Multimedia Systems (2016) 22:181–195
DOI 10.1007/s00530-014-0439-0

REGULAR PAPER

Tracking 3-D motion from straight lines with trifocal tensors

Kai Ki Lee · Ying Kin Yu · Kin Hong Wong · Michael Ming Yuen Chang

Received: 21 June 2013 / Accepted: 24 November 2014 / Published online: 27 December 2014

Abstract We present a novel approach to track the position and orientation of a stereo camera using line features in the images. The method combines the strengths of trifocal tensors and Bayesian filtering. The trifocal tensor provides a geometric constraint to lock line features among every three frames. It eliminates the explicit reconstruction of the scene even if the 3-D scene structure is not known. Such a trifocal constraint thus makes the algorithm fast and robust. The twist motion model is applied to further improve its computation efficiency. Another major contribution is that our approach can obtain the 3-D camera motion using as little as 2 line correspondences instead of 13 in the traditional approaches. This makes the approach attractive for realistic applications. The performance of the proposed method has been evaluated using both synthetic and real data with encouraging results. Our algorithm is able to estimate 3-D camera motion in real scenarios accurately having little drifting from an image sequence longer than a 1,000 frames.

Communicated by Q. Tian.

K. K. Lee (✉) · M. M. Y. Chang
Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China
e-mail: kklee6@ie.cuhk.edu.hk

M. M. Y. Chang
e-mail: mchang@ie.cuhk.edu.hk

Y. K. Yu
Hong Kong, China
e-mail: ykyu.hk@gmail.com

K. H. Wong
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China
e-mail: khwong@cse.cuhk.edu.hk

Keywords Trifocal tensor · Kalman filtering · Pose tracking · Stereo vision · Structure · Motion · Robot vision · Twist

1 Introduction

This paper tackles the problem of camera motion tracking from a stereo image sequence. It aims at recovering the position and orientation of a camera using line features. It is considered as a challenging task in computer vision especially if the 3-D structure of the scene is unknown. The recovered motion is useful for a wide range of applications such as augmented reality, robot navigation and human–computer interaction. The novelty of this work is the integration of trifocal tensor with a Bayesian tracking algorithm for high-speed 3-D motion estimation. The use of these techniques results in enhanced robustness and accuracy.

Point features can easily be found and detected in common scenes. Previous research in the recovery of 3-D camera motion using points can be found in [1, 6, 7, 13, 14, 16, 22–26, 28, 30–32].In this paper, we consider straight lines instead of interest points in the images. Actually, lines are as widely available as point features in our living environments. They can normally be determined more accurately than points due to multi-pixel support. Feature extraction error and accumulated drift become smaller. With the use of infinite lines in replacement of line segments, the problem of occlusions can be alleviated when the view of a line is partially blocked.

Trifocal tensor encapsulates the projective geometric relations among three views. This mathematical relation is independent of the 3-D scene structure. It is analogous to the epipolar geometry, which is established between two

image views. It is known that corresponding lines in two images do not provide enough information to find a unique camera pose without the knowledge of a known model [29]. At least three views are necessary to fulfill the requirement. The trifocal tensor provides a stronger constraint than the fundamental matrix for features in the images as it involves additional information from the third view. Previously, line features and trifocal tensor were used to compute camera relations based on a traditional RANSAC framework [9] and a robust second-order minimization method [28]. Conventional three-view approaches making use of trifocal tensor require at least 13 line correspondences. Such a requirement discourages the application of trifocal tensor to methods that use a minimum of three lines [4, 8, 15].

Beside trifocal tensor, some researchers applied numerical approaches [2, 7] to estimate the object pose given a known CAD model. These methods need a good initial guess and are time-consuming. There is a complicated optimization algorithm in literature [5]. It determines the object motion from three corresponding lines. Its high computation cost prohibits it from being used in real-time applications such as virtual and projected reality systems. Although there are high-speed 3-D motion estimation methods [8, 15] using a minimum of three lines, they impose a number of limitations on the scene. A special geometric configuration among the lines is required. The approach in [8] computes 3-D pose from a "primitive configuration", in which three lines with two of them are parallel and their directions are orthogonal to the third line. It is not guaranteed that such a condition occurs in an indoor or urban environment, hence limiting their applicability.

The contributions of this paper are: (1) trifocal tensors have been employed in our recursive motion tracking approach using straight lines. The nature is quite different from those approaches that utilize point features with trifocal tensors [23, 25]. A line transfer function instead of point transfer formula is applied to the proposed method. (2) Thanks to the use of trifocal tensors, the algorithm is model-free. Even without the 3-D model of the scene, the reconstruction of the 3-D scene structure is not necessary. This increases the algorithm speed and at the same time reduces the space complexity as there is no need to keep track of the 3-D structure. (3) The minimum number of lines required by the proposed method is two. It is much smaller than most of the existing approaches that need 13 or more based on the trifocal tensor. This is due to the simultaneous use of two trifocal tensors, forming a quadrifocal constraint across four views. (4) With the help of an extended Kalman filter, our method operates at a high speed. This differentiates itself from other non-recursive or numerical approaches that consume much more time in computing the solutions.

Experimental results show that our approach is more accurate than a previous method that uses point features with trifocal tensors [25], and a state-of-the-art approach [8] that makes use of three lines. It is found that our algorithm is able to estimate 3-D camera motion in real scenarios accurately with little drifting from an image sequence surpassed a 1,000 frames.

The rest of this paper is organized as follows. In Sect. 2, the problem of camera motion tracking is defined and the system geometry is introduced. In Sect. 3, an overview of the proposed algorithm is given. The procedures for the line correspondence matching are outlined. In Sect. 4, the details of applying trifocal tensor to the extended Kalman filter are described. In Sect. 5, an empirical comparison among our approach and other state-of-the-art methods in [8, 19, 25] were made using synthetic data. In addition, the proposed method was tested with real image sequences taken by a robot and a hand-held stereo rig.

2 Problem modeling

A graphical illustration of the geometric system is shown in Fig. 1. I_t, I'_t are the images taken by the left and right camera at time t, respectively. I_1, I'_1 are regarded as the reference image pair. Lines are extracted from images. $l_{m,t}$ is the mth line extracted from image I_t while $l'_{m,t}$ is from I'_t. Lines $l_{m,t}$ and $l'_{m,t}$ are the projection of the 3-D line L_m on the left and right view at time t, respectively. The geometric relationships among a 3-D point $P_m^W = [x_m^W, y_m^W, z_m^W, 1]^T$ on line L_m in the 3-D structure and its projection $\tilde{p}_{m,t}$ on the left and $\tilde{p}'_{m,t}$ on the right view can be obtained as:

$$\tilde{p}_{m,t} = [\tilde{u}_{m,t}, \tilde{v}_{m,t}, \tilde{w}_{m,t}]^T = K[x_m^W, y_m^W, z_m^W, 1]^T \quad (1)$$

$$\tilde{p}'_{m,t} = [\tilde{u}'_{m,t}, \tilde{v}'_{m,t}, \tilde{w}'_{m,t}]^T = KEM_t[x_m^W, y_m^W, z_m^W, 1]^T \quad (2)$$

where K encodes the intrinsic parameters of a camera such as the focal length f and the principal point $[s_u\ s_v]$. It is a 3×3 upper triangular matrix and is assumed fixed during tracking. E is a 3×4 matrix represents the rigid transformation between two cameras in the stereo system. Both K and E are found by the camera calibration toolbox in [12]. The matrix M_t, or equivalently the twist vector ζ_t, encapsulates the pose information that transforms the 3-D structure from the world frame to the reference (left) camera at time instance t.

The twist motion model [27] is used to provide an elegant linear representation of the 3-D motion in our pose tracking algorithm. A twist has two representations: (1) a 6-dimensional vector denoted by ζ_t or (2) a 4×4 matrix $\tilde{\zeta}_t$ with the upper 3×3 component as a skew-symmetric matrix.

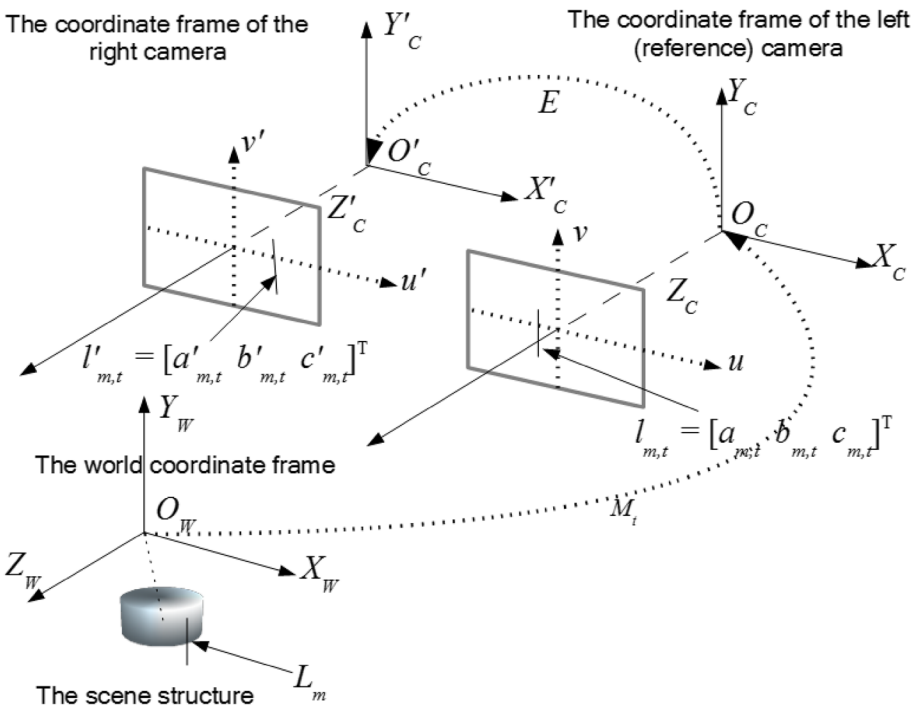

Fig. 1 The system geometry

$$\xi_t = \begin{bmatrix} x_t & y_t & z_t & \alpha_t & \beta_t & \gamma_t \end{bmatrix}^T \tag{3}$$

$$\tilde{\xi}_t = \begin{bmatrix} 0 & -\gamma_t & \beta_t & x_t \\ \gamma_t & 0 & -\alpha_t & y_t \\ -\beta_t & \alpha_t & 0 & z_t \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

x_t, y_t and z_t are the translation along the x, y and z axis and α, β, γ are the rotations about the x, y and z axis, respectively. The twist motion can be converted into the rigid transformation matrix M_t with the exponential map as follows:

$$M_t = e^{\tilde{\xi}_t} = I + \tilde{\xi}_t + \frac{(\tilde{\xi}_t)^2}{2!} + \frac{(\tilde{\xi}_t)^3}{3!} + \cdots \tag{5}$$

The normalized form $\tilde{l}_{m,t}$ and $\tilde{l}'_{m,t}$ of line $l_{m,t}$ on the left view and $l'_{m,t}$ on the right view are, respectively, given by

$$\tilde{l}_{m,t} = \begin{bmatrix} \tilde{a}_{m,t} \\ \tilde{b}_{m,t} \\ \tilde{c}_{m,t} \end{bmatrix} = \begin{bmatrix} a_{m,t}/f \\ b_{m,t}/f \\ (c_{m,t} + s_u a_{m,t} + s_v b_{m,t})/f^2 \end{bmatrix} \tag{6}$$

$$\tilde{l}'_{m,t} = \begin{bmatrix} \tilde{a}'_{m,t} \\ \tilde{b}'_{m,t} \\ \tilde{c}'_{m,t} \end{bmatrix} = \begin{bmatrix} a'_{m,t}/f \\ b'_{m,t}/f \\ (c'_{m,t} + s_u a'_{m,t} + s_v b'_{m,t})/f^2 \end{bmatrix} \tag{7}$$

$\theta_{m,t}$ and $\lambda_{m,t}$ are the slope and polar radius of line $l_{m,t}$ while $\theta'_{m,t}$ and $\lambda'_{m,t}$ are the slope and polar radius of line $l'_{m,t}$, respectively.

$$\begin{cases} \theta_{m,t} = \tan^{-1}(\tilde{a}_{m,t}/ - \tilde{b}_{m,t}) \\ \theta'_{m,t} = \tan^{-1}(\tilde{a}'_{m,t}/ - \tilde{b}'_{m,t}) \end{cases} \tag{8}$$

$$\begin{cases} \lambda_{m,t} = \frac{\tilde{b}_{m,t}\tilde{c}_{m,t}}{\tilde{b}_{m,t}\sin(\theta_{m,t}) + \tilde{a}_{m,t}\cos(\theta_{m,t})} \\ \lambda'_{m,t} = \frac{\tilde{b}'_{m,t}\tilde{c}'_{m,t}}{\tilde{b}'_{m,t}\sin(\theta'_{m,t}) + \tilde{a}'_{m,t}\cos(\theta'_{m,t})} \end{cases} \tag{9}$$

The goal of the proposed method is to find the motion of the camera system, i.e., ξ_t and M_t, at each time-step t given the lines $l_{m,t}$, and $l'_{m,t}$ in image measurements.

3 Outline of the algorithm

An outline of the proposed method is shown in Fig. 2. In the initialization step, line segments in the reference image pair are detected and matched. Matched line correspondences are tracked from one frame to the next. The details will be explained in Sect. 3.1. In each time-step, the quadrifocal constraint, which is in the form of trifocal tensors, is used to lock the line features. Its configuration will be discussed in Sect. 3.2. The tracked lines are passed to the extended Kalman filter (EKF) with a validation gate for recursive computation of the 3-D camera motion. The formulation of the EKF will be presented in Sect. 4.

3.1 Line extraction, matching and tracking

Line features are first detected in the reference image pair by a conventional edge detection algorithm such as the Canny algorithm [3]. Then, line correspondences between two stereo views of the reference image pair are

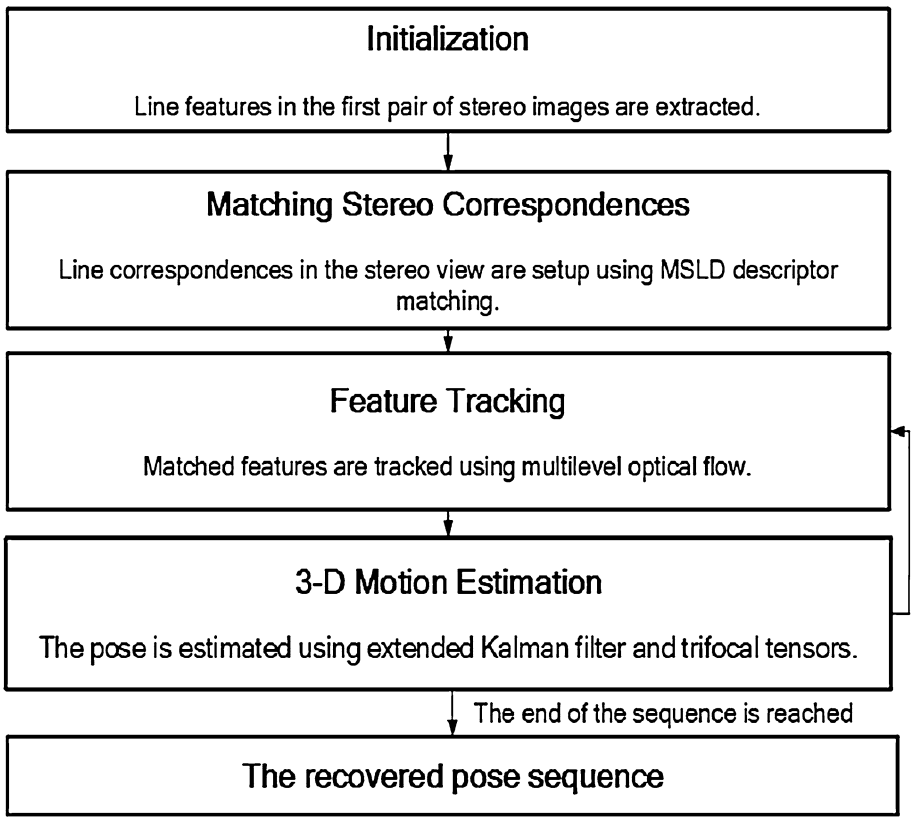

Fig. 2 An outline of the proposed method

established. Prior to line matching, each line segment is oriented according to the sign of intensity gradient such that its brighter side is always on one side. We apply the method by Schmid and Zisserman [17] that uses mean-standard deviation line descriptors (MSLD) to set up stereo correspondences [20].

Their method employs the epipolar beam constraint to find tentative matches. Let us consider the line segment s on the left view with homogenous end points denoted by x and y Similarly, line segment s' on the right image has end points x' and y'. s and s' satisfy the epipolar beam constraint if and only if the following conditions [21] hold:

$$\det(e,x,y)\det(e',x',y') < 0$$
$$\det(e,x,y)(Fy)^T x' < 0$$
$$\det(e,x,y)(Fy)^T y' < 0$$

where F, e, and e' are the fundamental matrix, epipole of the left view and right view, respectively. After the matching procedure, we calculate the general form $l_{m,t}$ and $l'_{m,t}$ of the mth line segment on the left and right image plane. A line pair is regarded as a corresponding match if it satisfies the epipolar beam constraint above and its photometric cross-correlation score c [18] is higher than 0.6.

Matched correspondences in the stereo images of the reference pair are tracked from one frame to the next using Lucas–Kanade optical flow [10].

3.2 Arrangement of the trifocal tensors in the stereo system

A trifocal tensor, which is similar to the essential matrix, is the intrinsic geometry relating three views [11]. A trifocal tensor pair T and T' is used to constrain the slopes and displacements of the line features in the image views. The first tensor T relates the images of the first stereo pair I_1, I'_1 and the current image I_t taken by the left camera. The second trifocal tensor T' is formed with the current image pair I_t, I'_t together with the reference image I_1 from the left view. A graphical illustration of the configuration is shown in Fig. 3.

3.3 Handling feature replacement

A simple scheme catering for new features coming into the scene and old features moving out of the views is devised. Line correspondences extracted and matched in the stereo images are related by the trifocal tensors. They are fed into the EKF to find the innovation residual and will be discussed in the next section. If the number of available features is below any greater-than-2 integer k_c defined by the user, the views at the current time-step will be set as the new reference image pair and the tracker will be bootstrapped. The quadrifocal constraint is able to characterize the rigid motion of the camera

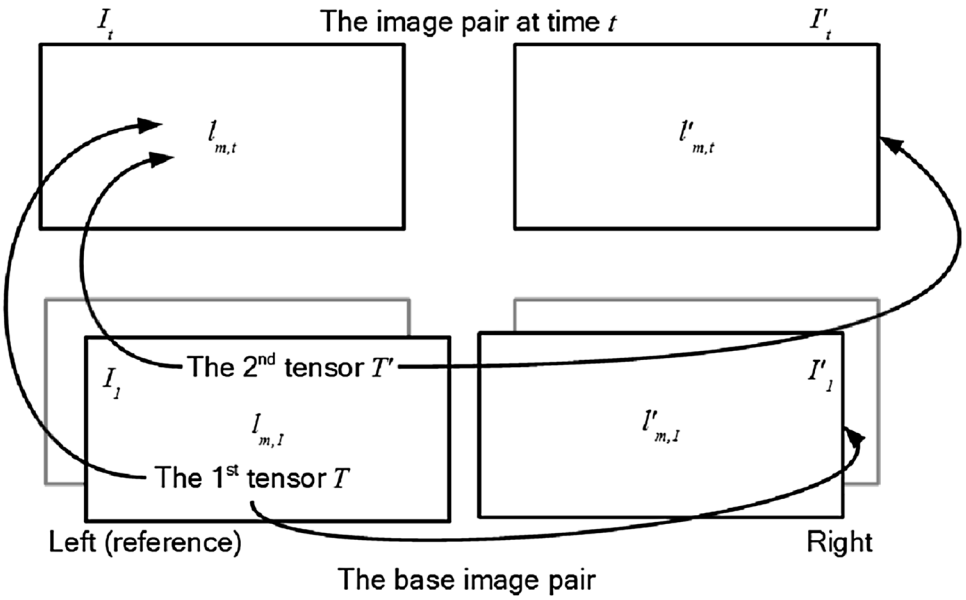

Fig. 3 The arrangement of the trifocal tensors. The first tensor T involves images I_1, I'_1 and I_t. The second tensor T' involves views I_1, I_t and I'_t

with two or more line correspondences across four views under standard EKF conditions. An EKF can be regarded as a recursive solution to a non-linear system of equations. The EKF is applied to estimate the six velocity values, and in turn the six pose parameters, of the camera system. With the use of two trifocal tensors across four views, a line feature that is represented by its slope and polar radius is able to provide four independent equations with line transfer function (16). A minimum of two lines present in the scene, which gives eight independent equations, is enough to make our system over determined. A larger value of the constant k_c can result in a higher stability.

4 Camera motion tracking using extended Kalman filter and trifocal tensors

An extended Kalman filter (EKF) is used to estimate the velocity of the motion in our system. To make the explanation clear, features in the reference image pair I_1 and I'_1 are assumed to be observable throughout the sequence in this section. Let us define the mathematical notations further. Lines $L_{m,t}$ and $L'_{m,t}$ are the mth line obtained from the left image I_t and right image I'_t at time t, respectively.

The formulation of our EKF is as follows. The state vector $\dot{\xi}_t$ representing the pose is defined as:

$$\dot{\xi}_t = \begin{bmatrix} \dot{x}_t & \dot{y}_t & \dot{z}_t & \dot{\alpha}_t & \dot{\beta}_t & \dot{\gamma}_t \end{bmatrix}^T \tag{10}$$

$\dot{x}_t, \dot{y}_t, \dot{z}_t$ are translation velocities along the axes. $\dot{\alpha}_t, \dot{\beta}_t, \dot{\gamma}_t$ are the angular velocities of the motion on the x, y and z axis, respectively. M_t can be regarded as an integral of velocity from the initial frame to the current time-step. The acceleration is modeled as zero-mean Gaussian noise η. The dynamic system equations of the filter are

$$\begin{cases} M_t = M_{t-1}\exp(\dot{\xi}_t) \\ \dot{\xi}_t = \dot{\xi}_{t-1} + \eta \end{cases}. \tag{11}$$

Relatively high sampling rate of the measurements is assumed such that motion of the object between successive images is small. With the first-order Taylor expansion, the exponential map of $\dot{\xi}_t$ in Eq. (5) can be approximated by

$$M_t = M_{t-1}(I + \tilde{\dot{\xi}}_t) \tag{12}$$

The measurement model, which relates the pose M_t, and the measurements $[\varepsilon_t, \varepsilon'_t]$ taken from the system, is defined as:

$$\begin{cases} \varepsilon_1 = g_t(M_t, \varepsilon'_1, \varepsilon_t) + \nu_t \\ \varepsilon_1 = g'_t(M_t, \varepsilon_t, \varepsilon'_t) + \nu_t \end{cases} \tag{13}$$

$$g_t(M_t, \varepsilon'_1, \varepsilon_t) = \begin{bmatrix} \theta_{1,1} & \lambda_{1,1} & \cdots & \theta_{m,1} & \lambda_{m,1} & \cdots & \theta_{N,1} & \lambda_{N,1} \end{bmatrix}^T \tag{14}$$

$$g'_t(M_t, \varepsilon_t, \varepsilon'_t) = \begin{bmatrix} \theta_{1,1} & \lambda_{1,1} & \cdots & \theta_{m,1} & \lambda_{m,1} & \cdots & \theta_{N,1} & \lambda_{N,1} \end{bmatrix}^T \tag{15}$$

ε_t and ε'_t are the measurements acquired by the left and right camera at time-step t, respectively. ν_t is a $2N \times 1$ that represents zero-mean Gaussian noise imposed on the images captured. g_t and g'_t are the $2N \times 1$ output line transfer function, where N is the number of extracted line features.

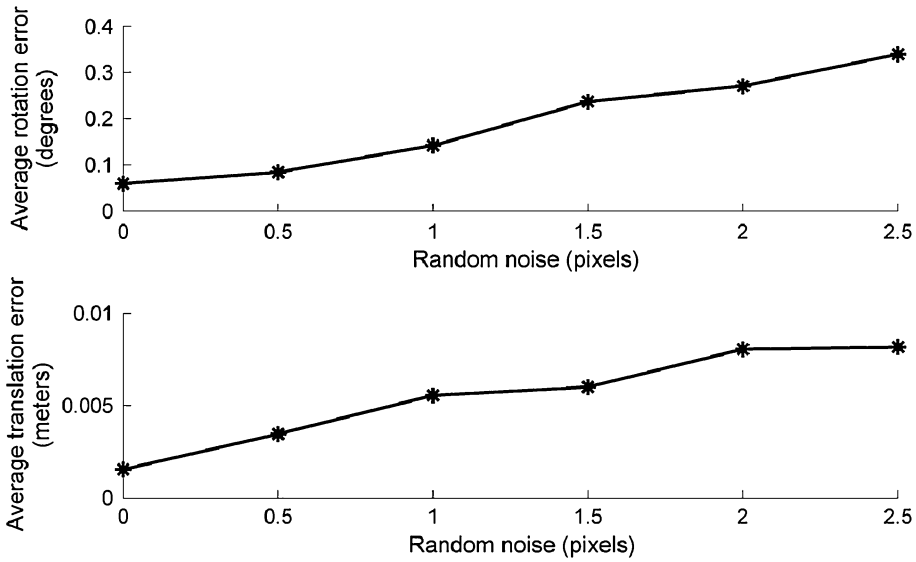

Fig. 4 The accuracy of the recovered poses varies as a function of measurement noise

The trifocal tensor line transfer formulae can be written in the tensor notation as:

$$\begin{cases} [L_{m,1}]^i = [L_{m,t}]^k [L'_{m,1}]_j T_i^{jk} \\ [L_{m,1}]^i = [L'_{m,t}]^k [L_{m,t}]_j T_i^{jk} \end{cases} \tag{16}$$

$L_{m,1}, L'_{m,1}, L_{m,t}$ and $L'_{m,t}$ are respectively, the normalized homogenous general form of $l_{m,1}, l'_{m,1}, l_{m,t}$ and $l'_{m,t}$. With the normalized 2-D coordinates, T and T' can be broken down as:

$$\begin{cases} T_i^{jk} = a_i'^j a_4^k - a_4'^j a_i^k \\ T_i'^{jk} = a_i'^j a_4''^k - a_4'^j a_i''^k \end{cases}. \tag{17}$$

The formulae above are in tensor notation. $a_i^j, a_i'^j, a_i''^j$ are respectively, the extrinsic parameters E of the stereo system, the elements of the upper 3×4 component of the camera motion matrix M_t, and the matrix product EM_t. More specifically, $E = [a_i^j], [I_{3\times 3} \quad 0_{3\times 1}]M_t = [a_i'^j]$ and $EM_t = [a_i''^j]$.

With Eqs. (16) and (17), 2-D lines can be transferred back to the first left camera image with the values of line measurements from the first right camera ε'_1, the measurements from the current left camera ε_t, the predicted motion M_t, and the rigid transformation E. Similarly, 2-D lines can be back-transferred to the first left view using lines from the current stereo image pair, the predicted motion M_t, and the extrinsic matrix E.

According to the dynamic system (11) and measurement model (13), the core equations for calculating the optimal estimates and correcting the estimates can be derived. The prediction equations of the EKF are:

$$\begin{cases} \hat{\dot{\xi}}_{t,t-1} = \hat{\dot{\xi}}_{t-1,t-1} \\ P_{t,t-1} = P_{t-1,t-1} + Q_t \end{cases}. \tag{18}$$

The update equations are:

$$\begin{cases} \hat{\dot{\xi}}_{t,t} = \hat{\dot{\xi}}_{t,t-1} + W\left(\left[\varepsilon_t - g_t(M_t, \varepsilon'_1, \varepsilon_t) \;\; \varepsilon'_t - g'_t(M_t, \varepsilon_t, \varepsilon'_t) \right]^T \right) \\ P_{t,t} = P_{t,t-1} - W \nabla g_M P_{t,t-1} \\ W = P_{t,t-1} \nabla g_M^T (\nabla g_M P_{t,t-1} \nabla g_M^T + R_t)^{-1} \end{cases} \tag{19}$$

$\hat{\dot{\xi}}_{t,t-1}$ and $\hat{\dot{\xi}}_{t,t}$ are the estimates of state $\dot{\xi}_t$ after prediction and update, respectively. $P_{t,t-1}$ and $P_{t,t}$ are respectively, the 6×6 covariance matrices of $\hat{\dot{\xi}}_{t,t-1}$ and $\hat{\dot{\xi}}_{t,t}$. R_t is the covariance of the measurement noise v_t while Q_t is the covariance of system noise η_t. W is the $6 \times 4N$ Kalman gain. ∇g_M is the Jacobian of the measurement functions g_t and g'_t evaluated at $\hat{\dot{\xi}}_{t,t-1}$.

The performance of the EKF can be improved in terms of robustness by adding a validation gate. It is necessary to prevent a line transfer from degeneracy. Such a constraint is used to exclude outliers and lines in the trifocal plane. Lines near the epipole are also excluded. It is based on the following inequality

$$r_{m,t}^T S_{m,t}^{-1} r_{m,t} < g. \tag{20}$$

$r_{m,t}$ is the innovation of an individual measurement pair at time t and $S_{m,t}$ is the corresponding residual covariance. g is the validation threshold defined by the user. Lines that do not satisfy Eq. (20) are dropped and ignored in the EKF. We set g to 0.2 in our implementation.

5 Experiment results

5.1 Experiments with synthetic data

The first simulation experiment was designed to determine how the accuracy of our novel method varies as the errors

in the image measurements increase. In the simulation, 40 synthetic lines, centered at 0.5 m away from the camera, were randomly generated. The camera had a focal length of 4.6 mm. The pixel dimensions were 5.42×10^{-3} mm by 5.42×10^{-3} mm. The two cameras in the stereo system were 0.1 m apart. They were pointing towards the positive direction of the z axis. The motion parameters per frame were randomly set with their maximum changes in rotation as 1.2° and 0.010 m in translation. A uniformly distributed random image error having a maximum of ±2.5 pixels was added to the endpoints of the line segments. These parameters modeled the situation as realistic as possible. The length of each synthetic sequence was 100 frames. For each test case, 20 trials were run to compute the average errors.

Figure 4 shows the relationship between the pose parameter errors and measurement noise. The errors in the plots are the accumulated total rotation and translation errors measured at the 100th frame. The errors of the recovered pose increase proportionally with the random image errors. Even with a noise of ±2.5 pixels, the translation and rotation errors were less than 0.01 m and 0.4°, respectively. A non-zero average error happened at zero measurement noise because the EKF could give a wrong initial guess at the beginning of the estimation process

The second experiment was performed to investigate the effects of inaccuracy in calibration data, such as focal length and camera projection center, on the proposed algorithm. The parameters in the simulation were similar to the previous experiment. No random error was added to the

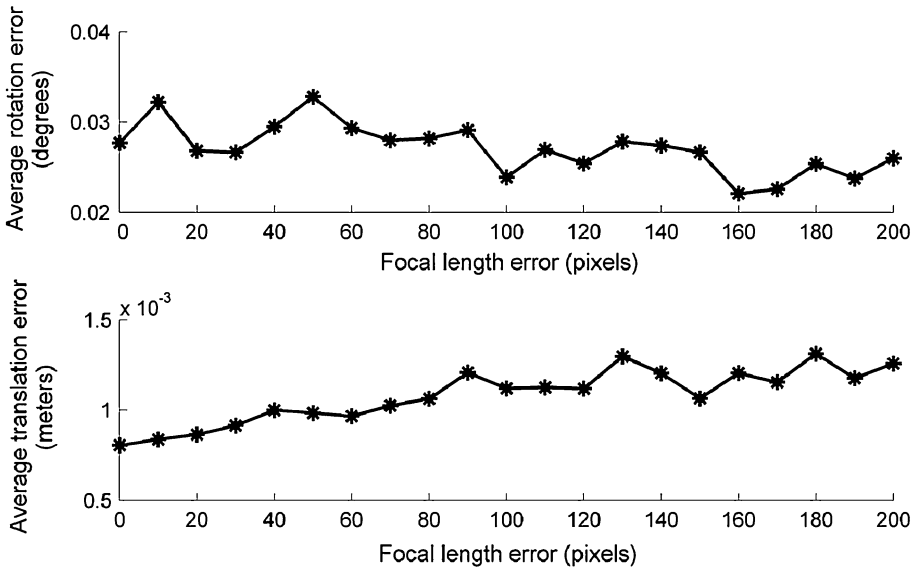

Fig. 5 The accuracy of the recovered poses varies as a function of errors in the calibrated focal length

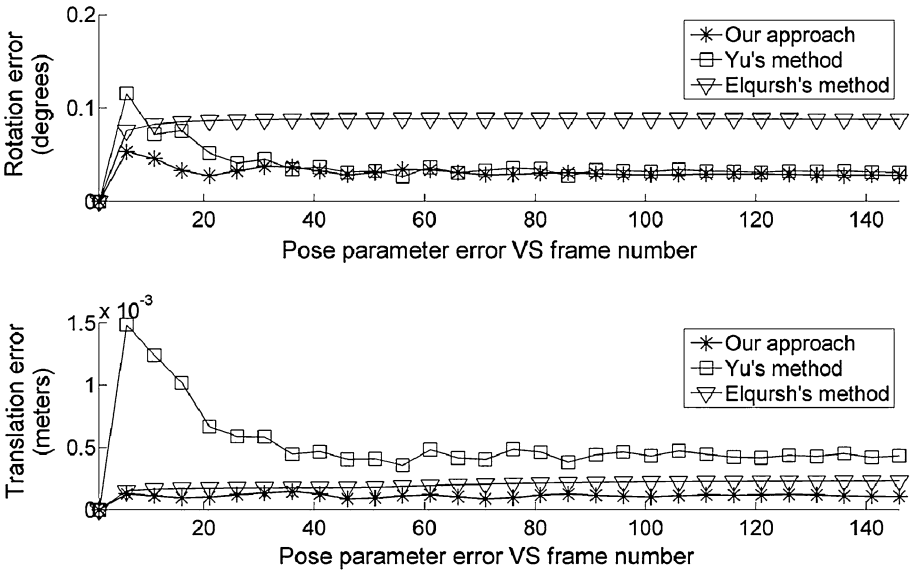

Fig. 6 The mean accumulated rotation (*top*) and translation (*bottom*) errors versus frame number of the algorithms under comparison. For the sake of clear presentation, results of method [19] are not included in the graphs

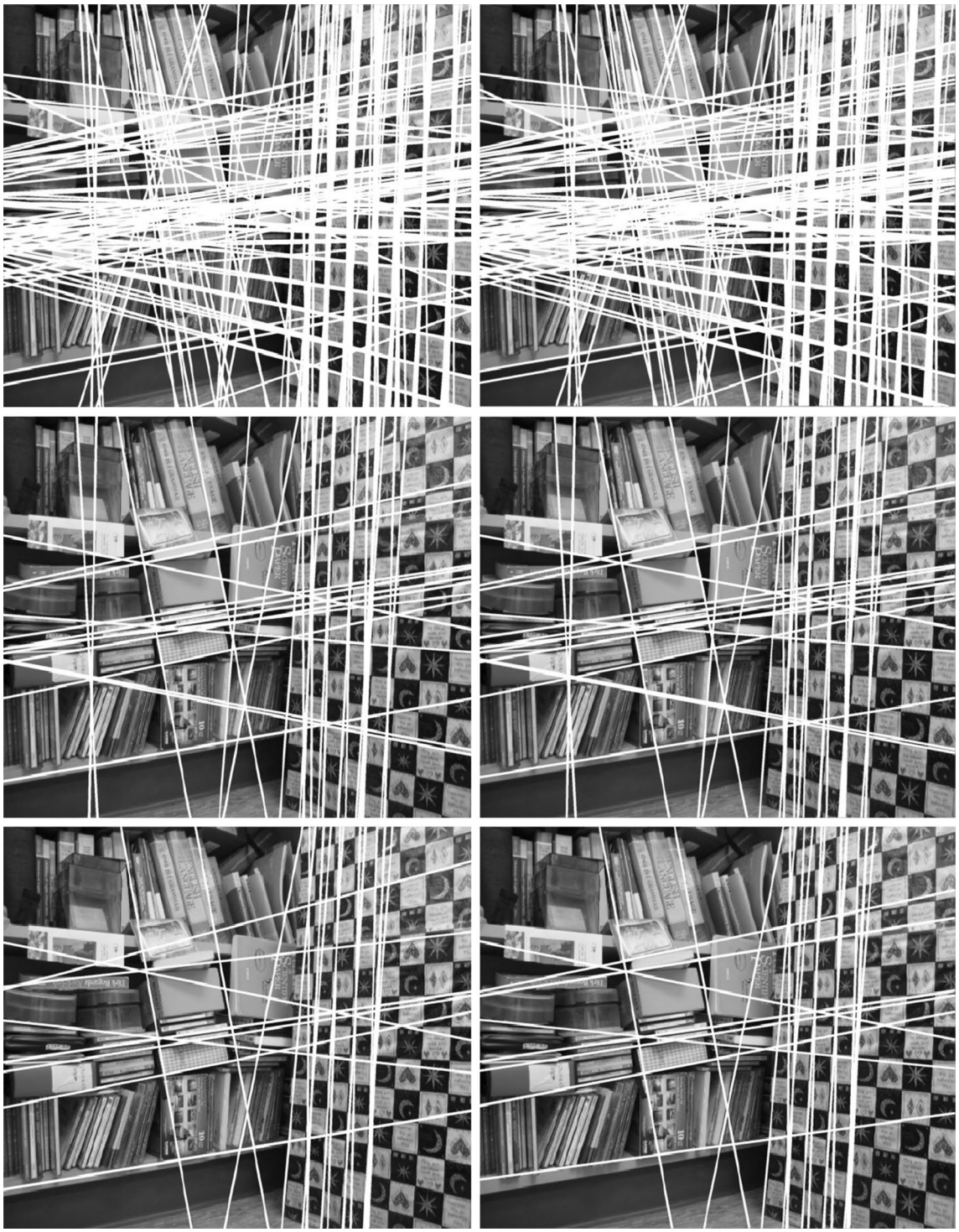

Fig. 7 The *top row* detected infinite lines in the first image pair in *white color*. The *middle row* results of line re-projection in the 35th image pair with the recovered camera motion. The *bottom row* results of line re-projection in the 90th stereo pair

endpoints. The true focal length was replaced with erroneous values. Figure 5 shows the results. Our algorithm was not sensitive to the deviations in focal length. With an error of 200 pixels, the average errors of the recovered camera rotation and translation were less than 0.03° and 1.5 mm, respectively.

In the third experiment, we want to compare the performance of the proposed algorithm, the point-based approach

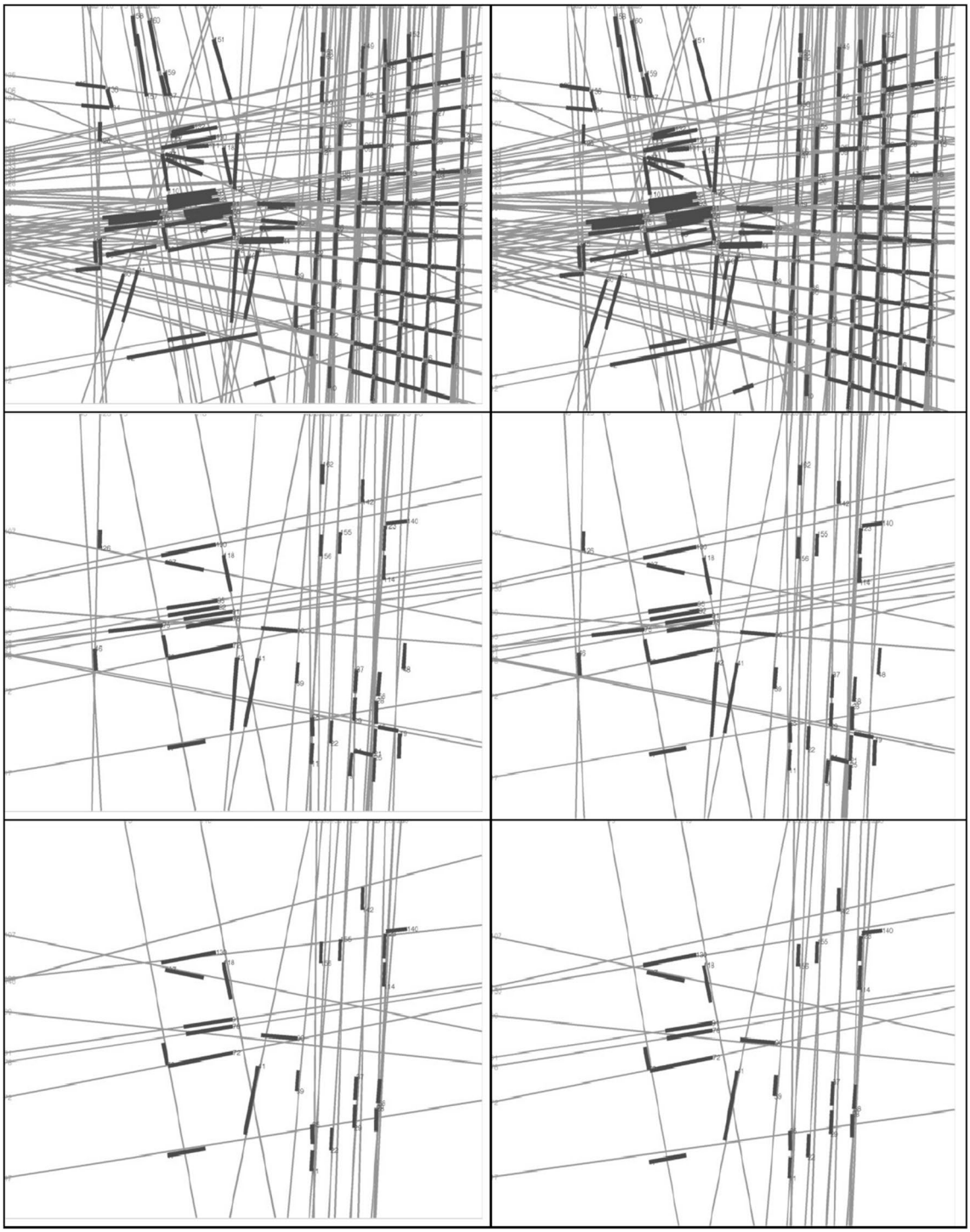

Fig. 8 The *top row* the detected line segments in *black* and the corresponding infinite lines in *gray* in the 1st stereo view. The *middle row* the tracked line segments in *black* and the re-projected lines in *gray* in the 35th image pair. The *bottom row* the tracked lines in *black* and the re-projection of lines in *grey* in the 90th image view

by Yu et al. [25], the line-based structure from motion approach by Taylor et al. [19], and the latest method that uses three lines by Elqursh and Elgammal [8]. The setting was similar to the first one. The endpoints of the line segments were treated as the point features for comparing with Yu's algorithm. Figure 6 shows the results. The lines with asterisks, squares and triangles are for our approach, Yu's algorithm [25] and Elqursh's method [8], respectively.

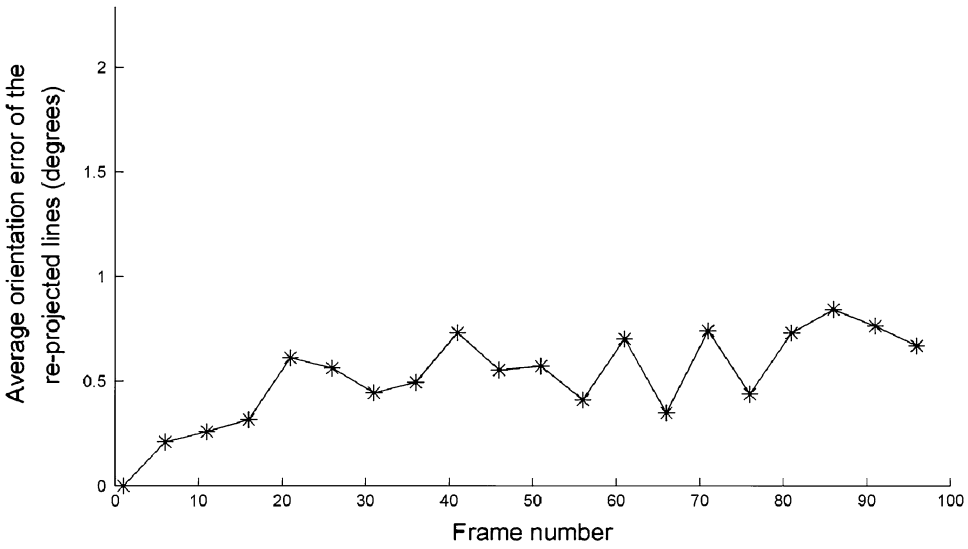

Fig. 9 Average re-projection errors of the line features versus frame number. The errors are measured in terms of the angles between the tracked and re-projected lines

Table 1 A comparison of computational time among the four algorithms

Algorithm\feature number	Minimum number	10	20	30
The proposed approach	0.0014 s	0.0025 s	0.0049 s	0.0078 s
Yu's method	0.0008 s	0.0016 s	0.0032 s	0.0051 s
Elqursh's method	0.0002 s	N/A	N/A	N/A
Taylor's method	>1 s	>1 s	>1 s	>1 s

The time presented is the average duration required to recover the pose from one stereo image in seconds

The errors presented in the figures are the mean differences between the actual and recovered pose in terms of rotation and translation. For the sake of clear presentation, results of Taylor's algorithm [19] are not included in the figure due to its relatively large error values. Indeed, its average rotation and translation errors were larger than 0.5° and 2 mm, respectively. From the results, our method had the lowest errors among the four methods.

The proposed method outperformed Elqursh's approach [8] in the comparison mainly because the latter one requires a special arrangement of lines, which is known as primitive configuration, for the computation of an accurate camera rotation. Such configuration comprises three lines with two of the lines parallel and their directions orthogonal to the third line. This scenario does not always occur in real environments and is not guaranteed in our simulation experiment. Our algorithm does not need such an assumption to acquire an accurate pose in the tracking process.

Another reason for a better performance is that our algorithm considers all the line features available in the scene. They are taken into account for a period of time until they are occluded or disappear. The camera pose is then tracked recursively using an EKF. A dynamic model is applied to relate the recovered motion in the sequence and filter off noisy results from time to time. For Elqursh's method [8], the calculation of the camera motion is independent in every time-step. It estimates the 3-D motion of an image sequence on an image pair basis. Three features are randomly picked to compute the relative rotation between two images under the RANSAC framework. The triplets of features used are different among the image pairs throughout the sequence. The proposed method is able to give a more accurate and reliable estimation of 3-D pose due to its ability to build up a continuous relation of features and motion in the tracking process.

The use of line features instead of points makes our method perform better than Yu's approach [25]. Lines can normally be determined more accurately than points due to multi-pixel support. Using infinite lines, the problem of occlusions can be alleviated when a part of the line is occluded. This cannot be achieved with the use of point features.

The computation time of the four approaches has been tabulated in Table 1. The comparison was done with a machine using a 1.7-GHz Intel processor. With minimum number of line features, the core part of the proposed approach took 0.0014 s to recover the 3-D motion from a stereo image, which was faster than Taylor's algorithm but slower than Elqursh's and Yu's method. Nevertheless, our algorithm can operate at a rate higher than 700 Hz. This is enough for a real-time application.

5.2 Experiments with real images

First, we are going to show that the 3-D motion of a robot can be accurately estimated with our algorithm. The image

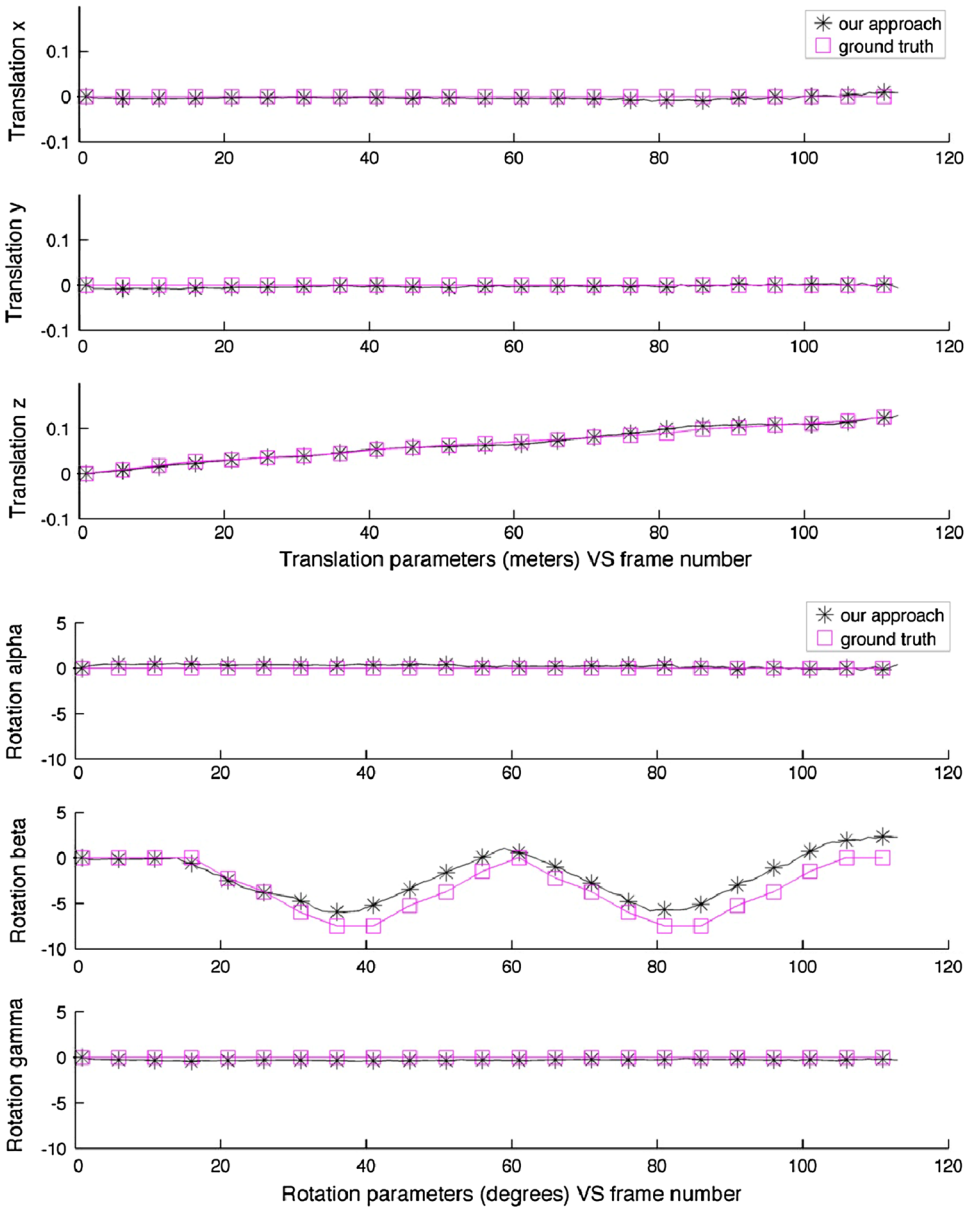

Fig. 10 A comparison of the recovered motion parameters from the 1st real sequence with the ground truth

sequence was taken by a robot driven by two stepping motors. The robot moved in front of the bookshelf and the ground truth of motion was recorded. A stereo camera was mounted on the top. The image resolution was 640 by 480 pixels. The ground-truth information was used to compare with the motion recovered by our method.

We verify the resulting pose information by re-projecting lines from the first (reference) stereo pair to the succeeding image frames in the stereo sequence. To do this, line features in the first and the *t*th image pairs were obtained. The trifocal tensors were computed using the recovered pose parameters. Then line features from the first image frames were projected to succeeding images with the trifocal tensors. Figure 7 shows the results of re-projection overlaid on the real images. Line features in the first image pair were transferred to the 35th and 90th stereo frames using the recovered 3-D pose. It is shown that the re-projected lines, which are in white, stick to the same position in the background in all the three image pairs. Figure 8 compares the 2-D position of the tracked and the re-projected line features, which are, respectively, indicated by black and gray lines. It can be observed that the positions of these two kinds of features are very close. Figure 9 shows the re-projection error in a quantitative manner. The discrepancies in orientation between the tracked and re-projected lines against frame number were plotted. The errors were so small and the angles between lines were within 1°. It means that the recovered 3-D camera motion was accurate.

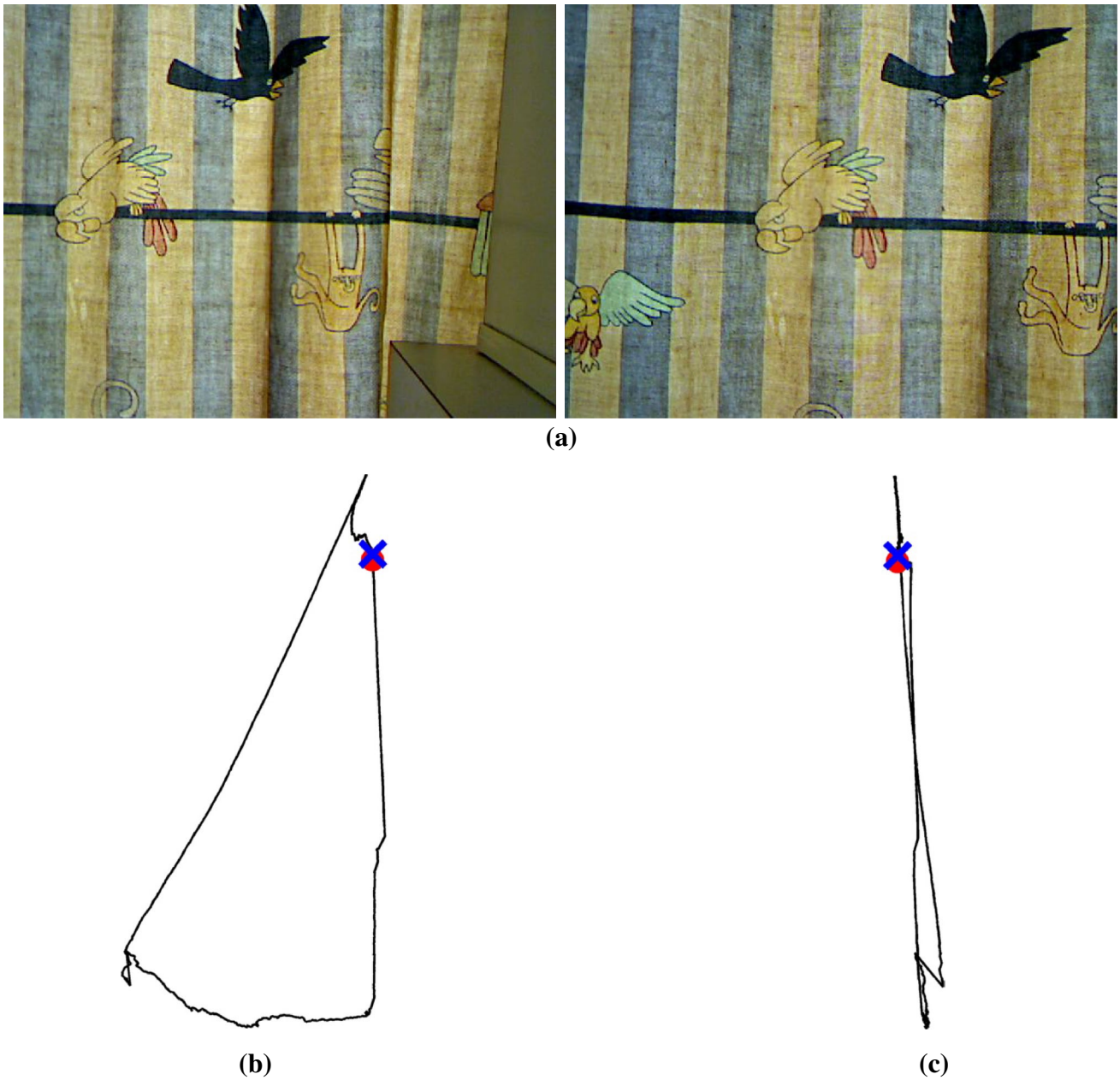

Fig. 11 The Curtain stereo sequence consisting 1,428 frames acquired by a stereo camera. The traversed distance was 1.5 m. *Red dots* and *blue crosses* indicate the starting and ending points, respectively. **a** Two sample images from the Curtain sequence consisting of 1,428 frames, **b** plan view of the recovered trajectory, **c** side view of the recovered trajectory (color figure online)

Figure 10 shows a comparison of resulting 3-D motion with the ground truth. One can find that the recovered motion closely follows the actual motion. It is believed that these errors were mainly due to the deviations of the calibrated values of the system parameters like focal length from the actual values.

The second test used the Curtain sequence consisting of 1,428 frames. It was acquired by a stereo camera pair mounted on a wheeled plate. The initial position was marked in the scene. The camera was moved on a planar surface and returned to the marker at the end. The traversed distance was 1.5 m. Figure 11 shows the recovered trajectory. Neither iterative optimization nor bundle adjustment was applied. Red dots and blue crosses are, respectively, the starting and finishing points. The distance between these points were small having a value of 0.01827 m, indicating that the proposed algorithm was accurate in this scenario.

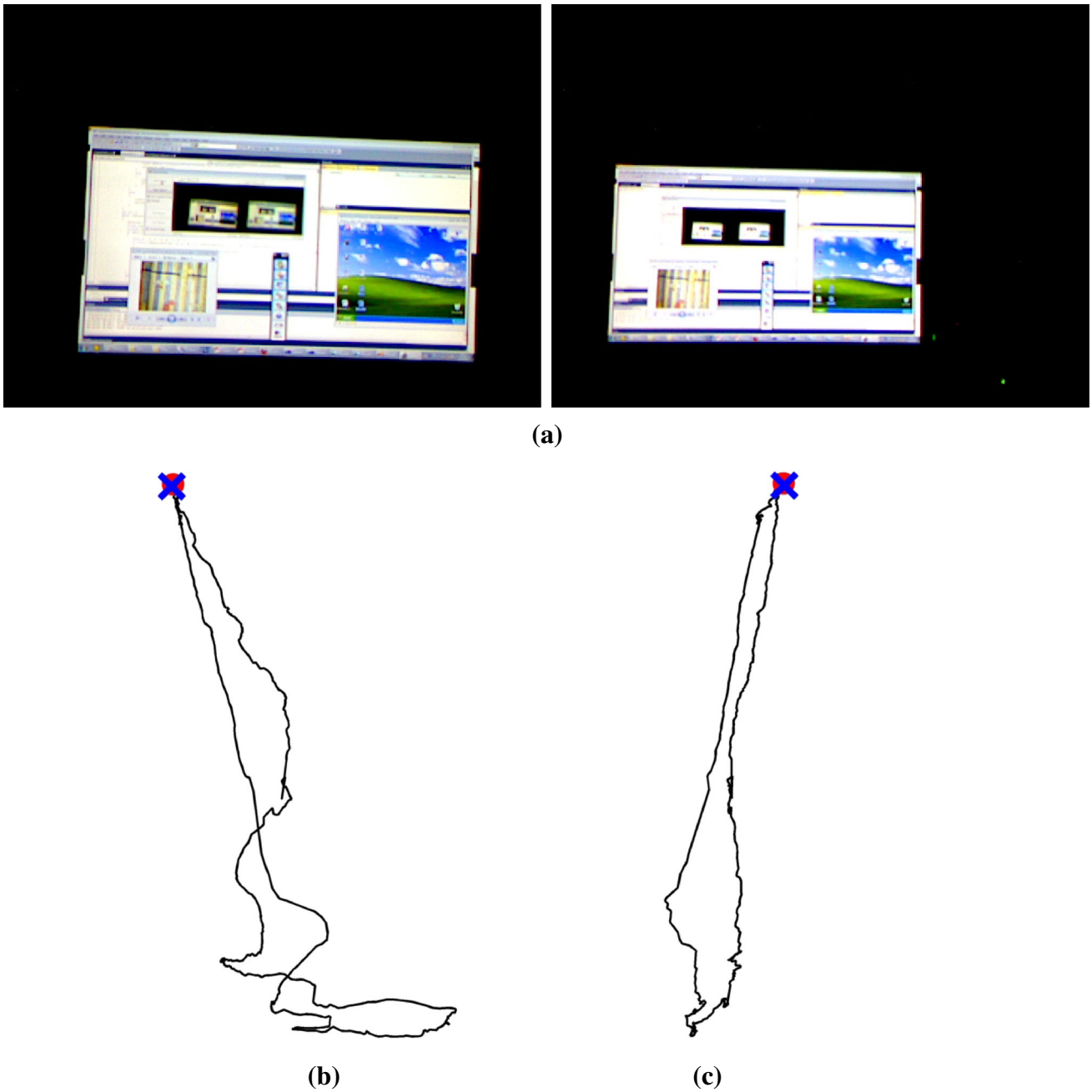

Fig. 12 The Monitor sequence consisting of 570 frames. The traversed distance was 3 m. *Red dots* and *blue crosses* represent the starting and finishing points, respectively. **a** Two sample images from the Monitor sequence consisting of 570 frames, **b** plan view of the recovered trajectory, **c** side view of the recovered trajectory (color figure online)

The third test used the Monitor sequence taken with a hand-held stereo camera. The camera was moved in front of a monitor screen arbitrarily and finally returned to the original position. The content displayed in the monitor was changing. This actually injected outlying features into the image sequence. The sequence had 570 frames and the total traversed distance of the camera system was 3 m. Figure 12 shows the trajectory recovered. Red dots and blue crosses shown in the trajectory indicate the starting and ending points, respectively. One can notice that the initial and ending positions were very close. The distance between them was 0.03172 m. It means that the problem of drifting of the proposed approach was very little even with the presence of outlying features in the scene. This was mainly due to the use of validation gate in the EKF.

Indeed, our approach has a few limitations in real situations. It is sensitive to the calibration error of the stereo camera system. To ensure reliable pose tracking results, the extrinsic parameters relating the two cameras on the stereo rig should be accurate. This is because trifocal tensors are employed to lock line features among the stereo views across different time-steps. Deviation of the extrinsic parameters from the real values will affect the computation of line transfer function of the measurement model in the filter, leading to an erroneous estimation of 3-D motion.

The 2-D motion of the lines in successive image frames should be small enough with a relatively high sampling rate. It is due to the fact that a Kanade's optical flow-based technique [10] is used to track lines in stereo views from time to time. Also, an EKF is adopted as the core for recursively acquiring 3-D pose in the image sequence. Given the above condition, the optical flow-based method is able to work normally and the EKF can predict the motion reliably in the process.

The proposed method cannot estimate the camera motion in a natural scene, such as inside a forest, where straight lines are not available. It could be our next step to devise an algorithm that considers curves in pose tracking to make our system work in such a natural environment.

6 Conclusion

We present a novel recursive approach for the computation of 3-D camera motion from a stereo image sequence. Line features instead of interest points in stereo images are utilized. The quadrifocal constraint, which is in the form of a trifocal tensor pair, is incorporated into the system to eliminate the step of 3-D structure reconstruction. This in return increases the speed and reduces the memory complexity without the scarification of the algorithm accuracy compared to the structures from motion-based methods. The system uses a minimum of two line correspondences while previous methods based on the trifocal tensor require at least thirteen. With the use of extended Kalman filter, our algorithm runs at a high speed. The core part of our method can operate at more than 700 Hz with 2 line features. This meets the requirement of most real-time applications. The validation gate in the Kalman filter discards outlying line features in the images, further improving the robustness of our algorithm. The proposed method outperformed a previous model-free approach [25] and a state-of-the-art method using three lines [8] in terms of accuracy in the synthetic data experiment. It is demonstrated in our real image test that our method was able to compute the 3-D motion precisely with little drifting from an image sequence surpassed a thousand image frames.

References

1. Aldoma, A., Marton, Z.C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R.B., Gedikli, S., Vincze, M.: Tutorial: point cloud library: three-dimensional object recognition and 6 dof pose estimation. IEEE Robot. Autom. Mag. **19**(3), 80–91 (2012)
2. Andreff, N., Espiau, B., Horaud, R.: Visual servoing from lines. Int. J. Robot. Res. **21**(8), 679–699 (2002)
3. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**, 679–698 (1986)
4. Chandraker, M., Lim, J., Kriegman, D.: Moving in stereo: efficient structure and motion using lines. In: Proceedings of the 12th International Conference on Computer Vision 2009, pp. 1741–1748 (2009)
5. Chen, H.H.: Pose determination from line-to-plane correspondences: existence condition and closed-form solutions. In: Proceedings of the Third International Conference on Computer Vision 1990, pp. 374–378 (1990)
6. Chiuso, A., Favaro, P., Jin, H., Soatto, S.: Structure from motion causally integrated over time. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 523–535 (2002)
7. Christy, S., Horaud, R.: Iterative pose computation from line correspondences. Comput. Vis. Image Underst. **73**(1), 137–144 (1999)
8. Elqursh, A., Elgammal, A.: Line-based relative pose estimation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2011, pp. 3049–3056 (2011)
9. Hartley, R.I.: Lines and points in three views and the trifocal tensor. Int. J. Comput. Vis. **22**(2), 125–140 (1997)
10. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Image Understanding Workshop, pp. 121–130 (1981)
11. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
12. Bouguet, J.Y.: Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/ (2010)
13. Lippiello, V., Siciliano, B., Villani, L.: Position and orientation estimation based on Kalman filtering of stereo images. In: Proceedings of the IEEE International Conference on Control Applications, 2001, pp. 702–707 (2001)
14. Lowe, D.G.: Fitting parameterized three-dimensional models to images. IEEE Trans. Pattern Anal. Mach. Intell. **13**(5), 441–450 (1991)
15. Qin, L.-J., Zhu, F.: A new method for pose estimation from line correspondences. Acta Automatica Sinica **34**(2), 130–134 (2008)
16. Rhee, S.-M., Lee, Y.-B., Kim, J.D., Rhee, T.: Pose estimation of a depth camera using plane features. In: IEEE International Conference Consumer Electronics 2013, pp. 133–134 (2013)
17. Schmid, C. Zisserman, A.N.: Automatic line matching across views. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1997, pp. 666–671 (1997)
18. Schmid, C., Zisserman, A.: The geometry and matching of lines and curves over multiple views. Int. J. Comput. Vis. **40**(3), 199–233 (2000)
19. Taylor, C.J., Kriegman, D.J.: Structure and motion from line segments in multiple images. IEEE Trans. Pattern Anal. Mach. Intell. **17**(11), 1021–1032 (1995)
20. Wang, Z., Wu, F., Hu, Z.: MSLD: a robust descriptor for line matching. Pattern Recogn. **42**(5), 941–953 (2009)
21. Werner, T., Pajdla, T.: Oriented matching constraints. In: Proceedings of British Machine Vision Conference, pp. 441–450 (2001)

22. Ye, M., Wang, X., Yang, R., Ren, L., Pollefeys, M.: Accurate 3d pose estimation from a single depth image. In: Proceedings of International Conference on Computer Vision, pp. 731–738 (2011)
23. Yu, Y.K., Wong, K.H., Chang, M.M.Y., Or, S.H.: Recursive camera-motion estimation with the trifocal tensor. IEEE Trans. Syst. Man Cybern. B Cybern. **36**(5), 1081–1090 (2006)
24. Yu, Y.K., Wong, K.H., Chang, M.M.Y.: Pose estimation for augmented reality applications using genetic algorithm. IEEE Trans. Syst. Man Cybern. B Cybern. **35**(6), 1295–1301 (2005)
25. Yu, Y.K., Wong, K.H., Or, S.H., Chang, M.M.Y.: Robust 3-d motion tracking from stereo images: a model-less method. IEEE Trans. Instrum. Meas. **57**(3), 622–630 (2008)
26. Zhang, Z., Shan, Y.: Incremental motion estimation through modified bundle adjustment. In: Proceedings of International Conference on Image Processing, vol. 2, p. 343 (2003)
27. Li, Z., Sastry, S.S., Murray, R.: A mathematical introduction to robotic manipulation. CRC Press (1994)
28. Comport, A.I., Malis, E., Rives, P.: Accurate quadrifocal tracking for robust 3d visual odometry. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 40–45 (2007)
29. Weng, J., Huang, T.S., Ahuja, N.: Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization. IEEE Trans. Pattern Anal. Mach. Intell. **14**(3), 318–336 (1992)
30. Yu, Y.K., Wong, K.H., Or, S.H., Junzhou, C.: Controlling virtual cameras based on a robust model-free pose acquisition technique. IEEE Trans Multimed 11(1), 184–190 (2009)
31. Yu, Y.K., Wong, K.H., Chang, M.M.Y.: Merging artificial objects with marker-less video sequences based on the interacting multiple model method. IEEE Trans Multimed 8(3), 521–528 (2006)
32. Yu, Y.K., Wong, K.H., Chang, M.Y.Y.: A fast recursive 3D model reconstruction algorithm for multimedia applications. In: Proceeding of the International Conference on Pattern Recognition, vol. 2, pp. 241–244 (2004)