to a point sequence obtained from an ellipse, but we may obtain a hyperbola when the point sequence is very short in the presence of large noise; a parabola results only when the solution exactly satisfies $AC - B^2 = 0$, but this possibility can be excluded in real numerical computation using noisy data. If a hyperbola results, we can ignore it in practice, inasmuch as it indicates that the data do not have sufficient information to define an ellipse. From a theoretical interest, however, schemes to force the fit to be an ellipse have also been studied.

2.6.2 Method of Fitzgibbon et al.

A well-known method for fitting only ellipses is the following method of Fitzgibbon et al.

Procedure 2.8 (Method of Fitzgibbon et al.)

1. Compute the 6×6 matrices

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N \boldsymbol{\xi}_{\alpha} \boldsymbol{\xi}_{\alpha}^{\top}, \quad \mathbf{N} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (2.55)$$

2. Solve the generalized eigenvalue problem

$$\mathbf{M}\boldsymbol{\theta} = \lambda\mathbf{N}\boldsymbol{\theta}, \quad (2.56)$$

and return the unit generalized eigenvector $\boldsymbol{\theta}$ for the generalized eigenvalue λ of the smallest absolute value.

Comments This is an algebraic method, minimizing the algebraic distance $\sum_{\alpha=1}^N (\xi_\alpha, \theta)^2$ subject to $AC - B^2 = 1$ such that Eq.(2.54) is satisfied. The matrix $\mathbf{N}$ in Eq.(2.56) is not positive definite, as in the case of renormalization and hyper-renormalization, therefore we rewrite Eq.(2.56) in the form of Eq.(2.24) and use a standard numerical tool to compute the unit eigenvector θ for the generalized eigenvalue $1/\lambda$ of the largest absolute value.

2.6.3 Method of Random Sampling

One way always to end up with an ellipse is first to do ellipse fitting without considering the ellipse condition (e.g., by hyper-renormalization) and modify the result to an ellipse if it is not an ellipse. A simple and effective method is the use of random sampling: we randomly sample five points from the input sequence, fit an ellipse to them, and repeat this many times to find the best ellipse. The actual procedure is as follows.

Procedure 2.9 (Random sampling)

1. Fit an ellipse without considering the ellipse condition. If the solution θ satisfies

$$\theta_1\theta_3 - \theta_2^2 > 0, \quad (2.57)$$

return it as the final result.

2. Else, randomly select five points from the input sequence. Let $\xi_1, \dots, \xi_5$ be the corresponding vectors of the form of Eq.(2.3).
3. Compute the unit eigenvector θ of the following matrix for the smallest eigenvalue.

$$\mathbf{M}_5 = \sum_{\alpha=1}^5 \xi_\alpha \xi_\alpha^\top. \quad (2.58)$$

4. If that θ does not satisfy Eq.(2.57), discard it and select a new set of five points randomly for fitting an ellipse.
5. If θ satisfies Eq.(2.57), store it as a candidate. Also, evaluate the corresponding Sampson error J of Eq.(2.36), and store its value.
6. Repeat this process many times, and return the value θ whose Sampson error J is the smallest.

Comments In Step 2, we select five integers over $[1, N]$ by sampling from a uniform distribution; overlapped values are discarded. Step 3 computes the ellipse passing through the given five points. We may solve the simultaneous linear equations in $A, B, \dots, F$ obtained by substituting the point coordinates to Eq.(2.1); the solution is determined up to scale, which is then normalized. However, using least squares as shown above is easier to program; we compute the unit eigenvector of the matrix $\mathbf{M}$ in the first row of Eq.(2.32). The coefficient $1/N$ is omitted in the above matrix $\mathbf{M}_5$, but the computed eigenvectors are the same. In Step 4, we judge whether the

solution is an ellipse. Theoretically, any selected five points may define a hyperbola, for example, when all the points are on a hyperbola. We ignore such an extreme case. In Step 5, if the evaluated J is larger than the stored J , we go directly on to the next sampling without storing θ and J . We stop the sampling if the current value of J is not updated consecutively after a fixed number of iterations.

In practice, however, if a standard method returns a hyperbola, it forces the solution to be an ellipse by an ellipse-specific method such as the method of Fitzgibbon et al. and random sampling does not have much meaning, as shown in the example below.

2.7 Outlier Removal

For fitting an ellipse to real image data, we must first extract a sequence of edge points from an elliptic arc. However, edge segments obtained by an edge detection filter may not belong to the same ellipse; some may be parts of other object boundaries. We call points of such segments *outliers*; those coming from the ellipse under consideration are called *inliers*. Note that we are not talking about random dispersions of edge pixels due to image processing inaccuracy, although such pixels, if they exist, are also called outliers. Our main concern is the parts of the extracted edge sequence that do not belong to the ellipse in which we are interested.

Methods that are not sensitive to the existence of outliers are said to be *robust*. The general idea of robust fitting is to find an ellipse such that *the number of points close to it is as large as possible*. Then those points that are not close to it are removed as outliers. A typical such method is *RANSAC*. The procedure is as follows.

Procedure 2.10 (RANSAC)

1. Randomly select five points from the input sequence. Let $\xi_1, \dots, \xi_5$ be their vector representations in the form of Eq. (2.3).
2. Compute the unit eigenvector θ of the matrix

$$M_5 = \sum_{\alpha=1}^5 \xi_\alpha \xi_\alpha^\top, \quad (2.59)$$

for the smallest eigenvalue, and store it as a candidate.

3. Let n be the number of points in the input sequence that satisfy

$$\frac{(\xi, \theta)^2}{(\theta, V_0[\xi]\theta)} < d^2, \quad (2.60)$$

where ξ is the vector representation of the point in the sequence in the form of Eq. (2.3), $V_0[\xi]$ is the normalized covariance matrix defined by Eq. (2.15), and d is a threshold for admissible deviation from the fitted ellipse; for example, $d = 2$ (pixels). Store that n .

4. Select a new set of five points from the input sequence, and do the same. Repeat this many times, and return from among the stored candidate ellipses the one for which n is the largest.

Comments The random sampling in Step 1 is done in the same way as in Sect. 2.6.3. Step 2 computes the ellipse that passes through the five points. In Step 3, we are measuring the distance of the points from the ellipse by Eq. (2.35). As in Sect. 2.6.3, we can go directly on to the next sampling if the count n is smaller than the stored count, and we can stop if the current value of n is not updated over a fixed number of samplings. In the end, those pixels that do not satisfy Eq. (2.60) for the chosen ellipse are removed as outliers.

2.8 Examples

Figure 2.3a shows edges extracted from an image that contains a circular object. An ellipse is fitted to the 160 edge points indicated there, and Fig. 2.3b shows ellipses obtained using various methods superimposed on the original image, where the occluded part is artificially composed for visual ease. We can see that least squares and iterative reweight produce much smaller ellipses than the true shape. All other fits are very close to the true shape, and the geometric distance minimization gives the best fit in this example.

Figure 2.4a is another edge image of a scene with a circular object. An ellipse is fitted to the 140 edge points indicated there, using different methods. Figure 2.4b shows the resulting fits. In this case, hyper-renormalization returns a hyperbola, and the method of Fitzgibbon et al. fits a small and flat ellipse. As an alternative ellipse-specific method, Szpak et al. minimized the Sampson error of Eq. (2.36) with a penalty term such that it diverges to infinity as the solution approaches a hyperbola. In this case, their method fits a large ellipse close to the hyperbola. The random sampling fit is somewhat in between, but it is still not very close to the true shape. In general, if a standard method returns a hyperbola, it indicates data insufficiency

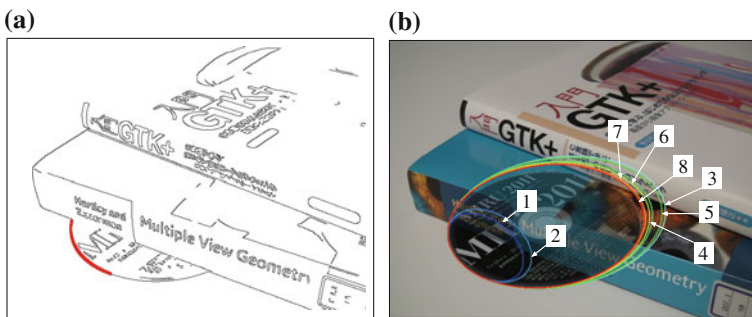


Fig. 2.3 **a** An edge image of a scene with a circular object. An ellipse is fitted to the 160 edge points indicated. **b** Fitted ellipses superimposed on the original image. The occluded part is artificially composed for visual ease: (1) least squares, (2) iterative reweight, (3) Taubin method, (4) renormalization, (5) HyperLS, (6) hyper-renormalization, (7) FN, and (8) FNS with hyper-accurate correction

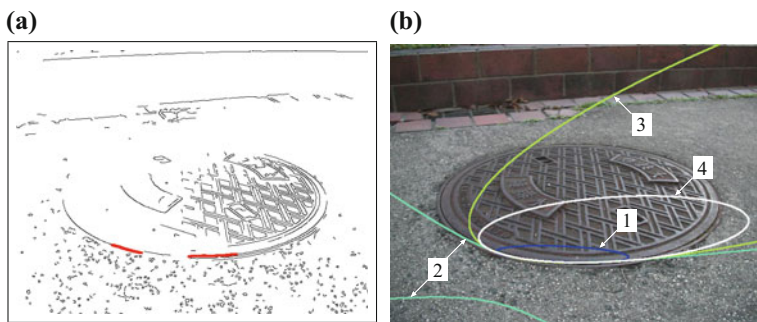


Fig. 2.4 **a** An edge image of a scene with a circular object. An ellipse is fitted to the 140 edge points indicated. **b** Fitted ellipses superimposed on the original image. (1) Fitzgibbon et al., (2) hyper-renormalization, (3) Szpak et al., (4) random sampling

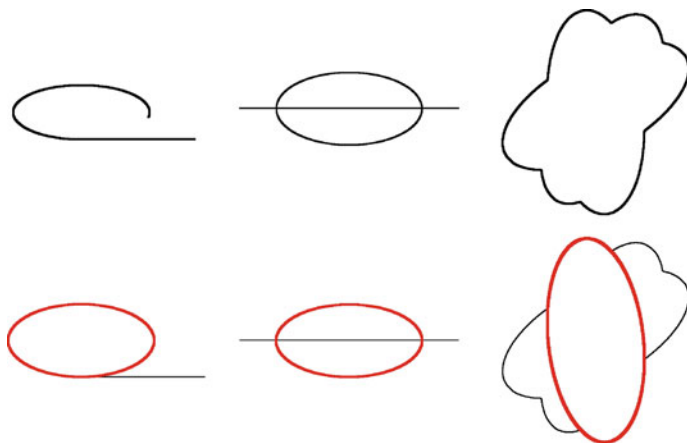


Fig. 2.5 *Upper row: input edge segments. Lower row: ellipses fitted by RANSAC*

for ellipse fitting. Hence, although ellipse-specific methods have theoretical interest, they are not suitable for practical applications.

The upper row of Fig. 2.5 shows examples of edge segments that do not entirely consist of elliptic arcs. Regarding those nonelliptic parts as outliers, we fit an ellipse to inlier arcs, using RANSAC. The lower row of Fig. 2.5 depicts fitted ellipses. We see that the segments that constitute an ellipse are automatically selected.

2.9 Supplemental Note

The description in this chapter is rather sketchy. For a more comprehensive description, see the authors' book [17], where detailed numerical experiments for accuracy comparison of different methods are also shown. The renormalization method was proposed by Kanatani in [5], and details are discussed in [6]. The Taubin method was

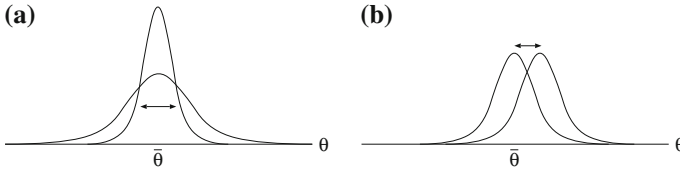


Fig. 2.6 **a** The matrix M controls the covariance of θ . **b** The matrix N controls the bias of θ

proposed in [28], but his derivation was heuristic without considering the statistical properties of image noise in terms of the covariance matrix. However, the procedure can be interpreted as described in this chapter. The HyperLS method was proposed by the author's group in [11, 12, 24], which was later extended to hyper-renormalization in [9, 10].

The fact that all known algebraic methods can be generalized in the form of Eq. (2.31) was pointed out in [10]. Because noise is assumed to occur randomly, the noisy data are random variables having probability distributions. Therefore the value θ computed from them is also a random variable and has its probability distribution, and the choice of the matrices M and N affects the probability distribution of the computed θ . According to the detailed error analysis in [10], the choice of M controls the covariance of θ (Fig. 2.6a), whereas the choice of N controls its bias (Fig. 2.6b). If we choose M to be the second row of Eq. (2.32), we can show that the covariance matrix of θ attains the theoretical accuracy bound called the *KCR (Kanatani-Cramer-Rao) lower bound* (this is discussed in Chap. 16) except for $O(\sigma^4)$ terms. If we choose N to be the fifth row of Eq. (2.33), we can show that the bias of θ is 0 except for $O(\sigma^4)$ terms (this is discussed in Chap. 14). Hence, the accuracy of hyper-renormalization cannot be improved any further except for higher-order terms of σ . In this sense, hyper-renormalization is theoretically optimal. It has also been shown in [16] that the Sampson error minimization with hyper-accurate correction has theoretically the same order of accuracy (this is discussed in Chap. 15). For detailed performance comparison of various ellipse fitting methods, see [17].

The FNS scheme for minimizing the Sampson error was introduced by Chojnacki et al. [1]. They called Eq. (2.36) the *ALM (approximated maximum likelihood)*, but later the term “Sampson error” came to be used widely. This name originates from Sampson [26] who studied ellipse fitting in early days. The minimization scheme that Sampson used is essentially the iterative reweight (Procedure 2.2), but it does not strictly minimize Eq. (2.36), because the numerator part and the denominator part are updated separately. In Step 4 of Procedure 2.5, we can compute the unit eigenvector for the eigenvalue of the smallest absolute value, but it was experimentally found by Chojnacki et al. [1] and the authors' group [13] that the iterations converge faster using the smallest eigenvalue. There exists an alternative iterative scheme closely resembling FNS called *HEIV (heteroscedastic errors-in-variables)* [18, 21]; the computed solution is the same as FNS. The general strategy of minimizing the geometric distance by repeating Sampson error minimization was presented by the authors in [15], and its use for ellipse fitting was presented in [14].

When the noise distribution is Gaussian, the geometric distance minimization is equivalent to what is called in statistics *maximum likelihood* (ML; we discuss this in Chap. 15). The fact that the solution has statistical bias has been pointed out by many people. The bias originates from the fact that an ellipse is a convex curve: if a point on the curve is randomly displaced, the probability of falling outside is larger than the probability of falling inside. Given a datapoint, its closest point on the ellipse is the foot of the perpendicular line from it to the ellipse, and the probability of on which side its true position is more likely depends on the shape and the curvature of the ellipse in the neighborhood. Okatani and Deguchi [22] analyzed this for removing the bias. They also used a technique called the *projected score* to remove bias [23]. The hyper-accurate correction described in this chapter was introduced by the authors in [7, 8, 16].

The ellipse-specific method of Fitzgibbon et al. was published in [3]. The method of Szpak et al. was presented in [27], and the method of random sampling was given in [19]. For their detailed performance comparison, see [17].

The RANSAC technique for outlier removal was proposed by Fischler and Bolles [2]. Its principle is to select randomly a minimum set of datapoints to fit a curve (or any shape) many times, each time counting the number of datapoints nearby, and to output the curve to which the largest number of datapoints is close. Hence, a small number of datapoints are allowed to be far apart, and they are regarded as outliers. Alternatively, instead of counting the number of points nearby, we may sort the square distances of all the datapoints from the curve, evaluate the median, and output the curve for which the median is the smallest. This is called *LMedS (least median of squares)* [25]. This also ignores those datapoints far apart from the curve. Instead of ignoring far apart points, we may use a distance function that does not penalize far apart points very much. Such a method is generally called *M-estimation*, for which various types of distance functions have been studied [4]. An alternative approach for the same purpose is the use of the L_p -norm for $p < 2$, typically $p = 1$. These methods make no distinction regarding whether the datapoints are continuously connected, but techniques for classifying edge segments into elliptic and nonelliptic arcs have also been studied [20].

Problems

2.1 Fitting a line in the form of $n_1x + n_2y + n_3f_0 = 0$ to points (x_α, y_α) , $\alpha = 1, \dots, N$, can be viewed as the problem of computing $\mathbf{n}$, which can be normalized to a unit vector, such that $(\mathbf{n}, \xi_\alpha) \approx 0$, $\alpha = 1, \dots, N$, where

$$\xi_\alpha = \begin{pmatrix} x_\alpha \\ y_\alpha \\ f_0 \end{pmatrix}, \quad \mathbf{n} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}. \quad (2.61)$$

- (1) Write down the least-squares procedure for computing this.
- (2) If the noise terms Δx_α and Δy_α are subject to a mutually independent Gaussian distribution of mean 0 and variance σ^2 , how is their covariance matrix $V[\xi_\alpha]$ defined?

2.2 Specifically write down the procedure for the Taubin method as the computation of the initial solution of Procedure 2.3.

2.3 Specifically write down the HyperLS procedure as the computation of the initial solution of Procedure 2.4.

2.4 (1) Define the vector $\mathbf{x}$ and the symmetric matrix $\mathbf{Q}$ by

$$\mathbf{x} = \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} A & B & D \\ B & C & E \\ D & E & F \end{pmatrix}. \quad (2.62)$$

Show that the ellipse equation of Eq. (2.1) is written in the form

$$(\mathbf{x}, \mathbf{Q}\mathbf{x}) = 0. \quad (2.63)$$

(2) Show that the square distance d_α^2 of point (x_α, y_α) from the ellipse of Eq. (2.63) is written, up to high-order small noise terms, in the form

$$d_\alpha^2 \approx \frac{f_0^2}{4} \frac{(\mathbf{x}_\alpha, \mathbf{Q}\mathbf{x}_\alpha)^2}{(\mathbf{Q}\mathbf{x}_\alpha, \mathbf{P}_k \mathbf{Q}\mathbf{x}_\alpha)}, \quad (2.64)$$

where $\mathbf{P}_k$ is the matrix defined by

$$\mathbf{P}_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (2.65)$$

(3) Show that in terms of the vectors $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$ Eq. (2.64) is written as

$$d_\alpha^2 \approx \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}. \quad (2.66)$$

2.5 (1) Show that the derivative of Eq. (2.36) is given by Eq. (2.41).

(2) Show that the eigenvalue λ in Eq. (2.39) is 0 after the FNS iterations have converged.

2.6 (1) Show that for the ellipse given by Eq. (2.63), the point $(\bar{x}_\alpha, \bar{y}_\alpha)$ that minimizes Eq. (2.47) is written up to high-order terms in $\hat{x}_\alpha - \bar{x}_\alpha$ and $\hat{y}_\alpha - \bar{y}_\alpha$ as

$$\begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \end{pmatrix} = \begin{pmatrix} x_\alpha \\ y_\alpha \end{pmatrix} - \frac{(\hat{\mathbf{x}}_\alpha, \mathbf{Q}\hat{\mathbf{x}}'_\alpha) + 2(\mathbf{Q}\hat{\mathbf{x}}'_\alpha, \tilde{\mathbf{x}}_\alpha)}{2(\mathbf{Q}\hat{\mathbf{x}}_\alpha, \mathbf{P}_k \mathbf{Q}\hat{\mathbf{x}}_\alpha)} \begin{pmatrix} A & B & D \\ B & C & E \end{pmatrix} \begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \\ f_0 \end{pmatrix}, \quad (2.67)$$

where A, B, C, D , and E are the elements of the matrix $\mathbf{Q}$ in Eq. (2.62), and $\hat{\mathbf{x}}_\alpha$ and $\tilde{\mathbf{x}}_\alpha$ are defined as follows (the point $(\tilde{x}_\alpha, \tilde{y}_\alpha)$ is defined by Eq. (2.48)).

$$\hat{\mathbf{x}}_\alpha = \begin{pmatrix} \hat{x}_\alpha/f_0 \\ \hat{y}_\alpha/f_0 \\ 1 \end{pmatrix}, \quad \tilde{\mathbf{x}}_\alpha = \begin{pmatrix} \tilde{x}_\alpha/f_0 \\ \tilde{y}_\alpha/f_0 \\ 0 \end{pmatrix}. \quad (2.68)$$

- (2) Show that if we define the 9D vector ξ_α^* by Eq. (2.42), Eq. (2.67) can be written in the form

$$\begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \end{pmatrix} = \begin{pmatrix} x_\alpha \\ y_\alpha \end{pmatrix} - \frac{2(\xi_\alpha^*, \theta)}{(\theta, V_0[\hat{\xi}_\alpha] \theta)} \begin{pmatrix} \theta_1 & \theta_2 & \theta_4 \\ \theta_2 & \theta_3 & \theta_5 \end{pmatrix} \begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \\ f_0 \end{pmatrix}, \quad (2.69)$$

where $V_0[\hat{\xi}_\alpha]$ is the normalized covariance matrix obtained by replacing $(\bar{x}, \bar{y})$ in Eq. (2.15) by $(\hat{x}_\alpha, \hat{y}_\alpha)$.

- (3) Show that by replacing $(\bar{x}_\alpha, \bar{y}_\alpha)$ in Eq. (2.34) by $(\hat{x}_\alpha, \hat{y}_\alpha)$, we can rewrite the geometric distance S in the form of Eq. (2.43).

References

1. W. Chojnacki, M.J. Brooks, A. van den Hengel, D. Gawley, On the fitting of surfaces to data with covariances. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(11), 1294–1303 (2000)
2. M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981)
3. A. Fitzgibbon, M. Pilu, R.B. Fisher, Direct least squares fitting of ellipses. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5), 476–480 (1999)
4. P.J. Huber, *Robust Statistics*, 2nd edn. (Wiley, Hoboken, 2009)
5. K. Kanatani, Renormalization for unbiased estimation, in *Proceedings of 4th International Conference on Computer Vision*, Berlin, Germany (1993), pp. 599–606
6. K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice* (Elsevier, Amsterdam, The Netherlands, 1996) (Reprinted by Dover, New York, U.S., 2005)
7. K. Kanatani, Ellipse fitting with hyperaccuracy. *IEICE Trans. Inf. Syst.* **E89-D**(10), 2653–2660 (2006)
8. K. Kanatani, Statistical optimization for geometric fitting: theoretical accuracy bound and high order error analysis. *Int. J. Comput. Vis.* **80**(2), 167–188 (2008)
9. K. Kanatani, A. Al-Sharadqah, N. Chernov, Y. Sugaya, Renormalization returns: hyper-renormalization and its applications, in *Proceedings of 12th European Conference on Computer Vision*, Firenze, Italy, October 2012
10. K. Kanatani, A. Al-Sharadqah, N. Chernov, Y. Sugaya, Hyper-renormalization: non-minimization approach for geometric estimation. *IPSP Trans. Comput. Vis. Appl.* **6**, 143–159 (2014)
11. K. Kanatani, P. Rangarajan, Hyper least squares fitting of circles and ellipses. *Comput. Stat. Data Anal.* **55**(6), 2197–2208 (2011)
12. K. Kanatani, P. Rangarajan, Y. Sugaya, H. Niitsuma, HyperLS and its applications. *IPSP Trans. Comput. Vis. Appl.* **3**, 80–94 (2011)
13. K. Kanatani, Y. Sugaya, Performance evaluation of iterative geometric fitting algorithms. *Comput. Stat. Data Anal.* **52**(2), 1208–1222 (2007)
14. K. Kanatani, Y. Sugaya, Compact algorithm for strictly ML ellipse fitting, in *Proceedings of 19th International Conference on Pattern Recognition*, Tampa, FL, U.S. (2008)
15. K. Kanatani, Y. Sugaya, Unified computation of strict maximum likelihood for geometric fitting. *J. Math. Imaging Vis.* **38**(1), 1–13 (2010)

16. K. Kanatani, Y. Sugaya, Hyperaccurate correction of maximum likelihood for geometric estimation. *IPSJ Trans. Comput. Vis. Appl.* **5**, 19–29 (2013)
17. K. Kanatani, Y. Sugaya, K. Kanazawa, *Ellipse Fitting for Computer Vision: Implementation and Applications* (Morgan and Claypool, San Rafael, 2016)
18. Y. Leedan, P. Meer, Heteroscedastic regression in computer vision: problems with bilinear constraint. *Int. J. Comput. Vis.* **37**(2), 127–150 (2000)
19. T. Masuzaki, Y. Sugaya, K. Kanatani, High accuracy ellipse-specific fitting, in *Proceedings of 6th Pacific-Rim Symposium on Image and Video Technology*, Guanajuato, Mexico (2013), pp. 314–324
20. T. Masuzaki, Y. Sugaya, K. Kanatani, Floor-wall boundary estimation by ellipse fitting, in *Proceedings IEEE 7th International Conference on Robotics, Automation and Mechatronics*, Angkor Wat, Cambodia (2015), pp. 30–35
21. J. Matei, P. Meer, Estimation of nonlinear errors-in-variables models for computer vision applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(10), 1537–1552 (2006)
22. T. Okatani, K. Deguchi, On bias correction for geometric parameter estimation in computer vision, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, U.S. (2009), pp. 959–966
23. T. Okatani, K. Deguchi, Improving accuracy of geometric parameter estimation using projected score method, in *Proceedings of 12th International Conference on Computer Vision*, Kyoto, Japan (2009), pp. 1733–1740
24. P. Rangarajan, K. Kanatani, Improved algebraic methods for circle fitting. *Electron. J. Stat.* **3**(2009), 1075–1082 (2009)
25. P.J. Rousseeuw, A.M. Leroy, *Robust Regression and Outlier Detection* (Wiley, New York, 1987)
26. P.D. Sampson, Fitting conic sections to “very scattered” data: an iterative refinement of the Bookstein Algorithm. *Comput. Vis. Image Process.* **18**(1), 97–108 (1982)
27. Z.L. Szpak, W. Chojnacki, A. van den Hengel, Guaranteed ellipse fitting with a confidence region and an uncertainty measure for centre, axes, and orientation. *J. Math. Imaging Vis.* **52**(2), 173–199 (2015)
28. G. Taubin, Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(11), 1115–1138 (1991)

Guide to 3D Vision Computation

Geometric Analysis and Implementation

Kanatani, K.; Sugaya, Y.; Kanazawa, Y.

2016, XI, 321 p. 54 illus., 10 illus. in color., Hardcover

ISBN: 978-3-319-48492-1