Monocular Visual Odometry Based on Trifocal Tensor Constraint

Y J Chen, G L Yang, Y X Jiang and X Y Liu
Science and Technology on Inertial Lab, Beihang University, 37Xueyuan Road, Beijing, China
cyj_tju@163.com

Abstract. For the problem of real-time precise localization in the urban street, a monocular visual odometry based on Extend Kalman fusion of optical-flow tracking and trifocal tensor constraint is proposed. To diminish the influence of moving object, such as pedestrian, we estimate the motion of the camera by extracting the features on the ground, which improves the robustness of the system. The observation equation based on trifocal tensor constraint is derived, which can form the Kalman filter alone with the state transition equation. An Extend Kalman filter is employed to cope with the nonlinear system. Experimental results demonstrate that, compares with Yu’s 2-step EKF method, the algorithm is more accurate which meets the needs of real-time accurate localization in cities.

1. Introduction
Visual odometry (VO) and real-time monocular Simultaneous Localization and Mapping (SLAM) have become increasingly popular research topic [1]. They estimates the motion of the object relying on the input information of the camera. With the further development of image processing techniques, these algorithms have been gradually applied to robotics, unmanned aerial vehicles, and virtual reality applications [1].

Visual odometry can roughly be divided into monocular odometry and stereo odometry, or into feature-based method and direct method according to the algorithm [2]. Nister et al. [3] who proposed sparse methods for estimating the small movement of monocular by sequential frame-to-frame matching. More recently, Forster et al. [4] proposed semi-direct visual odometry (SVO), a real-time capable, direct method that be applied to micro-aerial-vehicle. They deducted the Gaussian mixture distribution model of the depth and reconstruct the 3D scene. The most recent Kalman filter based methods are by Yu et al [5, 6], the full covariance extended Kalman filter (EKF) is decoupled such that the computation efficiency is increased as a tradeoff in accuracy.

In this paper, we proposed a monocular visual odometry based on trifocal tensor constraint [7] which is real-time capable on CPUs. Our method tracks the interest points with Lucas-Kanade optical-flow [8] and form Kalman filter model using trifocal tensor constraint. It concurrently estimates the rigid-body motion through Extend Kalman filter. The proposed approach can improve the precision of the optical-flow and enhance system robust.

2. Preliminaries

2.1. Camera model
One point $P_w = (x_w, y_w, z_w)^T \in S$ in the world coordinates $S \in \mathbb{R}^3$ maps to the image coordinates $P_w = (u, v)^T \in \Pi$ through the projection model as follows, where $\Pi \in \mathbb{R}^2$ is the image domain.

$$\tilde{P}_w = P \tilde{P}_w = K[R \ t] \tilde{P}_w$$

(1)

Where $\tilde{P}_w$ is in homogeneous coordinates and $K$ denotes the camera intrinsic parameters which can be gained from calibration. The motion of the camera is expressed with its rotation matrix $R$ and translation vector $t$. $P$ denotes the camera matrix.

2.2. Trifocal tensor constraint

The trifocal tensor to the three views is similar to the fundamental matrix to the two views. It describes the imaging geometric relationship which is independent of scene structure among three views. The constraint allows us to estimate the motion of the camera without recovering the 3D points from the world coordinates which can avoid the error from triangularization.

For the sake of simplicity, we assume that we get consecutive three images from one camera. A point $X$ from world coordinates map to the three images $X_A$, $X_B$, $X_C$ and the corresponding transformation matrix are denoted as $T_A$, $T_B$, $T_C$, respectively. And the camera matrixes of the three views are as follows:

$$P_A = KT_A$$

(2)

$$P_B = KT_B$$

(3)

$$P_C = KT_C$$

(4)

And the corresponding trifocal tensors are indicated below.

$$T_i^{jk} = (-1)^{i+1} \cdot \det \begin{pmatrix} \sim a^i \\ b^q \\ c^k \end{pmatrix}$$

(5)

Where $\sim a^i$ denotes $P_A$ without $i^{th}$ row, $b^q$ denotes $q^{th}$ row of $P_B$, $c^k$ denotes $k^{th}$ row of $P_C$.

After calculating the tensor matrix, we can track the corresponding feature points in the third image through the matched feature points from first and second image. As shown in figure 1, we construct a line $l_{B,j}$ going through a feature point $X_B$ in second image and project this line on the scene surface to form a plane which construct a planar scene between first and second image. So we can track the $X_C$ by estimating a homography.

$$X_C^l = X_A^l \cdot l_{B,j} \cdot T_i^{jk}$$

(6)

Figure 1. The feature correspondences in three views construct the trifocal tensor constraint.

3. Monocular visual odometry based on trifocal tensor constraint
The algorithm consists of three major modules: initialization, trifocal tensor tracking, and motion estimation based on EKF as visualized in Figure 2.

3.1. Initialization

We define the notation as follows. The image collected at timestep $t$ is denoted with $f_t$, the rate of frame is $\Delta t$. The feature correspondences of $f_1, f_2, \text{and} f_t$ is denoted as $x_i, x'_i, x''_i$ $(i = 1, 2, ..., N)$.

Because of scale uncertainty, monocular visual odometry needs initialization and to be given sufficient translational camera movement between the first and second frame [1, 9]. In practice, we estimate the motion using the camera pointing to the ground, which can calculate the initial pose by homography. Based on plane hypothesis, we can recognize the points on the ground by calculating the primary homography and measure the inliers by RANSAC [10].

Firstly, we extract the FAST corner features from frame $f_1$ and track the feature correspondences by L-K optical flow in the next frame. As shown in Figure 3, the pedestrians and cars are general at the top of the images since ground accounts for the main part of the images.

![Figure 2](image_url)

**Figure 2.** This figure provides an overview of the whole algorithm. The algorithm can be divided into three modules: initialization, trifocal tensor tracking, and motion estimation based on EKF.

![Figure 3](image_url)

**Figure 3.** This figure shows the result of the optical flow, which illustrates the features extracted from the ground are detection of optical flow points on the ground.

Based on these assumptions, we choose the four groups of correspondences at the bottom of the image as inliers and calculate the homography between these correspondences which denotes as $H$. Then we calculate the re-projection error for the rest of the points selected at random.

$$E(x_i, x'_i, H) = \|x'_i - Hx_i\|$$

(7)

The point is considered an outlier if the error exceeds the threshold we set before and deleted, which means that all the points used to estimate the motion of the camera are inliers. The initial guess
of the \(f_2\)'s pose is required through homography matrix decomposition and refined in the process of EKF.

### 3.2. Trifocal tensor tracking

After tracking the current frame, the frame \(f_1, f_2\) and current frame compose the three views which are used to construct trifocal tensor constraint. The detailed trifocal tensor constraint can refer to [9]. The number of the feature correspondences tracked may decrease with the angle view change, so we need to restart EKF when the number of the features is under a minimum size. In practice, the process followed is that the tracking algorithm will restart with frames \(f_{t-1}\) and \(f_t\). We will recalculate the feature correspondences. And then we reset \(t\) to 2. The new EKF process is then activated by frame \(f_{t+1}\). The roll-back method keep the same scale among the whole algorithm.

### 3.3. Motion estimation based on EKF

The model parameters to be estimated are given:

\[
y_k = \begin{bmatrix} t_{x_k} & i_{x_k} & i_{y_k} & i_{z_k} & \alpha_k & \beta_k & \gamma_k & t_{x_2} & t_{y_2} & t_{z_2} & \alpha_2 & \beta_2 & \gamma_2 \end{bmatrix}
\]

(8)

Where \(t_{x_k}, i_{x_k}, i_{y_k}, i_{z_k}\) are the displacement alone x, y, z axis relative to the world coordinates, respectively. \(t_{x_2}, i_{x_2}, i_{y_2}, i_{z_2}\) are corresponding velocity. \(\alpha_k, \beta_k, \gamma_k\) are yaw, pitch and roll of the current frame, respectively. And \(\dot{\alpha}_k, \dot{\beta}_k, \dot{\gamma}_k\) describe corresponding angular velocity.

The state transition equation and measurement equation are as follows:

\[
y_{k+1} = f(y_k) + w_k
\]

(9)

\[
z_{k+1} = h(y_{k+1}) + v_{k+1}
\]

(10)

The motion of the camera is complex, resulting in that \(f(\cdot)\) is nonlinear state transition function. \(h(\cdot)\) is nonlinear measurement function constructed by trifocal tensor constraint.

\[
z_{k+1} = \begin{bmatrix} u_{1,r} & v_{1,r} & u_{2,r} & v_{2,r} & ... & u_{N,r} & v_{N,r} \end{bmatrix}^T
\]

denotes 2N-dimensional measurement vector, which describes feature points in the current frame. \(w_k \sim N(0, Q_k)\) denotes system error, \(v_{k+1} \sim N(0, R_{k+1})\) denotes observation noise and \(N\) is the number of the feature correspondences.

To solve this model, the equation should linearization. In general, the motion of the camera is regarded as uniform motion with same speed and angular velocity based on the assumption of the high rate of frame. The initial guess of frame \(f_2\)'s pose is close to the true value. So the state equation can be given:

\[
y_{k+1} = Ay_k + w_k
\]

(11)

\[
A = \text{diag}\left( \begin{bmatrix} 1 & \Delta t & 1 & \Delta t & 1 & \Delta t \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \right)
\]

(12)

The measurement equation is general linearized by Taylor expansion and calculating the Jacobian matrix, which is called extend Kalman filter.

For the measurement equation \(h(y_{k+1})\):

\[
x'_i = h(T_{1, i_2}, x_i)
\]

(13)

According to the above analysis, the state variable updating procedure is shown as follows:

\[
\hat{w}_{k+1,k} = A\hat{w}_{k,k}
\]

(14)

\[
P_{k+1,k} = AP_{k,k}A' + Q_k
\]

(15)
\[
\dot{\hat{w}}_{k+1|k|} = \hat{w}_{k+1|k} + K(z_{k+1} - h(\hat{w}_{k+1|k})) \tag{16}
\]
\[
P_{k+1|k+1} = P_{k+1|k} - KJ_{y}(\hat{w}_{k+1|k})^T
\]
\[
K = P_{k+1|k} J_{y}(\hat{w}_{k+1|k})^T + R_{k+1}^{-1} \tag{18}
\]

Where \( \nabla J_{y} \) is the Jacobian matrix of measurement equation at \( \hat{w}_{k+1|k} \).

4. Experiment and result

Experiments were performed on videos recorded from a downward-looking camera attached to an electromobile in the school. Differential GPS is used to provide the ground-truth with 10Hz sampling frequency and 0.01m location error. The detailed parameters configuration is shown in table 1.

| Table 1. Experiment configuration. |
|-----------------------------------|
| Camera   | Image resolution | Frame-rate |
|----------|------------------|------------|
| bumblebee2 | 640×480         | 20Hz       |

4.1. Accuracy

We evaluate the accuracy on a video mentioned before and the result is illustrated in figure 4. The ground-truth for the trajectory originates from a differential GPS. The trajectory is 350 meters long and the car drives on average 7.56km/h. We contrast our approach with Yu’s 2-step EKF algorithm and generate the plots. Figure 5 illustrates the position error over frame.

![Figure 4. Comparison against the groundtruth.](image)

![Figure 5. Position drift of the proposed method and comparison against Yu’s method.](image)

Overall, the proposed approach is more accurate than Yu’s approach. We suspect the main reason is that our method combine the Kalman filter with trifocal tensor constraint without calculating the scene structure which eliminate the influence of triangularization and improve the system robustness. And in real scene, recognizing the ground reduce the influence of moving object, which increase the accuracy as well.

4.2. Running time

Table 2 show the running time required to operate the algorithm on the CPUs. Similarly, we compare the performance of proposed algorithm with Yu’s 2-step EKF. From the table, we can see Yu’s...
method is faster than our method. The main reason for that is that trifocal tensor constraint need to run at every frame.

**Table 2. Running time comparison.**

| Proposed approach | Yu’s 2-step EKF |
|-------------------|----------------|
| Running time (in second) | 1.08 | 0.52 |

5. Conclusion

A monocular visual odometry based on trifocal tensor constraint has been proposed in this paper. It has a great improvement on the accuracy due to the combination of EKF and trifocal tensor constraint. The gain in accuracy is due to the fact that the scene structure doesn’t need to be calculate which eliminate the influence of triangularization and improve the system robustness. The algorithm can be used in real-time precise localization in the urban street as it is capable of high frame-rate motion estimation and provide robust output combined with RANSAC.

Acknowledgments

The work described in this paper was supported by a grant from Science and Technology on Inertial Lab in Beihang University.

References

[1] Engel, J., Schöps, T., & Cremers, D. (2014). Lsd-slam: large-scale direct monocular slam. , 8690, 834-849.
[2] Nister, D., Naroditsky, O., & Bergen, J. (2004). Visual odometry. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on (Vol.1, pp.I-652-I-659 Vol.1). IEEE.
[3] Royer, E., Lhuillier, M., Dhome, M., & Lavest, J. M. (2007). Monocular vision for mobile robot localization and autonomous navigation. International Journal of Computer Vision, 74(3), 237-260.
[4] Forster, C., Pizzoli, M., & Scaramuzza, D. (2014). SVO: Fast semi-direct monocular visual odometry. IEEE International Conference on Robotics and Automation (pp.15-22). IEEE.
[5] Yu, Y. K., Wong, K. H., & Chang, M. Y. (2008). A fast and robust simultaneous pose tracking and structure recovery algorithm for augmented reality applications. International Conference on Image Processing (Vol.2, pp.1029-1032 Vol.2). IEEE.
[6] Yu, Y. K., Wong, K. H., Ming, M., Chang, Y., & Or, S. H. (2006). Computing pose sequences directly from videos. 2773-2776.
[7] Hartley, R., & Zisserman, A. (2000). Multiple view geometry in computer vision. (Vol.30, pp.1865 - 1872). Cambridge University Press.
[8] Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. International Joint Conference on Artificial Intelligence (Vol.73, pp.674-679). Morgan Kaufmann Publishers Inc.
[9] Scaramuzza, D., & Fraundorfer, F. (2011). Visual odometry: part i: the first 30 years and fundamentals. IEEE Robotics & Automation Magazine.
[10] Watanabe, T., Kamai, T., & Ishimaru, T. (2015). Robust estimation for camera homography by fuzzy RANSAC algorithm with reinforcement learning. International Symposium on Soft Computing and Intelligent Systems (Vol.19, pp.712-717). IEEE.