Vision-Aided Inertial Navigation System with Point and Vertical Line Observations for Land Vehicle Applications

Zhenbo Liu, Qifan Zhou, Yongyuan Qin and Naser El-Sheimy

Abstract Various aiding sensors can be integrated with the inertial navigation system (INS) to reduce its error growth when the vehicle is operating in GNSS denied environments. This paper developed a method to use the vanishing point from vertical line observations of building blocks in order to further improve point-based visual-inertial navigation system (VINS) for land vehicle applications. First, we presented the formulations of tightly coupled point-based VINS based on the Multi-State Constraint Kalman Filter (MSCKF) in the local-level frame. Second, we developed the relationship between the INS roll angle and vanishing point coordinates from vertical line observations. The roll angle measurement model is formulated. Finally, loosely coupled vertical line aiding module is added to the existing VINS, and the integration scheme is presented. Real world experiments demonstrated the validity of the mixed VINS method and the improved accuracy of the attitude and position estimation when compared with the solution without vertical line vanishing point aiding.

Keywords Visual-inertial navigation system · Vanishing point · Roll angle · MSCKF · Localization

1 Introduction

In the last decade there has been an increasing use of Autonomous Ground Vehicles (AGVs) in many navigation applications. One of the representative civil application is in intelligent transport systems (ITSs); more specifically in autonomous cars or driverless cars [1]. In the vehicle's Guidance, Navigation, and Control (GNC) system, the navigation state estimation is an essential component, not only

Z. Liu (✉) · Q. Zhou · N. El-Sheimy
Department of Geomatics Engineering, University of Calgary, Calgary, Canada
e-mail: zhenbo.liu2@ucalgary.ca

Z. Liu · Y. Qin
School of Automation, Northwestern Polytechnical University, Xi'an, China

J. Sun et al. (eds.), *China Satellite Navigation Conference (CSNC) 2017 Proceedings: Volume II*, Lecture Notes in Electrical Engineering 438,
DOI 10.1007/978-981-10-4591-2_36

for the vehicle's autonomous navigation, but also for cooperative safety communications among vehicles [2–4]. Among different navigation systems, the inertial navigation system (INS) is the only one that has the capability to produce a complete and continuous set of navigation state data with high precision during a short time span. However, low-cost INS has to be integrated with other aiding sensors to reduce its error growth. INS and Global Navigation Satellite System (GNSS) integration is commonly used for vehicle outdoor navigation. However, GNSS signal can suffer from obstruction and multipath errors in city centers and mountainous regions, and is prone to the possibility of being jammed or spoofed [5].

To mitigate the error growth of INS in GNSS denied environment, aiding sensors such as wheel odometers and motion constraints (e.g., Non-holonomic Constraints (NHC) as a "virtual" sensor) were utilized for land vehicles [6, 7]. Recently, researchers proposed to use vision sensor aiding in the navigation system, in loosely coupled form, with the visual odometry (VO) module [8, 9], or in tightly coupled form [10–13]. Most research used point features in the tightly coupled VINS, including simultaneous localization and mapping (SLAM) based methods [10, 14] and structure-less methods, which focus on the ego-motion only [12, 15]. The representative algorithm of filter based methods is called multi-state constraint Kalman filter (MSCKF) [12], which augments camera poses (a sliding window of poses) into the state vector to make most use of measurements of feature points in the geometry constraint. Apart from the point-based VINS, there are approaches using the vanishing point (VP) module [16, 17], another attitude aiding source for INS working in the environment with structured buildings or in corridors. These methods rely on the observation of parallel lines in structured buildings.

In this paper, we present MSCKF in the local-level frame (LLF), which is common to the navigation community. To improve the performance of point-based MSCKF, we propose to add a VP module, which observes vertical lines of building blocks, for land vehicle navigation. Based on that, a roll angle measurement model is derived. Therefore, a mixed VINS using points and line observations is developed, and its performance is verified by real data experiments.

2 MSCKF Algorithm in Local-Level Frame

MSCKF is considered as a structure-less visual-inertial navigation system, which means the vehicle localization is of the main interest, rather than recovering the structure of the environment. They take into account all camera poses at which observations of the same 3D point occurred. MSCKF augments camera poses (a sliding window of poses) into the state vector to make most use of measurements of feature points in the geometry constraint. Every feature has a corresponding stack of misclosure vectors (residuals). Feature position error states are cancelled out by projecting the residual stack on the left null space of the design matrix corresponding to the feature error state [12]. Here we present the MSCKF algorithm in LLF with the notations common to navigation engineers and researchers, in order to bridge between the navigation community and robotics community. It should be mentioned that some components in the filtering formulations should be modified with respect to original MSCKF.

2.1 MSCKF States and Covariance

For the near-Earth navigation, the navigation frame is usually the local-level frame (n-frame; east-north-up is used in this paper). The navigation state is presented by the position (latitude L, longitude λ and height h), velocity (east velocity, north velocity, and up velocity), and attitude (pitch, roll, and azimuth). Differential equations of position $\boldsymbol{P}$, velocity $\boldsymbol{V}^n$ and attitude matrix $\boldsymbol{C}_b^n$ are given by

$$\dot{\boldsymbol{P}} = \boldsymbol{D}^{-1}\boldsymbol{V}^n = \begin{bmatrix} 0 & \frac{1}{R_M+h} & 0 \\ \frac{1}{(R_N+h)\cos L} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{V}^n \tag{2.1}$$

$$\dot{\boldsymbol{V}}^n = \boldsymbol{f}^n - \left(2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\right) \times \boldsymbol{V}^n + \boldsymbol{g}^n \tag{2.2}$$

$$\dot{\boldsymbol{C}}_b^n = \boldsymbol{C}_b^n\left[\boldsymbol{\omega}_{ib}^b \times\right] - \left[\left(\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n\right)\times\right]\boldsymbol{C}_b^n \tag{2.3}$$

where R_M and R_N are meridian radius of curvature and prime vertical radius respectively, $\boldsymbol{f}^n$ is the specific force in n-frame, $\boldsymbol{\omega}_{ie}^n$ is the earth rotation angular rate vector, $\boldsymbol{\omega}_{en}^n$ is the angular rate of the navigation frame with respect to (w.r.t.) ECEF frame, and $\boldsymbol{\omega}_{ib}^b$ is the angular rate of the body frame (b) w.r.t. the inertial frame (i). The notation $(\boldsymbol{a})\times$ is the asymmetric matrix of a vector $\boldsymbol{a}$. The navigation states are propagated using Eqs. (2.1–2.3), which is called INS mechanization where the IMU measurement is the input signal.

The navigation states can be considered as the full states, and the associated errors are called error states. For convenience, let $\boldsymbol{X}$ denote the error states, and the full states will be presented by their quantities notions. The state vector of the MSCKF is:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_I^T & \boldsymbol{X}_{C_1}^T & \boldsymbol{X}_{C_2}^T & \cdots & \boldsymbol{X}_{C_m}^T \end{bmatrix}^T \tag{2.4}$$

where $\boldsymbol{X}_I$ is the current INS error states, and $\boldsymbol{X}_{C_i}, i = 1\ldots m$ are camera poses errors at the times the last m images were recorded. INS error states are defined as

$$\boldsymbol{X}_I = \begin{bmatrix} \delta\boldsymbol{V}^{nT} & \boldsymbol{\phi}^{nT} & \delta\boldsymbol{P}^T & \boldsymbol{b}_g^T & \boldsymbol{b}_a^T \end{bmatrix}^T \tag{2.5}$$

where $\delta\boldsymbol{V}^n$, $\boldsymbol{\phi}^n$, and $\delta\boldsymbol{P}$ are the velocity error, misalignment error, and position error (latitude, longitude and height error). $\boldsymbol{b}_g$ and $\boldsymbol{b}_a$ are biases of gyroscopes and accelerometers. In this case, the dimension of the state vector is 15 + 6 m. Misalignment error $\boldsymbol{\phi}^n$ (considered as the small angle) is defined by

$$\hat{\boldsymbol{C}}_b^n = [\boldsymbol{I} - \boldsymbol{\phi}^n \times]\boldsymbol{C}_b^n \tag{2.6}$$

For EKF, the INS error state propagation is based on the INS error dynamic model. The IMU sensor biases are modelled as random constants, random walks or Gauss-Markov processes. The *i*th camera pose errors are defined as:

$$\boldsymbol{X}_{C_i} = \begin{bmatrix} \boldsymbol{\phi}_{C_i}^{nT} & \delta\boldsymbol{P}_{C_i}^T \end{bmatrix}^T \tag{2.7}$$

where $\boldsymbol{\phi}_{C_i}^n$ and $\delta\boldsymbol{P}_{C_i}$ are the camera's attitude error and position error at the *i*th recorded instance. The augmented *i*th camera pose errors are modeled as random constant, since the previous camera poses will not change over time.

The camera pose is augmented into the state when recording a new image. It is calculated from current IMU pose and the known relative pose between IMU and the camera using Eqs. 2.8–2.9.

$$\hat{\boldsymbol{C}}_C^n = \hat{\boldsymbol{C}}_b^n \boldsymbol{C}_C^b \tag{2.8}$$

$$\hat{\boldsymbol{P}}_C = \hat{\boldsymbol{P}} + \boldsymbol{D}^{-1} \hat{\boldsymbol{C}}_b^n \boldsymbol{P}_{bC}^b \tag{2.9}$$

where $\boldsymbol{C}_C^b$ and $\boldsymbol{P}_{bC}^b$ are the rotation matrix and lever-arm between the camera and the IMU; $\hat{\boldsymbol{P}}_C$ and $\hat{\boldsymbol{C}}_c^n$ are the position and attitude matrix of the camera. Then the camera pose estimate is appended to the state vector, and the covariance matrix is augmented accordingly:

$$\boldsymbol{X}_{(15+6(m+1))} \leftarrow \left[\boldsymbol{X}_{(15+6m)\times 1}; \boldsymbol{X}_{C(6\times 1)}\right]^T \tag{2.10}$$

$$\boldsymbol{P}_{k|k} \leftarrow \begin{bmatrix} \boldsymbol{I}_{15+6m} \\ \boldsymbol{J}_{6\times(15+6m)} \end{bmatrix} \boldsymbol{P}_{k|k} \begin{bmatrix} \boldsymbol{I}_{15+6m} \\ \boldsymbol{J}_{6\times(15+6m)} \end{bmatrix}^T \tag{2.11}$$

where

$$\boldsymbol{J}_{6\times(15+6m)} = \begin{bmatrix} \boldsymbol{O}_{3\times 3} & \boldsymbol{I}_{3\times 3} & \boldsymbol{O}_{3\times 3} & \boldsymbol{O}_{3\times 6} & \boldsymbol{O}_{3\times 6m} \\ \boldsymbol{O}_{3\times 3} & \boldsymbol{D}^{-1}\left(\widehat{\boldsymbol{C}}_b^n \boldsymbol{P}_{bC}^b\right)\times & \boldsymbol{I}_{3\times 3} & \boldsymbol{O}_{3\times 6} & \boldsymbol{O}_{3\times 6m} \end{bmatrix}$$

2.2 MSCKF Procedures

Assuming that the *i*th feature, which has been observed in a set of M_i images, is no longer detected. An EKF measurement update will be triggered after forming a residual corresponding to this ith feature. Suppose we have the measurement $\tilde{z}_i^j$ for $j = 1\ldots M_i$. The predicted measurement of the ith feature in jth image is

$$\hat{z}_i^j = h\left(\widehat{\boldsymbol{P}}_{C_j}, \widehat{\boldsymbol{C}}_{C_j}^n, \widehat{\boldsymbol{P}}_{f_i}\right) \tag{2.12}$$

where $h(\cdot)$ is the nonlinear camera projection equation, and $\widehat{\boldsymbol{P}}_{f_i}$ is the computed feature position. So the residuals of ith feature are

$$\delta z_i^j = \hat{z}_i^j - \tilde{z}_i^j \approx \boldsymbol{H}_{X_i}^j \boldsymbol{X} + \boldsymbol{H}_{f_i}^j \delta \boldsymbol{P}_{f_i} + n_i^j \tag{2.13}$$

where the Jacobian matrix is

$$\boldsymbol{H}_{X_i}^j = \begin{bmatrix} \boldsymbol{O}_{2\times15} & \boldsymbol{O}_{2\times6} & \cdots & \boldsymbol{J}_i^j\left(\widehat{\boldsymbol{P}}_{f_i}^{C_j}\times\right)\widehat{\boldsymbol{C}}_n^{c_j} & -\boldsymbol{J}_i^j\widehat{\boldsymbol{C}}_n^{c_j}\boldsymbol{D} & \cdots & \boldsymbol{O}_{2\times6} \end{bmatrix}_{2\times(15+6\mathrm{m})}$$

$$\boldsymbol{H}_{f_i}^j = \boldsymbol{J}_i^j \widehat{\boldsymbol{C}}_n^{c_j} \boldsymbol{D}$$

where

$$\boldsymbol{J}_i^j = \frac{1}{\widehat{Z}_i^{C_j}} \begin{bmatrix} 1 & 0 & -\frac{\widehat{X}_i^{C_j}}{\widehat{Z}_i^{C_j}} \\ 0 & 1 & -\frac{\widehat{Y}_i^{C_j}}{\widehat{Z}_i^{C_j}} \end{bmatrix}, \quad \begin{bmatrix} \widehat{X}_i^{C_j} \\ \widehat{Y}_i^{C_j} \\ \widehat{Z}_i^{C_j} \end{bmatrix} = \widehat{\boldsymbol{P}}_{f_i}^{C_j} = \widehat{\boldsymbol{C}}_n^{C_j}\boldsymbol{D}\left(\widehat{\boldsymbol{P}}_{f_i} - \widehat{\boldsymbol{P}}_{C_j}\right)$$

Stack the δz_i^j of all M_i measurements of the ith feature to form the block residual vector:

$$\delta z_i = \boldsymbol{H}_{X_i}\boldsymbol{X} + \boldsymbol{H}_{f_i}\delta\boldsymbol{P}_{f_i} + \boldsymbol{n}_i \tag{2.14}$$

To remove the effect of feature position error, define a residual vector $\delta z_i^O = \boldsymbol{A}^{\mathrm{T}}\delta z_i$, where $\boldsymbol{A}$ is a matrix whose columns form the basis of the left null space of $\boldsymbol{H}_{f_i}$. The new residual will be

$$\delta z_i^O = \boldsymbol{A}^{\mathrm{T}}\delta z_i = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{H}_{X_i}\boldsymbol{X} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{n}_i = \boldsymbol{H}_i^O\boldsymbol{X} + \boldsymbol{n}_i^O \tag{2.15}$$

After the measurement update in the Kalman filter, do the feedback corrections to the navigation states and camera poses.

The MSCKF can be implemented and operated in different ways [18]. For example, minimum feature track lengths can be set to trigger an update. Large minimum feature track length means that the landmarks will be observed with a larger base line, while some near landmarks will be neglected. Potentially, large minimum feature track may have better accuracy than the small minimum feature track, but the former will also have the problem of losing useful information, which is often better than none. Usually, a large minimum feature track length will require more features to be detected to have superior accuracy than the small minimum feature track length.

3 Roll Angle Aiding with Vertical Line Observations

Most approaches of VINS have focused on the point features observation. Line features are omnipresent and can be reliably extracted and tracked in structured urban environments. Here we propose to use vanishing point coordinates from buildings' vertical line observations as an attitude aiding to improve point-based VINS on-board the land vehicles.

3.1 Roll Measurement from Vertical Line Observations

The advantage of using buildings' vertical line observations is that the measurement is absolute roll, since the directions of these vertical lines are known. A vanishing point is the intersection of image projections of a set of parallel 3D lines in the scene. Each set of parallel lines is associated to a VP in an image [19].

The vertical lines in the world frame (local-level frame here) intersect at the infinity point which can be represented as $\boldsymbol{U} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^{\boldsymbol{T}}$. The corresponding vanishing point $\boldsymbol{u}$ in the image plane is the projection of the infinity point $\boldsymbol{U}$. The projection equation is written as:

$$\boldsymbol{u} = \boldsymbol{K}\begin{bmatrix} \boldsymbol{C}_n^C & \boldsymbol{T} \end{bmatrix}\boldsymbol{U} \tag{3.1}$$

where $\boldsymbol{K}$ is the camera calibration matrix, $\boldsymbol{C}_n^C$ is the camera rotation matrix, and $\boldsymbol{T}$ is the translation vector of the camera. Equation (3.1) can be rewritten as

$$\boldsymbol{u} = \boldsymbol{K}\boldsymbol{C}_b^C\boldsymbol{C}_n^b\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{3.2}$$

Then INS roll (γ) and pitch (θ) can be represented as

$$\begin{bmatrix} -\sin\gamma\cos\theta \\ \sin\theta \\ \cos\gamma\cos\theta \end{bmatrix} = \boldsymbol{C}_C^b\boldsymbol{K}^{-1}\boldsymbol{u} \tag{3.3}$$

The roll angle can be calculated as

$$\gamma = \tan^{-1}\left(-\frac{\boldsymbol{e}_1^T \boldsymbol{C}_C^b \boldsymbol{K}^{-1}\boldsymbol{u}}{\boldsymbol{e}_3^T \boldsymbol{C}_C^b \boldsymbol{K}^{-1}\boldsymbol{u}}\right) \tag{3.4}$$

where $\boldsymbol{e}_1 = [\,1, \quad 0, \quad 0,]^T$, and $\boldsymbol{e}_3 = [\,0, \quad 0, \quad 1\,]^T$.

Suppose the camera has been calibrated, so $\boldsymbol{C}_C^b \boldsymbol{K}^{-1}$ is known. Vanishing point $\boldsymbol{u}$ is determined by the coordinates of the intersection of multiple detected vertical lines in the image. In this paper, we use the Line Segment Detection (LSD) [20] to get the lines segments, which are then clustered into multiple VP classes based on the J-Linkage model [21]. The vertical VP $\boldsymbol{u}$ is one of the dominant VP classes, and can be easily selected and estimated [22]. Note that some erroneous line segments exist in the vertical VP class. They can be further cancelled by making additional assumptions, for example on the orientation or length of the line segments.

3.2 Roll Angle Measurement Model

In the INS mechanization, the roll angle γ can be calculated by

$$\hat{\gamma} = \tan^{-1}\frac{-\hat{c}_{31}}{\hat{c}_{33}} \tag{3.5}$$

where $\hat{c}_{31}$ and $\hat{c}_{33}$ are elements of attitude matrix $\widehat{\boldsymbol{C}}_b^n$, which can be written as

$$\widehat{\boldsymbol{C}}_b^n = [\boldsymbol{I} - \boldsymbol{\phi}^n \times]\boldsymbol{C}_b^n = \begin{bmatrix} 1 & \phi_U & -\phi_N \\ -\phi_U & 1 & \phi_E \\ \phi_N & -\phi_E & 1 \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{3.6}$$

So the roll can be expressed as

$$\hat{\gamma} = \tan^{-1}\frac{-\hat{c}_{31}}{\hat{c}_{33}} = \tan^{-1}\frac{-(c_{11}\phi_N - c_{21}\phi_E + c_{31})}{c_{13}\phi_N - c_{23}\phi_E + c_{33}} \tag{3.7}$$

Then the INS roll error $\delta\gamma$ can be expressed as

$$\delta\gamma = \frac{\partial\hat{\gamma}}{\partial\phi_E}\phi_E + \frac{\partial\hat{\gamma}}{\partial\phi_N}\phi_N + \frac{\partial\hat{\gamma}}{\partial\phi_U}\phi_U \tag{3.8}$$

where

$$\frac{\partial\hat{\gamma}}{\partial\phi_E} \approx \frac{\hat{c}_{21}\hat{c}_{33} - \hat{c}_{31}\hat{c}_{23}}{\hat{c}_{31}^2 + \hat{c}_{33}^2}, \quad \frac{\partial\hat{\gamma}}{\partial\phi_N} \approx \frac{\hat{c}_{31}\hat{c}_{13} - \hat{c}_{11}\hat{c}_{33}}{\hat{c}_{31}^2 + \hat{c}_{33}^2}, \quad \frac{\partial\hat{\gamma}}{\partial\phi_U} = 0$$

Subtracting the VP derived roll angle from the INS roll angle, we have the residual

$$Z_{\gamma} = \hat{\gamma} - \tilde{\gamma} = \delta\gamma - v_{\gamma} \tag{3.9}$$

Therefore, the design matrix of the roll measurement is

$$\boldsymbol{H}_{\gamma} = \left[\boldsymbol{O}_{1\times 3} \quad \frac{\partial\hat{\gamma}}{\partial\phi_E} \quad \frac{\partial\hat{\gamma}}{\partial\phi_N} \quad \frac{\partial\hat{\gamma}}{\partial\phi_U} \quad \boldsymbol{O}_{1\times 9} \quad \boldsymbol{O}_{1\times 6m} \right] \tag{3.10}$$

As we can see from the Jacobian matrices, the roll measurement will contribute to the estimation of both horizontal attitude errors but not the azimuth error.

3.3 Navigation Procedure

The navigation flow chat is illustrated in Fig. 1. When a new image is observed, some key functions in MSCKF and VP aiding will work: state augmentation and feature detection and matching, and vertical line detection, respectively. If the measurement update is triggered in either vision modules, the states will be corrected.

4 Results and Analysis

The point-based MSCKF and vanishing point aiding from vertical lines algorithm described in the previous sections was tested using ‘2011_09_26_drive_0036’ dataset from the KITTI benchmark [23]. The synchronized IMU data (10 Hz) and rectified grayscale images (10 Hz, global shutter) were used to verify the algorithm. The biases of gyroscopes and accelerometers are 0.01°/s and 1 mg (see more details in [24]). The reference was the IMU/GPS results (OXTS RT3003) with the L1/L2 RTK positioning accuracy of 0.02 m and pitch/roll accuracy of 0.03°. We used 250 images with traveled distance about 180 m, as shown on Fig. 2.

To illustrate the performance of proposed algorithms, the GPS data was totally blocked of whole trajectory except for the initial value of navigation states. We compared the performance of the following navigation schemes.

- Free INS: Only INS mechanization is performed to calculate the navigation states.
- MSCKF-VINS with minimum 5 tracking poses: minimum 5 poses in the tracking are required to trigger a measurement update.
- MSCKF-VINS with minimum 20 tracking poses: minimum 20 poses in the tracking are required to trigger a measurement update.

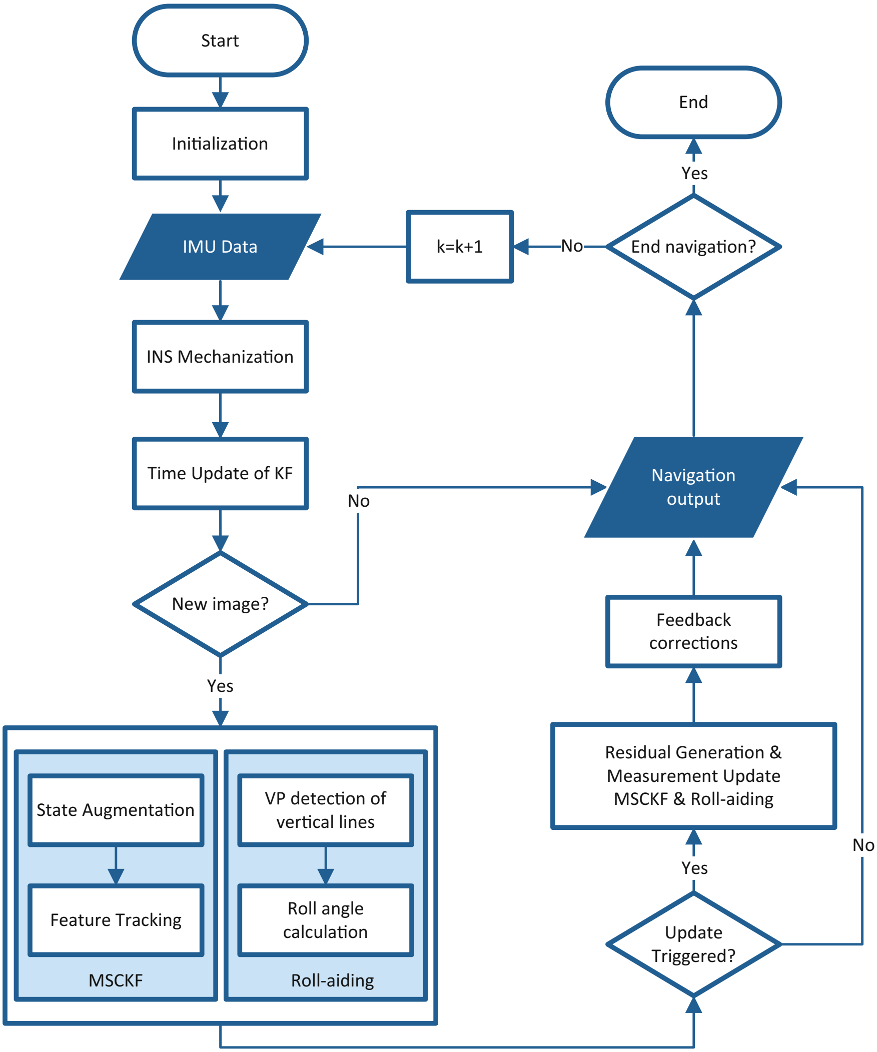

Fig. 1 Navigation flowchart

- MSCKF-VINS with minimum 20 tracking poses and with vertical line aiding: Vanishing point aiding module is added into the system where minimum 20 poses in the tracking are required.

The trajectory results of the 4 navigation schemes are shown in Fig. 2. The horizontal attitude errors and position errors are presented in Figs. 3 and 4. We can see that there is just minor difference between MSCKF-VINS with minimum 5 and 20 tracking poses. A slight better pitch estimation of MSCKF-VINS with 20 tracking poses can be seen in Fig. 3 when there are abundant features (before 14 s).

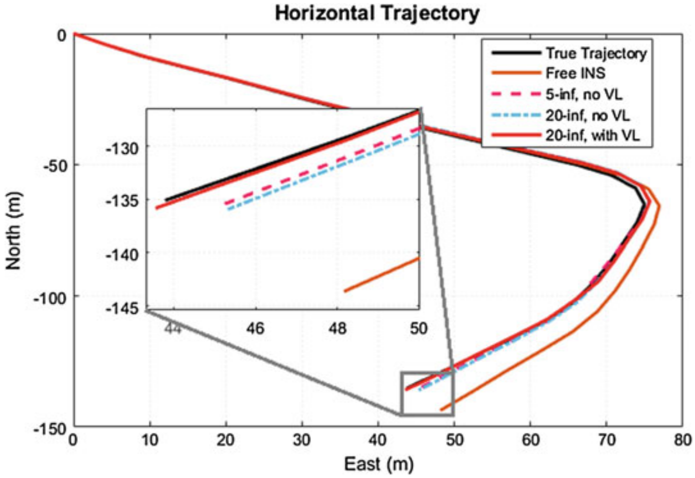

Fig. 2 Reference and calculated trajectory

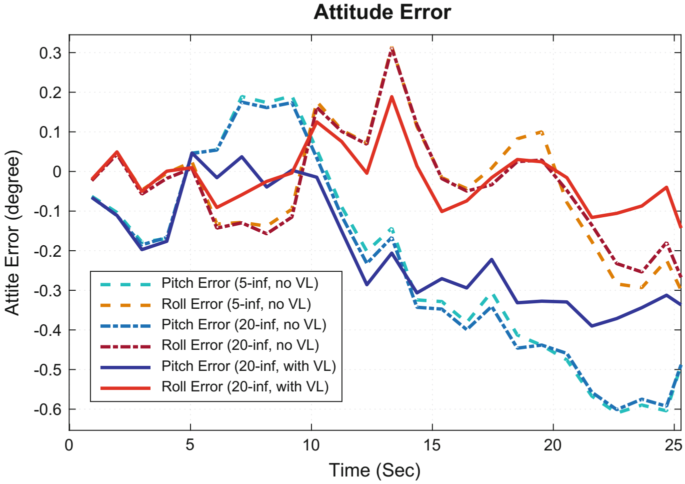

Fig. 3 Attitude errors (pitch and roll)

One reason can be that the longer tracking will observe further distance landmarks, which is beneficial for the pitch angle estimation.

An example of vertical line clustering is shown in Fig. 5, where the green line segments correspond to the vertical VP. When adding the vertical line vanishing point aiding into the system, both pitch and roll estimations are improved to a great extent (up to 0.2°) compared with MSCKF-only methods. The horizontal position error is also less than MSCKF-only methods in general, within 1.2 m. The position estimation improvement results from better attitude estimation, because the attitude

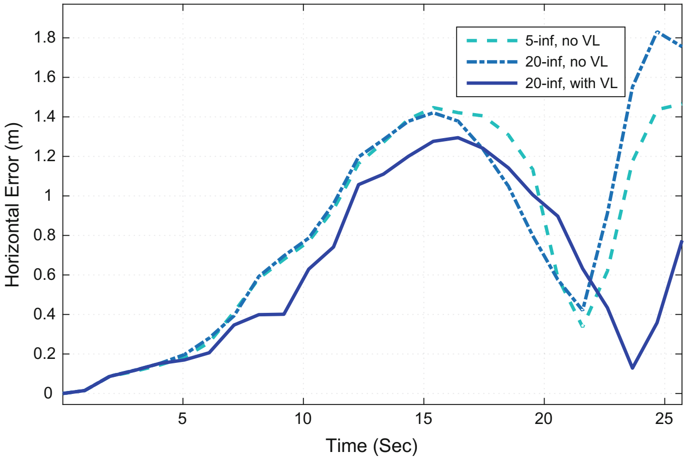

Fig. 4 Horizontal position errors

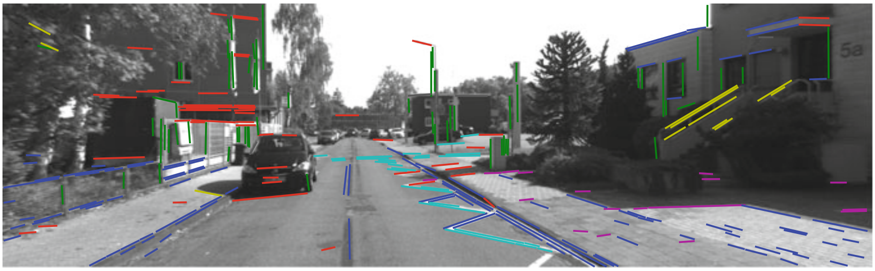

Fig. 5 Line segments groups with different colors (Color figure online)

error will produce horizontal acceleration error by projecting the gravity to the horizontal plane.

5 Conclusion

In this paper, we use MSCKF in the local-level frame and present it in a way that is common to navigation community. To improve the performance of point-based MSCKF, we proposed to add a VP module which observes the vertical lines of building blocks for land vehicle navigation. Based on that, the roll angle measurement model was derived. Therefore, a mixed VINS using points and line observations was developed, and its performance was verified by real data experiments. Horizontal attitude and position estimations are improved by using vertical line vanishing point aiding module.

References

1. Vinet L, Zhedanov A (2012) Handbook of intelligent vehicles. Springer London, London
2. Zolghadri A, Henry D, Cieslak J (2014) Fault diagnosis and fault-tolerant control and guidance for aerospace vehicles. Springer, London
3. Liu Z, El-Sheimy N, Qin Y, Yu C, Zhang J (2016) Partial state feedback correction for smoothing navigational parameters. In: Sun J, Liu J, Fan S, Wang F (eds) China satellite navigation conference (CSNC) 2016 proceedings, vol II. Springer Singapore, Singapore, pp 461–472
4. Ahmed-Zaid F, Bai F, Bai S, Basnayake C, Bellur B, Brovold S, Brown G, Caminiti L, Cunningham D, Elzein H, Hong K, Ivan J, Jiang D, Kenney J, Krishnan H, Lovell J, Maile M, Masselink D, McGlohon E, Mudalige P, Popovic Z, Rai V, Stinnett J, Tellis L, Tirey K, and VanSickle S (2011) Vehicle safety communications–applications (VSC-A)
5. Chowdhary G, Johnson EN, Magree D, Wu A, Shein A (2013) GPS-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft. J F Robot 30 (3):415–438
6. Dissanayake G, Sukkarieh S, Nebot E, Durrant-Whyte H (2001) The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. IEEE Trans Robot Autom 17(5):731–747
7. Niu X, Nassar S, El-Sheimy N (2007) An accurate land-vehicle MEMS IMU/GPS navigation system using 3D auxiliary velocity updates. J Inst Navig 54(3):177–188
8. Nistér D, Naroditsky O, Bergen J (2006) Visual odometry for ground vehicle applications. J F Robot 23(1):3–20
9. Scaramuzza D, Fraundorfer F (2011) Visual odometry [tutorial]. IEEE Robot Autom Mag 18 (4):80–92
10. Veth M, Raquet J (2006) Two-dimensional stochastic projections for tight integration of optical and inertial sensors for navigation, Natl Tech Meet Proc Inst Navig, pp 587–596
11. Veth MJ (2008) Fusion of imaging and inertial sensors for navigation. Air Force Institute of Technology, USA
12. Mourikis AI Roumeliotis SI, (2007) A multi-state constraint Kalman filter for vision-aided inertial navigation, In: Proceedings-IEEE international conference on robotics and automation, pp 3565–3572
13. Leutenegger S, Lynen S, Bosse M, Siegwart R, Furgale P (2015) Keyframe-based visual-inertial odometry using nonlinear optimization. Int J Rob Res 34(3):314–334
14. Veth M, Raquet J, Pachter M (2006) Stochastic constraints for efficient image correspondence search. IEEE Trans Aerosp Electron Syst 42(3):973–982
15. Forster C, Carlone L, Dellaert F, Scaramuzza D (2015) On-manifold pre-integration theory for fast and accurate visual-inertial navigation, IEEE Trans Robot, pp 1–18
16. Kim S-B, Bazin J-C, Lee H-K, Choi K-H, Park S-Y (2011) Ground vehicle navigation in harsh urban conditions by integrating inertial navigation system, global positioning system, odometer and vision data. IET Radar Sonar Navig 5(8):814
17. Camposeco F, Pollefeys M (2015) Using vanishing points to improve visual-inertial odometry. In: 2015 IEEE international conference on robotics and automation (ICRA), pp 5219–5225
18. Clement LE, Peretroukhin V, Lambert J Kelly J (2015) The Battle for filter supremacy: a comparative study of the multi-state constraint Kalman filter and the sliding window filter. In: Proceedings of 2015 12th conference on computer and robot vision, CRV 2015, pp 23–30
19. Hartley R, Zisserman A (2003) Multiple view geometry in computer vision. Cambridge University Press, Cambridge
20. Grompone Von Gioi R, Jakubowicz J, Morel JM, Randall G (2010) LSD: a fast line segment detector with a false detection control. IEEE Trans Pattern Anal Mach Intell 32(4):722–732

21. Toldo R, Fusiello A Informatica D (2008) Robust multiple structures estimation with J-Linkage. In: Computer vision-ECCV: 10th European conference on computer vision, marseille, France, Oct 12–18, Proceedings, Part I, pp 537–547
22. Feng C, Deng F Kamat VR (2010) Semi-automatic 3D reconstruction of piecewise planar. In: 10th international conference construction applications of virtual reality, pp 1–9
23. Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: the KITTI dataset. Int J Rob Res 32(11):1231–1237
24. RT3000 inertial and GPS navigation system. Available: http://www.oxts.com/Downloads/Support/Brochures/rt3kpres.pdf. Accessed: 17-Jan-2017