Image and Vision Computing 32 (2014) 155–168

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Editor's Choice Article

Euclidean upgrading from segment lengths: DLT-like algorithm and its variants[☆]

Kunfeng Shi [*], Qiulei Dong, Fuchao Wu

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 13 March 2013
Received in revised form 7 December 2013
Accepted 30 December 2013

Keywords:
DLT-like algorithm
Euclidean upgrading
Multi-camera calibration
Segment lengths
Weighted DLT-like algorithm

ABSTRACT

In this paper, how to calibrate a fixed multi-camera system and simultaneously achieve a Euclidean reconstruction from a set of segments is addressed. It is well known that only a projective reconstruction could be achieved without any prior information. Here, the known segment lengths are exploited to upgrade the projective reconstruction to a Euclidean reconstruction and simultaneously calibrate the intrinsic and extrinsic camera parameters. At first, a DLT(Direct Linear Transformation)-like algorithm for the Euclidean upgrading from segment lengths is derived in a very simple way. Although the intermediate results in the DLT-like algorithm are essentially equivalent to the quadric of segments (QoS), the DLT-like algorithm is of higher accuracy than the existing linear algorithms derived from the QoS because of a more accurate way to extract the plane at infinity from the intermediate results. Then, to further improve the accuracy of Euclidean upgrading, two weighted DLT-like algorithms are presented by weighting the linear constraint equations in the original DLT-like algorithm. Finally, using the results of these linear algorithms as the initial values, a new weighted nonlinear algorithm for Euclidean upgrading is explored to recover the Euclidean structure more accurately. Extensive experimental results on both the synthetic data and the real image data demonstrate the effectiveness of our proposed algorithms in Euclidean upgrading and multi-camera calibration.

1. Introduction

With an uncalibrated multi-camera system, a projective reconstruction of the 3D points in the common field of view is possible to be obtained from the correspondence between their projections. Rectifying this projective reconstruction to a Euclidean reconstruction, i.e. performing Euclidean upgrading, is equivalent to calibrating the intrinsic and extrinsic parameters of the cameras. In literature, camera calibration methods can be divided into two categories: self-calibration and reference-object-based calibration. The self-calibration methods do not use any geometry information about the scene and they usually require some sort of constraints on the intrinsic parameters [1–6], or constraints on the motion of cameras [7–9]. The reference-object-based methods are mainly based on some calibration objects of known size, such as 3D calibration object [10,11], 2D calibration object [12,13], and 1D calibration object [14–17]. Compared with the self-calibration methods, the reference-object-based methods can provide higher calibration accuracy due to the use of Euclidean knowledge.

In the case of calibrating multi-camera systems [18,19], both 3D calibration objects and 2D calibration objects are prone to be self-occluded so that the marker points on them cannot be observed by all the referred cameras simultaneously, whereas 1D calibration objects are immune to self-occlusion. In [14,16,17], the 1D objects contain at least three marker points in order to calibrate one or more cameras, where the marker points are required to be exactly collinear and the distances between these marker points need to be measured with high precision. Liebowitz and Carlsson [20] proposed a method to recover the Euclidean structure from an affinely distorted space by the knowledge of segment lengths. This method can be used to calibrate multiple affine cameras with a segment undergoing general motions. This calibration object, a segment with two end points, is more flexible than the conventional 1D objects, since only the distance between the two end points needs to be known, and the collinearity requirement is not involved anymore. Moreover, even without the knowledge about the segment length, a Euclidean reconstruction can still be obtained despite that its global scale is undetermined.

Can the Euclidean structure be recovered from a projectively distorted space from the knowledge of segment lengths? Ronda and Valdés [21] gave this question an affirmative answer and proposed three algorithms based on the quadric of segments (QoS) defined in a higher-dimensional space by a set of segments of fixed length. These algorithms based on the QoS need to solve a set of

[☆] Editor's Choice Articles are invited and handled by a select rotating 12 member Editorial Board committee. This paper has been recommended for acceptance by Peter Sturm.

[*] Corresponding author at: National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, P.O. Box 2728, Beijing 100190, China. Tel.: +86 10 62535412; fax: +86 10 62551993.

E-mail addresses: kfshi@nlpr.ia.ac.cn (K. Shi), qldong@nlpr.ia.ac.cn (Q. Dong), fcwu@nlpr.ia.ac.cn (F. Wu).

0262-8856/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.imavis.2013.12.012

homogeneous linear equations in 55 variables, and thus require at least 54 segments in the projective space. The work of Ronda and Valdés [21] is very interesting but involves some very complex mathematics. The main contributions of them are that the relation of the QoS with the standard geometry associated to the Euclidean structure of space is given and the explicit formulae are derived to obtain the dual absolute quadric and the absolute quadratic complex from the QoS.

In this paper, we find that for the Euclidean upgrading from segment lengths, it is not necessary to use such complex mathematics as in [21], or in other words, a DLT (Direct Linear Transformation)-like algorithm can be derived in a very simple way. The derived algorithm consists of two main steps: First, a set of 4-order polynomial equations on the plane at infinity and the image of the absolute conic (IAC) are introduced from a given projective reconstruction of the scene consisting of segments with known lengths. Then, these 4-order polynomial equations are transformed into a set of inhomogeneous linear equations in 54 variables by a simple linearization technique, from which the least squares solution to the 54 variables can be computed in a linear way. Although the intermediate results in the DLT-like algorithm are essentially equivalent to the QoS, the DLT-like algorithm extracts the plane at infinity more accurately than the existing linear algorithms derived from the QoS and thus has higher accuracy in Euclidean upgrading. In addition, considering that the constraint equations in the above DLT-like algorithm have different reliabilities due to the measurement errors, we design two strategies to assign weights on these constraint equations, resulting in two weighted DLT-like algorithms. At last, a new weighted nonlinear algorithm for Euclidean upgrading is explored to refine the obtained results by these linear algorithms.

The rest of this paper is organized as follows. In Section 2 we describe the notations and related work. Section 3 presents a DLT-like algorithm for Euclidean upgrading from segment lengths and discusses the relationship between the DLT-like algorithm and existing algorithms. Then, two weighted DLT-like algorithms are proposed in Section 4. Section 5 introduces a weighted nonlinear algorithm to refine the results by these linear algorithms. The experimental results are reported in Section 6, followed by some concluding marks in Section 7.

2. Preliminaries

2.1. Notations

Here, the homogeneous coordinates of a space point are denoted by $\mathbf{X} = [x_1, x_2, x_3, 1]^{\mathrm{T}}$, and the corresponding inhomogeneous coordinates are denoted by $\widetilde{\mathbf{X}} = [x_1, x_2, x_3]^{\mathrm{T}}$. A segment is determined by its two end points $\{\mathbf{X}, \mathbf{Y}\}$. Given the images of m ($m \geq 4$) segments in general position under n ($n \geq 2$) views, the projective camera matrices and the projective reconstruction of the segments can be obtained from the correspondences between the images of the $2m$ end points [5], which are related to the real Euclidean structure by a 4×4 transformation matrix. The projective camera matrices of the n views are denoted by P_j $(1 \leq j \leq n)$, and the projective reconstruction of the end points of segments $\{\mathbf{X}_i, \mathbf{Y}_i\}$ $(1 \leq i \leq m)$ are denoted by $\mathbf{X}_{pi} = \left(\widetilde{\mathbf{X}}_{pi}^{\mathrm{T}}, 1\right)^{\mathrm{T}}$ and $\mathbf{Y}_{pi} = \left(\widetilde{\mathbf{Y}}_{pi}^{\mathrm{T}}, 1\right)^{\mathrm{T}}$.

2.2. Linear algorithms derived from the QoS

Ronda and Valdés [21] proposed three algorithms for Euclidean upgrading with a set of segments of known lengths. These algorithms are designed based on the quadric of segments (QoS): For segments $\{\mathbf{X}, \mathbf{Y}\}$, let $\sigma(\mathbf{X}, \mathbf{Y})$ be a 10-vector composed of the elements of the symmetric matrix $(\mathbf{X}\mathbf{Y}^{\mathrm{T}} + \mathbf{Y}\mathbf{X}^{\mathrm{T}})$. Then, for all the segments of fixed length d, $\sigma(\mathbf{X}, \mathbf{Y})$ lie on a quadric called the QoS of length d. The QoS can be written as $\mathrm{C}_1 + \frac{d^2}{2}\mathrm{C}_2$, where the dimensions of the linear spaces spanned by C_1 and C_2 are respectively 20 and 35. Given a segment $\{\mathbf{X}_i, \mathbf{Y}_i\}$ of length d_i, the constraint equation on C_1 and C_2 is obtained as

$$\sigma^{\mathrm{T}}\left(\mathbf{X}_{pi}, \mathbf{Y}_{pi}\right)\left(\mathrm{C}_1 + \frac{d_i^2}{2}\mathrm{C}_2\right)\sigma\left(\mathbf{X}_{pi}, \mathbf{Y}_{pi}\right) = 0. \tag{1}$$

Given at least 54 segments of known lengths, C_1 and C_2 can be determined uniquely. Then, C_1 and C_2 are used respectively to extract the Euclidean structure in the following three algorithms (C1S, C1A, and C2A):

2.2.1. C1S and C1A

C_1 in Eq. (1) determines the real structure up to a Euclidean transformation with an unknown scale. An explicit formula to derive the absolute quadratic complex from C_1 is given in [21], where the absolute quadratic complex is the quadric in terms of the Plücker coordinates of the lines which intersect the absolute conic [6]. A rectifying matrix is then extracted from the absolute quadratic complex to recover a Euclidean reconstruction [6], where the scale ambiguity can be determined by changing the overall scale of the structure such that the length constraints are satisfied. This algorithm is called C1S.

The scale ambiguity of the Euclidean reconstruction can also be reduced with the affine adjustment algorithm [20]. Here, an affine transformation matrix is computed linearly to minimize the total reconstruction error of segment lengths. The algorithm C1S followed by the affine adjustment step is called C1A. And the experimental results in [21] show that the affine adjustment step can improve the accuracy of Euclidean upgrading.

2.2.2. C2A

C_2 in Eq. (1) encodes the information of the plane at infinity. In the implementation of the algorithm C2A in [21], the plane at infinity is computed as the polar plane of the center of a sphere with respect to the sphere. For a given space point $\mathbf{Y}$, the sphere $\mathrm{A}_{\mathbf{Y}}$ with center $\mathbf{Y}$ and radius infinity is extracted from C_2 through

$$(\mathrm{A}_{\mathbf{Y}})_{ij} = \sigma^{\mathrm{T}}(\mathbf{e}_i, \mathbf{Y})\mathrm{C}_2\sigma\left(\mathbf{e}_j, \mathbf{Y}\right),$$

where $(\mathrm{A}_{\mathbf{Y}})_{ij}$ is the element in the i-th row and the j-th column of $\mathrm{A}_{\mathbf{Y}}$, and $\mathbf{e}_i$ is the i-th column of the 4×4 identity matrix. And then the plane at infinity is obtained as the polar plane of $\mathbf{Y}$ with respect to $\mathrm{A}_{\mathbf{Y}}$, i.e.

$$\pi_\infty = \mathrm{A}_{\mathbf{Y}}\mathbf{Y}. \tag{2}$$

When C_2 is not accurate due to the error in the projective reconstruction of the segments, the result of Eq. (2) is dependent on the choice of $\mathbf{Y}$. Therefore, multiple values of $\mathbf{Y}$ are used to compute a matrix which is composed of multiple estimations of the plane at infinity and then a final result is obtained by the SVD of this matrix. Next, a rectifying matrix is constructed from the plane at infinity to transform the projective reconstruction to an affine reconstruction. At last, the affine adjustment [20] is performed to refine the affine reconstruction to a Euclidean reconstruction.

A simpler way to directly extract the plane at infinity from C_2 was also given in [21]. In this simpler way, the plane at infinity is computed as

$$\pi_\infty = \begin{pmatrix} \sigma^{\mathrm{T}}(\mathbf{e}_1, \mathbf{e}_\alpha)\mathrm{C}_2\sigma\left(\mathbf{e}_\beta, \mathbf{e}_\gamma\right) \\ \vdots \\ \sigma^{\mathrm{T}}(\mathbf{e}_4, \mathbf{e}_\alpha)\mathrm{C}_2\sigma\left(\mathbf{e}_\beta, \mathbf{e}_\gamma\right) \end{pmatrix}, \tag{3}$$

where α, β, and γ represent three arbitrary numbers in {1,2,3,4}. When the projective reconstruction of the segments is influenced by noise, the estimation of C_2 in Eq. (1) does not satisfy its internal constraints and the accuracy of the computed π_∞ in Eq. (3) is dependent on the choices of α, β, and γ. The influence of the choices of α, β, and γ was not tested in [21], and we will give some analyses in the later part of this paper.

3. DLT-like algorithm

This section proposes a DLT-like algorithm for Euclidean upgrading from segment lengths, and discusses the relationship between the DLT-like algorithm and the existing linear algorithms derived from the QoS.

3.1. Constraint equations

Given a projective reconstruction of the segments and the corresponding projective camera matrices, we can always choose a projective coordinate system such that the first projective camera projection matrix has the canonical form $P_1 = (I|0)$. Let $\pi_\infty = (\mathbf{n}^T,1)^T$ be the coordinates of the plane at infinity in this projective space and K_j $(1 \le j \le n)$ the intrinsic matrix of the j-th view. Then, according to the theory of stratified reconstruction [5], the Euclidean camera projection matrices of the n views and the Euclidean reconstruction $\{\mathbf{X}_{ei},\mathbf{Y}_{ei}\}$ of the segments $\{\mathbf{X}_i,\mathbf{Y}_i\}$ can be expressed respectively as

$$P_{ej} \sim P_j \begin{pmatrix} sK_1 & 0 \\ -s\mathbf{n}^T K_1 & 1 \end{pmatrix}, \quad (1 \le j \le n), \tag{4}$$

$$\mathbf{X}_{ei} \sim \begin{pmatrix} (sK_1)^{-1} & 0 \\ \mathbf{n}^T & 1 \end{pmatrix} \mathbf{X}_{pi}, \ \ \widetilde{\mathbf{X}}_{ei} = \frac{(sK_1)^{-1}\widetilde{\mathbf{X}}_{pi}}{1+\mathbf{n}^T\widetilde{\mathbf{X}}_{pi}}, (1 \le i \le m), \tag{5}$$

$$Y_{ei} \sim \begin{pmatrix} (sK_1){-}1 & 0 \\ n^T & 1 \end{pmatrix} Y_{pi}, \widetilde{Y}_{ei} = \frac{(sK_1)^{-1}\widetilde{Y}_{pi}}{1+n^T\widetilde{Y}_{pi}}, (1 \le i \le m), \tag{6}$$

where "~" denotes equality up to an unknown scalar multiplication, and s is the overall scale.

Since $\left\{\widetilde{\mathbf{X}}_{ei}, \widetilde{\mathbf{Y}}_{ei}\right\}$ is the Euclidean reconstruction of the segments $\{\mathbf{X}_i, \mathbf{Y}_i\}$ of length d_i, the following equation holds

$$\left(\widetilde{\mathbf{X}}_{ei} - \widetilde{\mathbf{Y}}_{ei}\right)^T \left(\widetilde{\mathbf{X}}_{ei} - \widetilde{\mathbf{Y}}_{ei}\right) = d_i^2. \tag{7}$$

Denote $\omega = (sK_1)^{-T}(sK_1)^{-1}$ as the image of the absolute conic under the first view. Substituting Eqs. (5) and (6) into Eq. (7), a constraint equation on $(\mathbf{n},\omega)$ is obtained as

$$\left(\frac{\widetilde{\mathbf{X}}_{pi}}{1+\mathbf{n}^T\widetilde{\mathbf{X}}_{pi}} - \frac{\widetilde{\mathbf{Y}}_{pi}}{1+n^T\widetilde{\mathbf{Y}}_{pi}}\right)^T \omega \left(\frac{\widetilde{\mathbf{X}}_{pi}}{1+n^T\widetilde{X}_{pi}} - \frac{\widetilde{\mathbf{Y}}_{pi}}{1+n^T\widetilde{\mathbf{Y}}_{pi}}\right) = d_i^2. \tag{8}$$

Eq. (8) can be rewritten as

$$\left[\left(1+n^T\widetilde{Y}_{pi}\right)\widetilde{\mathbf{X}}_{pi} - \left(1+n^T\widetilde{\mathbf{X}}_{pi}\right)\widetilde{\mathbf{Y}}_{pi}\right]^T \omega \left[\left(1+n^T\widetilde{\mathbf{Y}}_{pi}\right)\widetilde{\mathbf{X}}_{pi} - \left(1+n^T\widetilde{\mathbf{X}}_{pi}\right)\widetilde{\mathbf{Y}}_{pi}\right] = d_i^2\left(1+n^T\widetilde{\mathbf{X}}_{pi}\right)^2\left(1+n^T\widetilde{\mathbf{Y}}_{pi}\right)^2. \tag{9}$$

This is a 4-order polynomial equation in 9 variables (3 variables in $\mathbf{n}$ and 6 different variables in the symmetric matrix ω). Given a projective reconstruction of 9 segments, a nonlinear solution to the Euclidean upgrading can be obtained from Eq. (9) in theory. However, it is very difficult in practice to solve such a set of 4-order polynomial equations in 9 variables. In order to obtain a linear solution of $(\mathbf{n},\omega)$, a linearization of Eq. (9) will be derived in the next subsection.

3.2. Linearization

For $\mathbf{n} = [n_1,n_2,n_3]^T$, its anti-symmetric matrix is defined as

$$[\mathbf{n}]_\times = \begin{pmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{pmatrix}.$$

Define a 6 × 6 matrix Λ and a 6-dimensional vector $p\left(\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right)$ as

$$\Lambda = \begin{pmatrix} \omega & \omega[n]_\times^T \\ [n]_\times \omega & [n]_\times^T \omega [n]_\times \end{pmatrix}, p\left(\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right) = \begin{pmatrix} \widetilde{\mathbf{Y}}_{pi} - \widetilde{\mathbf{X}}_{pi} \\ \widetilde{\mathbf{X}}_{pi} \times \widetilde{\mathbf{Y}}_{pi} \end{pmatrix}, \tag{10}$$

then, the left hand side of Eq. (9) is rewritten as

$$\begin{aligned} & \left[\left(1+\mathbf{n}^T\widetilde{\mathbf{Y}}_{pi}\right)\widetilde{\mathbf{X}}_{pi} - \left(1+\mathbf{n}^T\widetilde{\mathbf{X}}_{pi}\right)\widetilde{\mathbf{Y}}_{pi}\right]^T \omega \\ & \quad \left[\left(1+\mathbf{n}^T\widetilde{\mathbf{Y}}_{pi}\right)\widetilde{\mathbf{X}}_{pi} - \left(1+\mathbf{n}^T\widetilde{\mathbf{X}}_{pi}\right)\widetilde{\mathbf{Y}}_{pi}\right] \\ & = \left\{\left(\widetilde{\mathbf{X}}_{pi} - \widetilde{\mathbf{Y}}_{pi}\right) + \left[\left(n^T\widetilde{\mathbf{Y}}_{pi}\right)\widetilde{\mathbf{X}}_{pi} - \left(n^T\widetilde{\mathbf{X}}_{pi}\right)\widetilde{\mathbf{Y}}_{pi}\right]\right\}^T \omega \\ & \quad \left\{\left(\widetilde{\mathbf{X}}_{pi} - \widetilde{\mathbf{Y}}_{pi}\right) + \left[\left(n^T\widetilde{Y}_{pi}\right)\widetilde{\mathbf{X}}_{pi} - \left(n^T\widetilde{\mathbf{X}}_{pi}\right)\widetilde{\mathbf{Y}}_{pi}\right]\right\} \\ & = \left\{\left(\widetilde{\mathbf{X}}_{pi} - \widetilde{\mathbf{Y}}_{pi}\right) + [n]_\times\left(\widetilde{\mathbf{X}}_{pi} \times \widetilde{\mathbf{Y}}_{pi}\right)\right\}^T \omega \\ & \quad \left\{\left(\widetilde{\mathbf{X}}_{pi} - \widetilde{\mathbf{Y}}_{pi}\right) + [n]_\times\left(\widetilde{\mathbf{X}}_{pi} \times \widetilde{\mathbf{Y}}_{pi}\right)\right\} \\ & = p^T\left(\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right) \Lambda p\left(\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right). \end{aligned} \tag{11}$$

For $\mathbf{a} = (a_1,a_2,a_3)^T$, define vector $\mathbf{q}(\mathbf{a})$ as

$$\mathbf{q}(\mathbf{a}) = \left(a_1^2, a_2^2, a_3^2, \sqrt{2}a_1a_2, \sqrt{2}a_1a_3, \sqrt{2}a_2a_3, \sqrt{2}a_1, \sqrt{2}a_2, \sqrt{2}a_3, 1\right)^T.$$

The symmetric matrix $\Gamma = \mathbf{q}(\mathbf{n})\mathbf{q}^T(\mathbf{n})$ is given by

$$\Gamma = \begin{pmatrix} n_1^4 & n_1^2n_2^2 & n_1^2n_3^2 & \sqrt{2}n_1^3n_2 & \sqrt{2}n_1^3n_3 & \sqrt{2}n_1^2n_2n_3 & \sqrt{2}n_1^3 & \sqrt{2}n_1^2n_2 & \sqrt{2}n_1^2n_3 & n_1^2 \\ n_1^2n_2^2 & n_2^4 & n_2^2n_3^2 & \sqrt{2}n_1n_2^3 & \sqrt{2}n_1n_2^2n_3 & \sqrt{2}n_2^3n_3 & \sqrt{2}n_1n_2^2 & \sqrt{2}n_2^3 & \sqrt{2}n_2^2n_3 & n_2^2 \\ n_1^2n_3^2 & n_2^2n_3^2 & n_3^4 & \sqrt{2}n_1n_2n_3^2 & \sqrt{2}n_1n_3^3 & \sqrt{2}n_2n_3^3 & \sqrt{2}n_1n_3^2 & \sqrt{2}n_2n_3^2 & \sqrt{2}n_3^3 & n_3^2 \\ \sqrt{2}n_1^3n_2 & \sqrt{2}n_1n_2^3 & \sqrt{2}n_1n_2n_3^2 & 2n_1^2n_2^2 & 2n_1^2n_2n_3 & 2n_1n_2^2n_3 & 2n_1^2n_2 & 2n_1n_2^2 & 2n_1n_2n_3 & \sqrt{2}n_1n_2 \\ \sqrt{2}n_1^3n_3 & \sqrt{2}n_1n_2^2n_3 & \sqrt{2}n_1n_3^3 & 2n_1^2n_2n_3 & 2n_1^2n_3^2 & 2n_1n_2n_3^2 & 2n_1^2n_3 & 2n_1n_2n_3 & 2n_1n_3^2 & \sqrt{2}n_1n_3 \\ \sqrt{2}n_1^2n_2n_3 & \sqrt{2}n_2^3n_3 & \sqrt{2}n_2n_3^3 & 2n_1n_2^2n_3 & 2n_1n_2n_3^2 & 2n_2^2n_3^2 & 2n_1n_2n_3 & 2n_2^2n_3 & 2n_2n_3^2 & \sqrt{2}n_2n_3 \\ \sqrt{2}n_1^3 & \sqrt{2}n_1n_2^2 & \sqrt{2}n_1n_3^2 & 2n_1^2n_2 & 2n_1^2n_3 & 2n_1n_2n_3 & 2n_1^2 & 2n_1n_2 & 2n_1n_3 & \sqrt{2}n_1 \\ \sqrt{2}n_1^2n_2 & \sqrt{2}n_2^3 & \sqrt{2}n_2n_3^2 & 2n_1n_2^2 & 2n_1n_2n_3 & 2n_2^2n_3 & 2n_1n_2 & 2n_2^2 & 2n_2n_3 & \sqrt{2}n_2 \\ \sqrt{2}n_1^2n_3 & \sqrt{2}n_2^2n_3 & \sqrt{2}n_3^3 & 2n_1n_2n_3 & 2n_1n_3^2 & 2n_2n_3^2 & 2n_1n_3 & 2n_2n_3 & 2n_3^2 & \sqrt{2}n_3 \\ n_1^2 & n_2^2 & n_3^2 & \sqrt{2}n_1n_2 & \sqrt{2}n_1n_3 & \sqrt{2}n_2n_3 & \sqrt{2}n_1 & \sqrt{2}n_2 & \sqrt{2}n_3 & 1 \end{pmatrix}. \tag{12}$$

Then, the right hand side of Eq. (9) becomes

$$d_i^2\left(1+\mathbf{n}^{\mathrm{T}}\widetilde{\mathbf{X}}_{pi}\right)^2\left(1+\mathbf{n}^{\mathrm{T}}\widetilde{\mathbf{Y}}_{pi}\right)^2 = d_i^2\mathbf{q}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi}\right)\Gamma\mathbf{q}\left(\widetilde{\mathbf{Y}}_{pi}\right). \tag{13}$$

According to Eqs. (11) and (13), Eq. (9) can be rewritten as

$$\mathbf{p}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)\Lambda\mathbf{p}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)-d_i^2\mathbf{q}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi}\right)\Gamma\mathbf{q}\left(\widetilde{\mathbf{Y}}_{pi}\right)=0. \tag{14}$$

The 6 × 6 symmetric matrix Λ contains 21 different elements. Let $\Lambda_{i,j}$ be the element in the i-th row and the j-th column of Λ. It is noted that $\omega[\mathbf{n}]_\times^{\mathrm{T}}$ (i.e. the upper right 3 × 3 sub-matrix of Λ) has zero trace, which means $\Lambda_{3,6}$ is equal to $-\Lambda_{1,4}-\Lambda_{2,5}$. Therefore, Λ can be linearly represented by the following 20 elements:

$$\lambda = (\Lambda_{1,1},\ldots,\Lambda_{16},\Lambda_{22},\ldots,\Lambda_{26},\Lambda_{33},\ldots,\Lambda_{35},\Lambda_{44},\ldots,\Lambda_{46}, \\ \Lambda_{55},\Lambda_{56},\Lambda_{66})^{\mathrm{T}}. \tag{15}$$

There are 34 different monomials $n_1^i n_2^j n_3^k (1 \le i+j+k \le 4)$ in Γ, and all these monomials can form the following vector when sorted in the graded lexicographic order:

$$\gamma = (n_1^4, n_1^3n_2, n_1^3n_3, n_1^2n_2^2, n_1^2n_2n_3, n_1^2n_3^2, n_1n_2^3, n_1n_2^2n_3, \\ n_1n_2n_3^2, n_1n_3^3, n_2^4, n_2^3n_3, n_2^2n_3^2, n_2n_3^3, n_3^4, n_1^3, n_1^2n_2, \\ n_1^2n_3, n_1n_2^2, n_1n_2n_3, n_1n_3^2, n_2^3, n_2^2n_3, n_2n_3^2, n_3^3, n_1^2, \\ n_1n_2, n_1n_3, n_2^2, n_2n_3, n_3^2, n_1, n_2, n_3)^{\mathrm{T}}. \tag{16}$$

Let $\mathbf{c}_\lambda\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)$ and $\mathbf{c}_\gamma\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)$ be the coefficient vectors of λ and γ in Eq. (14), and then the following linear equation in 54 variables (20 variables in λ and 34 variables in γ) is obtained from Eq. (14):

$$\left(\mathbf{c}_\lambda\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)\mathbf{c}_\gamma\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)\right)\begin{pmatrix}\lambda\\ \gamma\end{pmatrix} = d_i^2. \tag{17}$$

3.3. Algorithm overview

Based on the derivations in Sections 3.1 and 3.2, the DLT-like algorithm (abbreviated as DLT-like) for Euclidean upgrading from segment lengths is performed as follows: From the projective reconstruction $\left\{\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right\}$ of a set of segments of length d_i, a set of linear equations is established by Eq. (17):

$$\begin{pmatrix}\mathbf{c}_\lambda\left(\widetilde{\mathbf{X}}_{p1},\widetilde{\mathbf{Y}}_{p1}\right) & \mathbf{c}_\gamma\left(\widetilde{\mathbf{X}}_{p1},\widetilde{\mathbf{Y}}_{p1}\right)\\ \vdots & \vdots \\ \mathbf{c}_\lambda\left(\widetilde{\mathbf{X}}_{pm},\widetilde{\mathbf{Y}}_{pm}\right) & \mathbf{c}_\gamma\left(\widetilde{\mathbf{X}}_{pm},\widetilde{\mathbf{Y}}_{pm}\right)\end{pmatrix}\begin{pmatrix}\lambda\\ \gamma\end{pmatrix} = \begin{pmatrix}d_1^2\\ \vdots \\ d_m^2\end{pmatrix}. \tag{18}$$

Denote the coefficient matrix and the constant vector in Eq. (18) as C and **d** respectively, and then the least squares solution to Eq. (18) is

$$\begin{pmatrix}\lambda\\ \gamma\end{pmatrix} = \mathrm{C}^{+}\mathbf{d}.$$

where C^{+} denotes the pseudo-inverse of C.

According to Eq. (16), the last three elements of $\gamma = (\gamma_1,\gamma_2,\ldots,\gamma_{34})^{\mathrm{T}}$ determine the normal vector of the plane at infinity:

$$\mathbf{n} = (\gamma_{32},\gamma_{33},\gamma_{34})^{\mathrm{T}}. \tag{19}$$

And according to Eqs. (16) and (30), the IAC is extracted from $\lambda = (\lambda_1,\lambda_2,\ldots,\lambda_{20})^{\mathrm{T}}$ by

$$\omega = \begin{pmatrix}\lambda_1 & \lambda_2 & \lambda_3\\ \lambda_2 & \lambda_7 & \lambda_8\\ \lambda_3 & \lambda_8 & \lambda_{12}\end{pmatrix}.$$

Then, the intrinsic matrix K_1 and the overall scale s are determined by the Cholesky decomposition of $\omega^{-1} = s^2\mathrm{K}_1\mathrm{K}_1^{\mathrm{T}}$.

After computing ($\mathbf{n}$, $s\mathrm{K}_1$), the Euclidean camera projection matrices and the Euclidean reconstruction of the segments are obtained according to Eqs. (4), (5), and (6). The affine adjustment algorithm [20] is then performed to refine the obtained Euclidean structure and the camera projection matrices. Finally, the intrinsic matrices and extrinsic parameters (K_j,R_j,$\mathbf{C}_j$) ($1 \le j \le n$) are obtained from the decompositions of P_{ej} in the form of $\mathrm{P}_{ej} \sim \mathrm{K}_j\mathrm{R}_j[\mathrm{I} - \mathbf{C}_j]$.

It is noted that the coefficient matrix C in Eq. (18) is quartic in the coordinates of the end points and the large condition number of C may cause instability of the DLT-like algorithm [22]. Therefore, the end points need to be normalized to improve the stability. Similarly, the normalization step is also necessary to Ronda and Valdés's linear algorithms derived from the QoS [21]. A matter of concern is that some end points in the projective reconstruction of the segments may be near infinity and then the computation of the normalization matrix will be dominated by the influence of those points. A solution to this problem is to first transform the projective reconstruction to a quasi-affine reconstruction [5] and then construct the normalization matrix. This two-step transformation has another benefit. Because the resulting space points are not split across the plane at infinity [5] and the centroid of these points is at the origin of the coordinate frame [22], the plane at infinity will not pass through the origin. It is clear that the last coordinate of the plane at infinity will not be zero and therefore parametrizing the plane at infinity as $\pi_\infty = (n_1,n_2,n_3,1)^{\mathrm{T}}$ will not cause instability.

3.4. Relationship between DLT-like and the existing algorithms derived from the QoS

The DLT-like algorithm and the existing linear algorithms (C1S, C1A, and C2A) in [21] are all derived from the knowledge of segment lengths, and they all require at least 54 segments. Indeed, DLT-like is closely related to these linear algorithms.

As mentioned in [21], the constraint equation on the QoS in Eq. (1) can be rewritten as

$$\left(\mathbf{X}_{pi}\wedge\mathbf{Y}_{pi}\right)^{\mathrm{T}}\Sigma\left(\mathbf{X}_{pi}\wedge\mathbf{Y}_{pi}\right)+\frac{d_i^2}{2}\sigma^{\mathrm{T}}\left(\mathbf{X}_{pi},\mathbf{Y}_{pi}\right)\mathrm{C}_2\sigma\left(\mathbf{X}_{pi},\mathbf{Y}_{pi}\right) \\ =0, \tag{20}$$

where ($\mathbf{X}_{pi} \wedge \mathbf{Y}_{pi}$) is the Plücker coordinates of the line containing the segment $\{\mathbf{X}_{pi},\mathbf{Y}_{pi}\}$ and Σ is the 6 × 6 matrix representing the absolute quadratic complex [6]. The relationship between the involved vectors and matrices in Eqs. (14) and (20) is summarized as below:

1. According to the representation of the Plücker coordinates by Stan Birchfield [23], $\mathbf{p}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)$ in Eq. (10) are exactly the Plücker coordinates ($\mathbf{X}_{pi} \wedge \mathbf{Y}_{pi}$) of the line passing through $\widetilde{\mathbf{X}}_{pi}$ and $\widetilde{\mathbf{Y}}_{pi}$. Besides, by writing Λ and Σ in terms of the elements of $\mathbf{n}$ and ω with the help of symbolic computation software, it is observed that Λ is the absolute quadratic complex Σ when the Plücker coordinates are defined as in Eq. (10). Therefore, it is obvious that

$$\mathbf{p}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)\Lambda\mathbf{p}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)\sim\left(\mathbf{X}_{pi}\wedge\mathbf{Y}_{pi}\right)^{\mathrm{T}}\Sigma\left(\mathbf{X}_{pi}\wedge\mathbf{Y}_{pi}\right). \tag{21}$$

As seen from Eq. (21), the first term in Eq. (14) is equivalent to the first term in Eq. (20) up to a scale factor.

2. It is easy to verify that

$$\mathrm{q}\left(\widetilde{\mathbf{X}}_{pi}\right) = \mathrm{H}_2\sigma\left(\mathbf{X}_{pi}, \mathbf{X}_{pi}\right), \quad \mathrm{q}\left(\widetilde{\mathbf{Y}}_{pi}\right) = \mathrm{H}_2\sigma\left(\mathbf{Y}_{pi}, \mathbf{Y}_{pi}\right), \tag{22}$$

where

$$\mathrm{H}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

After writing Γ and C_2 in terms of the elements of $\mathbf{n}$, we can verify that Γ can be obtained by reordering the elements of C_2 as

$$\Gamma \sim \mathrm{H}_2^{-\mathrm{T}}\mathrm{C}_2\mathrm{H}_2^{-1}, \tag{23}$$

According to Eqs. (22), (23), and $\sigma^{\mathrm{T}}(\mathbf{X}_{pi},\mathbf{X}_{pi})\mathrm{C}_2\sigma(\mathbf{Y}_{pi},\mathbf{Y}_{pi}) = \sigma^{\mathrm{T}}(\mathbf{X}_{pi},\mathbf{Y}_{pi})\mathrm{C}_2\sigma(\mathrm{X}_{pi},\mathbf{Y}_{pi})$, it is obvious that the second term in Eq. (14) is equivalent to the second term in Eq. (20) up to a scale factor,

$$\mathbf{q}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi}\right)\Gamma\mathbf{q}\left(\widetilde{\mathbf{Y}}_{pi}\right) \sim \sigma^{\mathrm{T}}\left(\mathbf{X}_{pi}, \mathbf{Y}_{pi}\right)\mathrm{C}_2\sigma\left(\mathbf{X}_{pi}, \mathbf{Y}_{pi}\right). \tag{24}$$

From the results in Eqs. (21) and (24), we can see that Eq. (14) is equivalent to Eq. (20) up to a scale factor. In addition, since Eq. (20) is equivalent to Eq. (1), we see that Eq. (14) and Eq. (1) are also equivalent up to a scale factor. Actually, our DLT-like algorithm and the linear algorithms in [21] use different parametrization strategies to represent the same constraint equations. According to the proof in [21], C_1 and C_2 can be uniquely determined from enough number of segments, and therefore Λ and Γ in the DLT-like algorithm can be uniquely determined as well. Since the parametrizations of Eqs. (14) and (20) are related by a linear transformation, the computation results of (Λ,Γ) and (Σ,C_2) will have the same first-order error propagation model [24]. Besides, data normalization on the end points can reduce the second-order and higher-order error terms. Therefore, these intermediate results will have similar accuracy.

It is worth pointing out that in spite of the mathematical equivalence between the constraint equations of the DLT-like algorithm and the existing linear algorithms in [21], the accuracies of Euclidean upgrading in these linear algorithms are different as shown in the experiments in Section 6 mainly because their strategies to extract $(\mathbf{n},\omega)$ from the intermediate results are different. Furthermore, since DLT-like, C1A, and C2A all use affine adjustment as the last step of Euclidean upgrading and the results of affine adjustment are dependent only on the estimations of $\mathbf{n}$, the different accuracies among these three linear algorithms are mainly caused by their different estimations of $\mathbf{n}$.

The computation result in Eq. (3) is a 4-dimensional vector composed of four elements of C_2, where the positions of these four elements are dependent on the choices of α, β, and γ. Besides, we find that when α, β, and γ are all equal to 4, the positions of these four elements in C_2 will correspond to the positions of $(n_1,n_2,n_3,1)$ in Γ by the relationship in Eq. (23), and therefore this strategy to extract π_∞ from C_2 will be equivalent to our strategy to extract $\mathbf{n}$ from Γ in Eq. (19). As a result, the DLT-like algorithm has the same first-order error propagation model as the linear algorithm that extracts the plane at infinity from C_2 by (3) with $\alpha = 4, \beta = 4$, and $\gamma = 4$. However, the implementation of the DLT-like algorithm is easier and the parametrization in the DLT-like algorithm is more direct to show the relation between $(\mathbf{n},\omega)$ and the intermediate matrices.

If the plane at infinity is extracted from C_2 using Eq. (3) with other choices of α, β, and γ, the estimation of the plane at infinity may be unstable. When at least one element of $\mathbf{n}$ is zero, if we choose the values of α, β, and γ such that $n_\alpha n_\beta n_\gamma$ is zero, then it is easy to verify that the estimation of the plane at infinity by Eq. (3) with noise-free C_2 will be a 4-dimensional zero vector $(0,0,0,0)^{\mathrm{T}}$. As a result, $\mathbf{n}$ cannot be recovered by computing the inhomogeneous coordinates of this 4-dimensional zero vector, and therefore a degeneracy occurs. It is unavoidable that the computation result of C_2 is influenced by noise in practice, and thus a 4-dimensional nonzero vector will be obtained by Eq. (3) with inaccurate C_2. The estimation of $\mathbf{n}$ recovered by computing the inhomogeneous coordinates of this 4-dimensional nonzero vector is only determined by random noise, and therefore is a wrong estimation. After the original projective reconstruction is transformed to quasi-affine and then normalized, the magnitudes of the elements of $\mathbf{n}$ are usually small. When at least one element of $\mathbf{n}$ is close to zero, recovering $\mathbf{n}$ by computing the inhomogeneous coordinates of the result of Eq. (3) is close to a degeneracy for the referred choices of α, β, and γ, and therefore the resulting estimation of $\mathbf{n}$ has high sensitivity to noise.

4. Weighted DLT-like algorithms

The DLT-like algorithm is a least squares algorithm, where the cost function is the sum of squared residuals. This cost function is not statistically meaningful [5], and the different reliabilities of the constraint equations in the DLT-like algorithm are not taken into account. To further improve the accuracy of the DLT-like algorithm, we assign each constraint equation with a weight which reflects its reliability to some extent.

It is obvious that the accuracies of the end points $\left\{\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right\}$ $(i = 1, 2, \ldots, m)$ have an important influence on the reliabilities of the constraint equations in Eq. (18). The more accurate the end points $\left\{\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right\}$ are, the more reliably the corresponding constraint equation is used for Euclidean upgrading. Therefore, a natural idea to design weights on these constraint equations is to make use of the different accuracies of $\left\{\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right\}$. Unfortunately, the true errors of $\left\{\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right\}$ cannot be known except in the case of synthetic data experiments. However, the covariance matrices of the end points can be used to represent their uncertainties. Thus, we design the weights on these constraint equations based on the covariance matrices of $\widetilde{\mathbf{X}}_{pi}$ and $\widetilde{\mathbf{Y}}_{pi}$. Here, their covariance matrices $\mathrm{V}_{\widetilde{\mathbf{X}}_{pi}}$ and $\mathrm{V}_{\widetilde{\mathbf{Y}}_{pi}}$ are computed using the first-order error propagation method [25,5], where the noise on the image points is supposed to be independent, identically distributed, and isotropic Gaussian. After computing the covariance matrices of the end points, we propose two methods to compute the weights on the constraint equations with and without a given initial value of $(\mathbf{n},\omega)$ respectively, resulting in two weighted DLT-like algorithms.

4.1. Weighted DLT-like algorithm with an initial value

Suppose we already have an initial value of $(\mathbf{n},\omega)$ (e.g. the result calculated by the DLT-like algorithm), and then we weight the constraint equations in the DLT-like algorithm in a similar way to the well-known "iterative weighted least squares" [26], where the weights on the constraint equations are computed as the inverses of the standard derivations of their residuals.

According to Eqs. (14) and (17), the residual of the i-th constraint equation in Eq. (18) can be rewritten as

$$\epsilon_i = \mathbf{p}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right)\Sigma\mathbf{p}\left(\widetilde{\mathbf{X}}_{pi}, \widetilde{\mathbf{Y}}_{pi}\right) - d_i^2\mathbf{q}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi}\right)\Gamma\mathbf{q}\left(\widetilde{\mathbf{Y}}_{pi}\right).$$

Then, the Jacobians of ϵ_i with respect to $\widetilde{\mathbf{X}}_{pi}$ and $\widetilde{\mathbf{Y}}_{pi}$ are given by

$$\frac{\partial\epsilon_i}{\partial\widetilde{\mathbf{X}}_{pi}} = 2\mathbf{p}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)\Sigma\frac{\partial\mathbf{p}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)}{\partial\widetilde{\mathbf{X}}_{pi}} - d_i^2\mathbf{q}^{\mathrm{T}}\left(\widetilde{\mathbf{Y}}_{pi}\right)\Gamma\frac{\partial\mathbf{q}\left(\widetilde{\mathbf{X}}_{pi}\right)}{\partial\widetilde{\mathbf{X}}_{pi}},$$

$$\frac{\partial\epsilon_i}{\partial\widetilde{\mathbf{Y}}_{pi}} = 2\mathbf{p}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)\Sigma\frac{\partial\mathbf{p}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)}{\partial\widetilde{\mathbf{Y}}_{pi}} - d_i^2\mathbf{q}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}_{pi}\right)\Gamma\frac{\partial\mathbf{q}\left(\widetilde{\mathbf{Y}}_{pi}\right)}{\partial\widetilde{\mathbf{Y}}_{pi}},$$

where Λ and Γ are computed with the initial value of $(\mathbf{n},\omega)$, and the Jacobians of $\mathbf{p}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)$, $\mathbf{q}\left(\widetilde{\mathbf{X}}_{pi}\right)$, and $\mathbf{q}\left(\widetilde{\mathbf{Y}}_{pi}\right)$ are

$$\frac{\partial\mathbf{p}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)}{\partial\widetilde{\mathbf{X}}_{pi}} = \begin{pmatrix} -\left[\widetilde{\mathbf{Y}}_{pi}\right]_\times \\ \mathrm{I} \end{pmatrix}, \frac{\partial\mathbf{p}\left(\widetilde{\mathbf{X}}_{pi},\widetilde{\mathbf{Y}}_{pi}\right)}{\partial\widetilde{\mathbf{Y}}_{pi}} = \begin{pmatrix} \left[\widetilde{\mathbf{X}}_{pi}\right]_\times \\ -\mathrm{I} \end{pmatrix},$$

$$\frac{\partial\mathbf{q}\left(\widetilde{\mathbf{X}}_{pi}\right)}{\partial\widetilde{\mathbf{X}}_{pi}} = \sqrt{2}\begin{pmatrix} \sqrt{2}x_1 & 0 & 0 & x_2 & x_3 & 0 & 1 & 0 & 0 & 0 \\ 0 & \sqrt{2}x_2 & 0 & x_1 & 0 & x_3 & 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{2}x_3 & 0 & x_1 & x_2 & 0 & 0 & 1 & 0 \end{pmatrix}^{\mathrm{T}},$$

$$\frac{\partial\mathbf{q}\left(\widetilde{\mathbf{Y}}_{pi}\right)}{\partial\widetilde{\mathbf{Y}}_{pi}} = \sqrt{2}\begin{pmatrix} \sqrt{2}y_1 & 0 & 0 & y_2 & y_3 & 0 & 1 & 0 & 0 & 0 \\ 0 & \sqrt{2}y_2 & 0 & y_1 & 0 & y_3 & 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{2}y_3 & 0 & y_1 & y_2 & 0 & 0 & 1 & 0 \end{pmatrix}^{\mathrm{T}}.$$

Then, the weight on ϵ_i is computed as

$$w_i = \frac{1}{\mathrm{std}(\epsilon_i)} = \frac{1}{\sqrt{\frac{\partial\epsilon_i}{\partial\widetilde{\mathbf{X}}_{pi}}\mathrm{V}_{\widetilde{\mathbf{X}}_{pi}}\left(\frac{\partial\epsilon_i}{\partial\widetilde{\mathrm{X}}_{pi}}\right)^{\mathrm{T}} + \frac{\partial\epsilon_i}{\partial\widetilde{\mathbf{Y}}_{pi}}\mathrm{V}_{\widetilde{\mathbf{Y}}_{pi}}\left(\frac{\partial\epsilon_i}{\partial\widetilde{\mathrm{Y}}_{pi}}\right)^{\mathrm{T}}}} \tag{25}$$

and the weighted DLT-like algorithm is given by

$$\mathrm{WC}\begin{pmatrix}\lambda \\ \gamma\end{pmatrix} = \mathrm{W}\mathbf{d}, \tag{26}$$

where $\mathrm{W} = \mathrm{diag}(w_1, w_2, \ldots, w_m)$, and the computations of C and $\mathbf{d}$ are the same as in Eq. (18). The least squares solution to Eq. (26) is

$$\begin{pmatrix}\lambda \\ \gamma\end{pmatrix} = (\mathrm{WC})^{+}(\mathrm{W}\mathbf{d}).$$

After computing the weights on the linear equations in the DLT-like algorithm, a set of weighted linear equations can be constructed and solved. To reduce the computational complexity, the weights in our algorithm are not recomputed and thus the set of weighted linear equations is solved only once. The Euclidean reconstruction is then obtained in the same manner as the DLT-like algorithm, and affine adjustment is also performed as the last step. The above weighted DLT-like algorithm is named as WDLT-Like1.

4.2. Weighted DLT-like algorithm without an initial value

To avoid the requirement of an initial value of $(\mathbf{n},\omega)$ in the computation of weights in Eq. (25), some approximation must be made to replace the Jacobians of ϵ_i with respect to $\widetilde{\mathbf{X}}_{pi}$ and $\widetilde{\mathbf{Y}}_{pi}$. Without any prior information about $(\mathbf{n},\omega)$, it is intuitive to treat the six unknown variables in the Jacobians of ϵ_i equally. We assume that the typical values of $\partial\epsilon_i/\partial\widetilde{\mathbf{Y}}_{pi}$ and $\partial\epsilon_i/\partial\widetilde{\mathbf{Y}}_{pi}$ are both (1,1,1) multiplied by an unknown and unimportant factor. Define $\mathbf{e}$ as the 3-dimensional vector of ones, $\mathbf{e} = (1,1,1)^{\mathrm{T}}$. Then, an approximation of Eq. (25) is given by

$$w_i = \frac{1}{\sqrt{\mathrm{e}^{\mathrm{T}}\mathrm{V}_{\widetilde{\mathbf{X}}_{pi}}\mathrm{e} + \mathrm{e}^{\mathrm{T}}\mathrm{V}_{\widetilde{\mathbf{Y}}_{pi}}\mathrm{e}}}. \tag{27}$$

It is noted that the expression (Eq. (25)) is invariant under affine transformations on the projective reconstruction but Eq. (27) is not invariant even under orthogonal transformations. Therefore, the expression for the weights is not satisfactory and some further modifications are necessary. Since the covariance matrices $\mathrm{V}_{\widetilde{\mathbf{X}}_{pi}}$ and $\mathrm{V}_{\widetilde{\mathbf{Y}}_{pi}}$ are positively definite, their elements with the largest magnitude are on the diagonal. We approximate them with their diagonal matrices, and then $\mathbf{e}^{\mathrm{T}}\mathrm{V}_{\widetilde{\mathbf{X}}_{pi}}\mathrm{e}$ and $\mathbf{e}^{\mathrm{T}}\mathrm{V}_{\widetilde{\mathbf{Y}}_{pi}}\mathrm{e}$ are approximated with the traces of $\mathrm{V}_{\widetilde{\mathbf{X}}_{pi}}$ and $\mathrm{V}_{\widetilde{\mathbf{Y}}_{pi}}$. The new approximation of Eq. (25) is given by

$$w_i = \frac{1}{\sqrt{\mathrm{trace}\left(\mathrm{V}_{\widetilde{\mathbf{X}}_{pi}}\right) + \mathrm{trace}\left(\mathrm{V}_{\widetilde{\mathbf{Y}}_{pi}}\right)}}. \tag{28}$$

It is easy to check that this expression is invariant under orthogonal transformations. This way of computing weights depends only on the covariance matrices of the reconstructed end points, and it is easier to compute than Eq. (25). After computing the weights using Eq. (28), a set of weighted linear equations can be constructed and then solved in a similar manner to the implementation in WDLT-Like1. And this new weighted DLT-like algorithm is named as WDLT-Like2.

Remark. It is noted that compared with WDLT-Like1, the computation of weights in WDLT-Like2 has lower computational cost and the initial value of $(\mathbf{n},\omega)$ is not needed in WDLT-Like2. However, WDLT-Like2 has a little lower accuracy as shown in our experimental results in Section 6.

5. Nonlinear algorithms to reinforce length constraints

5.1. Basic optimization step

As discussed in Section 3.1, the minimal number of segments required to recover the Euclidean structure by solving polynomial equations is 9 in theory, but the linear algorithms (C1A, C2A, DLT-like, WDLT-Like1, and WDLT-Like2) require at least 54 segments. In

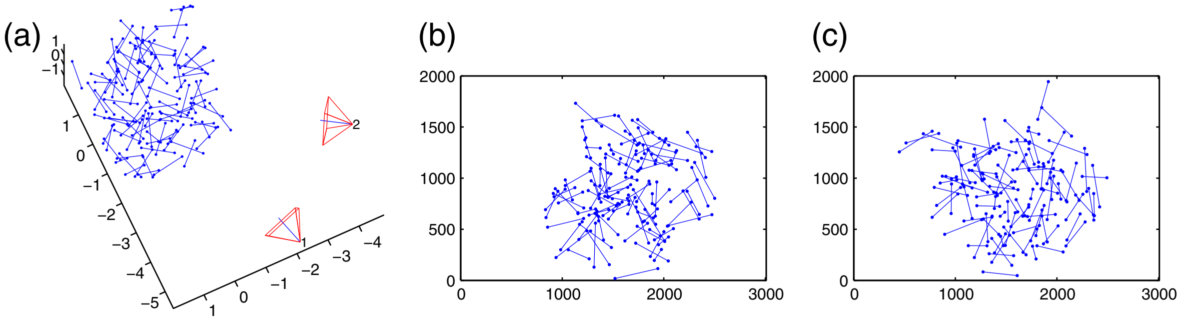

Fig. 1. Images of synthetic segments. (a) Segments and cameras. (b) Left image. (c) Right image.

these linear algorithms, severe over-parametrizations are introduced to linearly represent C_1, C_2, Λ, and Γ. To avoid over-parametrization reducing the accuracy of Euclidean upgrading, we introduce a nonlinear optimization step (abbreviated as OS) to reinforce the length constraints, where the cost function to be minimized is the total squared residuals of the lengths of the reconstructed segments,

$$f(\mathbf{n}, s\mathrm{K}_1) = \sum_{i=1}^{m} \epsilon_i^2 = \sum_{i=1}^{m} \left\{ \left\| \left(\frac{(s\mathrm{K}_1)^{-1}\widetilde{\mathbf{X}}_{pi}}{1+\mathbf{n}^{\mathrm{T}}\widetilde{\mathbf{X}}_{pi}} - \frac{(s\mathrm{K}_1)^{-1}\widetilde{\mathbf{Y}}_{pi}}{1+\mathbf{n}^{\mathrm{T}}\widetilde{\mathbf{Y}}_{pi}} \right) \right\| - d_i \right\}^2. \quad (29)$$

The cost function (Eq. (29)) can be minimized by the Levenberg–Marquardt algorithm [5], and the initial value of $(\mathbf{n}, s\mathrm{K}_1)$ needed to start the iterative minimization is provided by a linear algorithm.

5.2. Weighted optimization step

As discussed in Section 4, the reliabilities of the linear equations in the DLT-like algorithm are different. Similarly, the reconstruction residuals ϵ_i $(1 \le i \le m)$ in Eq. (29) also have different reliabilities. Therefore, to further improve the accuracy of the basic optimization

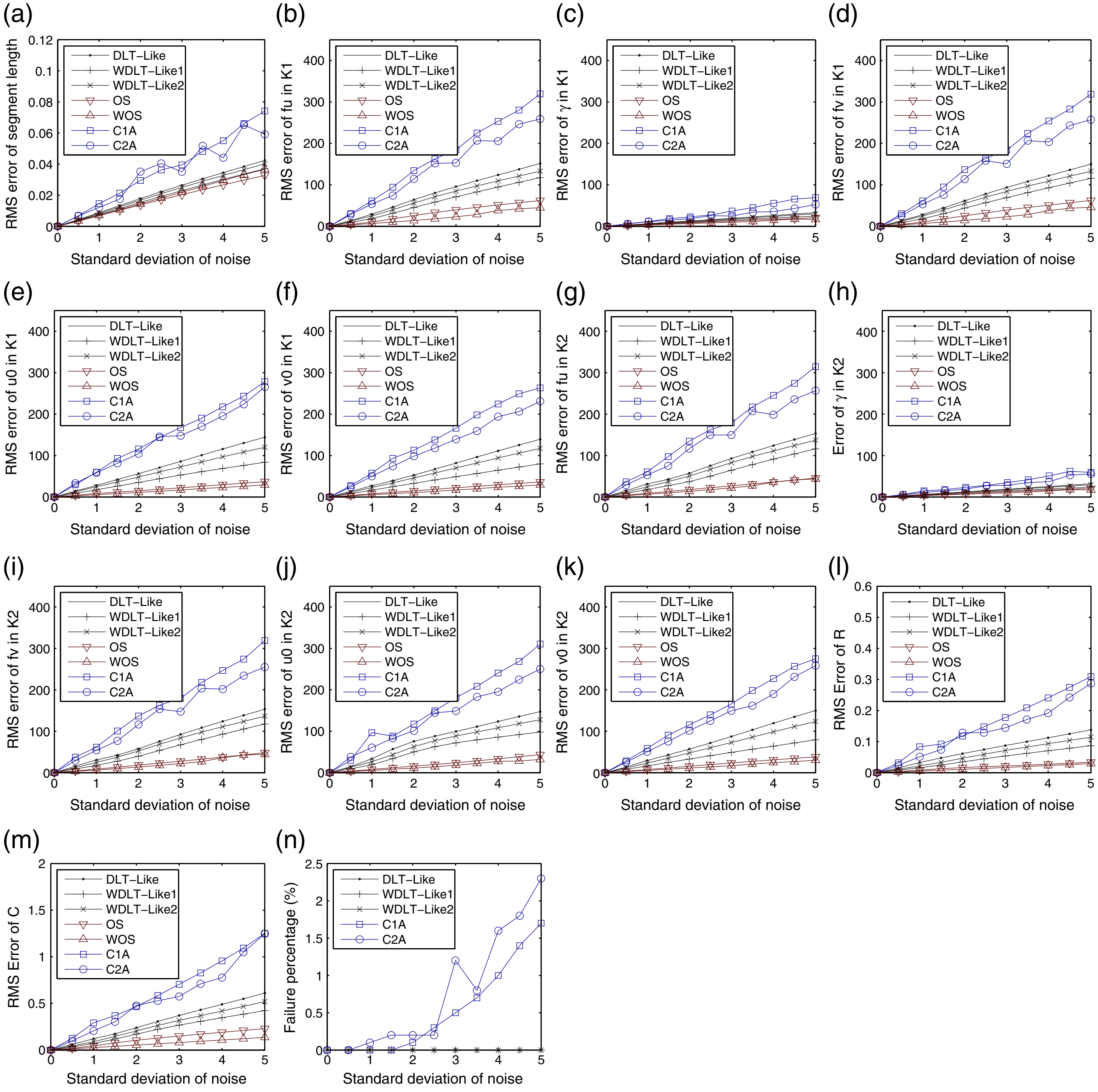

Fig. 2. Experimental results on the synthetic images with different noise levels. (a) Error comparison on the reconstructed segment lengths. (b)–(f) Error comparison on the intrinsic parameters in the first view. (g)–(k) Error comparison on the intrinsic parameters in the second view. (l) Error comparison on the rotation matrix. (m) Error comparison on the translation. (n) Failure percentage comparison.

step, we assign different weights to the residuals ϵ_i, resulting in a weighted optimization step (abbreviated as WOS) where the form of the cost function is

$$f(\mathbf{n}, s\mathrm{K}_1) = \sum_{i=1}^{m} (w_i \epsilon_i)^2 = \sum_{i=1}^{m} \left\{ w_i \left[\left\| \left(\frac{(s\mathrm{K}_1)^{-1} \widetilde{\mathbf{X}}_{pi}}{1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{X}}_{pi}} - \frac{(s\mathrm{K}_1)^{-1} \widetilde{\mathbf{Y}}_{pi}}{1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{Y}}_{pi}} \right) \right\| - d_i \right] \right\}^2. \tag{30}$$

Since it is necessary to give beforehand an initial value to start an iterative optimization procedure, the weights w_i $(1 \le i \le m)$ in Eq. (30) are computed with this initial value of $(\mathbf{n}, s\mathrm{K}_1)$ in the same manner as WDLT-Like1. Here, the weight on residual ϵ_i is computed as

$$w_i = \frac{1}{\mathrm{std}(\epsilon_i)} = \frac{1}{\sqrt{\frac{\partial \epsilon_i}{\partial \widetilde{\mathbf{X}}_{pi}} \mathrm{V}_{\widetilde{\mathbf{X}}_{pi}} \left(\frac{\partial \epsilon_i}{\partial \widetilde{\mathbf{X}}_{pi}} \right)^{\mathrm{T}} + \frac{\partial \epsilon_i}{\partial \widetilde{\mathbf{Y}}_{pi}} \mathrm{V}_{\widetilde{\mathbf{Y}}_{pi}} \left(\frac{\partial \epsilon_i}{\partial \widetilde{\mathbf{Y}}_{pi}} \right)^{\mathrm{T}}}},$$

where the Jacobians of ϵ_i with respect to $\widetilde{\mathbf{X}}_{pi}$ and $\widetilde{\mathbf{Y}}_{pi}$ are

$$\frac{\partial \epsilon_i}{\partial \widetilde{\mathbf{X}}_{pi}} = \frac{1}{d_i} \left(\frac{\widetilde{\mathbf{X}}_{pi}}{1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{X}}_{pi}} - \frac{\widetilde{\mathbf{Y}}_{pi}}{1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{Y}}_{pi}} \right)^{\mathrm{T}} \frac{\omega \left[\left(1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{X}}_{pi} \right) \mathrm{I} - \widetilde{\mathbf{X}}_{pi} \mathrm{n}^{\mathrm{T}} \right]}{\left(1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{X}}_{pi} \right)^2},$$

$$\frac{\partial \epsilon_i}{\partial \widetilde{\mathbf{Y}}_{pi}} = \frac{1}{d_i} \left(\frac{\widetilde{\mathbf{X}}_{pi}}{1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{X}}_{pi}} - \frac{\widetilde{\mathbf{Y}}_{pi}}{1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{Y}}_{pi}} \right)^{\mathrm{T}} \frac{\omega \left[- \left(1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{Y}}_{pi} \right) \mathrm{I} + \widetilde{\mathbf{Y}}_{pi} \mathbf{n}^{\mathrm{T}} \right]}{\left(1 + \mathbf{n}^{\mathrm{T}} \widetilde{\mathbf{Y}}_{pi} \right)^2}.$$

The weighted cost function in Eq. (30) can be minimized in the same way as in the basic optimization step. Then, the Euclidean structure is recovered with the obtained estimation of $(\mathbf{n}, s\mathrm{K}_1)$. Similar to WDLT-Like1, the weights in the weighted optimization step are not recomputed with the refined result of $(\mathbf{n}, s\mathrm{K}_1)$ to reduce computational complexity.

6. Experimental results

6.1. Synthetic data

In this experiment, a set of segments of unit length are randomly generated in a cube of width 4, where the center of the cube is at the origin of a predefined coordinate frame. The intrinsic parameter matrices of the two synthetic cameras are set to

$$\mathrm{K}_{1\mathrm{true}} = \mathrm{K}_{2\mathrm{true}} = \begin{pmatrix} 2000 & 0 & 1504 \\ 0 & 2000 & 1000 \\ 0 & 0 & 1 \end{pmatrix},$$

and the extrinsic parameters of the two cameras are also randomly chosen such that images of the segments are in the range of [0,3008] × [0,2000]. Fig. 1 shows an example of the left and right images of 100 segments. Gaussian noise with mean zero and standard deviation σ is then added to the image of each end point. The projective reconstruction of the scene is computed by the following steps: First, a projective reconstruction is obtained from the point correspondences between the two images using the extended fundamental numerical scheme (EFNS) [27] and the optimal triangulation [28]. Then, a transformation matrix is computed to change the projective reconstruction to a quasi-affine reconstruction [5]. At last, the end points are normalized such that their centroid is at the origin of the coordinate frame and their average distance from the origin is equal to $\sqrt{3}$ [22].

The Euclidean upgrading algorithms C1A, C2A, DLT-like, WDLT-Like1, WDLT-Like2, OS, and WOS are evaluated. Here, C1S is not evaluated since it was reported to have lower accuracy than C1A in [21]. In the implementation of C2A, the plane at infinity is extracted as the polar plane of the center of a sphere with respect to the sphere, following the implementation in [21]. The initial values of $(\mathbf{n}, \omega)$ required in WDLT-Like1, OS, and WOS are provided by DLT-like. Let $\left\{ \widetilde{\mathbf{X}}_{ei}, \widetilde{\mathbf{Y}}_{ei} \right\}$ $(i = 1, \ldots, m)$ denote the Euclidean reconstruction of the segments, and let R_j and $\mathbf{C}_j$ $(j = 1, 2)$ denote the rotation matrices and the translations of the cameras. Then, let $\mathrm{R} = \mathrm{R}_2 \mathrm{R}_1^{\mathrm{T}}$ and $\mathbf{C} = \mathrm{R}_1(\mathbf{C}_2 - \mathbf{C}_1)$ denote the relative rotation matrix and the relative translation of the second view. The following items of errors are used to evaluate the accuracy of this Euclidean reconstruction: (i) The average error of reconstructed segment lengths is defined as $\sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(\| \widetilde{\mathbf{X}}_{ei} - \widetilde{\mathbf{Y}}_{ei} \| - d_i \right)^2}$; (ii) The ten error items of the ten intrinsic parameters $(f_u^j, \gamma^j, f_v^j, u_0^j, v_0^j)$ $(j = 1, 2)$ are defined as the differences between them and their ground truths; (iii) $\| \mathrm{R} - \mathrm{R}_{\mathrm{true}} \|_{\mathrm{Frob}}$ denotes the Frobenius norm of the difference between R and its ground truth $\mathrm{R}_{\mathrm{true}}$; and (iv) $\| \mathbf{C} - \mathbf{C}_{\mathrm{true}} \|$ denotes the norm of the difference between $\mathbf{C}$ and its ground truth $\mathbf{C}_{\mathrm{true}}$. The above items of errors of the seven algorithms are evaluated concerning the effects of the noise level σ, the number m of segments, and the length d of the segments. Under each configuration of σ, m, and d, 1000 independent trials of experiments are performed, and then we compute the root-mean-square (RMS) results of these error items as well as the failure percentages of the linear algorithms, where failure means an algorithm does not provide a positive definite estimation of the IAC.

6.1.1. Noise level

Here, the length d of the segments is fixed at 1, the number m of the segments is fixed at 100, and the noise level σ varies from 0 to 5 pixels with a step of 0.5 pixel. The experimental results of the seven algorithms are shown in Fig. 2.

As seen in Fig. 2(a)–(m), the errors of the evaluated algorithms approximately grow linearly with the increase of noise level. The proposed linear algorithm DLT-like consistently outperforms C1A and C2A. The two weighted algorithms WDLT-Like1 and WDLT-Like2 achieve higher accuracies than DLT-like, and WDLT-Like1 performs better than WDLT-Like2. In addition, OS provides more accurate results on both the reconstructed segment lengths and the camera parameters than all the linear algorithms. Moreover, the weighted nonlinear algorithm WOS estimates the intrinsic and extrinsic parameters more accurately than OS, which demonstrates the effectiveness of weighting in WOS. However, its performance on the reconstructed segment lengths is slightly worse than OS, which is because the used evaluation criterion on the reconstructed segment lengths in the experiments is the same as the designed cost function of OS. As seen from Fig. 2(n), C1A and C2A are prone to failure, whereas the other linear algorithms have zero failure rate in the experiments.

6.1.2. Number of segments

Here, the length d of the segments is fixed at 1, the noise level σ is fixed at 3 pixels, and the number m of the segments varies from 65 to 120 with a step of 5. The experimental results of the seven algorithms are shown in Fig. 3.

As seen in Fig. 3(a)–(m), the errors of the linear algorithms generally decrease with the increase of segment number m, and the errors of OS and WOS do not change significantly. The accuracy of DLT-like is consistently higher than those of C1A and C2A. It is also observed that the two weighted linear algorithms WDLT-Like1 and WDLT-Like2 successfully improve the accuracy of DLT-like, and the improvements on accuracy by the weighting in WDLT-Like1 and WDLT-Like2 become more obvious when more segments are used. Besides, the accuracy of WOS in terms of the intrinsic and extrinsic parameters is consistently higher than the accuracy of OS. Fig. 3(n) shows that C1A and C2A are prone to failure, and the proposed linear algorithms DLT-like, WDLT-Like1, and WDLT-Like2 have no case of failure in the experiments.

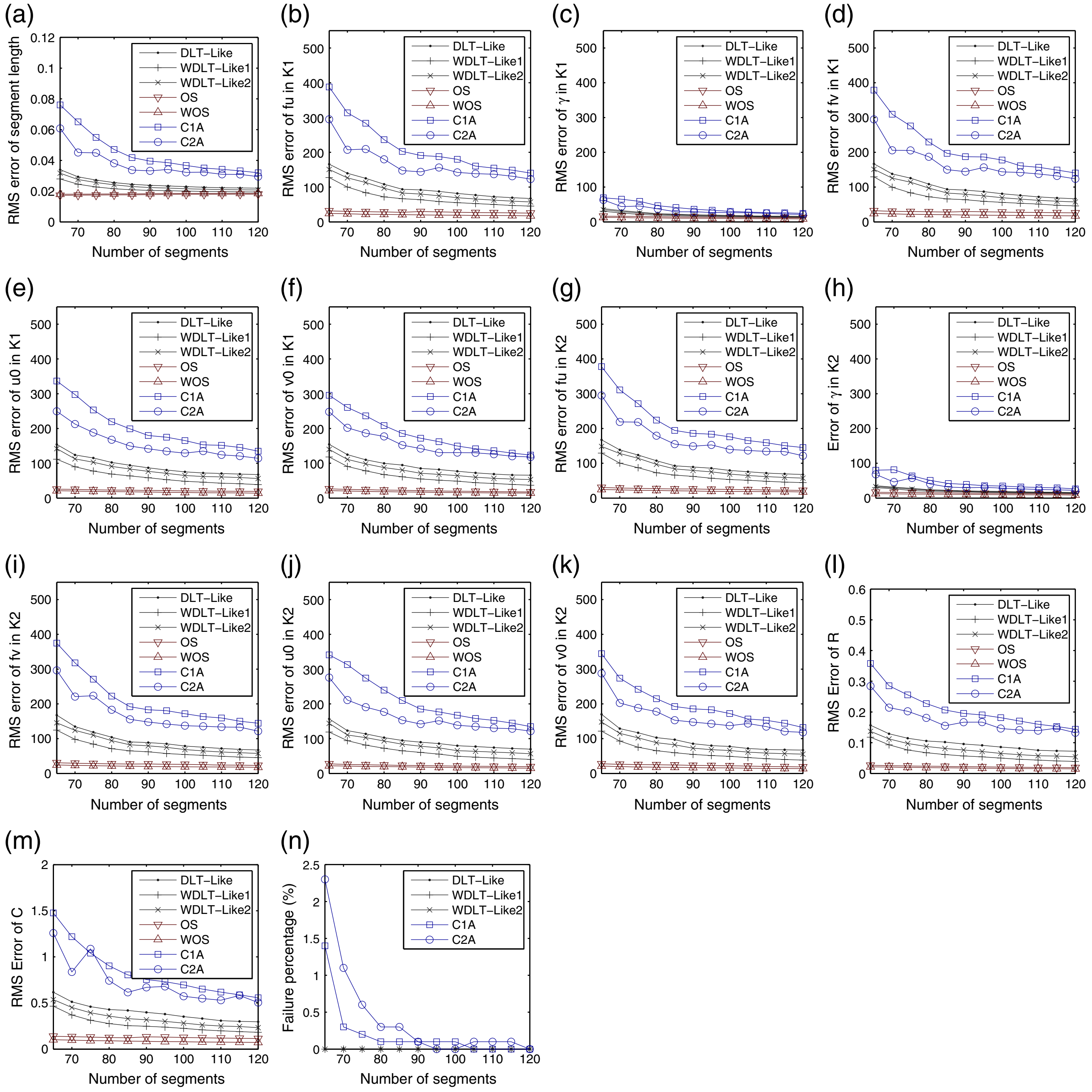

Fig. 3. Experimental results on the synthetic images with different numbers of segments. (a) Error comparison on the reconstructed segment lengths. (b)–(f) Error comparison on the intrinsic parameters in the first view. (g)–(k) Error comparison on the intrinsic parameters in the second view. (l) Error comparison on the rotation matrix. (m) Error comparison on the translation. (n) Failure percentage comparison.

6.1.3. Length of segments

In this subsection, the effects of the segment lengths on the accuracies of all the referred algorithms are tested. The experimental configuration is constructed as follows: The noise level σ is fixed at 3 pixels, the number m of the segments is fixed at 100, and the length d of the segments varies from 0.4 to 2 with a step of 0.2 while keeping the centers and orientations of the segments fixed. The experimental results of all the referred algorithms are shown in Fig. 4.

As seen in Fig. 4(a)–(m), for all the referred algorithms, the errors of the reconstructed segment lengths do not change significantly with the increase of segment length d, whereas the errors of camera parameters decrease with the increase of d. Moreover, it has to be pointed out that in our experimental setup, the coordinates of the images of some segments exceed the range of [0,3008] × [0,2000] when d is larger than 1, and the images of these segments are still used for Euclidean upgrading. In addition, the proposed DLT-like algorithm consistently outperforms C1A and C2A. The two weighted linear algorithms WDLT-

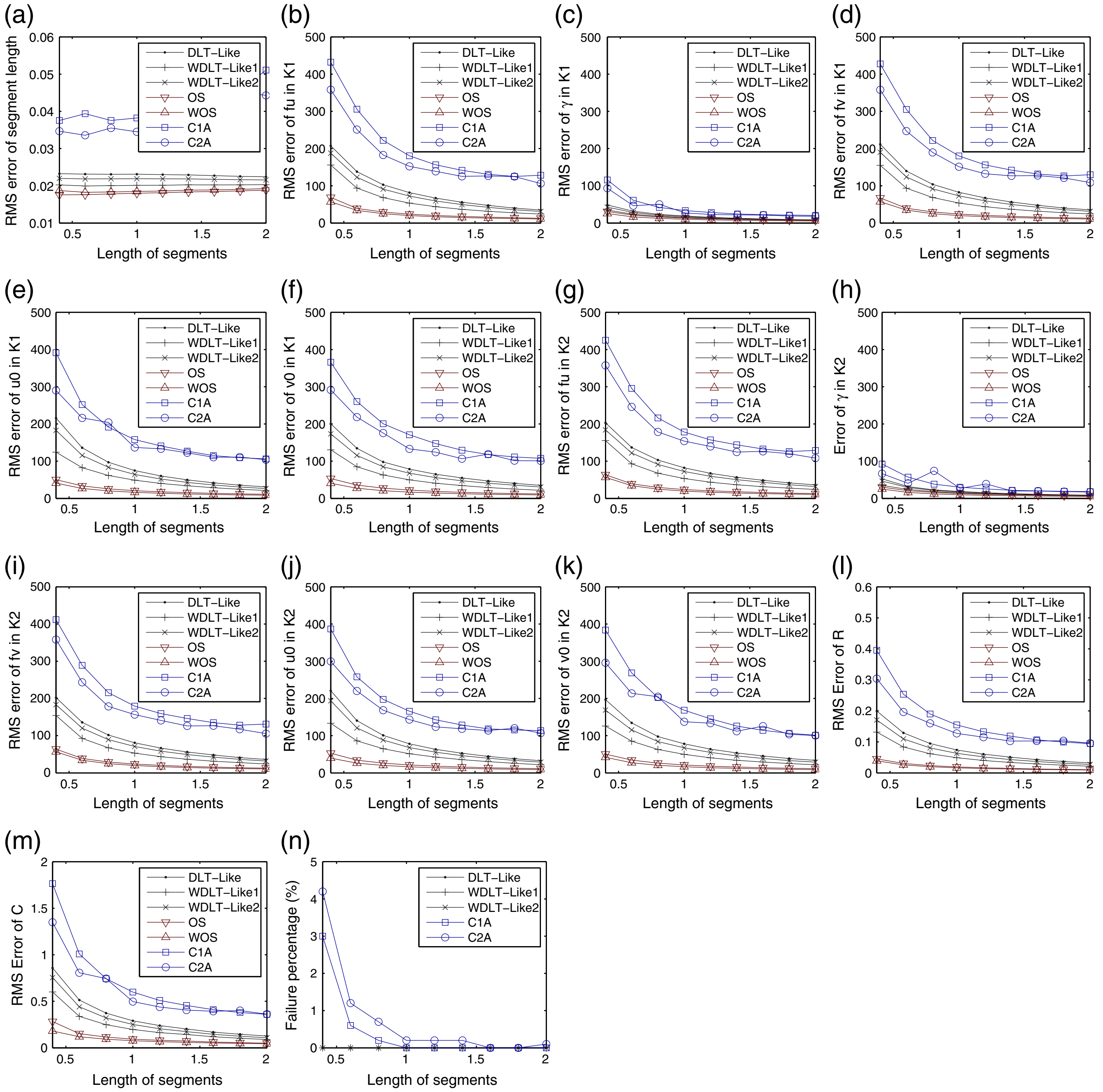

Fig. 4. Experimental results on the synthetic images with different lengths of the segments. (a) Error comparison on the reconstructed segment lengths. (b)–(f) Error comparison on the intrinsic parameters in the first view. (g)–(k) Error comparison on the intrinsic parameters in the second view. (l) Error comparison on the rotation matrix. (m) Error comparison on the translation. (n) Failure percentage comparison.

Like1 and WDLT-Like2 improve the accuracy of DLT-like further. The estimations on the intrinsic and extrinsic parameters by WOS are more accurate than those by OS. Fig. 4(n) shows that the failure rates of C1A and C2A decrease with the increase of d, whereas the proposed DLT-like, WDLT-Like1, and WDLT-Like2 have no case of failure in the experiments.

6.2. Real image data

In this experiment, a stick with two balls is used as the calibration object, where the distance between the centers of the two balls is 0.505 m. And the locations of two Nikon D40 cameras with resolution of 3008 × 2000 pixels are fixed during the experiment. The two cameras are calibrated with the 2D calibration algorithm [12], where the obtained ground truth of the two intrinsic matrices are

$$K_{1\text{true}} = \begin{pmatrix} 5829.4 & 0.1 & 1554.2 \\ 0 & 5832.8 & 1030.3 \\ 0 & 0 & 1 \end{pmatrix},$$

$$K_{2\text{true}} = \begin{pmatrix} 5038.2 & 0.3 & 1545.0 \\ 0 & 5042.6 & 1029.0 \\ 0 & 0 & 1 \end{pmatrix}.$$

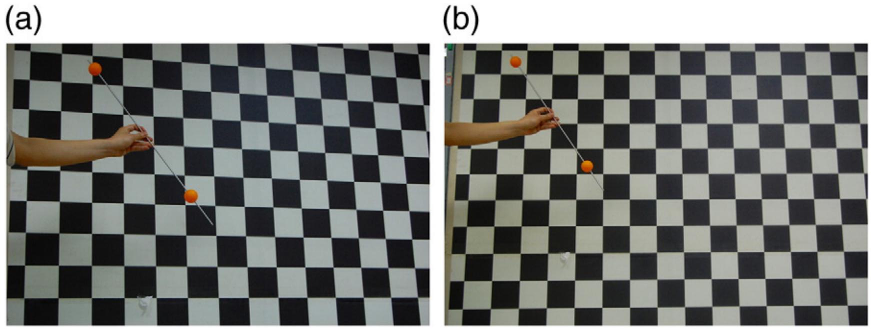

Fig. 5. Images of a segment under two views. (a) Left image. (b) Right image.

Define the world coordinate frame as the camera coordinate frame of the first view, and then the extrinsic parameters of the second view are

$$\mathrm{R}_{\mathrm{true}} = \begin{pmatrix} 0.97492 & 0.03328 & 0.22005 \\ -0.02470 & 0.99883 & -0.04163 \\ -0.22117 & 0.03515 & 0.97460 \end{pmatrix}, \mathbf{C}_{\mathrm{true}} = \begin{pmatrix} 0.6803 \\ 0.0213 \\ 0.0886 \end{pmatrix}.$$

A total of 146 poses of the stick are randomly generated and the corresponding images of the stick are captured by the two cameras. Fig. 5 shows two example images of the stick under a pose. The centers of the two balls in each image are detected as the images of the end points of the segment. Then, we obtain the detected projections of the 146 segments, as shown in Fig. 6. With the ground truth of the intrinsic and extrinsic parameters of the two cameras, the 146 segments are then reconstructed and shown in Fig. 7.

The experiments on Euclidean upgrading from segment lengths are performed as follows. For each m in {65, 70, ..., 120}, a subset of m segments is randomly selected from the total set of segments. The procedure to obtain the projective reconstruction of the m segments is the same as in the experiments on the synthetic data. Then, we test the accuracies of the algorithms: C1A, C2A, DLT-like, WDLT-Like1, WDLT-Like2, OS, and WOS. Here, the used criteria to evaluate the seven algorithms are the same as those in the experiments on the synthetic data. Fig. 8 shows the experimental results of the seven evaluated algorithms.

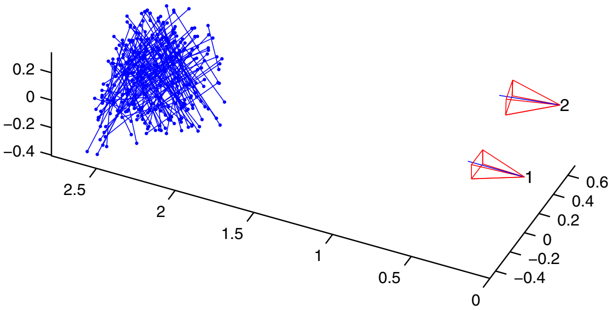

Fig. 7. The poses of the cameras and the reconstruction of the 146 segments.

As seen from Fig. 8, the experimental results on the real image data are generally similar to those on the synthetic data, but there are also two slight differences. In the real image data experiments, no failure occurs for all the seven evaluated algorithms possibly because of the well-controlled experimental condition. Besides, the advantages of the nonlinear algorithms OS and WOS over the linear algorithms are more obvious with the real image data.

In addition, the bundle adjustment method that minimizes the reprojection error of a set of segments with exact lengths is tested. Here, three different initializations for the bundle adjustment method are provided by DLT-like, OS, and WOS respectively, and the resulting Euclidean upgrading methods are denoted as DLT-like + BA, OS + BA, and WOS + BA. The corresponding experimental results including the results by OS and WOS are shown in Fig. 9. It is observed from Fig. 9(a) that OS has slightly lower error in terms of the reconstructed segment lengths than WOS, whereas DLT-like + BA, OS + BA, and WOS + BA have zero errors on the reconstructed segment lengths because the lengths of the segments are parametrized in the bundle adjustment method. As can be seen in Fig. 9(b)–(m): (i) WOS has higher accuracy than OS in estimating the intrinsic and extrinsic parameters. (ii) The accuracy of DLT-like + BA is lower than those of OS and WOS especially when the number of segments is small. (iii) OS + BA has slightly higher

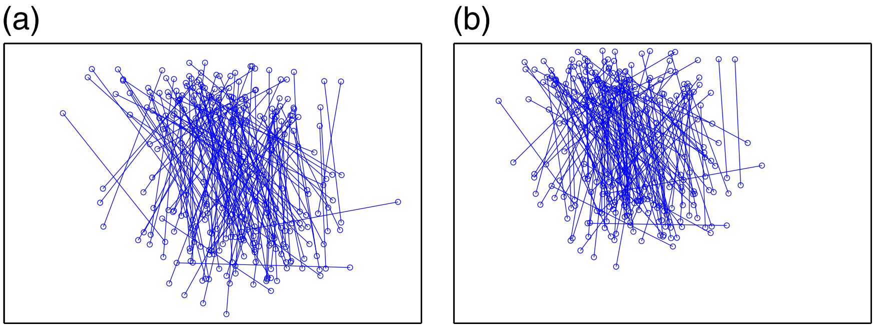

Fig. 6. Detected projections of the 146 segments. (a) Left image. (b) Right image.

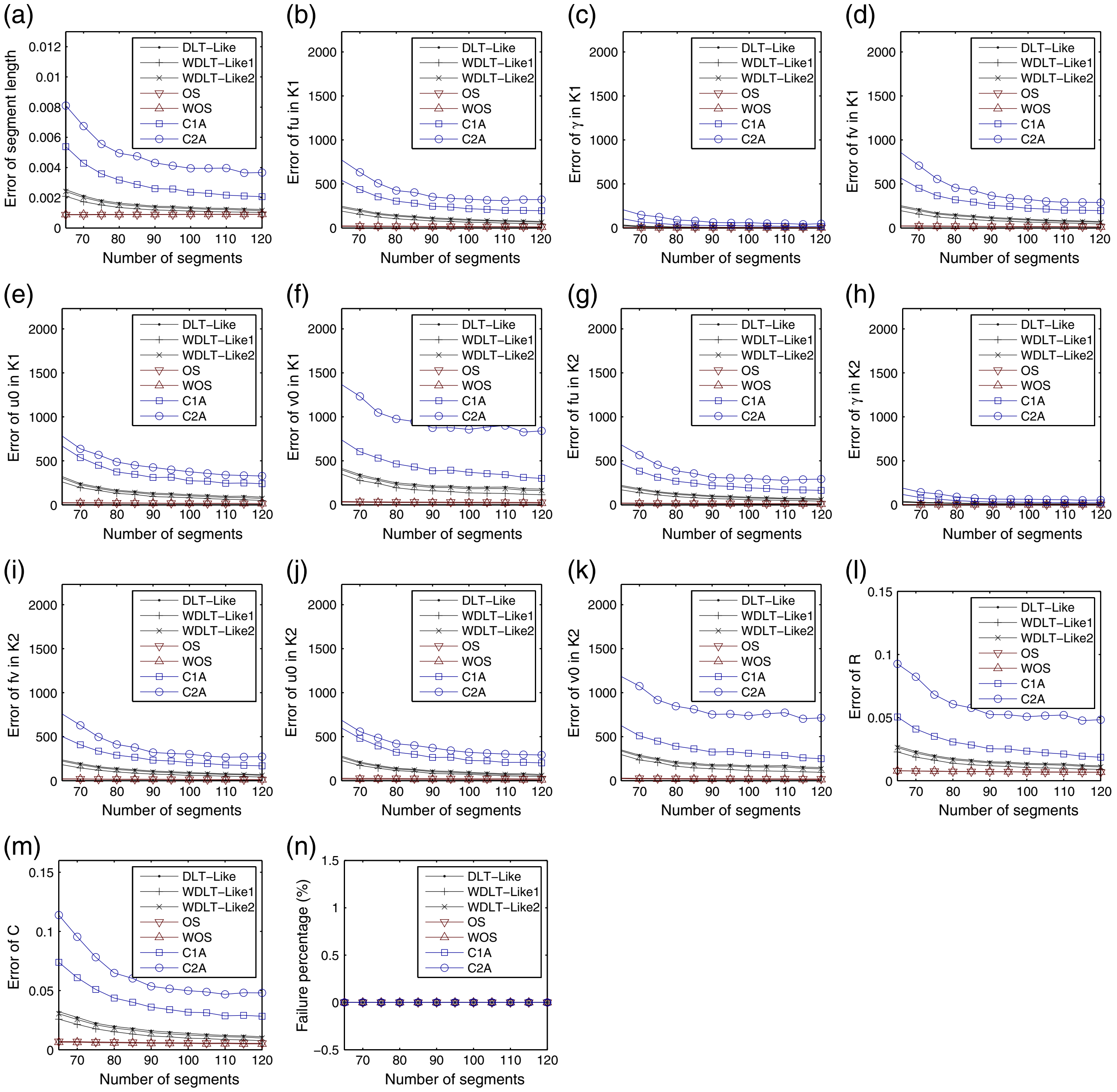

Fig. 8. Experimental results on the real images by all the referenced algorithms. (a) Error comparison on the reconstructed segment lengths. (b)–(f) Error comparison on the intrinsic parameters in the first view. (g)–(k) Error comparison on the intrinsic parameters in the second view. (l) Error comparison on the rotation matrix. (m) Error comparison on the translation. (n) Failure percentage comparison.

accuracy than OS, but its accuracy is still lower than that of WOS. (iv) The accuracy of WOS + BA is very close to that of WOS.

7. Conclusion

With a very simple mathematical treatment, we have presented a DLT-like algorithm for Euclidean upgrading from segment lengths. We have also shown that the intermediate results of the DLT-like algorithm are equivalent to the QoS, but the DLT-like algorithm has higher accuracy than the existing linear algorithms derived from the QoS because of a more accurate way to extract the plane at infinity. Two weighted DLT-like algorithms have also been proposed, which successfully achieve higher accuracy. Furthermore, we have developed a weighted nonlinear algorithm based on a basic nonlinear algorithm to refine the results of the linear algorithms. The experiments on both the synthetic data and the real image data have demonstrated the effectiveness of the proposed algorithms.

It is noted that although the proposed nonlinear algorithms require at least 9 segments, all the referred linear algorithms to provide initializations in this paper require at least 54 segments. In the future, we will investigate how to use fewer segments to perform Euclidean upgrading.

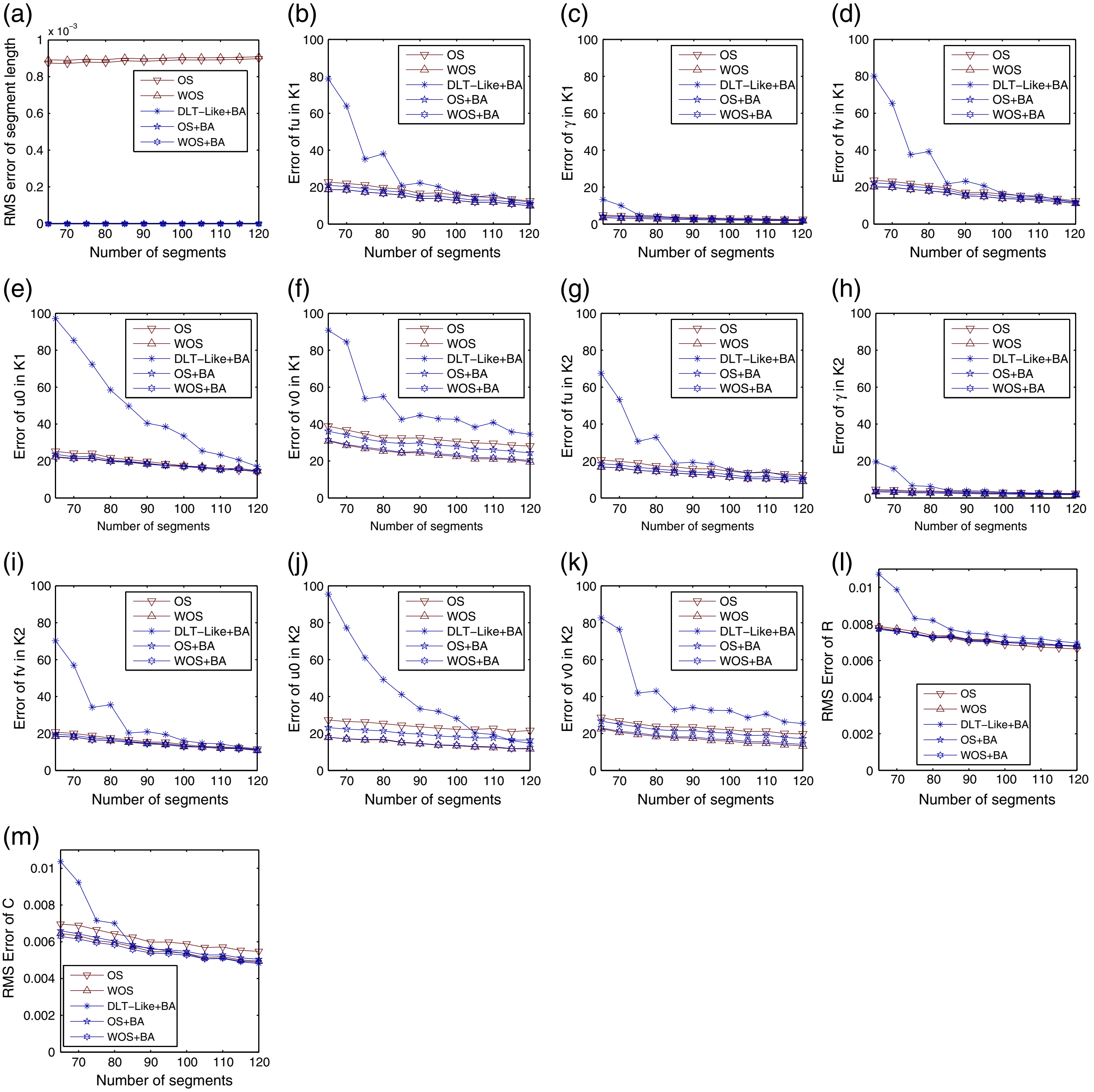

Fig. 9. Experimental results on the real images by the nonlinear algorithms. (a) Error comparison on the reconstructed segment lengths. (b)–(f) Error comparison on the intrinsic parameters in the first view. (g)–(k) Error comparison on the intrinsic parameters in the second view. (l) Error comparison on the rotation matrix. (m) Error comparison on the translation.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61333015, No. 91120012, and No. 61375042.

References

[1] S.J. Maybank, O.D. Faugeras, A theory of self-calibration of a moving camera, Int. J. Comput. Vis. 8 (1992) 123–151.

[2] M. Pollefeys, L. Van Gool, Stratified self-calibration with the modulus constraint, IEEE. Trans. Pattern Anal. Mach. Intell. 21 (8) (1999) 707–724.

[3] M. Pollefeys, R. Koch, L. Van Gool, Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters, Proc. Int. Conf. Comput. Vis., 1998, pp. 90–95.

[4] B. Triggs, Autocalibration and the absolute quadric, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 1997, pp. 609–614.

[5] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second edn Cambridge University Press, 2003.

[6] J.I. Ronda, A. Valdés, G. Gallego, Line geometry and camera autocalibration, J. Math. Imaging Vis. 32 (2) (2008) 193–214.

[7] M. Armstrong, A. Zisserman, R. Hartley, Self-calibration from image triplets, Proc. Eur. Conf. Comput. Vis., 1996, pp. 1–16.

[8] R. Hartley, Self-calibration from multiple views with a rotating camera, Proc. Eur. Conf. Comput. Vis., 1994, pp. 471–478.

[9] A. Fitzgibbon, G. Cross, A. Zisserman, Automatic 3D model construction for turn-table sequences, 3D Structure from Multiple Images of Large-Scale Environments, Lecture Notes in Computer Science, vol. 1506, Springer, Berlin Heidelberg, 1998, pp. 155–170.

[10] R. Tsai, An efficient and accurate camera calibration technique for 3D machine vision, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 1986, pp. 364–374.

[11] J. Heikkilä, Geometric camera calibration using circular control points, IEEE. Trans. Pattern Anal. Mach. Intell. 22 (2000) 1066–1077.

[12] Z. Zhang, A flexible new technique for camera calibration, IEEE. Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1330–1334.

[13] P. Sturm, S. Maybank, On plane-based camera calibration: a general algorithm, singularities, applications, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 1, 1999, pp. 432–437.

[14] Z. Zhang, Camera calibration with one-dimensional objects, IEEE. Trans. Pattern Anal. Mach. Intell. 26 (7) (2004) 892–899.

[15] K. Shi, Q. Dong, F. Wu, Weighted similarity-invariant linear algorithm for camera calibration with rotating 1-D objects, IEEE Trans. Image Process. 21 (8) (2012) 3806–3812.

[16] L. Wang, F. Wu, Z. Hu, Multi-camera calibration with one-dimensional object under general motions, Proc. Int. Conf. Comput. Vis., 2007, pp. 1–7.

[17] J.A. de França, M.R. Stemmer, M.B.d.M. França, J.C. Piai, A new robust algorithmic for multi-camera calibration with a 1D object under general motions without prior knowledge of any camera intrinsic parameter, Pattern Recognit. 45 (10) (2012) 3636–3647.

[18] L. Davis, E. Borovikov, R. Cutler, D. Harwood, T. Horprasert, Multi-perspective analysis of human action, Proc. Int. Workshop Cooperat. Distrib. Vis., 1999, pp. 1–35.

[19] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, ACM Trans. Graph. 23 (3) (2004) 600–608.

[20] D. Liebowitz, S. Carlsson, Uncalibrated motion capture exploiting articulated structure constraints, Int. J. Comput. Vis. 51 (2003) 171–187.

[21] J.I. Ronda, A. Valdés, Euclidean upgrading from segment lengths, Int. J. Comput. Vis. 90 (2010) 350–368.

[22] R.I. Hartley, In defense of the eight-point algorithm, IEEE. Trans. Pattern Anal. Mach. Intell. 19 (6) (1997) 580–593.

[23] S. Birchfield, An Introduction to Projective Geometry (for Computer Vision), http://robotics.stanford.edu/birch/projective/ 1998.

[24] K. Kanatani, Statistical optimization for geometric fitting: theoretical accuracy bound and high order error analysis, Int. J. Comput. Vis. 80 (2008) 167–188.

[25] J.C. Clarke, Modelling uncertainty: a primer, Tech. Rep. 2161/98, University of Oxford, Dept. Engineering Science, 1998.

[26] P.D. Sampson, fitting conic sections to “very scattered” data: an iterative refinement of the Bookstein algorithm, Comput. Graph. Image Process. 18 (1) (1982) 97–108.

[27] K. Kanatani, Y. Sugaya, Extended FNS for constrained parameter estimation, Proc. Meet. Image Recognit. Und., 2007, pp. 219–226.

[28] R.I. Hartley, P. Sturm, Triangulation, Comput. Vis. Image Underst. 68 (2) (1997) 146–157.