An Efficient Normal-Error Iterative Algorithm for Line Triangulation

Qiang Zhang[1,2], Yan Wu[1], Ming Liu[1], and Licheng Jiao[2]

[1] School of Electronic Engineering, Xidian University, Xi'an, 710071, China
[2] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an, 710071, China
zhangqiang@xidian.edu.cn

Abstract. In this paper, we address the problem of line triangulation, which is to find the position of a line in space given its three projections taken with cameras with known camera matrices. Because of measurement error in line extraction, the problem becomes difficult so that it is necessary to estimate a 3D line to optimally fit measured lines. In this work, the normal errors of measured line are presented to describe the measurement error and based on their statistical property a new geometric-distance optimality criterion is constructed. Furthermore, a simple iterative algorithm is proposed to obtain suboptimal solution of the optimality criterion, which ensures that the solution satisfies the trifocal tensor constraint. Experiments show that our iterative algorithm can achieve the estimation accuracy comparable with the Gold Standard algorithm, but its computational load is substantially reduced.

Keywords: Line triangulation, Normal error, Iterative algorithm, Suboptimal solution.

1 Introduction

In computer vision, triangulation is one of fundamental problems and in the past many algorithms [1-11] have been proposed to solve point triangulation. But, as the reconstruction of manmade environment becomes hot spot topic, lines in space as reconstruction feature draw more and more attention. This is not only because lines exist largely in manmade environments, also because they have some advantages that points haven't [1]. For example, a line is extracted with more accuracy, it has less chance to be occluded but more information to represent geometric structure of object, and etc. However, line triangulation is more difficult than point triangulation. To construct overconstrained problem, line triangulation needs at least trilinear constraint of three projections, compared with bilinear constraint in the point triangulation [2]. Moreover, because a line in space has 4 degrees of freedom, there is no trivial representation [12].

Although there are many difficulties, some researchers have begun the research of line triangulation. In the early works [13-15], the algorithms are proposed to solve line triangulation for calibrated cameras. But it is difficult for these algorithms to deal

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-319-02895-8_64

J. Blanc-Talon et al. (Eds.): ACIVS 2013, LNCS 8192, pp. 298–309, 2013.

© Springer-Verlag Berlin Heidelberg 2013

with the practical problem in which just the affine or projective camera matrices are known. To solve the problem in practice, with Plücker coordinates, a simple linear algorithm [12] is proposed. Because of its algebraic-error cost function and the Plücker correction, the solution of this algorithm is not accurate enough. The optimal algorithm in [16] can give global solution, but its optimality criterion just describes the statistics property of image line coordinates and its computation is very intensive. In the Gold Standard algorithm [17], geometric errors in images can be utilized to construct cost functions. But, because none of these cost functions for the Gold Standard algorithm are convex, even though the LM iteration method has large computational load, their solutions are not optimal.

In this paper, a novel algorithm is proposed for line triangulation of three views, equally valid in the projective, affine and metric reconstruction. The main difference of our algorithm from the previous algorithms is that its optimality criterion has both geometric and statistic meaning. This is because the normal errors of measured line are defined to describe measurement error in image, and based on the statistical property of the normal error, the likelihood function of the normal error is used to construct cost function. Furthermore, because the trifocal tensor constraint in the optimality criterion is simplified via the linear normal-error representation of the estimated line, a simple iterative algorithm is proposed to find suboptimal solution of this optimality criterion, which makes the constraint satisfied in each iterative cycle. Experiments show that our algorithm not only has lower computational load than the Gold Standard algorithm, but also has almost the same estimation accuracy with the Gold Standard algorithm.

The remainder of this paper is organized as follows. Section 2 is about the definition of the normal error and the normal-error representation of line. In Section 3, we give the normal-error distribution and construct optimality criterion. Further on in Section 4, an iterative algorithm is described to solve the optimality criterion. Section 5 reports experimental results. Section 6 concludes this paper.

2 Normal-Error Representation of Line

In this paper, the discussed situation is that the measured lines in images correspond to same 3D line and just have measurement error, so it is reasonable to assume that the estimated lines are not far from the measured lines and the angles between them are far less than 90 degrees. As shown in Fig 1, this means that the distances λ_1 and λ_2, which are perpendicular to the measured line $\mathbf{l}$ and from the measured endpoints $\mathbf{x},\mathbf{y}$ to the estimated line $\overline{\mathbf{l}}$, are finite. Because these two distances are measured along the normal direction of the measured line, we define them as normal errors between the measured line and the estimated line. Via the normal errors, two intersections $\overline{\mathbf{x}},\overline{\mathbf{y}}$ on the estimated line can be obtained and we define them as the estimated endpoints of the estimated line, which are represented as follows:

$$\overline{\mathbf{x}} = \mathbf{x} + \lambda_1 \begin{pmatrix} \mathbf{n} \\ 0 \end{pmatrix}, \overline{\mathbf{y}} = \mathbf{y} + \lambda_2 \begin{pmatrix} \mathbf{n} \\ 0 \end{pmatrix} \quad (1)$$

where $\mathbf{n}$ is the vector including the normalized first two elements of the measured line $\mathbf{l}$. Furthermore, via these two estimated endpoints, the normal-error representation of the estimated line $\overline{\mathbf{l}}$ can be written as

$$\begin{aligned}\overline{\mathbf{l}}(\lambda_1,\lambda_2) &= \overline{\mathbf{x}}\times\overline{\mathbf{y}} = \mathbf{l}+\lambda_1\mathbf{g}_1+\lambda_2\mathbf{g}_2 \\ &= \mathbf{x}\times\mathbf{y}+\lambda_1\begin{pmatrix}\mathbf{n}\\0\end{pmatrix}\times\mathbf{y}+\lambda_2\mathbf{x}\times\begin{pmatrix}\mathbf{n}\\0\end{pmatrix}\end{aligned} \tag{2}$$

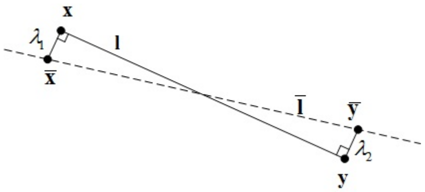


Fig. 1. Normal errors λ_1 and λ_2 between measured line $\mathbf{l}$ (solid line) and estimated line $\overline{\mathbf{l}}$ (dash line)

It is obvious that the normal-error representation (2) is a linear representation of the estimated line $\overline{\mathbf{l}}$ and its parameters are its two normal errors, which reduce the number of line coordinates and make the representation of error easier.

3 An Optimality Criterion about the Normal Error

3.1 Normal-Error Distribution

Before constructing the normal-error optimality criterion, it is necessary to give the statistical distribution of the normal error in this section. Therefore, two set of experiments are performed as follows. In these experiments, 1000000 random 3D lines are projected onto 1024×1024 image plane as standard lines. Then, each standard line is randomly sampled with 10 and 30 points (including two endpoints). It is common to assume that the extracted points in image are subject to Gaussian noise, we add Gaussian noise to these sampling points with zero mean and one-pixel standard deviation, the corresponding measured line is fitted to these noised sampling points using the least-squares and the measured endpoints are the nearest points from the two endpoints to the measured line. Thus, the normal error is the distance between the endpoint to the measured endpoint.

From Fig 2, we can see that no matter whether the experiment is about 10 or 30 sampling points, its histogram of the normal error can be perfectly approximated by a Gaussian distribution. Thus, it is reasonable to assume that each normal error of the measured line satisfies the Gaussian distribution.

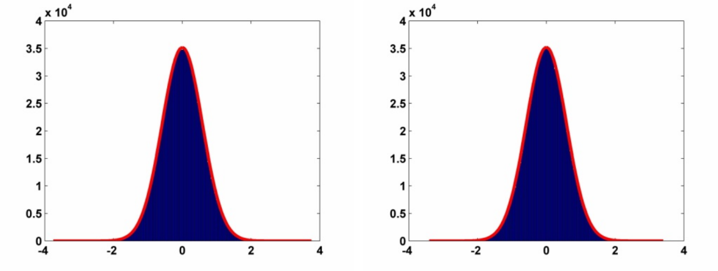


Fig. 2. The histograms of the normal error and their Gaussian distribution fittings for the 10-sampling-point experiment (left) and the 30-sampling-point experiment (right), where the horizontal axis represents the normal error level (pixel), the vertical axis represents the number of times each error level occurs.

Thus, based on maximum likelihood estimation, the most likely values for the normal errors in n images can be obtained by minimizing the function

$$L(\lambda_{11},\lambda_{12},\lambda_{21},\lambda_{22},\cdots,\lambda_{n1},\lambda_{n2})=\sum_{i=1}^{n}\lambda_{i1}^2+\lambda_{i2}^2 \tag{3}$$

where i stands for the i th image, subject to some constraint

$$T(\lambda_{11},\lambda_{12},\lambda_{21},\lambda_{22},\cdots,\lambda_{n1},\lambda_{n2})=0 \tag{4}$$

Once these normal errors are known, the corresponding estimated line can be found via (2).

3.2 The Optimality Criterion for Three Views

For three-view line triangulation, the constraint (4) can be described with trifocal tensor [17]. Given the measured line correspondence $(\mathbf{l}_1,\mathbf{l}_2,\mathbf{l}_3)$ and their endpoints $\{(\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2),(\mathbf{x}_3,\mathbf{y}_3)\}$ in three images, using the normal errors $(\lambda_{j1},\lambda_{j2})$ on the j th image $(j=1,2,3)$, the estimated endpoints of the line on the first image can be written as

$$\begin{aligned}(\bar{x}_1,\bar{x}_2,1)^T &= \mathbf{x}_1+\lambda_{11}(\mathbf{n}_1^T,0)^T\\ (\bar{y}_1,\bar{y}_2,1)^T &= \mathbf{y}_1+\lambda_{12}(\mathbf{n}_1^T,0)^T\end{aligned} \tag{5}$$

and the estimated lines $\bar{\mathbf{l}}_2$, $\bar{\mathbf{l}}_3$ on the last two images can be written as

$$\begin{aligned}\bar{\mathbf{l}}_2(\lambda_{21},\lambda_{22}) &= \mathbf{l}_2+\lambda_{21}\mathbf{g}_{21}+\lambda_{22}\mathbf{g}_{22}\\ \bar{\mathbf{l}}_3(\lambda_{31},\lambda_{32}) &= \mathbf{l}_3+\lambda_{31}\mathbf{g}_{31}+\lambda_{32}\mathbf{g}_{32}\end{aligned} \tag{6}$$

where $\mathbf{n}_i$ is the vector including the normalized first two elements of the measured line $\mathbf{l}_i$ $(i=1,2,3)$,

$$\begin{array}{l} \mathbf{g}_{21} = \begin{pmatrix} \mathbf{n}_2 \\ 0 \end{pmatrix} \times \mathbf{y}_2, \mathbf{g}_{22} = \mathbf{x}_2 \times \begin{pmatrix} \mathbf{n}_2 \\ 0 \end{pmatrix} \\ \mathbf{g}_{31} = \begin{pmatrix} \mathbf{n}_3 \\ 0 \end{pmatrix} \times \mathbf{y}_3, \mathbf{g}_{32} = \mathbf{x}_3 \times \begin{pmatrix} \mathbf{n}_3 \\ 0 \end{pmatrix} \end{array} .$$

thus, the trifocal tensor constraint can be written as following point-line-line correspondence

$$\begin{array}{l} C_1(\boldsymbol{\lambda}) = \bar{\mathbf{l}}_2^T (\bar{x}_1 \mathrm{T}_1 + \bar{x}_2 \mathrm{T}_2 + \mathrm{T}_3) \bar{\mathbf{l}}_3 = 0 \\ C_2(\boldsymbol{\lambda}) = \bar{\mathbf{l}}_2^T (\bar{y}_1 \mathrm{T}_1 + \bar{y}_2 \mathrm{T}_2 + \mathrm{T}_3) \bar{\mathbf{l}}_3 = 0 \end{array} \tag{7}$$

where $(\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3)$ is the trifocal tensor of the three images, $\boldsymbol{\lambda} = (\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}, \lambda_{31}, \lambda_{32})^T$ is called normal-error vector. From (3) and (4), we have the normal-error optimality criterion for three views

$$\begin{array}{ll} \min & \boldsymbol{\lambda}^T \boldsymbol{\lambda} \\ s.t. & C_1(\boldsymbol{\lambda}) = 0 \\ & C_2(\boldsymbol{\lambda}) = 0 \end{array} \tag{8}$$

Although via the normal-error representation of line, the degrees of both the two constraints in (8) are just three, the optimality criterion belongs to the nonlinear constraint quadratic programming and complex algorithm is needed to solve it but hardly to obtain the optimal solution. So, in next section, a simple iterative algorithm is proposed to obtain the suboptimal solution which satisfies the trifocal tensor constraint.

4 Iterative Algorithm

In this section, we regard the two constraints in the optimality criterion (8) as two 3-degree surfaces in the normal-error vector space, so the triangulation problem is converted to find the point on the intersection of the two surfaces and nearest to the origin. To simplify solving process and obtain the suboptimal solution, an iterative algorithm is used. In each iterative cycle, we define a pencil of lines through the origin, and the iteration solution is the nearest point to the origin, which is not only on the intersection of the two 3-degree surfaces but also on the pencil. Thus, the original criterion (8) can be converted to equation to solve and its process is stated as follows.

Suppose the solution $\boldsymbol{\lambda}^n$ of the nth iteration is known, where initial value $\boldsymbol{\lambda}^0 = 0$, the pencil in the $n+1$th iteration is defined by the linear span of the gradients of the two constraints at $\boldsymbol{\lambda}^n$ which are the normal vectors to the two degree 3 surfaces at point $\boldsymbol{\lambda}^n$ if $n \neq 0$. Therefore, the point $\boldsymbol{\lambda}$ on the pencil can be expressed

$$\boldsymbol{\lambda} = x\mathbf{p}_1 + y\mathbf{p}_2 \tag{9}$$

where

$$\mathbf{p}_1 = \partial C_1\left(\boldsymbol{\lambda}^n\right)/\partial\boldsymbol{\lambda} = \begin{bmatrix} \bar{\mathbf{l}}_2\left(\lambda_{21}^n,\lambda_{22}^n\right)^T (\sum_{i=1}^{2} n_{1i}\mathrm{T}_i)\bar{\mathbf{l}}_3\left(\lambda_{31}^n,\lambda_{32}^n\right) \\ 0 \\ \mathbf{g}_{21}^T(\sum_{i=1}^{3}\bar{x}_i^n\mathrm{T}_i)\bar{\mathbf{l}}_3\left(\lambda_{31}^n,\lambda_{32}^n\right) \\ \mathbf{g}_{22}^T(\sum_{i=1}^{3}\bar{x}_i^n\mathrm{T}_i)\bar{\mathbf{l}}_3\left(\lambda_{31}^n,\lambda_{32}^n\right) \\ \bar{\mathbf{l}}_2\left(\lambda_{21}^n,\lambda_{22}^n\right)^T (\sum_{i=1}^{3}\bar{x}_i^n\mathrm{T}_i)\mathbf{g}_{31} \\ \bar{\mathbf{l}}_2\left(\lambda_{21}^n,\lambda_{22}^n\right)^T (\sum_{i=1}^{3}\bar{x}_i^n\mathrm{T}_i)\mathbf{g}_{32} \end{bmatrix},$$

$$\mathbf{p}_2 = \partial C_2\left(\boldsymbol{\lambda}^n\right)/\partial\boldsymbol{\lambda} = \begin{bmatrix} 0 \\ \bar{\mathbf{l}}_2\left(\lambda_{21}^n,\lambda_{22}^n\right)^T (\sum_{i=1}^{2} n_{1i}\mathrm{T}_i)\bar{\mathbf{l}}_3\left(\lambda_{31}^n,\lambda_{32}^n\right) \\ \mathbf{g}_{21}^T(\sum_{i=1}^{3}\bar{y}_i^n\mathrm{T}_i)\bar{\mathbf{l}}_3\left(\lambda_{31}^n,\lambda_{32}^n\right) \\ \mathbf{g}_{22}^T(\sum_{i=1}^{3}\bar{y}_i^n\mathrm{T}_i)\bar{\mathbf{l}}_3\left(\lambda_{31}^n,\lambda_{32}^n\right) \\ \bar{\mathbf{l}}_2\left(\lambda_{21}^n,\lambda_{22}^n\right)^T (\sum_{i=1}^{3}\bar{y}_i^n\mathrm{T}_i)\mathbf{g}_{31} \\ \bar{\mathbf{l}}_2\left(\lambda_{21}^n,\lambda_{22}^n\right)^T (\sum_{i=1}^{3}\bar{y}_i^n\mathrm{T}_i)\mathbf{g}_{32} \end{bmatrix},$$

$$\left(\bar{x}_1^n,\bar{x}_2^n,\bar{x}_3^n\right)^T = \mathbf{x}_1 + \lambda_{11}^n\left(\mathbf{n}_1^T,0\right)^T, \left(\bar{y}_1^n,\bar{y}_2^n,\bar{y}_3^n\right)^T = \mathbf{y}_1 + \lambda_{12}^n\left(\mathbf{n}_1^T,0\right)^T, \left(n_{11},n_{12}\right) = \mathbf{n}_1^T.$$

Substituting $\boldsymbol{\lambda}$ into the two constraints of (8), we obtain two cubic equations in x, y

$$\left\{\begin{array}{l} a_1x^3 + a_2x^2y + a_3xy^2 + a_4x^2 + a_5y^2 + a_6xy + a_7x + a_8y + a_9 \\ = \left(\mathbf{u}_1^T\,\mathrm{M}_1\,\mathbf{v}_1\right)x^3 + \left(\mathbf{u}_1^T\,\mathrm{M}_1\,\mathbf{v}_2 + \mathbf{u}_2^T\,\mathrm{M}_1\,\mathbf{v}_1\right)x^2y + \left(\mathbf{u}_2^T\,\mathrm{M}_1\,\mathbf{v}_2\right)xy^2 + \left(\mathbf{u}_2^T\,\mathrm{M}_2\,\mathbf{v}_2\right)y^2 \\ +\left(\mathbf{l}_2^T\,\mathrm{M}_1\,\mathbf{v}_1 + \mathbf{u}_1^T\,\mathrm{M}_1\,\mathbf{l}_3 + \mathbf{u}_1^T\,\mathrm{M}_2\,\mathbf{v}_1\right)x^2 + \left(\mathbf{l}_2^T\,\mathrm{M}_1\,\mathbf{v}_2 + \mathbf{u}_1^T\,\mathrm{M}_2\,\mathbf{v}_2 + \mathbf{u}_2^T\,\mathrm{M}_1\,\mathbf{l}_3 + \mathbf{u}_2^T\,\mathrm{M}_2\,\mathbf{v}_1\right. \\ \quad\left.\right)xy + \left(\mathbf{l}_2^T\,\mathrm{M}_1\,\mathbf{l}_3 + \mathbf{l}_2^T\,\mathrm{M}_2\,\mathbf{v}_1 + \mathbf{u}_1^T\,\mathrm{M}_2\,\mathbf{l}_3\right)x + \left(\mathbf{l}_2^T\,\mathrm{M}_2\,\mathbf{v}_2 + \mathbf{u}_2^T\,\mathrm{M}_2\,\mathbf{l}_3\right)y + \left(\mathbf{l}_2^T\,\mathrm{M}_2\,\mathbf{l}_3\right) = 0 \\ b_1y^3 + b_2x^2y + b_3xy^2 + b_4x^2 + b_5y^2 + b_6xy + b_7x + b_8y + b_9 \\ = \left(\mathbf{u}_2^T\,\mathrm{N}_1\,\mathbf{v}_2\right)y^3 + \left(\mathbf{u}_1^T\,\mathrm{N}_1\,\mathbf{v}_1\right)x^2y + \left(\mathbf{u}_1^T\,\mathrm{N}_1\,\mathbf{v}_2 + \mathbf{u}_2^T\,\mathrm{N}_1\,\mathbf{v}_1\right)xy^2 + \left(\mathbf{u}_1^T\,\mathrm{N}_2\,\mathbf{v}_1\right)x^2 \\ +\left(\mathbf{l}_2^T\,\mathrm{N}_1\,\mathbf{v}_2 + \mathbf{u}_2^T\,\mathrm{N}_1\,\mathbf{l}_3 + \mathbf{u}_2^T\,\mathrm{N}_2\,\mathbf{v}_2\right)y^2 + \left(\mathbf{l}_2^T\,\mathrm{N}_1\,\mathbf{v}_1 + \mathbf{u}_1^T\,\mathrm{N}_2\,\mathbf{v}_2 + \mathbf{u}_1^T\,\mathrm{N}_1\,\mathbf{l}_3 + \mathbf{u}_2^T\,\mathrm{N}_2\,\mathbf{v}_1\right. \\ \quad\left.\right)xy + \left(\mathbf{l}_2^T\,\mathrm{N}_2\,\mathbf{v}_1 + \mathbf{u}_1^T\,\mathrm{N}_2\,\mathbf{l}_3\right)x + \left(\mathbf{l}_2^T\,\mathrm{N}_1\,\mathbf{l}_3 + \mathbf{l}_2^T\,\mathrm{N}_2\,\mathbf{v}_2 + \mathbf{u}_2^T\,\mathrm{N}_2\,\mathbf{l}_3\right)y + \left(\mathbf{l}_2^T\,\mathrm{N}_2\,\mathbf{l}_3\right) = 0 \end{array}\right. \quad (10)$$

where $\mathbf{u}_1 = p_{13}\mathbf{g}_{21} + p_{14}\mathbf{g}_{22}$, $\mathbf{u}_2 = p_{23}\mathbf{g}_{21} + p_{24}\mathbf{g}_{22}$, $\mathbf{v}_1 = p_{15}\mathbf{g}_{31} + p_{16}\mathbf{g}_{32}$, $\mathbf{v}_2 = p_{25}\mathbf{g}_{31} + p_{26}\mathbf{g}_{32}$, $\mathrm{M}_1 = p_{11}n_{11}\,\mathrm{T}_1 + p_{11}n_{12}\,\mathrm{T}_2$, $\mathrm{M}_2 = x_1\,\mathrm{T}_1 + x_2\,\mathrm{T}_2 + \mathrm{T}_3$,

$\mathrm{N}_1 = p_{22}n_{11}\,\mathrm{T}_1 + p_{22}n_{12}\,\mathrm{T}_2$, $\mathrm{N}_2 = y_1\,\mathrm{T}_1 + y_2\,\mathrm{T}_2 + \mathrm{T}_3$, $(x_1, x_2, 1)^T = \mathbf{x}_1$, $(y_1, y_2, 1)^T = \mathbf{y}_1$, $(p_{11}, 0, p_{13}, p_{14}, p_{15}, p_{16})^T = \mathbf{p}_1$, $(0, p_{22}, p_{23}, p_{24}, p_{25}, p_{26})^T = \mathbf{p}_2$.

Then the following problem is to solve the two cubic equations. From (10), it can be seen that the highest degree of x in the second equation is 2. Therefore, we rewrite the second one as

$$\left(b_2 y + b_4\right)x^2 + \left(b_3 y^2 + b_6 y + b_7\right)x + \left(b_1 y^3 + b_5 y^2 + b_8 y + b_9\right) = 0 \qquad (11)$$

And suppose y is known and $y \neq -b_4 / b_2$, the solutions of x are

$$x = \frac{-A(y) \pm \sqrt{B(y)}}{2C(y)} \qquad (12)$$

where $B(y) = \left(b_3 y^2 + b_6 y + b_7\right)^2 - 4\left(b_2 y + b_4\right)\left(b_1 y^3 + b_5 y^2 + b_8 y + b_9\right)$, $C(y) = b_2 y + b_4$, $A(y) = b_3 y^2 + b_6 y + b_7$. Substituting (12) into the first equation of (10), we obtain the equation

$$\begin{aligned}
&\Big(a_1\left(-A(y)^3 - 3A(y)B(y)\right) + 2C(y)\left(a_2 y + a_4\right)\left(A(y)^2 + B(y)\right) \\
&\quad - 4A(y)C(y)^2\left(a_3 y^2 + a_6 y + a_7\right) + 8C(y)^3\left(a_5 y^2 + a_8 y + a_9\right)\Big)^2 \qquad (13) \\
&= \Big(a_1\left(3A(y)^2 + B(y)\right) - 4A(y)C(y)\left(a_2 y + a_4\right) + 4C(y)^2\left(a_3 y^2 + a_6 y + a_7\right)\Big)^2 B(y)
\end{aligned}$$

This is an equation in y of degree 12, which can be solved with companion matrix [19].

Based on the above discussion, our iterative algorithm is outlined in Table 1. Because the iterative algorithm ensures that each iterative solution satisfies the trifocal tensor constraint, the suboptimal solution can be used to determine a 3D line directly. Therefore, substituting the suboptimal solution into (5) and (6) to obtain the estimated endpoints in the first image and the estimated lines in the other images, the 3D line can be computed by linear algorithm [12] without the Plücker correction.

Table 1. The Normal-Error Iterative Algorithm

Initiation: given a termination threshold ε and let $n = 0, \boldsymbol{\lambda}^0 = 0$
1. Compute $\mathbf{p}_1, \mathbf{p}_2$ in (9) with $\boldsymbol{\lambda}^n$ and the coefficients of cubic equations (10);
2. Construct equation (13) and solve it with the companion matrix, substitute roots of (13) which are different from $-b_4 / b_2$ into (12) to obtain x, and when $y = -b_4 / b_2$, $x = \left(b_1 b_4^3 - b_2 b_4^2 b_5 + b_2^2 b_4 b_8 - b_2^3 b_9\right) / \left(b_2 b_3 b_4^2 - b_2^2 b_4 b_6 + b_2^3 b_7\right)$;
3. Select the (x', y') from solutions that satisfies (10) and makes $\left\Vert x\mathbf{p}_1 + y\mathbf{p}_2 \right\Vert^2$ smallest, $\boldsymbol{\lambda}^{n+1} = x'\mathbf{p}_1 + y'\mathbf{p}_2$;
4. If $\left\Vert \boldsymbol{\lambda}^n - \boldsymbol{\lambda}^{n+1} \right\Vert \leq \varepsilon$, then output $\boldsymbol{\lambda}^{n+1}$ and terminate; otherwise, let $n = n + 1$ and return to 1.

5 Experiments

Both simulated and real experiments are carried out to evaluate the different four algorithms using MATLAB on a PC with Intel Core2 2.33 GHz CPU and 2GB RAM. These compared algorithms include LIN(linear algorithm)[12], GS+d(Golden Standard method with the distance from the measured endpoint to the estimated line), GS+nd(Golden Standard method with the normal distance), NIA(our normal-error iterative algorithm). In NIA, the termination threshold ε is 10^{-7}.

5.1 Simulated Data

As shown in Fig.3, the simulated experimental setup includes three cameras looking inward to a cube. The size of the angle between the adjacent cameras is chosen as 15 degrees, 30 degrees respectively to carry out two experiments. In all of these experiments, the distance between each camera centre and cube is 10 units and the side length of cube is 1 unit. In cube, 28 lines connecting every two vertices are chosen as test lines, which are projected onto 1024×1024 image plane. The internal matrices of the three cameras are set as:

$$\mathrm{K}=\begin{pmatrix}700 & 0 & 512\\ 0 & 700 & 512\\ 0 & 0 & 1\end{pmatrix}.$$

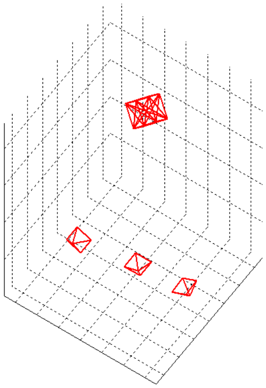
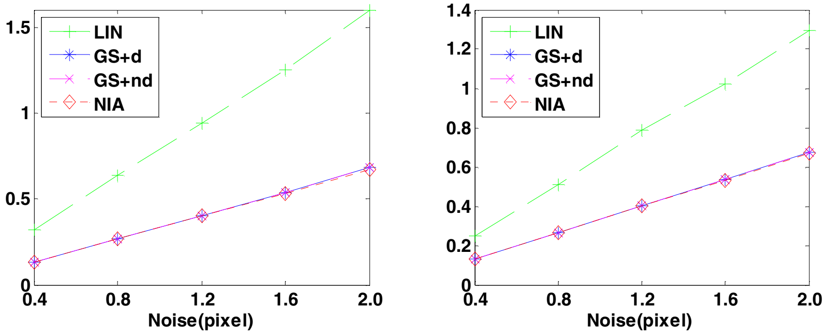


Fig. 3. Camera setup

In each of these experiments, 10 sampling points are extracted from test line randomly. After Gaussian noise with zero mean and standard deviation σ is added to each sampling point, the measured line is fitted to the 10 sampling points with the least-squares. The noise level σ varies from 0.4 to 2 with the steps of 0.4 pixels, 200 trails are performed at each noise level. The evaluated performances include: RMS (Root Mean Square) error of the normal distance; Average error of the space angle which is the angle between the real line and the estimated line in space; RMS error of the space distance which is the distance between the real line and the estimated line in space.

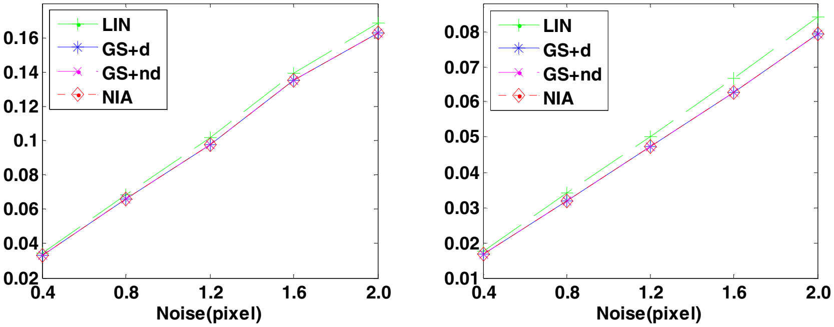


15 degrees 30 degrees

Fig. 4. RMS error of the normal distance (pixel)

Table 2. The difference between NIA and GS+nd for RMS error of the normal distance (pixel)

Noise (pixel)	0.4	0.8	1.2	1.6	2.0
15 degrees	1.68×10^{-7}	9.63×10^{-9}	2.08×10^{-12}	-0.007	-0.014
30 degrees	2.39×10^{-12}	8.67×10^{-9}	3.23×10^{-13}	-0.004	-0.007



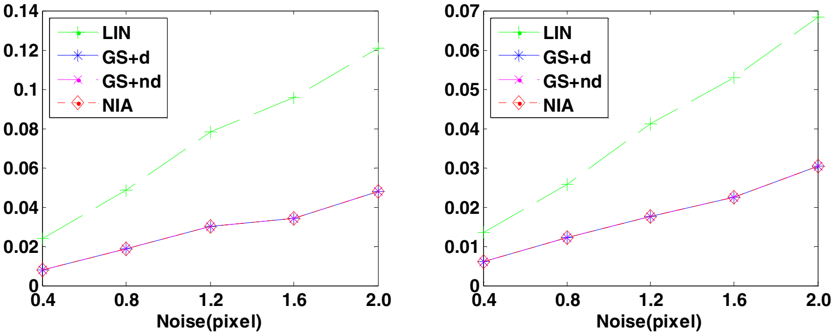
15 degrees 30 degrees

Fig. 5. Average error of the space angle (radian)

The error of normal distance in image and two errors of 3D reconstruction in space are shown in Fig.4, Fig.5 and Fig.6, respectively. From these errors, we can see that, in both experiments, the LIN has the worst performance and the other three geometric-distance algorithms have almost the same accuracy. In more detail, from Table2, it can be seen that the accuracy of the NIA is the similar to the GS+nd in the case of low noise level, but is better than it as the increase of noise level. From Table3 and Table4 which are about the differences of 3D reconstruction accuracy between the normal-distance algorithm (the NIA) and the point-line-distance algorithm (the GS+d), we can see that the NIA has better accuracy than the GS+d as the increase of noise level, not only in the case of the space angle but also in the case of the space distance.

Table 3. The difference between NIA and GS+d for average error of the space angle (radian)

Noise (pixel)	0.4	0.8	1.2	1.6	2.0
15 degrees	7.89×10^{-8}	-1.54×10^{-7}	9.66×10^{-8}	-6.94×10^{-5}	-0.00013
30 degrees	1.83×10^{-8}	-1.28×10^{-8}	-2.21×10^{-7}	-3.89×10^{-7}	-5.60×10^{-7}



15 degrees 30 degrees

Fig. 6. RMS error of the space distance (unit)

Table 4. The difference between NIA and GS+d for RMS error of the space distance (unit)

Noise (pixel)	0.4	0.8	1.2	1.6	2.0
15 degrees	3.94×10^{-8}	1.94×10^{-8}	-1.84×10^{-7}	-5.85×10^{-7}	-2.37×10^{-6}
30 degrees	-4.12×10^{-8}	-2.25×10^{-8}	-3.87×10^{-7}	-1.57×10^{-6}	-1.17×10^{-5}

5.2 Real Data

In real experiments, two sets of real images are used to test the algorithms mentioned above. One of the two sets is indoor objects as shown in Fig.7(left), in which the 36 measured lines are fitted to the points extracted by hand; the other is Tsinghua school as shown in Fig.8(left), in which the 196 measured lines are extracted by Canny edge detector. Their right figures show the reconstructions of indoor objects and Tsinghua school by NIA, respectively.

From Table 5 and Table 6 we can see that the accuracy of NIA is just the same as GS+d and GS+nd. But the running times of NIA are all less than $1/3$ times of them. Although the running time of the LIN is the least of all, its worst accuracy indicates that the LIN is not suitable to be utilized in practice.

Table 5. Results of indoor objects

	LIN	NIA	GS+d	GS+nd
RMS error of the normal distance (pixel)	0.76	0.29	0.29	0.29
average of running time(10^{-4} s)	2.06	45.90	146.74	183.02

Table 6. Results of Tsinghua school

	LIN	NIA	GS+d	GS+nd
RMS error of the normal distance (pixel)	3.54	0.19	0.19	0.19
average of the running time(10^{-4} s)	2.13	38.87	146.24	177.00

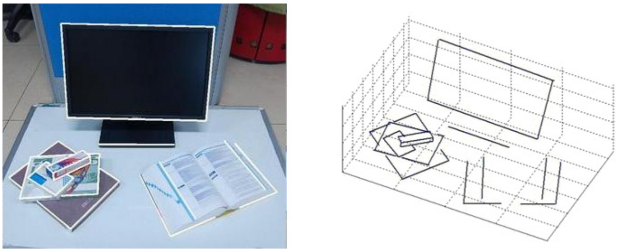


Fig. 7. Measured lines from indoor objects (white line) and its 3D reconstruction by NIA

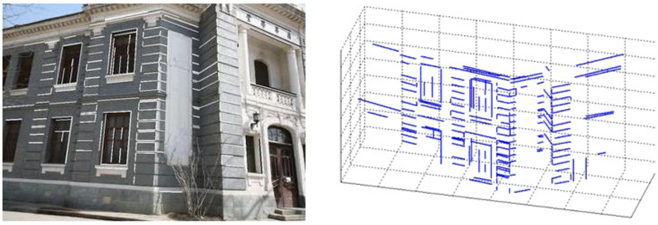


Fig. 8. Measured lines from Tsinghua school (white line in the left of pictures, black line in the right of pictures) and its 3D reconstruction by NIA

6 Conclusions

This paper discusses the line triangulation for three images. The main contributions include: the normal error is defined to describe measurement error of measured line and construct a linear representation of image line; based on the statistical property of the normal error, an optimality criterion about the normal error is presented; furthermore, a simple iterative algorithm is proposed to solve this optimality criterion, which makes the trifocal tensor constraint satisfied in each iterative cycle. Experiments show that our iterative algorithm can achieve comparable estimation accuracy with the Gold Standard algorithm, but with much less computational load.

Acknowledgements. This work was supported by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR)(201204243), the Fundamental Research Funds for the Central Universities(K5051302009), the Natural Science Foundation of China (No.61272281, No.61271297), and the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20110203110001).

Reference

[1] Beardsley, P.A., Zisserman, A., Murray, D.W.: Navigation using affine structure from motion. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 85–96. Springer, Heidelberg (1994)

[2] Hartley, R., Sturm, P.: Triangulation. Computer Vision and Image Understanding, 146–157 (1997)

[3] Wu, F., Zhang, Q., Hu, Z.: Efficient suboptimal solutions to the optimal triangulation. International Journal of Computer Vision, 77–106 (2011)

[4] Kanatani, K., Sugaya, Y., Niitsuma, H.: Triangulation from two views revisited: Hartley-Sturm vs. optimal correction. In: The British Machine Vision Conference, pp. 173–182. BMVC Press, Leeds (2008)

[5] Lindstrom, P.: Triangulation made easy. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1554–1561. IEEE Press, New York (2010)

[6] Olsson, C., Kahl, F., Oskarsson, M.: Branch and bound methods for Euclidean registration problems. IEEE Trans. Pattern Anal. Mach. Intell., 783–794 (2009)

[7] Olsson, C., Kahl, F., Hartley, R.: Projective least-squares: global solutions with local optimization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1216–1223. IEEE Press, New York (2009)

[8] Lu, F., Hartley, R.I.: A fast optimal algorithm for $\mathbf{L}_2$ triangulation. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 279–288. Springer, Heidelberg (2007)

[9] Hartley, R., Schaffalitzky, F.: $\mathbf{L}_\infty$ minimization in geometric reconstruction problems. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 504–509. IEEE Press, New York (2004)

[10] Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. IEEE Trans. Pattern Anal. Mach. Intell., 1834–1847 (2007)

[11] Kahl, F., Hartley, R.: Multiple-view geometry under the $\mathbf{L}_\infty$ -norm. IEEE Trans. Pattern Anal. Mach. Intell., 1603–1617 (2008)

[12] Bartoli, A., Sturm, P.: Structure-from-motion using lines: representation, triangulation, and bundle adjustment. Computer Vision and Image Understanding, 416–441 (2005)

[13] Spetsakis, M., Aloimonos, J.: Structure from motion using line correspondences. International Journal of Computer Vision, 171–183 (1990)

[14] Weng, J., Huang, T., Ahuja, N.: Motion and structure from line correspondences: closed-form solution, uniqueness, and optimization. IEEE Trans. Pattern Anal. Mach. Intell. 318–336 (1992)

[15] Taylor, C.J., Kriegman, D.J.: Structure and motion from line segments in multiple images. IEEE Trans. Pattern Anal. Mach. Intell. 1021–1032 (1995)

[16] Josephson, K., Kahl, F.: Triangulation of Points, Lines and Conics. Journal of Mathematical Imaging and Vision, 215–225 (2008)

[17] Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, Cambridge (2003)

[18] Ressl, C.: Geometry, Constraints and Computation of the Trifocal Tensor. PhD thesis, Technical University of Vienna (2003)

[19] Horn, R., Johnson, C.: Matrix Analysis. Cambridge University Press, Cambridge (2005)