American Journal of Engineering and Applied Sciences

Original Research Paper

Efficient Pose Tracking based on Line Segment Matching

Junqiu Wang, Jingjing Cui, Yanjie Gao, Yanxuan Ma, Zhichao Gan and Chao Yang

Beijing Changcheng Aviation Measurement and Control Institute, Aviation Industry Cooperation China, Beijing, China

Article history
Received: 29-10-2018
Revised: 22-11-2018
Accepted: 13-12-2018

Corresponding Author:
Junqiu Wang
Beijing Changcheng Aviation Measurement and Control Institute, Aviation Industry Cooperation China, Beijing, China
Email: jerywangjq@aliyun.com

Abstract: Pose tracking is a crucial issue for many applications such as robotic tasks and facility operations. Vision-based approaches with non-contact properties are appropriate choices for these tasks. However, vision-based approaches are not sufficiently robust and fast. In this work, we propose a vision-based pose tracking to deal with these problems. We estimate poses using Lie group and Lie algebra representation theory. Such operation is performed in a linearized space, therefore it is convenient for pose estimation. To provide reliable visual information for our pose estimation, we detect line segments. Our detection of line segment depends on semi-global image information. We describe all line segments and match those detected in consecutive frames. Our line segment detector and matching descriptor are good at discarding ambiguous line segments and finding real ones in noisy situations. The integration of group theory and line segment detection and matching plays an important role for developing a robust vision-based pose tracking system. Our system proves to be efficient and robust.

Keywords: Pose Tracking, Lie Algebra, Line Detection and Matching

Introduction

Vision-based pose estimation finds numerous applications, including robotic tasks, facility operations and automatic measurement. Real-time vision-based pose estimation is becoming practical in recent years. Despite of the development, real-time precise pose estimation is still a challenging task because the information in a video stream is huge. In addition, viewpoint variations, image noises and illumination changes are difficult to handle in many scenarios. We aim at developing a robust vision-based 6D pose estimation based on the projection of an object in a image sequence. There are two key ideas in our system development. The first one is pose estimation using Lie group and Lie algebra representation theory. The second one is detection and matching of line segment features. The integration of the two techniques is important to develop a robust vision-based pose tracking system.

Given an initialization for pose tracking, the transformation of continuous poses can be represented using 3D rotation and translation. The representation in a homogeneous coordinate setting is a 4×4 matrix. We estimate the transformation to update the relative pose between a camera mounted on a system an object in the camera's view. Given two sets of points $P = \{p_i\}$, one set is gotten by a Euclidean transformation and another is detection results in an image, we can estimate the motion. A good representation for estimating transformations needs to be composed, inverted and differentiated. Unfortunately, direct matrix operations do not meet the above demand. It is especially difficult to differentiate a transformation matrix. To deal with this problem, we represent object motion using Lie group and Lie algebra theory.

Group theory focuses on the algebraic structures that have certain properties on a set. A Lie group is a nonempty subset where a smooth manifold and a topological group are defined. Lie groups are differentiable manifolds (Sattinger and Weaver, 1998). The group operations of a Lie group are smooth. The inverse mapping of a Lie group is also smooth. A Special orthogonal Lie group SO(3) describes rotations in a 3D space. A Special Euclidean group SE(3) represents 3D rigid transformations that include linear transformation on homogeneous 6-vectors (Taylor and Kriegman, 1994). Lie algebra defined on the tangential space of a Lie group characterizes the local properties of the elements in the group. Lie algebra consists of a vector space over some field and a binary operation named as the Lie bracket operation. The Lie algebra $so(3)$ of SO(3) can be described by a 3D vector that can

be transformed into a skew-symmetric matrices. The Lie algebra *se*(3) associated with the Lie group SE(3) is able to represent motions. It is also possible to describe coordinate frame transformations using the ad-joint representation of a Lie group. We can find local representation and operation of a Lie group to estimate parameters in a linearized way.

To provide reliable information for pose estimation, feature correspondences between consecutive frames are crucial for the pose estimation results (Davison, 2003; Drummond and Cipolla, 2002; Kim *et al.*, 2016). Point matching has been widely used for tracking and pose estimation because it is easy to detect and find point feature correspondences in consecutive frames using an effective feature description. However, point features are not always consistent due to viewpoint and illumination variations. Although it is possible to find scale, rotation, or affine invariant features (Lowe, 2004; Mikolajczyk *et al*., 2005), such features are computationally expensive to be detected. Therefore, Invariant features are not good choices for practical applications such as tracking and pose estimation.

Edge features are abundant in images. The Canny edge detector can find edges in a heuristic way. The number of edges is adjusted by the thresholds. Edge features are searched in 1D when the direction of the searching can be calculated (Rosten and Drummond, 2005). Edge features are not always reliable especially in image regions with low contrast. In addition, some edge features are formulated due to local noises. Such edges bring outliers to pose estimation systems.

Lie group theory has been applied in a tracking system based on edge detection (Drummond and Cipolla, 2002). The edge detection is performed in a local region. The detection results might have drifts caused by the noises in local neighborhoods. The drift of edges can lead to tracking errors, which should be avoided. In contrast, we detect line segments based on semi-global information. These line segments are used to find correspondences with the projection of the 3D model.

This paper is organized as follows. After the related work in section 2, we introduces line segment detection, description and matching in section 3. Camera projection and Lie group and Lie algebra formulation are given in section 4. Experimental results on image sequences are demonstrated in section 5. Section 6 concludes this work.

Related Work

Our goal is to develop fast and robust pose tracking systems. Pose tracking has been handled based on different sensors. Ultra sound and laser sensors are useful in many applications (Talib *et al.*, 2007; Koolwal *et al.*, 2010). Although ultrasound sensors are cheap and convenient, they do not provide high accuracy. Laser sensors can be precise. Unfortunately, it is relatively difficult to measure an object at many points simultaneously using laser sensors. Vision-based pose tracking does not need expensive hardware. Moreover, vision-based approaches are flexible to add other functions such as recognition. They have better flexibility than other sensor-based approaches. Vision-based pose tracking has been dealt with using deterministic approaches (Wunsch and Hirzinger, 1997) and probabilistic approaches (Isard and Blake, 1998; Wang and Yagi, 2008). Using either approaches, motion has to be linearized to solve the problem. In this work, we handle this by using Lie group and Lie algebra. This bears certain similarities to a few works such as (Taylor and Kriegman, 1994; Drummond and Cipolla, 2002; Rosten and Drummond, 2005; Flint *et al.*, 2011).

However, our method is different from these works in three important aspects: First, we consider line segments as reliable input for pose tracking; second, we propose a line segment descriptor for matching; third, we consider the problem as a minimization using a robust fitting function. Line segments have several advantages compared with low-level features. Our line segment descriptor can discard ambiguous correspondences in consecutive frames, which is important for the input of the minimization process. The robust fitting function is useful for getting a stable solution to the minimization problem. Different from pose estimation and tracking based on point, edge and line features, we compute 3D relative pose by using line segments. In our system, line segments detection runs in a semi-global way. Given a local seed pixel, we find possible a line segment based on gradient orientation consistency. We also propose a line segment feature descriptor to match line segments. We discard ambiguous line segments based on matching results.

Line Segment Detection, Description and Matching

Line feature detection can be cast as a two-step issue: Detecting edges and connecting edges. Line features can be detected by connecting edge features with similar gradient orientation. This approach heavily depends on the edge detector. Edges in weak contrast regions are difficult to be detected. Although using a low threshold in Canny edge detector can increase the detection probability, it is at the cost of high false-positives. In addition, the computational cost is high using the two steps including the edge detection and connection.

Hough transform detects line features in a global way based on a transformation from feature space to parameter space. A Hough transform based method performs detection by splitting the input frame into a set of voting elements (Hough, 1962). The Hough vote for

each line hypothesis is simply obtained from the edges by observing the distribution of line parameters. Hough Transform has been extended for multiple object instances detection (Barinova *et al.*, 2012). Hough transform calculates all the votes in an image. Unnecessary information is considered for line detection. Hough transform cannot provide robust performance in image regions with low contrast because the votes from such regions cannot compete with the votes from other regions with high contrast.

Line segments contain more information than lines since they provide the ends of the lines. Line segments tend to be more reliable than lines in this application. Line segment detection can be done by proposing line hypotheses using corners found in an image (Rosten and Drummond, 2005). The features are detected using a corner detector. Then, many hypotheses are proposed based on the corners' position. The hypotheses are tested and those passing the test are considered as useful lines for pose estimation. This approach has to test a large number of hypotheses because line hypotheses are quadratic with respect to the number of the corners detected. In addition, many lines passed the test are overlapped. It is also difficult to eliminate the ambiguous lines.

Line Segment Detection

Line segments can be formulated by occlusion boundaries of foreground object and their background (Burns *et al.*, 1986). It is also possible that contrast in an object leads to a line segment. Line segments are sparse compared with the number of the pixels in an image. The direction of a line segment is closely related with the pixel gradient orientations in its neighborhood. They are approximately orthogonal to each other (Gioi *et al.*, 2010). The pixels following this rule are named as aligned pixels. They are important for line segment detection. Despite of the sparse property of line segments, there are a lot of line segment hypotheses when we consider a local region in an image. Most of the hypotheses will be discarded if a large support region is checked. A rectangular region along a line segment is defined as support region for the line segment support region. The pixels following the gradient consistency rule are used for line segment detection.

We calculate the difference between two pixels:

$$\Delta I(u,v) = I(u+1,v+1) - I(u,v) \tag{1}$$

where, $I(u,v)$ is the intensity value at pixel (u,v).

Then, the difference is used for calculating the gradients:

$$g_u(u,v) = \frac{I(u+1,v) + \Delta I(u,v) - I(u-1,v)}{2}, \tag{2}$$

$$g_v(u,v) = \frac{I(u,v+1) + \Delta I(u,v) - I(u,v-1)}{2}, \tag{3}$$

Since $\Delta I(u,v)$ appears in both (2) and (3). $\Delta I(u,v)$ is calculated once and used twice to reduce the computational cost. The magnitude of gradient is:

$$m(u,v) = \sqrt{g_u^2(u,v) + g_v^2(u,v)}, \tag{4}$$

We calculate the derivatives respectively:

$$\alpha_u(u,v) = \frac{g_u(u,v)}{m(u,v)}, \tag{5}$$

$$\alpha_v(u,v) = \frac{g_v(u,v)}{m(u,v)}. \tag{6}$$

The gradient orientation of a pixel (u,v) can be calculated by:

$$\alpha(u,v) = \arctan\left(\frac{g_v(u,v)}{g_u(u,v)}\right). \tag{7}$$

Gradient orientation is compared with the direction of a line segment hypotheses. We do not calculate the difference between two orientations directly because it is computationally expensive. The angle difference calculated directly can be ambiguous because of the periodicity property. Instead, we keep the two values $\alpha_u(u,v)$ and $\alpha_v(u,v)$ for each pixel.

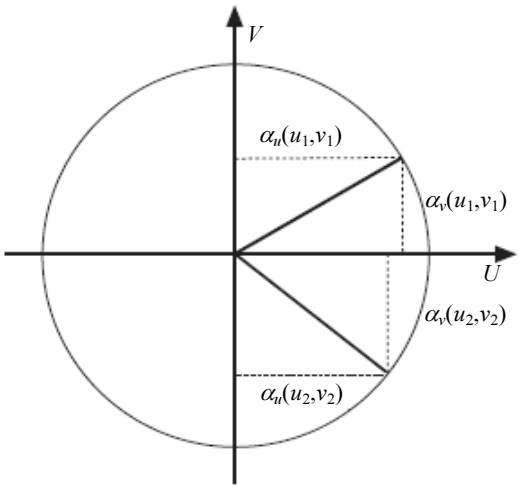


Fig. 1: Distance between two angles is calculated by measuring the Euclidean distance between the two points projected on a circle by the angles

Orientation difference of two pixels $I(u_1,v_1)$ and $I(u_2,v_2)$ is calculated by measuring the distance between $\Delta\alpha_u = \alpha_u(u_1,v_1)-\alpha_u(u_2,v_2)$ and $\Delta\alpha_v = \alpha_v(u_1,v_1)-\alpha_v(u_2,v_2)$.

The calculation shown in Fig. 1 is:

$$dist\left(\alpha\left(u_1,v_1\right)-\alpha\left(u_2,v_2\right)\right)=\sqrt{\left(\Delta\alpha_u\right)^2+\left(\Delta\alpha_v\right)^2}. \quad (8)$$

In the input image, the pixels on the edges have high gradient magnitudes than the pixels in the smooth regions. A few pixels with consistent gradient orientations are sampled as the seed set of a line segment. It is known that the gradient orientations of a line segment are consistent. We check the gradient orientation consistency between the selected pixel and its neighborhood. Once a pixel is sampled as a seed, it will grow from one pixel to many pixels that form a region. The pixels in the region have similar gradient orientations. The pixels in rectangle region are calculated to check the fitness of the region. Orientation difference calculated in (8) is computationally inexpensive. Moreover, it reflects the orientation difference in a better way. For example, the difference calculation between two angles is not straightforward when one angle's value is larger than 0 and another one is less than 0.

We sample several pixels in a region and calculate the orientations of these pixels. Then, we compute median of the orientations in the pixel set. The pixel that has the shortest distance to the median is considered as the seed for region growing. Pixels are checked along the direction given by the median. The pixels' gradient orientations within the range of orientations are added into the region. The orientation range is set to $\frac{\pi}{8}$ corresponding to a distance threshold 0.3902 using (8). The direction of the growing rectangular region Rec can be updated by accumulating i^{th} pixel's gradient orientations:

$$\arctan\left(\frac{\sum_{i\in \mathrm{Rec}}\alpha_v\left(u_i,v_i\right)}{\sum_{i\in \mathrm{Rec}}\alpha_u\left(u_i,v_i\right)}\right). \quad (9)$$

The weighted coordinate mean of the rectangular region is calculated by:

$$c^u_{\mathrm{Rec}}=\frac{\sum_{i\in \mathrm{Rec}}\left(m_i * u_i\right)}{\sum_{i\in \mathrm{Rec}} m_i}, \quad (10)$$

and:

$$c^v_{\mathrm{Rec}}=\frac{\sum_{i\in \mathrm{Rec}}\left(m_i * v_i\right)}{\sum_{i\in \mathrm{Rec}} m_i}. \quad (11)$$

The direction of the rectangular region is estimated by calculating the eigenvector of the association matrix defined by:

$$D_{\mathrm{Rec}}=\begin{bmatrix} d_{uu} & d_{uv} \\ d_{uv} & d_{vv} \end{bmatrix}, \quad (12)$$

where, the elements of the matrix d_{uu}, d_{uv}, d_{vv} are calculated by:

$$d_{uu}=\frac{\sum_{i\in \mathrm{Rec}} m_i\left(u_i-c^u_{\mathrm{Rec}}\right)^2}{\sum_{i\in \mathrm{Rec}} m_i}, \quad (13)$$

$$d_{uv}=\frac{\sum_{i\in \mathrm{Rec}} m_i\left(u_i-c^u_{\mathrm{Rec}}\right)\left(v_i-c^v_{\mathrm{Rec}}\right)}{\sum_{i\in \mathrm{Rec}} m_i}, \quad (14)$$

$$d_{vv}=\frac{\sum_{i\in \mathrm{Rec}} m_i\left(v_i-c^v_{\mathrm{Rec}}\right)^2}{\sum_{i\in \mathrm{Rec}} m_i}. \quad (15)$$

The eigenvector associated with the smallest eigenvalue of the matrix can be approximately calculated using an iterative algorithm within a few iterations (Gastal and Oliveira, 2012). The computational complexity of this eigenvector computation method is linear to the matrix dimension. In contrast, the computational complexities of the traditional methods are quadratic to the dimensionality of the matrix. In our implementation, we can estimate eigenvectors using 1 or 2 iterations. In most cases, only 1 iteration is sufficient for accurate estimation. Figure 2 shows the detection results of one image.

Line Segment Description and Matching

We can find the line segment correspondence for a projection of a 3D contour model. The searching is performed based on the line segments nearby the contour projection.

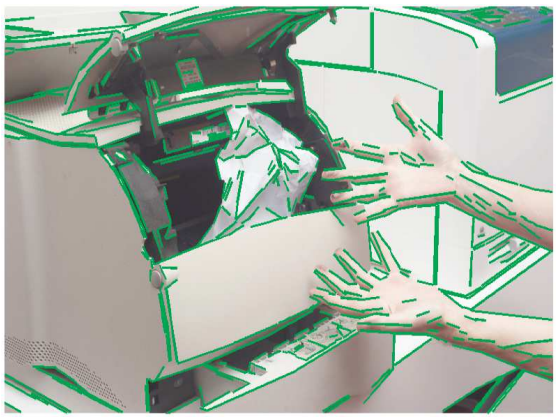


Fig. 2: Line segment detection results using our detector

However, this approach is not able to guarantee correct matching because multiple line segments are available in the neighborhood. Incorrect line segment can be selected for the pose estimation, which leads to failure of pose tracking. To deal with this problem, we match line segments detected in consecutive image frames. The line segments in the previous image frame are detected and described using a compact representation. They are matched with the line segments detected in the current image frame.

A good line segment descriptor should be compact and efficient for the comparison. We propose a descriptor by comparing the intensities of pixel pairs sampled in the rectangular region of a line segment. The intensity comparison of a pair of pixels is saved as 1 binary digit. The digit is set to 1 when the first pixels sampled is brighter than the second one. It is set to zero, vice versa. The matching is performed by calculating the Hamming distance between two line segment binary description vectors. The matching process greatly reduces the danger of incorrect matching. The binary description has been used in other matching approaches such as Sum of Squared Difference (SSD) are more expensive for line segment matching.

In addition, matching using SSD is sensitive to the small differences of pixel intensity variations. The matching results tend to be not very robust to noises and viewpoint changes. We estimate the relative position by minimizing an error term that is define as the sum of the distances between the projected features of the 3D model and their corresponding image features projected by the object.

Pose Tracking

We calibrate a camera using the method proposed by Zhang (2000). The pose is initialized by using the effective method (DeMenthon and Davis, 1995; Desolneux *et al.*, 2000). We detect line segments in an input image. The correspondences are found by using our line segment matching method in Section III-B. We linearize the motion using Lie algebra. The motion is calculated by solving a minimization problem. The flowchart of our system is shown in Fig. 3.

3D to 2D Projection

3D objects have projections in images. A camera model with intrinsic and extrinsic parameters describes the mapping from 3D to 2D. The intrinsic parameters of a camera is characterized in a 3×3 matrix **A**:

$$\mathbf{A} = \begin{bmatrix} f_u & & u_0 \\ & f_v & v_0 \\ & & 1 \end{bmatrix}, \tag{16}$$

where, f_u and f_v are the scale parameters of the camera; u_0 and v_0 are the coordinate deviations from the principal point in the image plane. The extrinsic parameters are described by a 3×4 matrix consisting of a rotation matrix **R** and a translation vector **T**:

$$\mathbf{E} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix}. \tag{17}$$

The projection from a 3D point $[X\ Y\ Z]^T$ to 2D image point $[u\ v]^T$ is determined by the intrinsic and extrinsic parameters. We use homogeneous coordinates to facilitate the computation:

$$\begin{bmatrix} \breve{u} & \breve{v} & \breve{w} \end{bmatrix}^T = \mathbf{AE}[X \quad Y \quad Z \quad 1]^T. \tag{18}$$

The coordinates in the image are calculated by:

$$\begin{bmatrix} u & v \end{bmatrix}^T = \begin{bmatrix} \breve{u} / \breve{w} & \breve{v} / \breve{w} \end{bmatrix}^T. \tag{19}$$

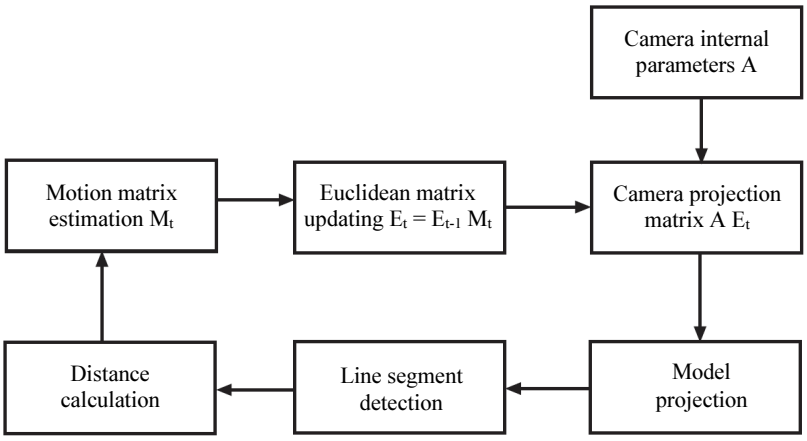


Fig. 3: The flow chart of our vision-based pose tracking system

Motions in consecutive frames are updated by right multiplying of the projection matrix $\mathbf{E}_{t\text{-}1}$ in the last frame by a Euclidean transformation:

$$\mathbf{E}_t = \mathbf{E}_{t-1}\mathbf{M}_t, \tag{20}$$

$\mathbf{M}_t$ is a 4×4 matrix composed of a rotation $\mathbf{R}_\mathrm{M}$ and a translation $\mathbf{t}_\mathrm{M}$ of the motion:

$$\mathrm{M} = \begin{bmatrix} \mathbf{R}_\mathrm{M} & \mathbf{T}_\mathrm{M} \\ \mathbf{q} & 1 \end{bmatrix}, \tag{21}$$

here q = [0 0 0].

We estimate the motion matrix based on the input of an input image. Then, the pose is calculated using the known transformation.

Minimization for Motion Tracking

The pose of the camera is calculated using the correspondences of the features. To make the estimation more precise, we perform M-estimation to minimize the fitting errors. The projection of the 3D model in an image p = $[u\ v]^T$ is calculated by (18) and (19). The corresponding point $\dot{p} = [\dot{u}\ \ \dot{v}]^T$ is found in the image of a line segment by using the line matching. The distance between the two points indicates the motion between consecutive frames:

$$\psi(q - p), \tag{22}$$

where, ψ(q-p) is a metric for measuring the distances. The simplest definition is measuring the Euclidean distance between the two points:

$$\psi(q - p) = \|q - p\|_2. \tag{23}$$

The matching of a single point pair is insufficient for the transformation estimation. In contrary, many correspondences are applied in the transformation estimation. The transformation is calculated by solving a least square problem:

$$d = \sum_{i=1}\psi(q_i - p_i). \tag{24}$$

To solve this problem, the derivative of the transformation matrix is necessary for the minimization. The derivative of the matrix should be calculated in a 6D space because the transformation is described by 3 rotation parameters and 3 translation parameters. Unfortunately, the transformation matrix is not closed for matrix addition. The result of two transformation matrix addition is not meaningful.

Motion Estimation using Lie Algebra

A Lie groups SO(3) represented in a $\mathbf{R}^{3\times4}$ matrix has 3 degrees of freedom corresponding to a rotation. A Lie group SE(3) described by a $\mathbf{R}^{4\times4}$ matrix has 6 degrees of freedom: $\mathbf{R}\in$SO(3) and $\mathbf{R}\in\mathbf{R}^3$. The Lie algebra se(3) of the Lie group SE(3) consists of the infinitesimal generators of rotations and translations. The generators are the elements of the tangent space of the manifold at the identity element of SE(3). The Lie algebra *se*(3) corresponding to the Lie group with 6 generators:

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{25}$$

$$G_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{26}$$

$$G_5 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G_6 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{27}$$

The mapping $\mathbf{R}^3 \rightarrow se(3)$ is:

$$\varphi, \omega \in R^3, \mathrm{alg}\begin{bmatrix} \varphi \\ \omega \end{bmatrix} = \begin{bmatrix} \omega_\times & \varphi \\ 0 & 0 \end{bmatrix}, \tag{28}$$

where, ϕ denotes the translation vector and ω denotes the rotation vector. $\omega_\times$ is the anti-symmetric matrix transformed from ω. alg is the linear combination of the generators.

Linear combination of the generators is specified by the coefficients:

$$a\mathrm{lg}: R^6 \rightarrow se(3) \subset R^{6\times6}, \mathrm{alg}(c) = \sum_{k=1}^{6} c_i G_i. \tag{29}$$

The motion matrix is approximated by the generators linearly:

$$M \approx \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \sum_{k=1}^{6} c_i G_i. \tag{30}$$

The partial derivatives corresponding to the motion parameters can be obtained by:

$$\left[\breve{u}' \quad \breve{v}' \quad \breve{w}'\right]^T = \mathbf{AE}\mathrm{G}_{\mathrm{i}}\left[X \quad Y \quad Z \quad 1\right]^T. \tag{31}$$

The derivatives with respect to the image coordinates are calculated by:

$$\left[u' \quad v'\right]^T = \left[\breve{u}'/\breve{w} - \breve{u}\breve{w}'/\breve{w}^2 \quad \breve{v}'/\breve{w} - \breve{v}\breve{w}'/\breve{w}^2\right]^T. \tag{32}$$

Motion Optimization

The distances measured between the projection of the 3D model the line segments are projected to the 6D subspace spanned by the transformation vectors. The solution to the 6D subspace transformation is obtained by minimizing the robust error term.

The standard least squares algorithm considers the residual values as the only evidence for the optimization. A few large residual values with quadratic penalty will make the parameter estimation far from the real solution. Unfortunately, such large residual values are due to wrong positions or correspondences. These are usually outliers that cannot be fitted into the transformation model. This problem can be addressed by adopting a robust error function. Instead of using the simple quadratic penalty term, we consider the residuals according to their values. A smaller penalty will be given to the residuals with large values. We use Huber cost function. The function in (22) is given by:

$$\begin{aligned} &\psi(\delta) = \delta^2, for\, \delta < b \\ &\psi(\delta) = 2b|\delta| - b^2, otherwise, \end{aligned} \tag{33}$$

where, the value b gives the range of δ for the approximation. The function is a hybrid between the $L1$ and least squares cost function.

Experimental Results

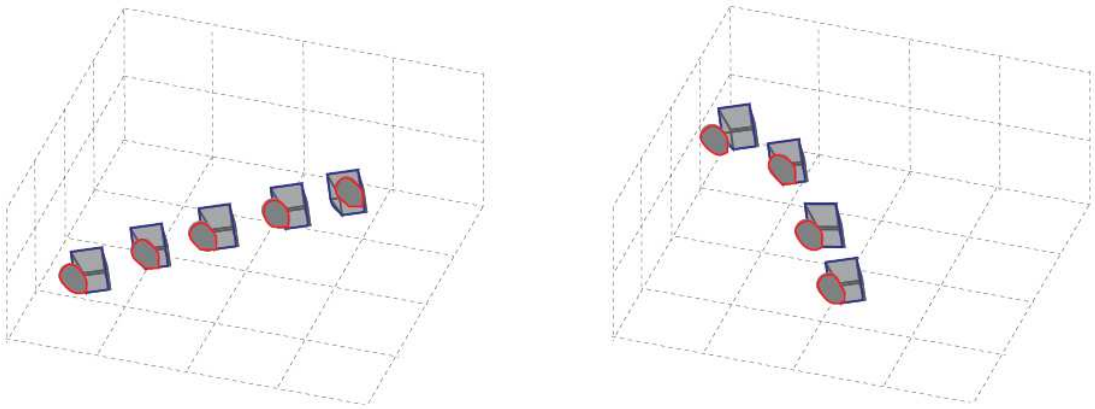
We implemented the proposed system and perform experiments. The camera is moved in different paths. In some of the experiments, the camera is mounted on a robot arm. The end effector on which a camera is mounted has a coordinate setting which is different from the camera coordinate system. The relationship of the two coordinate settings is calibrated by using the projection of calibration board in the images. A point p_c in the end effector coordinate is transformed to a point p_c in the camera coordinate by 4 transformation matrices: $\mathbf{p}_c = \mathbf{E}_c\, \mathbf{Q}_{bc}\, \mathbf{Q}_{eb}\, \mathbf{m}_e$ where E_c is the camera projection matrix composed of internal parameter matrix and external parameter matrix; Q_{bc} is the matrix describing the transformation from the camera coordinate system to the robot arm base coordinate system; Q_{eb} is the matrix describing the transformation from the robot arm base to its end effectors.

We use a calibration board to calibration the system. Several images of the calibration board are captured by the camera. The calibration board has same movement with the end effector which it is mounted on. The camera internal parameters are calibrated using the method proposed by Zhang (2000). We detect the points of the corners in the images. The correspondences between the image points and the points in the 3D space are utilized for calibrating the two transformations T_{ce} and T_{eb}. The transformation matrix can be converted into transformation vectors, which makes the representation compact. The transformation vector consists of 3 translation elements and 3 rotation elements.

The two transformations (T_{ce} and T_{eb}) are solved by using the interior point algorithm. There are noises in the images which lead to drift of the transformations. We calculate the distances between the image points and the fitted positions. The correspondences with large errors are discarded. Then, we run the optimization algorithm again to get the precise transformation parameters. We obtain the ground truth using a laser tracker measuring several points on the object. The pose is calculated by fitting these points. The results of the experiments are demonstrated in Fig. 4. The tracking accuracy is given in Table 1. The measured accuracy includes the translation and rotation errors. The translation errors are measured in millimeter and the rotation errors are measured in degrees. The accuracy given in Table 1 is the average error in a sequence.

Table 1: Results of experiments measuring accuracy of the proposed pose tracking method. (Trans in mm and Rot in degree)

TransX	TransY	TransZ	RotX	RotY	RotZ
0.13	0.16	0.56	0.032	0.019	0.035
0.23	0.18	0.47	0.033	0.021	0.029
0.20	0.21	0.76	0.043	0.015	0.036
0.15	0.19	0.49	0.029	0.030	0.045
0.26	0.20	0.36	0.041	0.027	0.019
0.21	0.28	0.58	0.043	0.015	0.036



(a) (b)

Fig. 4: Motion estimation results. The object in blue indicates the estimated pose and the object in red indicates the ground truth. The estimation error has been enlarged to make the difference can be seen

Conclusion

We presented a practical framework for 6*D* pose tracking. Our system finds applications in augmented reality and robotic applications. The line segment detector considers semi-global information for line segment parameter estimation. It is efficient to detect sufficient number of lines for matching. The detector is good at discarding ambiguous line segments while finds real ones in noisy situations. Line segments sets foundation for matching and pose tracking. We consider 3D frame transformations using Lie groups and the corresponding Lie algebras. The motion parameter estimation is performed in a linear space, thanks to the good properties of Lie algebras.

Funding Information

This work was partially supported by the project “Large Scale Aircraft Digital Assembly Techniques”, MIZ-2015-G-100.

Author Contributions

Junqiu Wang: Designed this research, implemented the tracking system, performed the experiments, wrote the paper.

Jingjing Cui, Yanjie Gao, Yanxuan Ma, Zhichao Gan and Chao Yang: Performed the experiments, wrote the paper.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Barinova, O., V.S. Lempitsky and P. Kohli, 2012. On detection of multiple object instances using hough transforms. IEEE Trans. Patt. Anal. Mach. Intell., 34: 1773-1784. DOI: 10.1109/TPAMI.2012.79

Burns, J.B., A.R. Hanson and E.M. Riseman, 1986. Extracting straight lines. IEEE Trans. Patt. Anal. Mach. Intell., 8: 425-455. DOI: 10.1109/TPAMI.1986.4767808

Davison, A.J., 2003. Real-time simultaneous localisation and mapping with a single camera. Proceedings of the IEEE International Conference on Computer Vision, Oct. 13-16, IEEE Xplore Press, Nice, pp: 1403-1410. DOI: 10.1109/ICCV.2003.1238654

DeMenthon, D. and L. Davis, 1995. Model-based object pose in 25 lines of code. Int. J. Comput. Vis., 15: 123-141. DOI: 10.1007/BF01450852

Desolneux, A., L. Moisan and JM. Morel, 2000. Meaningful alignments. Int. J. Comput. Vis., 40: 7-23. DOI: 10.1023/A:1026593302236

Drummond, T. and R. Cipolla, 2002. Real-time visual tracking of complex structures. IEEE Trans. Patt. Anal. Mach. Intell., 24: 932-946. DOI: 10.1109/TPAMI.2002.1017620

Flint, A., C. Mei, I. Reid and D.W. Murray, 2011. Manhattan scene understanding using monocular, stereo and 3D features. Proceedings of the International Conference on Computer Vision, Nov. 6-13, IEEE Xplore Press, Barcelona, pp: 2228-2235. DOI: 10.1109/ICCV.2011.6126501

Gastal, E. and M. Oliveira, 2012. Adaptive manifolds for real-time highdimensional filtering. ACM Trans. Graph., 31: 33:1-33:13. DOI: 10.1145/2185520.2335384

Gioi, R., J. Jakubowicz, J.M. Morel and G. Randall, 2010. LSD: A fast line segment detector with a false detection control. IEEE Trans. Patt. Anal. Mach. Intell., 32: 722-732. DOI: 10.1109/TPAMI.2008.300

Hough, P., 1962. Method and means for recognizing complex patterns. U.S. Patent 3,069,654.

Isard, M. and A. Blake, 1998. CONDENSATIOND: Conditional density propagation for visual tracking. Int. J. Comput. Vis., 29: 5-28. DOI: 10.1023/A:1008078328650

Kim, H., S. Leutenegger and A. Davison, 2016. Real-time 3D reconstruction and 6-DoF tracking with an event camera. Proceedings of the European Conference on Computer Vision, (CCV' 16), Springer Press, Amsterdam, pp: 349-364. DOI: 10.1007/978-3-319-46466-4_21

Koolwal, A., F. Barbagli, C. Carlson and D. Liang, 2010. An ultrasound-based localization algorithm for catheter ablation guidance in the left atrium. Int. J. Robot. Res., 29: 643-665. DOI: 10.1177/0278364909105332

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis., 60: 91-110. DOI: 10.1023/B:VISI.0000029664.99615.94

Mikolajczyk, K., T. Tuytelaars, C. Schmid, A. Zisserman and J. Matas *et al*., 2005. A comparison of affine region detectors. Int. J. Comput. Vis., 65: 43-72. DOI: 10.1007/s11263-005-3848-x

Rosten, E. and T. Drummond, 2005. Fusing points and lines for high performance tracking. Proceedings of the 10th IEEE International Conference on Computer Vision, Oct. 17-21, IEEE Xplore Press, Beijing, pp: 1508-1515. DOI: 10.1109/ICCV.2005.104

Sattinger, D. and O. Weaver, 1998. Lie groups and algebras with applications to physics, geometry and mechanics. Springer-Verlag.

Talib, H., M. Styner, T. Rudolph and G. Ballester, 2007. Dynamic registration using ultrasound for anatomical referencing. Proceedings of the 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Apr. 12-15, IEEE Xplore Press, Arlington, VA, USA, pp: 1164-1167. DOI: 10.1109/ISBI.2007.357064

Taylor, C. and D. Kriegman, 1994. Minimization on the lie group SO(3) and related manifolds. Technical Report 9405, Yale Univ.

Wang, J. and Y. Yagi, 2008. Integrating color and shapetexture features for adaptive real-time object tracking. IEEE Trans. Image Process., 17: 235-240. DOI: 10.1109/TIP.2007.914150

Wunsch, P. and G. Hirzinger, 1997. Real-time visual tracking of 3D objects with dynamic handling of occlusion. Proceedings of the IEEE International Conference on Robotics and Automation, Apr. 25-25, IEEE Xplore Press, Albuquerque, New Mexico, pp: 2868-2873. DOI: 10.1109/ROBOT.1997.606722

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Trans. Patt. Anal. Mach. Intell., 22: 1330-1334. DOI: 10.1109/34.888718