SCIENCE CHINA
Technological Sciences

• Article •

January 2016 Vol.59 No.1: 135–148
doi: 10.1007/s11431-015-5958-1

Pose optimization based on integral of the distance between line segments

ZHANG YueQiang[1,2], LI Xin[1,2], LIU HaiBo[1,2], SHANG Yang[1,2*] & YU QiFeng[1,2]

[1] *College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China;*
[2] *Hunan Provincial Key Laboratory of Image Measurement and Vision Navigation, Changsha 410073, China*

Received April 15, 2015; accepted August 3, 2015; published online November 12, 2015

In this paper, new solutions for the problem of pose estimation from correspondences between 3D model lines and 2D image lines are proposed. Traditional line-based pose estimation methods rely on the assumption that the noises (perpendicular to the line) for the two endpoints are statistically independent. However, these two noises are in fact negatively correlated when the image line segment is fitted using the least-squares technique. Therefore, we design a new error function expressed by the average integral of the distance between line segments. Three least-squares techniques that optimize both the rotation and translation simultaneously are proposed in which the new error function is exploited. In addition, Lie group formalism is utilized to describe the pose parameters, and then, the optimization problem can be solved by means of a simple iterative least squares method. To enhance the robustness to outliers existing in the match data, an M-estimation method is developed to convert the pose optimization problem into an iterative reweighted least squares problem. The proposed methods are validated through experiments using both synthetic and real-world data. The experimental results show that the proposed methods yield a clearly higher precision than the traditional methods.

machine vision, perspective-n-line problem, line distance function, pose optimization, M-estimation

Citation: Zhang Y Q, Li X, Liu H B, et al. Pose optimization based on integral of the distance between line segments. Sci China Tech Sci, 2016, 59: 135–148, doi: 10.1007/s11431-015-5958-1

1 Introduction

The goal of the pose estimation problem is to determine the position and attitude of a calibrated camera or the pose of an object with respect to the camera from known 3D features and their image projections (corresponding 2D features). This constitutes the core of many computer vision tasks and has numerous applications in other areas, such as robotic manipulation and augmented reality, as well as space rendezvous technology. Regarding point features, much noteworthy progress [1–5] has been made on the well-known Perspective-n-Point (PnP) problem in the last two decades.

However, from a practical point of view, it is frequently advantageous to use lines to estimate the pose of man-made objects having little texture. Indeed, line features are abundant in such objects and can be detected more accurately and reliably than points. Therefore, the solution of the Perspective-n-Line (PnL) problem is both desirable and indispensable in many applications. The use of 3D-2D line correspondences to recover the position and attitude of a calibrated camera or the pose of an object has received significant research attention for more than two decades. In the earliest papers on the subject, a closed-form solution for the PnL problem with three line correspondences was proposed [6,7]. Navab and Faugeras [8] gave the existence conditions and the maximum number of possible solutions for the P3L problem and showed that a maximum of three solutions

*Corresponding author (email: shangyang1977@nudt.edu.cn)

exists for three skew lines.

To find a unique solution for pose determination and obtain precise results when noise exists, more than three line correspondences should be established. Liu et al. [9] solved the rotation matrix and the translation vector separately. Eight or more line correspondences were utilized to linearly constrain the rotation matrix, and then, a linear least squares method was employed to obtain an up-to-scale estimation of the rotation. However, in the case of noisy observations, this approach does not necessarily achieve an appropriate orthonormal matrix. Adnan [10] proposed a method that employs the lifting approach to convert the polynomials determined by a four or more line correspondences constraint to a linear equation system in the components of the rotation matrix. However, this approach has $O(N^2)$ computational complexity, where N is the number of line correspondences used. Moreover, its performance is degraded severely with increasing measurement-noise variance. Recently, Mirzaei and Roumeliotis [11] utilized algebraic geometry techniques to directly solve a system of multivariate polynomial equations formed by the optimality conditions of the nonlinear least-squares problem. Although it has $O(N)$ computational complexity, the method is still computationally expensive and returns an excessive number of candidate solutions, with a maximum of 27. Elqursh and Elgammal [12] proposed an algorithm using three lines, where two are parallel and orthogonal to the third. They also designed a robust hypothesize-and-test framework using which such lines can be detected. However, their approach is suitable only for an environment where most of the lines are either parallel or orthogonal to each other. Zhang et al. [13] divided the reference lines into triplets by selecting a rotation axis in the camera frame, and then, recovered the optimum solution from the roots of the derivative of the 16th order cost function. This method also has $O(N)$ computational complexity. Liu et al. [14] proposed a similar linear method in which the angle depth of lines in the local coordinate system formed by two appropriate lines is recovered and the PnL problem is converted to an absolute orientation problem. However, this method is computationally expensive because the angle depth for each line has to be estimated.

Several iterative methods have also been presented to estimate the pose parameters from line correspondences using nonlinear least-squares techniques. Liu and Phong et al. [9,15] proposed an iterative method to estimate the rotation first and then the translation (called R_then_T). Kumar and Hanson [16] improved the iterative method by optimizing the rotation and translation simultaneously (called R_and_T) and showed that the performance of the R_and_T algorithm is far superior to that of the R_then_T algorithm. Moreover, an algorithm called R_and_T_mod was proposed in which the 2D alignment error between the image line segment and the 3D model line is minimized, which performs more robustly than the R_and_T_img method based on the infinite extended image line. Christy and Horaud [17] extended the iterative improvement of a linear camera model already used for points-based pose estimation [18] to lines and proposed an iterative algorithm to recover the pose parameters based on weak-perspective or para-perspective camera models. Hanek et al. [19] presented a noise model to describe the probabilistic relationship between 3D lines and their corresponding noisy finite image lines and utilized a maximum-likelihood approach to estimate the pose parameters. David et al. [20] improved the SoftPOSIT algorithm [21] designed for point features to iteratively estimate the pose parameters and the correspondences of model lines to image lines simultaneously. Inspired by the point-based orthogonal iteration algorithm [4], Zhang et al. [22] proposed an orthogonal iteration method for line features in which the objective function in the object space is iteratively optimized.

However, all these iterative methods minimize the error functions based either on the distance from any point in the line to the corresponding line [9,15,22,23] or the distance from the endpoints to the corresponding line [16,19,20]. The former error functions regard the infinite image lines as the observations and neglect significant information about the position and length of the observed line segments, whilst the latter rely on the assumption that the noise models of two endpoints for the image line segment along the normal vector are statistically independent. We prove that these two noises are incompletely independent and negatively correlated. Therefore, in this study a new error function between lines based on the mean of the distance integral was developed. In contrast to the traditional error function based on the distance from the endpoint to the corresponding line, the new error function contains the distance from the midpoint to the line inherently and gives it a weight 4 times that of the endpoint. According to the new error function between lines, we propose three pose optimization methods. Moreover, based on the error function presented in [24], we propose three additional methods.

2 Error function

2.1 Perspective projection

Throughout this study, a calibrated camera with a perspective projection model was utilized. As shown in Figure 1, the relationship between a 3D world $\boldsymbol{P}$ and its projection $\boldsymbol{p}$ can be given by

$$p_x = f_x \frac{(\boldsymbol{RP}+\boldsymbol{T})_x}{(\boldsymbol{RP}+\boldsymbol{T})_z}, p_y = f_y \frac{(\boldsymbol{RP}+\boldsymbol{T})_y}{(\boldsymbol{RP}+\boldsymbol{T})_z}, \tag{1}$$

where $\boldsymbol{p} = (p_x, p_y)^{\mathrm{T}}$, $\boldsymbol{P}^{\mathrm{C}} = \boldsymbol{RP} + \boldsymbol{T}$ is the coordinate in the camera frame for $\boldsymbol{P}$, and $(\bullet)_z$ is the z-coordinate. $\boldsymbol{R}$ and $\boldsymbol{T}$ are the rotation matrix and translation vector from the reference frame (the object frame) to the camera frame. f_x and f_y are the equivalent focal length.

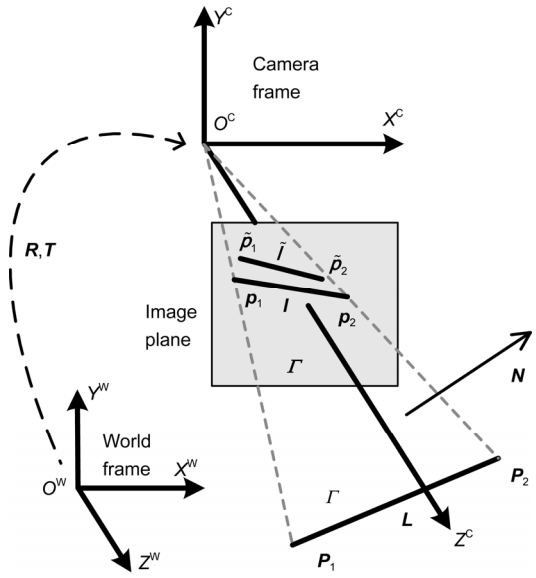


Figure 1 Perspective projection.

The 3D model line segment $\boldsymbol{L}$ and its projection $\boldsymbol{I}$ onto the image plane are represented by their endpoints ($\boldsymbol{P}_1$,$\boldsymbol{P}_2$) and ($\boldsymbol{p}_1$,$\boldsymbol{p}_2$). The image line segment $\tilde{\boldsymbol{l}}$ is represented by its two 2D endpoints $(\tilde{\boldsymbol{p}}_1, \tilde{\boldsymbol{p}}_2)$. The perspective projection of a 3D line segment is expressed by the projection of the two endpoints, as shown in Figure 1.

2.2 Error function according to endpoint distance

Figure 2 shows a typical distance function between two lines. The thin line represents the line segment, while the bold line represents the infinite line. Then, the error function between the line pair can be defined as the sum of the squares of the distances from two endpoints of one line segment to the other line, which is given by

$$d_{\mathrm{e}}(\tilde{\boldsymbol{l}},\boldsymbol{l}) = \left(s_1^2 + s_2^2\right), \tag{2}$$

where s_1 and s_2 are the distances from endpoints $\tilde{\boldsymbol{p}}_1$ and $\tilde{\boldsymbol{p}}_2$ of the line segment $\tilde{\boldsymbol{l}}$ to the line $\boldsymbol{l}$, which is hereafter referred to as the endpoint distance.

According to whether the image line is infinite or not, two methods can be used to develop error functions. On the basis of the perspective projection of line segments, a detailed discussion on obtaining the expression of the corresponding endpoint distances is introduced in the following.

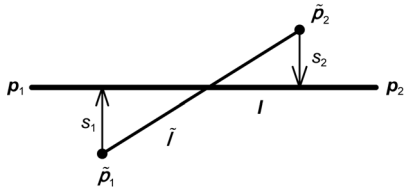


Figure 2 Distance from endpoint to line.

In our approach, an image line is represented in the form of (Θ, ρ) parameters:

$$p_x \cos\Theta + p_y \sin\Theta - \rho = 0, \tag{3}$$

where Θ is the angle between the normal vector of the image line and the horizontal axis and ρ is the distance from the origin to the image line.

Substituting eq. (1) into (3), we have

$$\frac{\boldsymbol{N}^{\mathrm{T}}(\boldsymbol{RP}+\boldsymbol{T})}{(\boldsymbol{RP}+\boldsymbol{T})_z} = 0, \tag{4}$$

where $\boldsymbol{N} = \left(f_x \cos\Theta, f_y \sin\Theta, -\rho\right)^{\mathrm{T}}$ is the normal vector of the projection plane formed by the optical center and the image line. The left side of eq. (4) corresponds to the distance from the endpoint of the projected model line segment to the infinite image line.

Let $\bar{\boldsymbol{N}} = \boldsymbol{N}/|\boldsymbol{N}|$, where $\bar{\boldsymbol{N}}$ is the unit normal vector of the projection plane; then, eq. (4) can be rewritten as

$$\bar{\boldsymbol{N}}^{\mathrm{T}}(\boldsymbol{RP}+\boldsymbol{T}) = 0. \tag{5}$$

The left side of eq. (5) corresponds to the distance from the endpoint of the model line segment to the projection plane.

In the two constraints described in eqs. (4) and (5) above, the 3D line segment is aligned with the infinitely extended image line. The image line segment can also be aligned with the projection of the 3D line. In the finite image line case, the normal vector of the projection plane can be given by

$$\boldsymbol{N} = k(\boldsymbol{RP}_1+\boldsymbol{T}) \times (\boldsymbol{RP}_2+\boldsymbol{T}), \tag{6}$$

where k is a scale factor.

Let $(\boldsymbol{RP}_1+\boldsymbol{T}) \times (\boldsymbol{RP}_2+\boldsymbol{T}) = (A_x, A_y, A_z)^{\mathrm{T}}$, and then

$$\cos\Theta = \frac{1}{f_x\lambda}A_x,\ \sin\Theta = \frac{1}{f_y\lambda}A_y,\ \rho = -\frac{1}{\lambda}A_z, \tag{7}$$

where $\lambda = \sqrt{\left(A_x/f_x\right)^2 + \left(A_y/f_y\right)^2}$.

Then, the constraint that the endpoints of the image line segment must lie in the projected model line can be given by

$$\frac{1}{\lambda}\bar{\boldsymbol{p}}^{\mathrm{T}}((\boldsymbol{RP}_1+\boldsymbol{T}) \times (\boldsymbol{RP}_2+\boldsymbol{T})) = 0, \tag{8}$$

where $\bar{\boldsymbol{p}} = (p_x/f_x, p_y/f_y, 1)^{\mathrm{T}}$. The distance from the endpoint of the image line segment to the projected model line is given by the left side of eq. (8).

2.3 Error function according to distance integral

When the localization of the image line segment is sufficiently accurate, the error functions defined by eqs. (4), (5) and (8) can suffice to optimize the pose successfully. However, the location of the endpoint and the continuity of the line may be unreliable because of the image noise, illumination, and self-occlusion, which has a severe effect on the distance functions defined above, in particular, those defined by eqs. (4) and (5). Moreover, the distance function defined by eq. (8) relies on the assumption that the noise models of two endpoints for the image line segment along the normal vector are statistically independent. It can be proved that these two noises are negatively correlated when the series of edge points are fitted into the line segment by using the least squares technique, as described in Appendix A.

In [24], the error function according to the distance integral as shown in Figure 3, is designed for recovering the structure and motion from line segments, given by eq. (9):

$$d_1(\boldsymbol{l}_1,\boldsymbol{l}_2)=\int_0^l s(t)^2\,\mathrm{d}t=\frac{l}{3}\left(s_1^2+s_1s_2+s_2^2\right), \tag{9}$$

where l is the length of the line segment $\boldsymbol{l}_1$.

As can be seen in eq. (9), the distance function is weighted by the line length explicitly, which is a desirable property since a longer line segment can be localized more accurately than a shorter one [24]. However, in some particular views, the length of a long line segment may be several times, even several decuples, that of a short one. In the above formulation of the distance function, the long line segment is overweighted, while the contribution of the short one can easily be undervalued. If the long line segment is not an image model line but rather an outlier near the projected model line, then the pose optimizer is disturbed.

Therefore, we design a new error function between line segments.

Figure 4 shows the new error function in the plane. $\boldsymbol{q}_1$, $\boldsymbol{q}_2$ are the endpoints of line segment $\boldsymbol{l}_1$. $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ are the endpoints of line segment $\boldsymbol{l}_2$. The midpoints of $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ are $\boldsymbol{q}_0$ and $\boldsymbol{p}_0$, respectively. Then, we define the error function between the two line segments as

$$d_{\mathrm{n}}(\boldsymbol{l}_1,\boldsymbol{l}_2)=\int_0^1\left\|\boldsymbol{q}(t)-\boldsymbol{p}(t)\right\|^2\mathrm{d}t=\frac{1}{6}\left(\boldsymbol{s}_1^{\mathrm{T}}\boldsymbol{s}_1+\boldsymbol{s}_2^{\mathrm{T}}\boldsymbol{s}_2+4\boldsymbol{s}_0^{\mathrm{T}}\boldsymbol{s}_0\right), \tag{10}$$

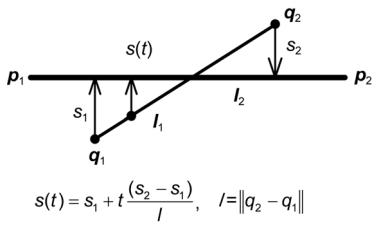


Figure 3 Error function between line segments according to the distance integral.

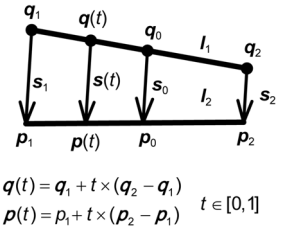


Figure 4 New error function between line segments according to the mean of the distance integral.

where $\boldsymbol{s}_1=\boldsymbol{q}_1-\boldsymbol{p}_1$, $\boldsymbol{s}_2=\boldsymbol{q}_2-\boldsymbol{p}_2$, $\boldsymbol{s}_0=\boldsymbol{q}_0-\boldsymbol{p}_0$.

It is clear that our new error function is not weighted by the length of the line segment explicitly, as compared with eq. (9). However, the feature of the proposed error function is discussed with reference to Figure 5 in the following.

In Figure 5(a), the two line segments have the same length and $\boldsymbol{s}_1$=$\boldsymbol{s}_2$. The distance between the two line segments is $\|\boldsymbol{s}_2\|^2$, whilst the distance in Figure 5(b) is $\frac{1}{3}\|\boldsymbol{s}_2\|^2$. When the line segments becoming longer, as seen in Figure 5(c), the distance remains unchanged, since the distances for both the endpoints and midpoint are the same. If the angle between the segments shown in Figure 5(b) is fixed and the lengths become longer, the distance becomes greater, as seen in Figure 5(d). It is clear that the proposed distance function is also weighted by the length of the line segments inherently.

According to eq. (9), three error functions between a 3D model line and a 2D image line can be defined. In the infinite image line case, two error functions can be given according to the distance from the projected model line segment in the image plane to the image line and the 3D distance from the model line segment to the projected model line in the projection plane.

$$d_1(\tilde{\boldsymbol{l}},\boldsymbol{L})=d_1(\boldsymbol{l},\tilde{\boldsymbol{l}}), \tag{11}$$

$$d_2(\tilde{\boldsymbol{l}},\boldsymbol{L})=d_1(\boldsymbol{L},\hat{\boldsymbol{L}}), \tag{12}$$

where $\hat{\boldsymbol{L}}$ is the projected extended line of the model line segment $\boldsymbol{L}$ in the projection plane along the normal vector.

In the finite image line case, the error function can be given by

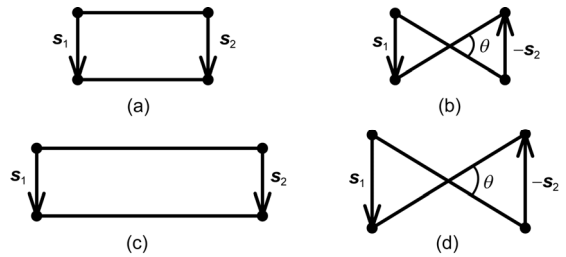


Figure 5 Examples of new error function between line segments. (a) Case 1; (b) Case 2; (c) Case 3; (d) Case 4.

$$d_3(\tilde{\boldsymbol{l}},\boldsymbol{L}) = d_1(\tilde{\boldsymbol{l}},\boldsymbol{l}). \tag{13}$$

According to the proposed distance function described in eq. (10), three additional error functions can be defined. In the infinite image line case, since only the part of the observed line nearest to the projected model line segment contributes to the distance measure, we can define the error function between the projected model line segment and the image line as

$$d_4(\tilde{\boldsymbol{l}},\boldsymbol{L}) = d_n((\tilde{\boldsymbol{p}}_1,\tilde{\boldsymbol{p}}_2),(\boldsymbol{p}_1,\boldsymbol{p}_2)), \tag{14}$$

where $\boldsymbol{p}_1, \boldsymbol{p}_2$ are the endpoints of the projected model line in the image plane. $\tilde{\boldsymbol{p}}_1, \tilde{\boldsymbol{p}}_2$ are the projections of $\boldsymbol{p}_1, \boldsymbol{p}_2$ on the line $\tilde{\boldsymbol{l}}$ along the normal direction of $\tilde{\boldsymbol{l}}$, respectively.

The error function between the model line segment and the projection plane is given by

$$d_5(\tilde{\boldsymbol{l}},\boldsymbol{L}) = d_n((\tilde{\boldsymbol{P}}_1,\tilde{\boldsymbol{P}}_2),(\boldsymbol{P}_1,\boldsymbol{P}_2)), \tag{15}$$

where $\tilde{\boldsymbol{P}}_1$, $\tilde{\boldsymbol{P}}_2$ is the projection of $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ on the projection plane along the normal vector.

In the finite image line case, the error function can be given by

$$d_6(\tilde{\boldsymbol{l}},\boldsymbol{L}) = d_n((\tilde{\boldsymbol{p}}_1,\tilde{\boldsymbol{p}}_2),(\boldsymbol{p}_1,\boldsymbol{p}_2)), \tag{16}$$

where $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ is the projection of $\tilde{\boldsymbol{P}}_1$, $\tilde{\boldsymbol{P}}_2$ on the projected model line along the normal vector.

3 Least squares techniques

3.1 Objective functions

Let the correspondences of the model and image line segments be $\left\{\left(\tilde{\boldsymbol{l}}_1,\boldsymbol{L}_1\right),\left(\tilde{\boldsymbol{l}}_2,\boldsymbol{L}_2\right),\cdots,\left(\tilde{\boldsymbol{l}}_N,\boldsymbol{L}_N\right)\right\}$, where N is the number of line pairs. The real pose can be estimated by minimizing the objective function described as

$$E = \sum_{i=1}^{N} d\left(\tilde{\boldsymbol{l}}_i,\boldsymbol{L}_i\right). \tag{17}$$

If eq. (2) is selected as the error function, then the optimization problem defined by eq. (17) can be modified as

$$E = \sum_{i=1}^{N} d_e\left(\tilde{\boldsymbol{l}}_i,\boldsymbol{L}_i\right) = \sum_{i=1}^{N}\sum_{j=1}^{2}\left(s_j^i(\boldsymbol{r}) - \overline{s}_j^i\right)^2, \tag{18}$$

where $s(\boldsymbol{r})$ is the observed data corresponding to the endpoint distance, $\boldsymbol{r}$ is the pose parameters, and $\overline{s}$ is the desired observed data, equal to 0. The minimization of eq. (18) can be handled using a least-squares approach.

In studies in the literature [16], the distance based on eqs. (4), (5), and (8) was selected as the endpoint distance $s(\boldsymbol{r})$.

If the endpoint distance is given by eq. (5), the first objective function is developed:

$$E_1 = \sum_{i=1}^{N}\sum_{j=1}^{2}\left(\overline{\boldsymbol{N}}_i^{\mathrm{T}}\left(\boldsymbol{R}\boldsymbol{P}_j^i + \boldsymbol{T}\right)\right)^2. \tag{19}$$

The resulting algorithm that minimizes E_1 is called R_and_T [16].

A similar endpoint distance defined in the image plane is given by eq. (4), and then, the second optional objective function can be obtained by

$$E_2 = \sum_{i=1}^{N}\sum_{j=1}^{2}\left(\frac{\boldsymbol{N}_i^{\mathrm{T}}(\boldsymbol{R}\boldsymbol{P}_j^i + \boldsymbol{T})}{(\boldsymbol{R}\boldsymbol{P}_j^i + \boldsymbol{T})_z}\right)^2. \tag{20}$$

The resulting algorithm that minimizes E_2 is called R_and_T_img [16].

If the endpoint distance is given by eq. (8), another objective function can be developed as

$$E_3 = \sum_{i=1}^{N}\sum_{j=1}^{2}\left(\frac{1}{\lambda_i}\left(\overline{\boldsymbol{p}}_j^i\right)^{\mathrm{T}}\left(\left(\boldsymbol{R}\boldsymbol{P}_1^i + \boldsymbol{T}\right)\times\left(\boldsymbol{R}\boldsymbol{P}_2^i + \boldsymbol{T}\right)\right)\right)^2. \tag{21}$$

The resulting algorithm that minimizes E_3 is called R_and_T_mod [16].

In contrast to the objective function defined by eq. (18), which contains only the distance from the endpoints to the corresponding line, three additional objective functions can be defined according to the error functions defined in eqs. (11)–(13).

$$E_4 = \sum_{i=1}^{N} d_2(\tilde{\boldsymbol{l}}_i,\boldsymbol{L}_j) = \sum_{i=1}^{N} d_1(\boldsymbol{L}_j,\tilde{\boldsymbol{L}}_i), \tag{22}$$

$$E_5 = \sum_{i=1}^{N} d_1(\tilde{\boldsymbol{l}}_i,\boldsymbol{L}_j) = \sum_{i=1}^{N} d_1(\boldsymbol{l}_j,\tilde{\boldsymbol{l}}_i), \tag{23}$$

$$E_6 = \sum_{i=1}^{N} d_3(\tilde{\boldsymbol{l}}_i,\boldsymbol{L}_j) = \sum_{i=1}^{N} d_1(\tilde{\boldsymbol{l}}_i,\boldsymbol{l}_j). \tag{24}$$

The resulting algorithms that minimize E_4, E_5, E_6 are called R_and_T_len, R_and_T_img _len, and R_and_T_ mod_len, respectively.

According to the proposed new error functions described in eqs. (14)–(16), three objective functions are defined:

$$E_7 = \sum_{i=1}^{N} d_5(\tilde{\boldsymbol{l}}_i,\boldsymbol{L}_j) = \sum_{i=1}^{N} d_n((\tilde{\boldsymbol{P}}_1,\tilde{\boldsymbol{P}}_2),(\boldsymbol{P}_1,\boldsymbol{P}_2)), \tag{25}$$

$$E_8 = \sum_{i=1}^{N} d_4(\tilde{\boldsymbol{l}}_i,\boldsymbol{L}_j) = \sum_{i=1}^{N} d_n((\tilde{\boldsymbol{p}}_1,\tilde{\boldsymbol{p}}_2),(\boldsymbol{p}_1,\boldsymbol{p}_2)), \tag{26}$$

$$E_9 = \sum_{i=1}^{N} d_6(\tilde{\boldsymbol{l}}_i,\boldsymbol{L}_j) = \sum_{i=1}^{N} d_n((\tilde{\boldsymbol{p}}_1,\tilde{\boldsymbol{p}}_2),(\boldsymbol{p}_1,\boldsymbol{p}_2)). \tag{27}$$

The resulting algorithms that minimize E_7, E_8, E_9 are called R_and_T_novel, R_and_T_img _novel, and R_and_T_mod_novel, respectively.

3.2 Nonlinear technique

In the initial case, the pose parameters can be provided by the PnL method [10,11,13]. At each iteration, the linearized error function is minimized to obtain the interframe motion vector for the rotation and translation terms. Then, the pose parameters are updated until the objective function converges to a minimum. Three endpoint distances defined by eqs. (4), (5), and (8) are utilized for the error function in E_1–E_9. The basic components of the nonlinear technique are to linearize the error terms about the current estimate for $\boldsymbol{R}$ and $\boldsymbol{T}$.

For the infinitely extended image line, we adopt the Lie group formulation to represent the rigid transformation from the reference frame to the camera frame. Assuming that we have a current estimation of pose $\boldsymbol{E}_t$, the posterior pose $\boldsymbol{E}_{t+1}$ can be computed from the prior pose $\boldsymbol{E}_t$ given the incremental motion $\boldsymbol{M}$:

$$\boldsymbol{E}_{t+1} = \boldsymbol{M}\boldsymbol{E}_t, \tag{28}$$

where $\boldsymbol{M}$ can be represented in the exponential map as

$$\boldsymbol{M} = \exp(\boldsymbol{\mu}) = \exp(\textstyle\sum_{j=1}^{6} \mu_j \boldsymbol{G}_j), \tag{29}$$

where $\boldsymbol{\mu}$ is the motion velocities corresponding to translations in the x, y, and z directions and rotations about the x, y, and z axis and $\boldsymbol{G}_j$ is the group generators [25].

Then, eq. (5) can be rewritten as

$$s_j^i(\boldsymbol{r}) = \bar{\boldsymbol{N}}_i^{\mathrm{T}}(\boldsymbol{R}_{t+1}\boldsymbol{P}_j^i + \boldsymbol{T}_{t+1}) = \left(\bar{\boldsymbol{N}}_i'\right)^{\mathrm{T}}(\boldsymbol{E}_{t+1}\bar{\boldsymbol{P}}_j^i), \tag{30}$$

where $\bar{\boldsymbol{N}}_i' = \begin{pmatrix} \bar{\boldsymbol{N}}_i^{\mathrm{T}} & 0 \end{pmatrix}^{\mathrm{T}}$, $\boldsymbol{E}_{t+1} = \begin{pmatrix} \boldsymbol{R}_{t+1} & \boldsymbol{T}_{t+1} \\ \boldsymbol{0} & 1 \end{pmatrix}$, $\bar{\boldsymbol{P}}_j^i = \begin{pmatrix} \left(\boldsymbol{P}_j^i\right)^{\mathrm{T}} & 1 \end{pmatrix}^{\mathrm{T}}$.

The partial derivative of $s_j^i(\boldsymbol{r})$ with respect the kth generating motion μ_k can be computed as

$$\frac{\partial s_j^i(\boldsymbol{r})}{\partial \mu_k} = \left(\bar{\boldsymbol{N}}_i'\right)^{\mathrm{T}} \frac{\partial}{\partial \mu_k}\left(\boldsymbol{M}\boldsymbol{E}_t\bar{\boldsymbol{P}}_j^i\right) = \left(\bar{\boldsymbol{N}}_i'\right)^{\mathrm{T}} \boldsymbol{G}_k\boldsymbol{E}_t\bar{\boldsymbol{P}}_j^i. \tag{31}$$

Eq. (4) can be rewritten as

$$s_j^i(\boldsymbol{r}) = \frac{\boldsymbol{N}_i^{\mathrm{T}}(\boldsymbol{R}_{t+1}\boldsymbol{P}_j^i + \boldsymbol{T}_{t+1})}{(\boldsymbol{R}_{t+1}\boldsymbol{P}_j^i + \boldsymbol{T}_{t+1})_z} = \frac{\left(\boldsymbol{N}_i'\right)^{\mathrm{T}}(\boldsymbol{M}\boldsymbol{E}_t\bar{\boldsymbol{P}}_j^i)}{(\boldsymbol{M}\boldsymbol{E}_t\bar{\boldsymbol{P}}_j^i)_z}, \tag{32}$$

where $\boldsymbol{N}_i' = \left(\boldsymbol{N}_i^{\mathrm{T}}, 0\right)^{\mathrm{T}}$.

Then, we have

$$\frac{\partial s_j^i(\boldsymbol{r})}{\partial \mu_k} = \left(\boldsymbol{N}_i'\right)^{\mathrm{T}} \frac{\partial}{\partial \mu_k}\left(\frac{\boldsymbol{M}\boldsymbol{E}_t\bar{\boldsymbol{P}}_j^i}{(\boldsymbol{M}\boldsymbol{E}_t\bar{\boldsymbol{P}}_j^i)_z}\right) = \left(\boldsymbol{N}_i'\right)^{\mathrm{T}} \frac{\partial \boldsymbol{Q}_j^i}{\partial \hat{\boldsymbol{P}}_j^i}\frac{\partial \hat{\boldsymbol{P}}_j^i}{\partial \mu_k}, \tag{33}$$

where $\boldsymbol{Q}_j^i = \dfrac{\hat{\boldsymbol{P}}_j^i}{\left(\hat{\boldsymbol{P}}_j^i\right)_z}$, $\hat{\boldsymbol{P}}_j^i = \boldsymbol{M}\boldsymbol{E}_t\bar{\boldsymbol{P}}_j^i$ and

$$\frac{\partial \boldsymbol{Q}_j^i}{\partial \hat{\boldsymbol{P}}_j^i} = \begin{pmatrix} \boldsymbol{L}_j^i & \boldsymbol{0}_{2\times 1} \\ \boldsymbol{0}_{2\times 3} & \boldsymbol{0}_{2\times 1} \end{pmatrix}_{4\times 4},$$

$$\boldsymbol{L}_j^i = \begin{pmatrix} \dfrac{1}{z_i^C} & 0 & \dfrac{-x_i^C}{\left(z_i^C\right)^2} \\ 0 & \dfrac{1}{z_i^C} & \dfrac{-y_i^C}{\left(z_i^C\right)^2} \end{pmatrix}, \quad \frac{\partial \hat{\boldsymbol{P}}_j^i}{\partial \mu_k} = \boldsymbol{G}_k\boldsymbol{E}_t\bar{\boldsymbol{P}}_j^i.$$

For the finite image line case, we adopt the Lie group formulation to represent the rotation. Assuming that we have a current estimation of the rotation $\boldsymbol{R}_t$, the posterior rotation $\boldsymbol{R}_{t+1}$ can be computed from the prior rotation $\boldsymbol{R}_t$ given the incremental rotation $\exp(\hat{\boldsymbol{\omega}})$:

$$\boldsymbol{R}_{t+1} = \exp(\hat{\boldsymbol{\omega}})\boldsymbol{R}_t, \tag{34}$$

where $\hat{\boldsymbol{\omega}}$ is the corresponding skew-symmetric matrix of $\boldsymbol{\omega} = \begin{pmatrix} \omega_0 & \omega_1 & \omega_2 \end{pmatrix}^{\mathrm{T}}$.

Then, eq. (8) can be rewritten as

$$s_j^i(\boldsymbol{r}) = \frac{1}{\lambda_t^i}\left(\boldsymbol{R}_{t+1}^{\mathrm{T}}\bar{\boldsymbol{p}}_j^i\right)^{\mathrm{T}}\left(\left(\boldsymbol{P}_1^i - \boldsymbol{T}_t^{\mathrm{W}} - \boldsymbol{\xi}\right)\times\left(\boldsymbol{P}_2^i - \boldsymbol{T}_t^{\mathrm{W}} - \boldsymbol{\xi}\right)\right), \tag{35}$$

where $\boldsymbol{T}^{\mathrm{W}} = -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{T}$ denotes the location of the origin of the camera frame in the world frame. $\boldsymbol{\xi} = \begin{pmatrix} \xi_0 & \xi_1 & \xi_2 \end{pmatrix}^{\mathrm{T}}$ represents the motion velocities corresponding to translation in the x, y, and z directions.

The partial derivative of the error function $s_j^i(\boldsymbol{r})$ with respect the kth motion velocities can be computed as

$$\frac{\partial s_j^i(\boldsymbol{r})}{\partial \omega_k} = -\frac{1}{\lambda_t^i}\left(\boldsymbol{R}_t^{\mathrm{T}}\hat{\boldsymbol{g}}_k\bar{\boldsymbol{p}}_j^i\right)^{\mathrm{T}}\left(\left(\boldsymbol{P}_1^i - \boldsymbol{T}_t^{\mathrm{W}}\right)\times\left(\boldsymbol{P}_2^i - \boldsymbol{T}_t^{\mathrm{W}}\right)\right)$$

$$\frac{\partial s_j^i(\boldsymbol{r})}{\partial \xi_k} = \frac{1}{\lambda_t^i}\left(\boldsymbol{R}_t^{\mathrm{T}}\bar{\boldsymbol{p}}_j^i\right)^{\mathrm{T}}\left(\hat{\boldsymbol{g}}_k\left(\boldsymbol{P}_1^i - \boldsymbol{P}_2^i\right)\right), \tag{36}$$

where $\boldsymbol{g}_1 = (1,0,0)^{\mathrm{T}}$, $\boldsymbol{g}_2 = (0,1,0)^{\mathrm{T}}$, $\boldsymbol{g}_3 = (0,0,1)^{\mathrm{T}}$.

$E_4, E_5, E_6, E_7, E_8, E_9$ can be minimized by modifying the same basic nonlinear problem defined by eq. (17). Therefore, only the technique for minimizing E_9 is presented. The objective function E_9 can be rewritten as

$$\begin{aligned} E_9 &= \sum_{i=1}^{N} d_6(\boldsymbol{l}_i, \boldsymbol{L}_i) = \sum_{i=1}^{N} d_n((\tilde{\boldsymbol{p}}_1, \tilde{\boldsymbol{p}}_2), (\boldsymbol{p}_1, \boldsymbol{p}_2)) \\ &= \frac{1}{6} \times \sum_{i=1}^{N} \left(s_1^i s_1^i + s_2^i s_2^i + 4 s_0^i s_0^i \right) \\ &\simeq \sum_{i=1}^{N} \sum_{j=0}^{2} \left(\alpha_j (s_j^i(\boldsymbol{r}) - \overline{s}_j^i)^2 \right), \quad \alpha_1 = \alpha_2 = 0.25\alpha_0, \end{aligned} \tag{37}$$

where s_1^i, s_2^i, and s_0^i denote the distances from the endpoints and midpoint of the image line segment to the corresponding projected model line, respectively. $s_j^i(\boldsymbol{r})$ is given by eq. (35).

3.3 Robust estimation

When the noise in the measurement data is Gaussian, the least squares method has been proven to be optimal and reliable. However, when outliers are present, the Gaussian assumption is not valid and the results of the least squares method are skewed. When outliers (incorrect 3D-2D line pairs) are present in the measures, a robust estimation is required. An M-estimator is considered a general form of maximum likelihood estimator that allows the likelihood that uncertain observations are utilized to be lower, and in some cases they are discarded. The robust optimization problem is then given by

$$E = \sum_{i=1}^{M} \rho\left(\delta_i / \sigma\right), \tag{38}$$

where $\rho(u)$ is a robust function that is symmetric and monotonically non-decreasing with increasing $|u|$ (see Figure 6). $\delta_i = s_i - \mathrm{Med}(s_i)$ is the normalized residue ($\mathrm{Med}(s)$ is the median operator). σ is the standard deviation of the inlier data.

Figure 6 shows several robust functions and their corresponding weight functions. Of the various robust functions, Tukey's hard trade-off was selected. For the Tukey function, $\rho(u)$ approximately varies as the square of $|u|$, when $|u|$ is a small value, and then, $\rho(u)$ tapers to a maximum value of $C^2/6$. In this function, the effect of any outliers is rejected and cannot be arbitrarily large. This is of interest in pose estimation since it means that a detected outlier has no effect on the optimization method. Therefore, we selected the Tukey function as the robust function in the M-estimation.

Then, the optimization problem defined by eq. (38) can be converted into an equivalent weighted least-squares problem:

$$E = \sum_{i=1}^{M} \omega\left(r_i^{(k-1)}\right) r_i^2, \tag{39}$$

where $r_i = \delta_i / \sigma$ and the superscript (k) indicates the iteration number. $\omega(\bullet)$ is the weight function corresponding to Tukey function,

$$\omega(u) = \begin{cases} \left(1 - \left(\dfrac{u}{C}\right)^2\right)^2, & \text{if } |u| \leqslant C, \\ 0, & \text{otherwise.} \end{cases} \tag{40}$$

The weight $\omega\left(r_i^{(k-1)}\right)$ should be recomputed after each iteration for use in the subsequent iteration. Eq. (39) can be solved according to

$$\boldsymbol{DJv} = \boldsymbol{e} = \boldsymbol{Ds}, \tag{41}$$

where $\boldsymbol{s} = \left(s_1(\boldsymbol{r}), s_2(\boldsymbol{r}), \cdots, s_m(\boldsymbol{r}), \cdots, s_M(\boldsymbol{r})\right)^{\mathrm{T}}$ is the error vector, $\boldsymbol{v}$ is the motion vector, $\boldsymbol{J}$ is the Jacobian matrix that links $\boldsymbol{s}$ to $\boldsymbol{v}$, and $\boldsymbol{D} = \mathrm{diag}(\omega_1, \omega_2, \cdots, \omega_m, \cdots, \omega_M)$ is the weight matrix.

Then, the solution of eq. (41) can be given by

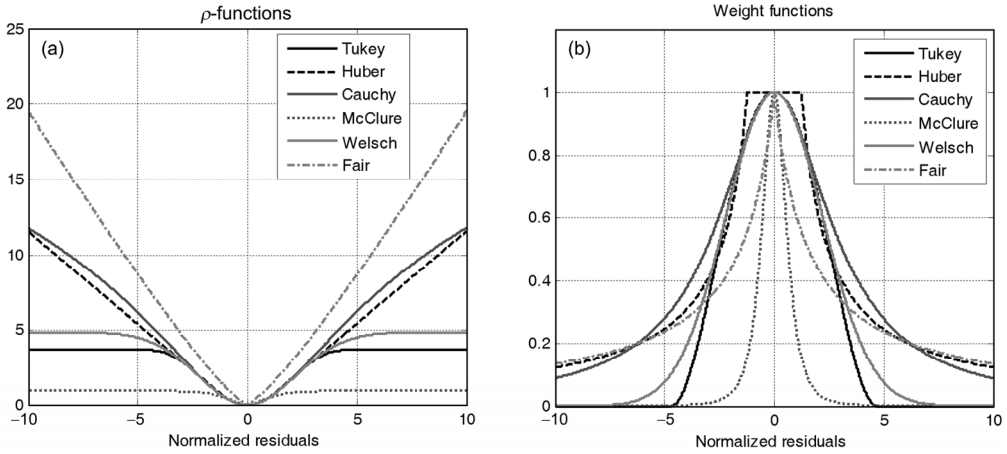


Figure 6 Robust function and weight function for different M-estimators. (a) Robust function; (b) weight function.

$$\boldsymbol{v} = \left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{J}\right)^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{e}. \tag{42}$$

4 Experimental results

In this section, we describe the validation of the proposed optimization algorithms by utilizing synthetic data and real images. All the related methods were implemented using a combination of C++ and OpenCV. The same total iterative number and terminal value, as well as initial pose, were set for all the related methods.

4.1 Synthetic data results

In this subsection, a number of comparisons are made with the proposed optimization methods based on the new error functions (referred to as R_and_T_novel, R_and_T_img_novel, and R_and_T_mod_novel), the proposed methods according to the error function presented by Taylor and Kriegman [24] (called R_and_T_len, R_and_T_img_len, and R_and_T_ mod_len), and three additional algorithms: R_and_T, R_and_T_img, and R_and_T_mod proposed by Kumar [16]. The 3D line model used for these experiments was a (3 m×3 m×2 m) cube. The virtual perspective camera was supposed to be 25 m from the cube. The resolution of the virtual camera was 1024×1024 with a view angle of 20°×20°. Given an input pose, nine model line segments were projected onto the image plane to generate 2D line segments. The endpoints of the 2D line segments were then corrupted by measurement noise, which can be decomposed into two components: noise perpendicular to the line and noise along the length of the line [16]. The first noise was modeled as a Gaussian random variable related to the orientation error, whilst the second was assumed to conform to uniform distribution related to line fragmentation.

In the first experiment, the related methods were compared under different levels of line fragmentation. The standard deviation of the noise perpendicular to the line was fixed to three pixels with the noises for the two endpoints negatively correlated. The upper bounding of the noise along the line was specified as a percentage of the length. Different levels of uniform noise ranging from 0 to 0.6 were added to the image line segments. For each noise level, 1000 test data sets were generated. The standard deviations of the pose parameters at different noise levels are shown in Figure 7.

In Figure 7, it can be seen that the related methods perform comparably when the noise level along the length of the line is low. However, when the noise level becomes 20% or more of the length, the three methods based on the finite image line segment, R_and_T_mod, R_and_T_mod_len, and R_and_T_mod_novel, significantly outperform the other methods. Moreover, the proposed method R_and_T_mod_novel performs slightly better than R_and_T_mod and R_and_T_mod_len. In addition, since the midpoint distance was utilized, the performance of the proposed methods

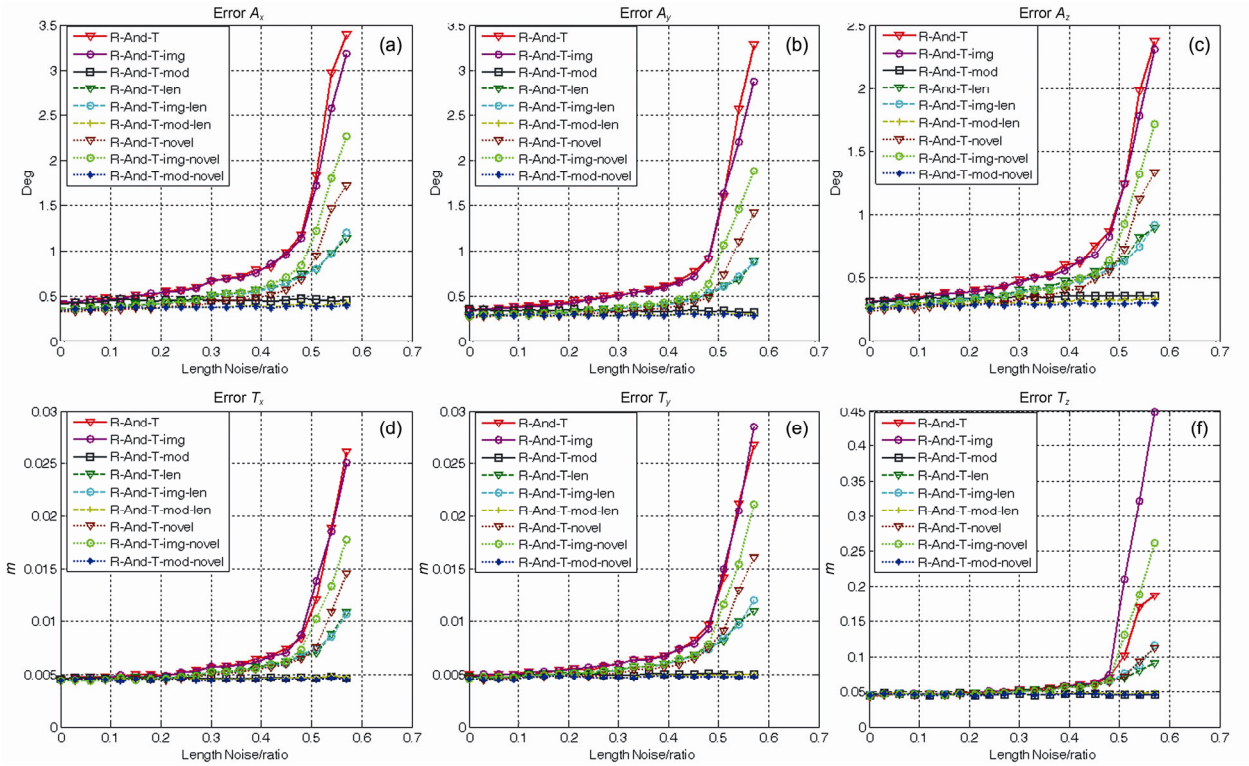


Figure 7 (Color online) Errors vs. the noise level along the image line. (a) Error for angle A_x; (b) error for angle A_y; (c) error for angle A_z; (d) error for angle T_x; (e) error for angle T_y; (f) error for angle T_z.

R_and_T_novel, R_and_T_img_novel, R_and_T_len, and R_and_T_img_len is superior to that of the other two methods, R_and_T and R_and_T_img, when significant line fragmentation exists in the image lines.

In the second experiment, the noise level along the line was fixed to 0.3. Different levels of uniform noise perpendicular to the line ranging from 0 to 6 pixels were added to the image line segments. In Figure 8, it can be seen that the performance of the related methods quickly degrades as the noise perpendicular to the line increases. However, the two methods R_and_T_mod_len and R_and_T_mod_novel provide a significantly better performance than the other methods. Moreover, the performance of the proposed method, R_and_T_mod_novel, is slightly better than that of R_and_T_mod_len. In addition, the performances of the proposed methods R_and_T_novel, R_and_T_img_novel, R_and_T_len, and R_and_T_img_len are superior to those of the other two methods, R_and_T and R_and_T_img, when significant line orientation noise exists in the image lines.

In addition, we present two additional sets of simulation results for the related pose optimization techniques integrated with M-estimation. In the simulation, synthetic data with outliers were used. The first set demonstrates the performance of the aforementioned methods for different levels of noise along the line ranging from 0 to 0.6, when the standard deviation of noise perpendicular to the line was fixed at 3 pixels. The second set shows the performance when the standard deviation of noise perpendicular to the line was varied in a range from 0 to 6 pixels, and the noise level along the line was fixed at 0.3. In Figures (9) and (10), it can be seen that the proposed methods according to the error functions containing the midpoint distance demonstrate significantly better robustness than the methods according to the endpoint distance.

4.2 Synthetic image results

In this subsection, we quantitatively compare the proposed methods, R_and_T_mod_len and R_and_T_mod_novel, with R_and_T_mod, using synthetic images generated by OpenGL with ground truth. In the experiment, the internal parameters of the virtual camera were not changed. The primary noise source of the procedure for generating synthetic images is the quantized error, since all the image points are in a unit of one pixel. After utilizing the multisampling technique in OpenGL, the influence of the quantized error on the accuracy of the compared methods can be neglected. The resolution of the virtual camera was 640×640 with a view angle 20°×20°. The rotation mode was derived using Euler angles with the order of *x*-*y*-*z*. The frame rate was 10 frames/s (10 Hz). There are around 350 frames in the over 35 s-long sequence; the distance along the optical axis ranged from 20 to 50 m. In all the tests, the image lines were detected using the LSD method [26] with the same parameters.

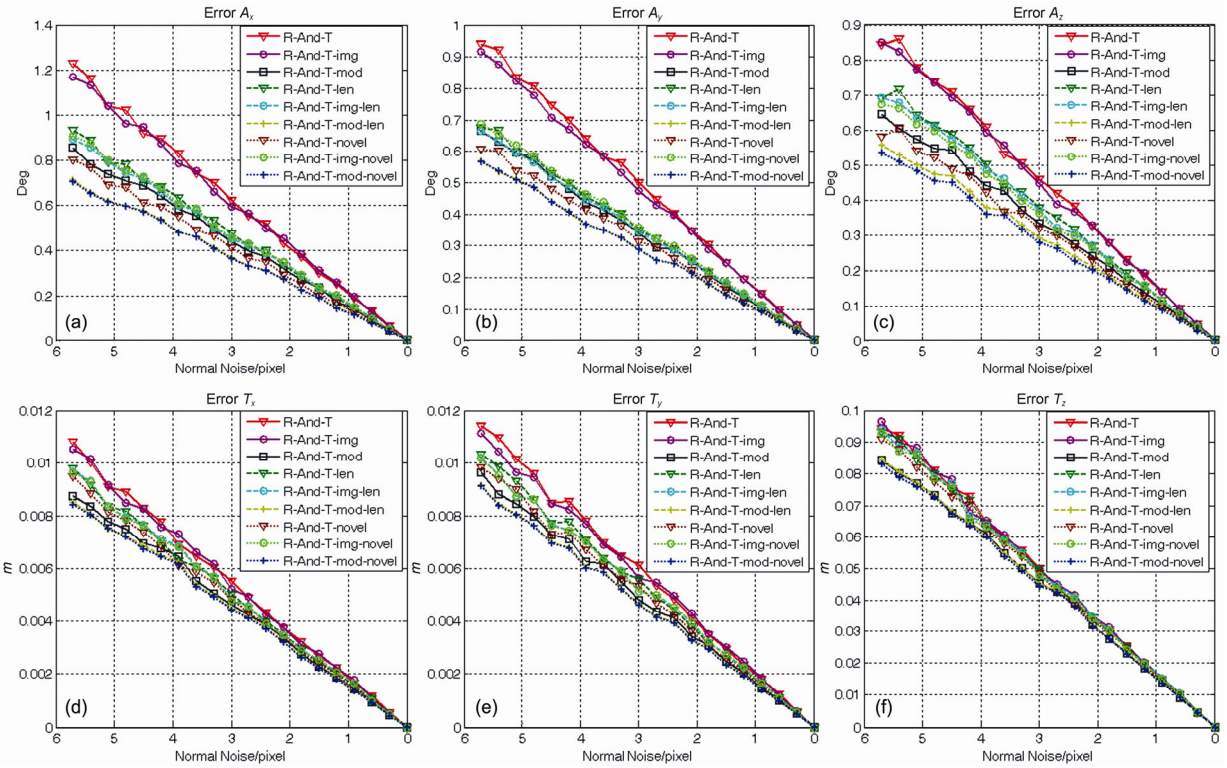


Figure 8 (Color online) Errors vs. the noise level perpendicular to the image line. (a) Error for angle A_x; (b) error for angle A_y; (c) error for angle A_z; (d) error for angle T_x; (e) error for angle T_y; (f) error for angle T_z.

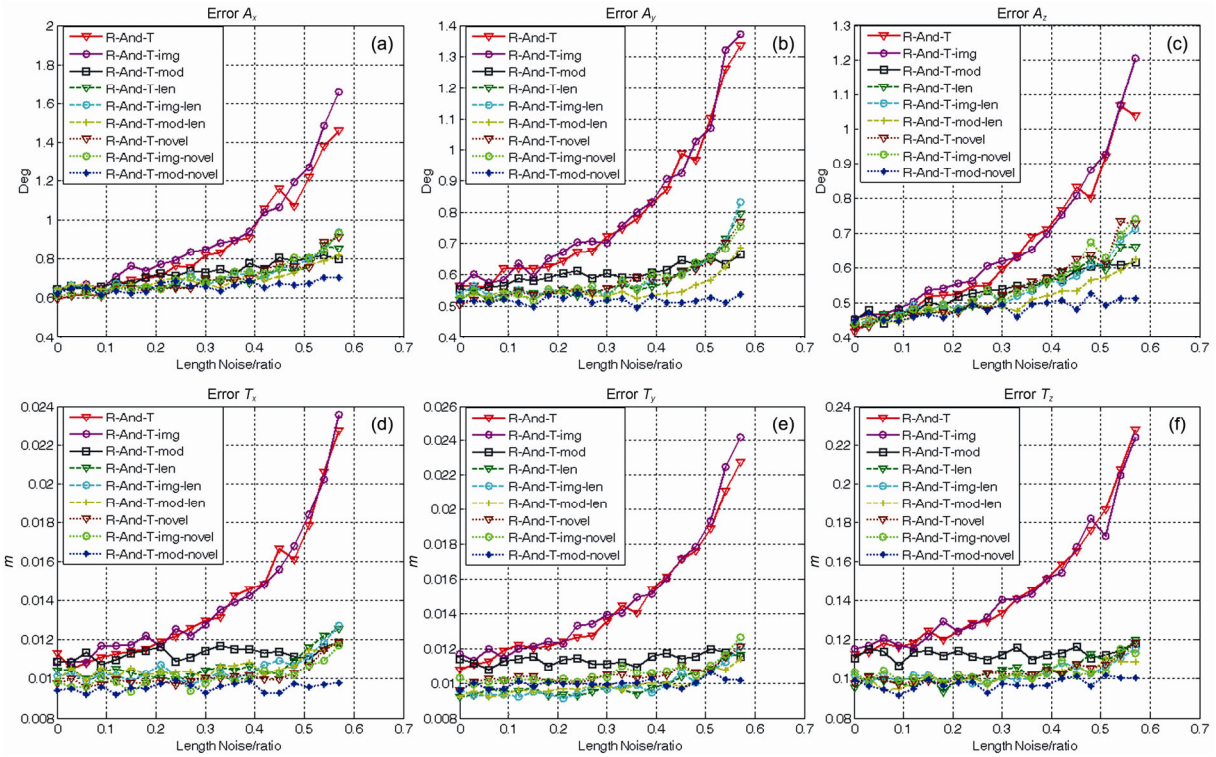


Figure 9 (Color online) Errors vs. the noise level along the image line (with M-estimation). (a) Error for angle A_x; (b) error for angle A_y; (c) error for angle A_z; (d) error for angle T_x; (e) error for angle T_y; (f) error for angle T_z.

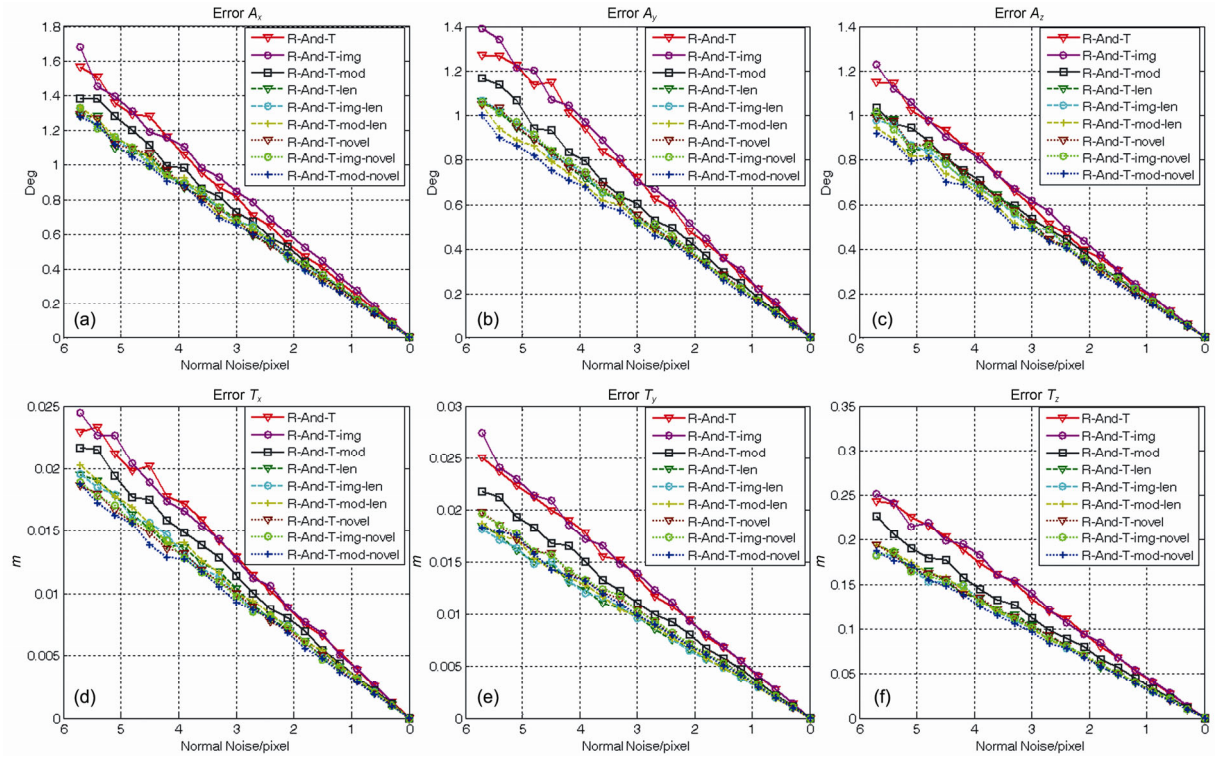


Figure 10 (Color online) Errors vs. the noise level perpendicular to the image line (with M-estimation). (a) Error for angle A_x; (b) error for angle A_y; (c) error for angle A_z; (d) error for angle T_x; (e) error for angle T_y; (f) error for angle T_z.

Figure 11 shows the errors (position and rotation) for the three methods. Table 1 represents the root mean square errors for the synthetic image sequence. As can be seen in Figure 11, the proposed methods perform better than R_and_

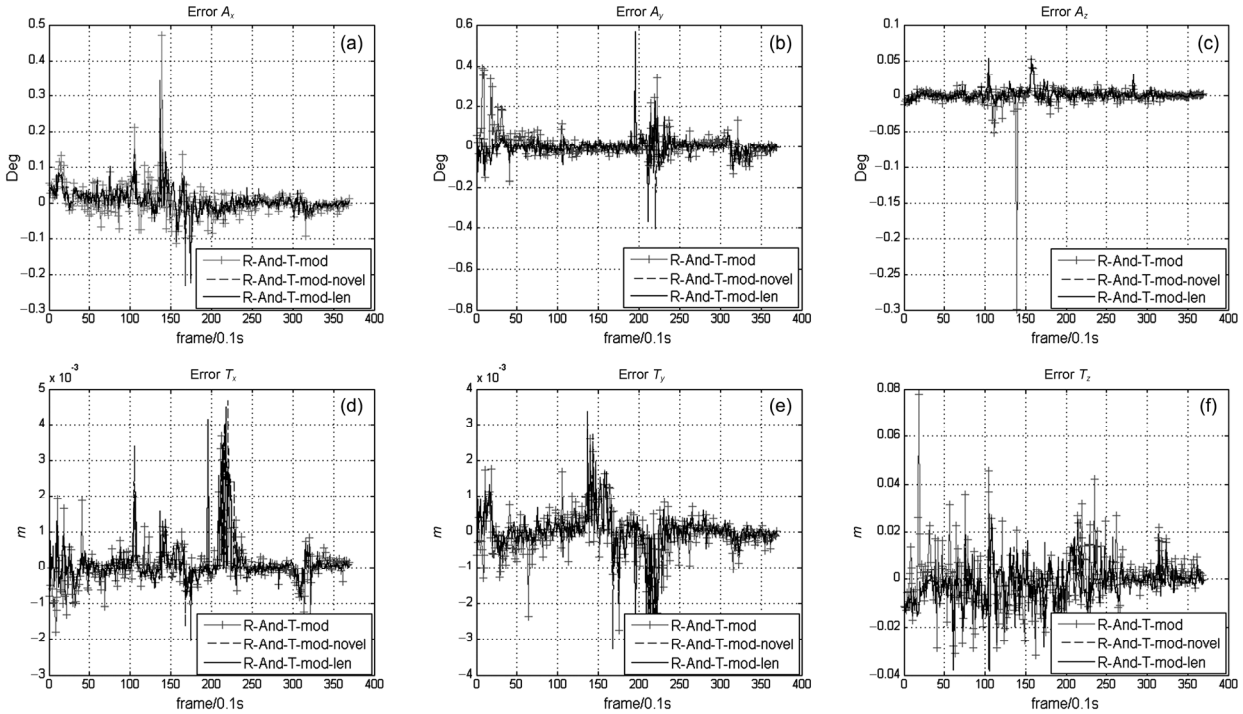


Figure 11 Comparisons for pose optimization over simulation sequence. (a) Error for angle A_x; (b) error for angle A_y; (c) error for angle A_z; (d) error for angle T_x; (e) error for angle T_y; (f) error for angle T_z.

T_mod for the test sequence and return a smaller error range. In addition, the performance of R_and_T_mod_novel based on the mean of the integral is equivalent to that of R_and_T_mod_len weighted by the length explicitly. In some views, the errors of R_and_T_mod_len are much larger than those of R_and_T_mod_novel, for the mismatched long line segment is drawn with a large weight whilst the correct shorter one can easily be underweighted in R_and_T_mod_len. According to the results in Table 1, the performances of the proposed methods are slightly better than that of R-and-T-mod, since the line segments for the synthetic images can be localized with high accuracy. Figure 12 shows the reprojected model lines for two selected simulated images according the pose optimized by the three methods.

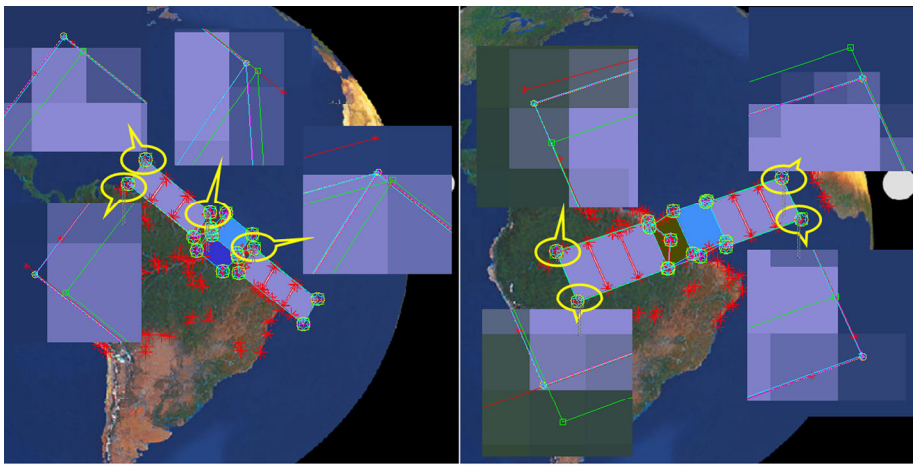
4.3 Real image results

In order to validate the aforementioned methods in real situations, we conducted two sets of experiments. First, we applied the three methods, R_and_T_mod_len and R_and_T_mod_novel together with R_and_T_mod, to five chessboard images, as shown in Figure 13 and calculated the RMS reprojection errors for each method. In the experiment, the model lines were the four edges of an A4 sheet of paper having a size of 297 mm×210 mm. For comparison purposes, we also estimated the pose of the camera from the chessboard corners by using the OI method [4]. Since the chessboard interior corners are a special case of the more general Harris corners, they can particularly be localized with subpixel accuracy. Therefore, we can treat the output of the OI method as the ground truth. Table 2 demonstrates that the reprojection errors of the proposed methods are lower than that of R_and_T_mod and close to the benchmark performance provided by the OI method.

Second, we applied R_and_T_mod_novel to two image sequences with known 3D line models to validate the robustness and accuracy of the proposed pose optimization method. For each image, the image lines were detected by using the LSD method [26]. The correspondences between the 3D model lines and the 2D image lines were established automatically by selecting the image line with the least distance to the projection of the visible model line as its correspondence. Then, the object pose was optimized using the R_and_T_mod_novel algorithm. In order to show the accuracy of the obtained results, 3D model lines were reprojected onto the image plane by using the optimized pose. Figure 14

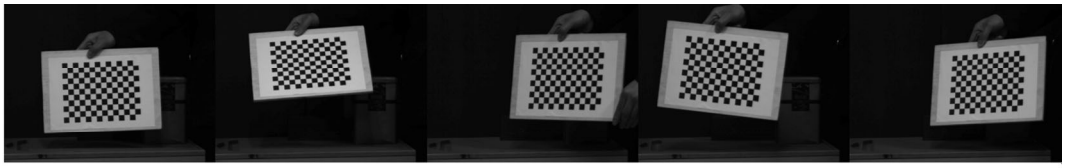
Table 1 Standard deviation of pose errors over simulation sequence

Error/1σ	A_x (°)	A_y (°)	A_z (°)	T_x (mm)	T_y (mm)	T_z (mm)
R_and_T_mod	0.045	0.066	0.018	0.6	0.7	12.1
R_and_T_mod_len	0.039	0.043	0.008	0.6	0.7	7.9
R_and_T_mod_novel	0.035	0.052	0.008	0.7	0.7	8.2



(a) Frame 17 (b) Frame 138

Figure 12 (Color online) Comparative reprojected results on simulated sequence. The endpoints of projected model lines using the true pose are drawn as hexagram and the endpoints of the detected line segment is drawn as asterisk. The projected endpoints of R_and_T_mod, R_and_T_mod_len and R_and_T_mod_novel are drawn as square, circle, and plus sign, respectively. The proposed methods, R_and_T_mod_novel and R_and_T_mod_len, achieve a similar performance and provide better matches of the projections of the visible model lines with the true positions than R-and-T-mod.

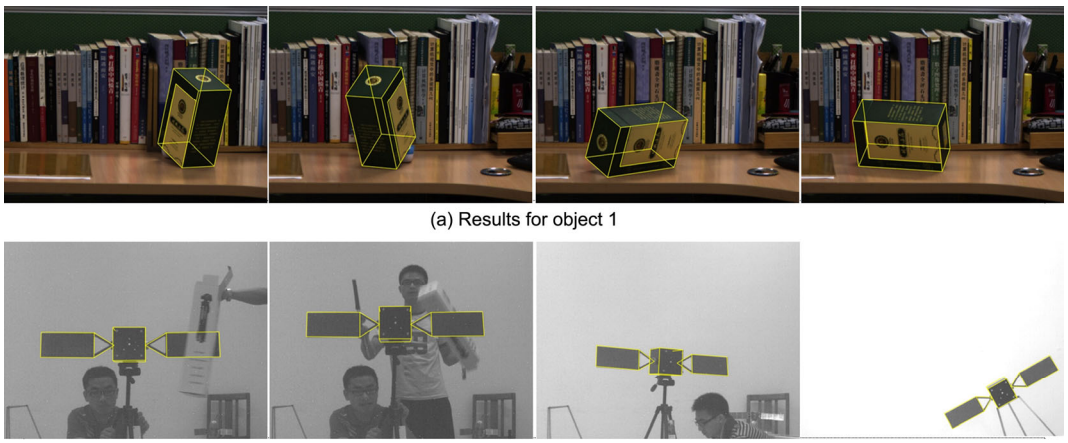


(a) Frame 1 (b) Frame 2 (c) Frame 3 (d) Frame 4 (e) Frame 5

Figure 13 Frames of the chessboard images.

Table 2 Reprojection errors for the chessboard images

Reprojection error (pixel)	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5
OI method	0.14	0.16	0.13	0.13	0.14
R_and_T_mod	0.36	0.52	0.34	0.53	0.36
R_and_T_mod_novel	0.34	0.48	0.32	0.48	0.32
R_and_T_mod_len	0.34	0.48	0.32	0.48	0.32



(a) Results for object 1

(b) Results for object 2

Figure 14 (Color online) Pose optimization for real images.

illustrates results from the image sequences. The first 3D model used was a tea box, as shown in Figure 14(a), while the second was a satellite model, as shown in Figure 14(b). These two objects were generated by using a structure and motion technique. In this case, the source of the noise was both the image measurement and the model lines because of the 3D position of the model line segments. The results demonstrate that the R_and_T_mod_novel method can successfully optimize the object pose.

5 Conclusions

In this paper, a new error function between line segments based on the mean of the distance integral was proposed, and then, its application to the problem of pose optimization from a set of matched 3D model and 2D image lines was described. First, based on the new error function, three pose optimization methods according to whether the image line is considered finite or not were proposed. Second, the pose optimization was converted into a simple iterative least squares problem by utilizing the Lie group to represent the pose parameters. Moreover, in order to handle data contaminated by outliers, an M-estimation method was employed to solve the pose optimization problem by means of iterative reweighted least squares techniques. The superior performance of the proposed methods as compared to previous approaches was verified through extensive simulations and experiments.

This work was supported by the National Basic Research Program of China ("973" Project) (Grant No. 2013CB733100), and National Natural Science Foundation of China (Grant No. 11332012).

1 Ferraz L, Binefa X, Moreno-Noguer F. Very fast solution to the pnp problem with algebraic outlier rejection. In: IEEE Conference on Computer Vision and Pattern Recognition, Columbus, 2014. 501–508
2 Li S, Xu C, Xie M. A robust o (n) solution to the perspective-n-point problem. IEEE Trans Pattern Anal Mach Intell, 2012, 34: 1444–1450
3 Lepetit V, Moreno-Noguer F, Fua P. Epnp: An accurate o (n) solution to the pnp problem. Int J Comput Vision, 2009, 81: 155–166
4 Lu C, Hager G D, Mjolsness E. Fast and globally convergent pose estimation from video images. IEEE T Pattern Anal Mach Intell, 2000, 6: 610–622
5 Li X, Long G C, Guo P Y, et al. Accurate mirror-based camera pose estimation with explicit geometric meanings. Sci China Tech Sci, 2014, 57: 2504–2513
6 Dhome M, Richetin M, Lapreste J, et al. Determination of the attitude of 3D objects from a single perspective view. IEEE T Pattern Anal Mach Intell, 1989, 11: 1265–1278
7 Chen H H. Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions. In: IEEE International Conference on Computer Vision, Osaka, 1990. 374–378
8 Navab N, Faugeras O. Monocular pose determination from lines: Critical sets and maximum number of solutions. In: IEEE Conference on Computer Vision and Pattern Recognition, New York, 1993. 254–260
9 Liu Y, Huang T S, Faugeras O D. Determination of camera location from 2-D to 3-D line and point correspondences. IEEE T Pattern Anal Mach Intell, 1990, 12: 28–37
10 Adnan A. Linear pose estimation from points or lines. IEEE T Pattern Anal Mach Intell, 2003, 25: 578–589
11 Mirzaei F M, Roumeliotis S I. Globally optimal pose estimation from line correspondences. In: IEEE International Conference on Robotics and Automation, Shanghai, 2011. 5581–5588
12 Elqursh A, Elgammal A. Line-based relative pose estimation. In: IEEE International Conference on Computer Vision, Providence, 2011. 3049–3056
13 Zhang L, Xu C, Lee K, et al. Robust and efficient pose estimation from line correspondences. In: Asian Conference on Computer Vision, Daejeon, 2013. 217–230
14 Liu J, Zhang X, Liu H, et al. New method for camera pose estimation based on line correspondence. Sci China Tech Sci, 2013, 56: 2787–2797
15 Phong Q T, Horaud R, Yassine A. Object pose from 2-D to 3-D point and line correspondences. Int J Comput Vision, 1995, 15: 225–243
16 Kumar R, Hanson A R. Robust methods for estimating pose and a sensitivity analysis. Cvgip Image Understand, 1994, 60: 313–342
17 Christy S, Horaud R. Fast and reliable object pose estimation from line correspondences. In: Computer Analysis of Images and Patterns, Kiel, 1997. 432–439
18 Horaud R, Dornaika F, Lamiroy B. Object pose: The link between weak perspective, paraperspective, and full perspective. Int J Comput Vision, 1997, 22: 173–189
19 Hanek R, Navab N, Appel M. Yet another method for pose estimation: A probabilistic approach using points lines and cylinders. In: IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, 1999. 544–550
20 David P, Dementhon D, Duraiswami R, et al. Simultaneous pose and correspondence determination using line features. In: IEEE Conference on Computer Vision and Pattern Recognition, Madison, 2003. 424–431
21 David P, Dementhon D, Duraiswami R, et al. SoftPOSIT: Simultaneous pose and correspondence determination. In: European Conference on Computer Vision, Copenhagen, 2002. 698–714
22 Zhang X, Zhang Z, Li Y, et al. Robust camera pose estimation from unknown or known line correspondences. Appl Optics, 2012, 51: 936–948
23 Christy S, Horaud R. Iterative pose computation from line correspondences. Comput Vis Image Understand, 1999, 73: 137–144
24 Taylor C J, Kriegman D J. Structure and motion from line segments in multiple images. IEEE T Pattern Anal Mach Intell, 1995, 17: 1021–1032
25 Drummond T, Cipolla R. Real-time visual tracking of complex structures. IEEE T Pattern Anal Mach Intell, 2002, 24: 932–946
26 Von Gioi Rafael G, Jérémie J, Morel J, et al. LSD: A fast line segment detector with a false detection control. IEEE T Pattern Anal Mach Intell, 2010, 32: 722–732

Appendix A

Let $\boldsymbol{p}_1=(x_1,y_1)$, $\boldsymbol{p}_2=(x_2,y_2),\cdots\boldsymbol{p}_n=(x_n,y_n)$ is a series of image edge points with the presence of the observation noise. Then the fitting model can be expressed as following:

$$y_i=ax_i+b+\varepsilon_i,\ i=1,2,\cdots,\ n, \tag{A1}$$

where $\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_i,\cdots,\varepsilon_n$ are Gaussian random variables with $\mathrm{E}\,\varepsilon_i=0,\ \mathrm{D}\,\varepsilon_i=\sigma^2$ and they are mutually independent.

Thus, the real line parameters can be estimated by the least squares technique as follows:

$$\hat{a}=C_{xy}/C_{xx},\ \hat{b}=\overline{y}-\hat{a}\overline{x}, \tag{A2}$$

where

$$C_{xx}=\sum_{i=1}^{n}\left(x_i-\overline{x}\right)^2,\ C_{xy}=\sum_{i=1}^{n}\left(x_i-\overline{x}\right)\left(y_i-\overline{y}\right)$$
$$\overline{x}=\frac{1}{n}\sum_{i=1}^{n}x_i,\ \overline{y}=\frac{1}{n}\sum_{i=1}^{n}y_i.$$

It is easy to obtain the variance and covariance for the estimated parameters $\hat{a}$ and $\hat{b}$.

$$\mathrm{D}(\hat{a})=\frac{\sigma^2}{C_{xx}},\ \mathrm{D}(\hat{b})=\sigma^2\left(\frac{1}{n}+\frac{\overline{x}^2}{C_{xx}}\right),\ \mathrm{Cov}(\hat{a},\hat{b})=-\frac{\overline{x}}{C_{xx}}\sigma^2. \tag{A3}$$

The estimates for the endpoints of the line segment along the normal direction can be provided by

$$\hat{y}_{\mathrm{s}}=\hat{a}x_1+\hat{b},\ \hat{y}_{\mathrm{e}}=\hat{a}x_n+\hat{b}. \tag{A4}$$

Thus, we can obtain the covariance for $\hat{y}_s$ and $\hat{y}_e$,

$$\begin{aligned}\mathrm{Cov}\left(\hat{y}_{\mathrm{s}},\hat{y}_{\mathrm{e}}\right)&=\mathrm{Cov}\left(\hat{a}x_1+\hat{b},\hat{a}x_n+\hat{b}\right)\\&=\frac{1}{nC_{xx}}\sigma^2\sum_{i=1}^{n}\left(x_i-x_1\right)\left(x_i-x_n\right).\end{aligned} \tag{A5}$$

Assuming that $0<x_1<x_2<\cdots<x_i<\cdots<x_n$, then we can obtain:

$$\mathrm{Cov}\left(\hat{y}_{\mathrm{s}},\hat{y}_{\mathrm{e}}\right)<0. \tag{A6}$$

Let the noises for the endpoints along the vertical direction be s_1 and s_2. Thus we have:

$$\mathrm{Cov}\left(s_1,s_2\right)=\mathrm{Cov}\left(\hat{y}_{\mathrm{s}},\hat{y}_{\mathrm{e}}\right)<0. \tag{A7}$$

If the true line is parallel with the horizontal axis, s_1 and s_2 correspond to the noise perpendicular to the line. Therefore, it is clear that two noises are negatively correlated.