INS Assisted Monocular Visual Odometry for Aerial Vehicles

Ji Zhang and Sanjiv Singh

Abstract The requirement to operate aircrafts in GPS denied environments can be met by use of visual odometry. We study the case that the height of the aircraft above the ground can be measured by an altimeter. Even with a high quality INS that the orientation drift is neglectable, random noise exists in the INS orientation. The noise can lead to the error of position estimate, which accumulates over time. Here, we solve the visual odometry problem by tightly coupling the INS and camera. During state estimation, we virtually rotate the camera by reprojecting features with their depth direction perpendicular to the ground. This allows us to partially eliminate the error accumulation in state estimation, resulting in a slow position drift. The method is tested with data collected on a full-scale helicopter for approximately 16km of travel. The estimation error is less than 1% of the flying distance.

1 Introduction

This paper addresses the problem of vision-based state estimation for an aerial vehicle. Typically, vision-based method is useful in the cases where GPS is unavailable or insufficiently accurate. On aerial vehicles, continuously accurate GPS positioning can be hard to ensure, especially when the vehicle flies at a high speed. Visual odometry [1, 2] becomes a supplemental method. Multiple cameras fixed on the aircraft can be used to recover 6DOF motion, but this requires that the baseline between the cameras to be at least a non-trivial fraction of the vehicle elevation above the ground. That is, if a small baseline is used, the cameras reduce to a monocular camera when the vehicle flies at a high altitude. If the cameras are separated significantly, camera calibration becomes hard and accuracy can be uninsured.

This paper uses a monocular camera looking downward toward the ground. The scale of translation is solved by the distance of the vehicle above the ground measured by an altimeter. We model the imaged ground as a locally flat patch with two

Ji Zhang (zhangji@cmu.edu) and Sanjiv Singh (ssingh@cmu.edu) are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA

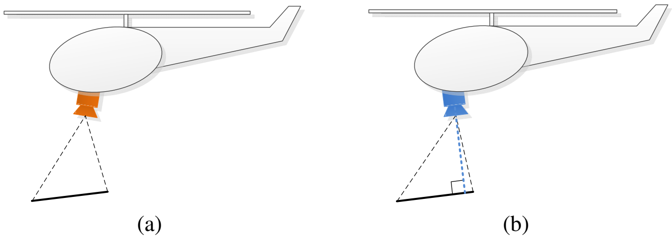


Fig. 1 Illustration of the proposed method. (a) shows the real position of the camera. During state estimation, features are reprojected with their depth direction perpendicular to the ground. This equals to rotating the camera to a virtual position pointing perpendicularly to the ground, as (b). We will show in this paper that using features associated with the virtual camera for the state estimation can decelerate the accumulation of the motion estimation error.

rotation DOFs. The proposed method estimates the translation and the inclination angles of the ground patch concurrently.

To deal with the noise in INS orientation, we propose to reproject features with their depth direction perpendicular to the ground patch. This equals to rotating the camera virtually to be perpendicular to the ground, as shown in Fig. 1. By doing this, we can partially eliminate the accumulation of translation estimation error– the accumulated translation error introduced by roll and pitch angle noise from the INS largely cancels itself overtime, especially when the vehicle flies at a constant altitude. Also, we find that it is hard to prevent propagation of the yaw angle noise and the altimeter noise in the state estimation, but we can only reduce the noise amount from the error sources. Correspondingly, we implement a Kalman filter [3] to reduce the yaw angle noise. We also adopt a high quality laser altimeter on the aircraft to obtain accurate elevation measurement. The result is state estimation with relative error less than 1% of the flying distance.

The rest of this paper is organized as follows. In section 2, we present related work. In section 3, we define assumptions and coordinate systems. The method is overviewed in Section 4, and solved in detail in Section 5. Analysis of error propagation is given in Section 6. Experimental results are presented in Section 7 and conclusion is made in Section 8.

2 Related Work

Vision based methods are now common for vehicle state estimation [4, 5]. Typically, the problem solves 6DOF camera motion in an arbitrary environment. When stereo cameras are used [6], the relative poses of the two cameras function as a constraint that helps solve the motion estimation problem. For example, Konolige, at al's stereo visual odometry recovers the camera motion from bundle adjustment [7]. The method is integrated with an IMU which handles orientation drift of the visual odometry in long distance navigation. For a monocular camera, if the camera motion is unconstrained and no prior knowledge is assumed about the environment, the scale ambiguity is generally unsolvable. Klein and Murray develop a visual SLAM method by parallel tracking and mapping of monocular imagery [8]. The method is improved by Weiss and modified to be visual odometry [9].

When using a monocular camera, if the camera motion follows certain constraint, the scale ambiguity can be solved in constrained cases. On the other hand, if certain a prior knowledge or constraint about the environment is available, it can also assist to solve the motion estimation problem. For example, Artieda, et al's visual SLAM uses a front looking camera mounted on an aerial vehicle [10]. The scale ambiguity is solved by assuming some of the feature points with known 3D coordinates. Conte and Doherty's visual navigation system works for flights at a relatively high altitude such that the ground is considered as flat and level [11]. The vehicle motion is solved by planar homography between images taken from the ground. The method also uses geo-referenced aerial images to fix the visual odometry drift. Caballero, et al's visual odometry also assumes flat ground and uses planar homography [12]. However, the method does not require the ground to be level and online recovers its orientation with respect to the vehicle. The scale is solved by the vehicle elevation above the ground measured by a range sensor.

Our method is similar to [12] in the sense that both assume the imaged ground to be locally flat but not necessary level. However, our method does not rely on planar homography. The orientation readings from the INS are directly used in solving the translation in a tightly coupled fashion. The result is that our method solves a problem with less DOFs. Further, the method is designed to be insensitive to the INS orientation errors, and therefore has a slow position drift.

3 Assumptions and Coordinate Systems

The visual odometry problem addressed in this paper is to estimate the state of an aerial vehicle using a monocular vision system, an INS and an altimeter. We assume that the camera is well modeled as a pinhole camera [13]. The camera intrinsic parameters are known from pre-calibration, and the lens distortion is removed. As a

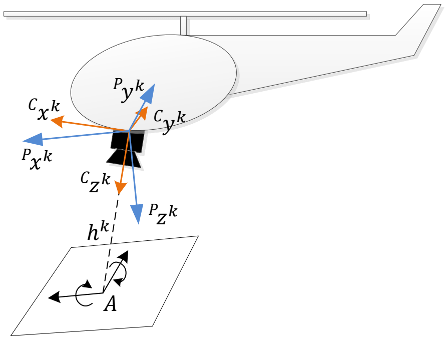


Fig. 2 Illustration of the coordinate systems and ground model. $\{C^k\}$ is the camera coordinate system at frame k. $\{P^k\}$ is a coordinate system with its $x-y$ plane parallel to the ground patch. A is the intersection of the z-axis of $\{C^k\}$ with the ground. The distance from A to the origin of $\{C^k\}$, h^k, is measured by an altimeter. The ground patch is modeled with roll and pitch DOFs around A.

convention in this paper, we use left uppercase superscription to indicate coordinate systems, and right superscription k, $k \in Z^+$ to indicate image frames. We use $\mathscr{I}$ to denote the set of feature points. We define two coordinate systems.

- Image coordinate system $\{I\}$ is a 2D coordinate system with its origin at the left upper corner of the image. The u- and v- axes in $\{I\}$ are pointing to the right and downward directions of the image. A point i, $i \in \mathscr{I}$, in $\{I^k\}$ is denoted as ${}^I\boldsymbol{x}_i^k$.
- Camera coordinate system $\{C\}$ is a 3D coordinate system. As shown in Fig. 2, the origin of $\{C\}$ is at the camera optical center with the z-axis coinciding with the camera principal axis. The $x-y$ plane is parallel to the camera image sensor with the x-axis pointing to the forward direction of the vehicle. A point i, $i \in \mathscr{I}$, in $\{C^k\}$ is denoted as ${}^C\boldsymbol{X}_i^k$.

We model the imaged ground as a locally flat patch, as shown in Fig. 2. Let A be the intersection of the z-axis of $\{C^k\}$ with the ground patch. The distance between A and the origin of $\{C^k\}$ is measured by an altimeter, denoted as h^k. The ground patch is modeled to have roll and pitch DOFs around A. Here, we define another coordinate system.

- Parallel to ground coordinate system $\{P\}$ is a 3D coordinate system. The origin of $\{P\}$ is coinciding with the origin of $\{C\}$, the $x-y$ plane is parallel to the ground with the x-axis pointing to the forward direction of the vehicle. The z-axis is pointing downward perpendicularly to the ground patch. A point i, $i \in \mathscr{I}$, in $\{P^k\}$ is denoted as ${}^P\boldsymbol{X}_i^k$.

4 Software System Diagram

The system diagram of the visual odometry software is shown in Fig. 3. The system takes the camera images, altimeter reading, orientation from the INS, and computes the translation and inclination of the ground. We will show that the translation estimation is insensitive to the noise in roll and pitch angles, but sensitive to yaw noise. Hence the visual odometry is particulary designed to take only the roll and pitch angles from the INS, and estimate the yaw angle by itself. Behind the visual odometry block, we implement a Kalman filter that integrates the yaw angle. The Kalman filter helps reduce the noise amount by taking the visual odometry estimate in the prediction steps and the INS measurement in the update steps. The integrated yaw angle can be used to register the translation in the world.

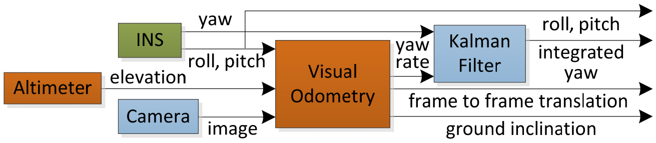


Fig. 3 Visual odometry software system diagram.

5 Visual Odometry Method

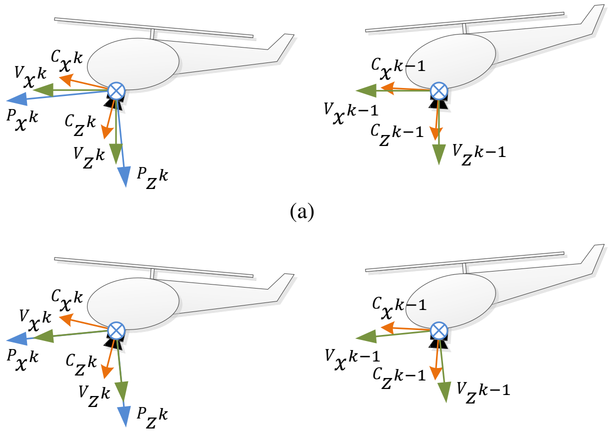
5.1 Method Intuition

Fig. 4 presents the key idea for the visual odometry method. We use two parallel coordinate systems at frames $k-1$ and k, respectively. Let $\{V^{k-1}\}$ be a coordinate system with its origin coinciding with the origin of $\{C^{k-1}\}$, and $\{V^{k}\}$ a coordinate system with its origin coinciding with that of $\{C^{k}\}$. Initially, $\{V^{k-1}\}$ and $\{V^{k}\}$ are rotated to the horizontal position as shown in Fig. 4(a), using orientation from the INS. Then, through nonlinear iterations, $\{V^{k-1}\}$ and $\{V^{k}\}$ are rotated to be parallel to $\{P^{k}\}$, the coordinate system parallel to the ground patch at frame k, as shown in Fig. 4(b). During the nonlinear iterations, $\{V^{k-1}\}$ and $\{V^{k}\}$ are kept parallel, and the features at both frames are projected into $\{V^{k-1}\}$ and $\{V^{k}\}$. The projected features are used to compute the vehicle frame to frame motion.

5.2 Mathematical Derivation

In this section, we present the mathematical derivation of the proposed method. The complete visual odometry algorithm is presented in the next section. From the pin-hole camera model, we have the following relationship between $\{I^{l}\}$ and $\{C^{l}\}$, $l \in \{k-1, k\}$,

$$\alpha\, {}^{I}\mathbf{X}_i^l = \mathbf{K}\, {}^{C}\mathbf{X}_i^l, \tag{1}$$



(a)

(b)

Fig. 4 Illustration of coordinate systems $\{V_{k-1}\}$ and $\{V_k\}$. As indicated by the green colored arrows, $\{V_{k-1}\}$ (in the right column) and $\{V_k\}$ (in the left column) are two parallel coordinate systems at frames $k-1$ and k, respectively. $\{V_{k-1}\}$ and $\{V_k\}$ are initialized at the horizontal position as illustrated in (a), using orientation from the INS. Then, through a nonlinear optimization, $\{V_{k-1}\}$ and $\{V_k\}$ are rotated to be parallel to $\{P_k\}$, as shown in (b). The figure only represents a planar case, while $\{V_{k-1}\}$ and $\{V_k\}$ have roll and pitch DOFs with respect to $\{C_{k-1}\}$ and $\{C_k\}$.

where α is a scale factor, and $\mathbf{K}$ is the camera intrinsic matrix, which is known from pre-calibration [13].

Let ${}^C_V\theta^l$ and ${}^C_V\psi^l$, $l \in \{k-1, k\}$, be the roll and pitch angles from $\{V^l\}$ to $\{C^l\}$. The relationship between $\{C^l\}$ and $\{V^l\}$ is expressed as

$$ {}^C\boldsymbol{X}_i^l = \mathbf{R}_x({}^C_V\theta^l)\mathbf{R}_y({}^C_V\psi^l)\ {}^V\boldsymbol{X}_i^l, \tag{2}$$

where $\mathbf{R}_x(\cdot)$ and $\mathbf{R}_y(\cdot)$ are rotation matrices around the x- and y- axes, respectively.

Let ${}^V\tilde{\boldsymbol{X}}_i^l$, $l \in \{k-1, k\}$, be the normalized term of ${}^V\boldsymbol{X}_i^l$, such that

$$ {}^V\tilde{\boldsymbol{X}}_i^l = {}^V\boldsymbol{X}_i^l / {}^V z_i^l, \tag{3}$$

where ${}^V z_i^l$ is the 3rd entry of ${}^V\boldsymbol{X}_i^l$. ${}^V\tilde{\boldsymbol{X}}_i^l$ can be computed by substituting (2) into (1) and scaling ${}^V\boldsymbol{X}_i^l$ such that the 3rd entry becomes one.

Let Δ_x^k, Δ_y^k, and Δ_z^k be the vehicle translation in the x-, y- and z- directions between frames $k-1$ and k, and let Δ_ϕ^k be the corresponding yaw rotation between the two frames. From the vehicle motion, we can establish a relationship between $\{V^{k-1}\}$ and $\{V^k\}$,

$$ {}^V\boldsymbol{X}_i^k = \mathbf{R}_z(\Delta_\phi^k)({}^V\boldsymbol{X}_i^{k-1} - \left[\Delta_x^k,\ \Delta_y^k,\ \Delta_z^k\right]^T), \tag{4}$$

where $\mathbf{R}_z(\cdot)$ is the rotation matrix around the z- axis.

Substituting (3) into (4) for both frames $k-1$ and k, and since Δ_ϕ^k is a small angle in practice, we perform linearization to obtain the following equations,

$$ s\ {}^V\tilde{x}_i^{k-1} = {}^V\tilde{x}_i^k - {}^V\tilde{y}_i^k\Delta_\phi^k + \Delta_x^k / {}^V z_i^k, \tag{5}$$

$$ s\ {}^V\tilde{y}_i^{k-1} = {}^V\tilde{y}_i^k + {}^V\tilde{x}_i^k\Delta_\phi^k + \Delta_y^k / {}^V z_i^k, \tag{6}$$

$$ s = 1 - \Delta_z^k / {}^V z_i^k, \tag{7}$$

where ${}^V\tilde{x}_i^l$ and ${}^V\tilde{y}_i^l$, $l \in \{k-1, k\}$, are the 1st and the 2nd entries of ${}^V\tilde{\boldsymbol{X}}_i^l$, respectively, ${}^V z_i^k$ is the 3rd entry of ${}^V\boldsymbol{X}_i^k$, and s is a scale factor.

Recall that h^k is the distance from the vehicle to the ground patch, and the ground patch has roll and pitch DOFs round point A in Fig. 2. Let ${}^V_P\theta^k$ and ${}^V_P\psi^k$ be the roll and pitch angles from $\{P^k\}$ to $\{V^k\}$. In (5)–(7), the feature depth ${}^V z_i^k$ can be computed from a simple geometry relationship,

$$ {}^V z_i^k = h^k(1 - ({}^V\tilde{x}_i^k - {}^C_V\psi^k){}^V_P\psi^k - ({}^V\tilde{y}_i^k - {}^C_V\theta^k){}^C_V\theta^k). \tag{8}$$

Combining (5), (6) and (8), we have,

$$ \frac{a\Delta_x^k + b\Delta_y^k}{c + d\ {}^V_P\psi^k + e\ {}^V_P\theta^k} + f\Delta_\psi^k + g = 0. \tag{9}$$

where

$$a = {}^{V}\tilde{y}_i^{k-1},\ b = -{}^{V}\tilde{x}_i^{k-1},\ c = h^k,\ d = -h^k({}^{V}\tilde{x}_i^{k} - {}^{C}_{V}\psi^k), \tag{10}$$

$$e = -h_k(\tilde{y}^{V}_{(k,i)} - {}^{C}_{V}\theta^k),\ f = -{}^{V}\tilde{x}_i^{k}\, {}^{V}\tilde{x}_i^{k-1} - {}^{V}\tilde{y}_i^{k}\, {}^{V}\tilde{y}_i^{k-1}, \tag{11}$$

$$g = {}^{V}\tilde{x}_i^{k}\, {}^{V}\tilde{y}_i^{k-1} - {}^{V}\tilde{y}_i^{k}\, {}^{V}\tilde{x}_i^{k-1}. \tag{12}$$

In (9), we have totally five unknowns, Δ_x^k, Δ_y^k, Δ_ϕ^k, ${}^{V}_{P}\theta^k$, ${}^{V}_{P}\psi^k$. The function can be solved using five or more feature points with a nonlinear method. However, in certain cases, we can consider ${}^{V}_{P}\theta^k$ and ${}^{V}_{P}\psi^k$ as known variables such that (9) can be solved linearly with three or more feature points. Next, we will provide a linear and a nonlinear way to solve the function. Both methods will be useful for the visual odometry algorithm presented in the next section.

5.2.1 Linear Method

Set ${}^{V}_{P}\theta^k$ and ${}^{V}_{P}\psi^k$ in (9) as known variables and treat Δ_x^k, Δ_y^k, Δ_ϕ^k as unknowns. For m, $m \geq 3$, feature points, stack (9) for each feature. This will give us a linear function in the form of

$$\mathbf{A}\boldsymbol{X}_L = \boldsymbol{b}, \tag{13}$$

where $\mathbf{A}$ is a $m \times 3$ matrix, $\boldsymbol{b}$ is a $m \times 1$ vector, and $\boldsymbol{X}_L$ contains the unknowns, $\boldsymbol{X}_L = [\Delta_x^k,\ \Delta_y^k,\ \Delta_\phi^k]^T$. Solving (13) with the singular value decomposition method [13], we can recover $\boldsymbol{X}_L$.

5.2.2 Nonlinear Method

For m, $m \geq 5$, feature points, stack (9) for each feature and reorganize the function into the following form,

$$\boldsymbol{f}(\boldsymbol{X}_N) = \boldsymbol{b}, \tag{14}$$

where $\boldsymbol{f}$ is a nonlinear function with 5 inputs and m outputs. $\boldsymbol{b}$ is a $m \times 1$ vector, and $\boldsymbol{X}_N$ contains the unknowns, $\boldsymbol{X}_N = [\Delta_x^k,\ \Delta_y^k,\ \Delta_\phi^k,\ {}^{V}_{P}\theta^k,\ {}^{V}_{P}\psi^k]^T$. Compute the Jacobian matrix of $\boldsymbol{f}$ with respect to $\boldsymbol{X}_N$, denoted as $\mathbf{J}$, where $\mathbf{J} = \partial f / \partial \boldsymbol{X}_N$. (14) can be solved through nonlinear iterations using the Levenberg-Marquardt method [13],

$$\boldsymbol{X}_N \leftarrow \boldsymbol{X}_N + (\mathbf{J}^T\mathbf{J} + \lambda\,\mathrm{diag}(\mathbf{J}^{\mathrm{T}}\mathbf{J}))^{-1}\mathbf{J}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{f}(\boldsymbol{X}_{\mathrm{N}})), \tag{15}$$

where λ is a scale factor.

5.3 Algorithm

Algorithm 1 presents the proposed visual odometry algorithm. The algorithm first initializes using readings from the INS. Let θ_{INS}^{k-1} and ψ_{INS}^{k-1} be the roll and pitch angles of the vehicle at frame $k-1$, measured by the INS, and let θ_{INS}^{k} and ψ_{INS}^{k} be the corresponding angles at frame k. On lines 4-5, we rotate $\{V^{k-1}\}$ and $\{V^k\}$ to the horizontal position using the INS orientation, and we project the feature points from $\{C^{k-1}\}$ and $\{C^k\}$ to $\{V^{k-1}\}$ and $\{V^k\}$, respectively. From now, $\{V^{k-1}\}$ and $\{V^k\}$ become parallel coordinate systems. Then, on line 6, we set ${}^{V}_{P}\theta_l, {}^{V}_{P}\psi_l \leftarrow 0$ and

Algorithm 1: Translation Estimation

```
1  **input** : ${}^{I}\boldsymbol{x}_i^{k-1}$, ${}^{I}\boldsymbol{x}_i^{k}$, $i \in \mathscr{I}$, $\theta_{INS}^{k-1}$, $\psi_{INS}^{k-1}$, $\theta_{INS}^{k}$, $\psi_{INS}^{k}$, $h^k$
2  **output** : $\Delta_x^k$, $\Delta_y^k$, $\Delta_z^k$
3  **begin**
4      Rotate $\{V^l\}$ to the horizontal position by ${}_V^C\theta^l \leftarrow \theta_{INS}^l$, ${}_V^C\psi^l \leftarrow \psi_{INS}^l$, $l \in \{k-1, k\}$;
5      Compute ${}^{V}\tilde{\boldsymbol{X}}_i^{k-1}$, ${}^{V}\tilde{\boldsymbol{X}}_i^{k}$ for $i \in \mathscr{I}$ based on (1-3);
6      Use $i \in \mathscr{I}$ to compute $\Delta_x^k$, $\Delta_y^k$, $\Delta_\phi^k$ linearly by setting ${}_P^V\theta^k, {}_P^V\psi^k \leftarrow 0$ based on (13);
7      **for** a number of iterations **do**
8          Compute image reprojection error (IRE) for $i \in \mathscr{I}$, then compute a weight for
           $i \in \mathscr{I}$ using the IREs;
9          **for** a number of iterations **do**
10             Use $i \in \mathscr{I}$ to update $\Delta_x^k$, $\Delta_y^k$, $\Delta_\phi^k$, ${}_P^V\theta^k$, ${}_P^V\psi^k$ for one iteration based on (15);
11             Rotate $\{V^l\}$ by ${}_V^C\theta^l \leftarrow {}_V^C\theta^l +{}_P^V \theta^k$ and ${}_V^C\psi^l \leftarrow {}_V^C\psi^l +{}_P^V \psi^k$, $l \in \{k-1, k\}$, then
               ${}_P^V\theta^k, {}_P^V\psi^k \leftarrow 0$;
12             Project ${}^{V}\tilde{\boldsymbol{X}}_i^{k-1}$, ${}^{V}\tilde{\boldsymbol{X}}_i^{k}$, $i \in \mathscr{I}$, $\Delta_x^k$, $\Delta_y^k$, $\Delta_\phi^k$ to the newly rotated $\{V^{k-1}\}$ and $\{V^k\}$;
13             **if** the nonlinear optimization converges **then**
14                 Break;
15             **end**
16         **end**
17         **if** the robust fitting converges **then**
18             Break;
19         **end**
20     **end**
21     Compute $\Delta_z^k$ based on (7) as the weighted average of the features;
22     Return $\Delta_x^k$, $\Delta_y^k$, $\Delta_z^k$;
23 **end**
```

compute Δ_x^k, Δ_y^k, Δ_ϕ^k linearly. The result is used as initialization for the nonlinear optimization between lines 7-20. The 5 unknowns Δ_x^k, Δ_y^k, Δ_ϕ^k, ${}_V^C\psi^k$, ${}_V^C\theta^k$ are updated on line 10. On lines 11-12, $\{V^{k-1}\}$ and $\{V^k\}$ are rotated to the newly updated orientation with the features reprojected into $\{V^{k-1}\}$ and $\{V^k\}$. The iterations finish if convergence is found or the maximum iteration number is met.

The algorithm is adapted to a robust fitting [14] to ensure robustness against features with large tracking errors. The algorithm assigns a weight for each feature (line 8), based on the image reprojection error (IRE). The features with larger IREs are assigned with smaller weights, while features with the IREs larger than a threshold are considered as outliers and assigned with zero weights. Note that the robust fitting only solves the x- and y- translation of the vehicle, Δ_x^k, Δ_y^k. To obtain the z-translation, Δ_z^k, we use (7) with the selected inlier features from the robust fitting (line 21). Δ_z^k is computed as the weighted average of the inlier features using the same weights generated by the robust fitting on line 8.

6 Analysis of Error Propagation

Here we show how the errors are propagated onto the vehicle motion estimation. We care about how the errors accumulate in the horizontal position estimate because vertical position drift can be largely corrected by reading of the altimeter. We will derive the upper bound of the accumulated position drift.

We start with the INS roll and pitch angles. Recall that θ^l_{INS}, ψ^l_{INS}, $l \in \{k-1, k\}$ are the roll and pitch inclination angles of the vehicle measured by the INS at fame l. Let us define $\hat{\theta}^l_{INS}$ and $\hat{\psi}^l_{INS}$ as their measurement values containing errors. Let e^l_θ and e^l_ψ be the corresponding errors, we have $e^l_\theta = \hat{\theta}^l_{INS} - \theta^l_{INS}$ and $e^l_\psi = \hat{\psi}^l_{INS} - \psi^l_{INS}$. By examining each step in Algorithm 1, we find that e^l_θ and e^l_ψ are introduced into the algorithm at the initialization step (line 4). With the INS measurements, the coordinate systems $\{V^{k-1}\}$ and $\{V^k\}$ are intended to be rotated to the horizontal position. However, because of e^l_ψ and e^l_θ, $\{V^{k-1}\}$ and $\{V^k\}$ are not exactly aligned with the horizontal position. The roll and pitch difference between $\{V^{k-1}\}$ and $\{V^k\}$ are $e^{k-1}_\theta - e^k_\theta$ and $e^{k-1}_\psi - e^k_\psi$, respectively. This angle difference is kept through the algorithm since the two coordinate systems are rotated simultaneously by the same angle. In the end, $\{V^k\}$ is rotated to be parallel to $\{P^k\}$, or the ground patch at frame k, and $\{V^{k-1}\}$ keeps an angular error to $\{P^k\}$. Let ${}^C_V\hat{\theta}^{k-1}$ and ${}^C_V\hat{\psi}^{k-1}$ be the measurement values of the roll and pitch angles from $\{V^{k-1}\}$ to $\{C^{k-1}\}$, ${}^C_V\theta^{k-1}$ and ${}^C_V\psi^{k-1}$, we can compute

$$
{}^C_V\hat{\theta}^{k-1} = {}^C_V\theta^{k-1} + e^{k-1}_\theta - e^k_\theta, \ {}^C_V\hat{\psi}^{k-1} = {}^C_V\psi^{k-1} + e^{k-1}_\psi - e^k_\psi. \tag{16}
$$

The errors in ${}^C_V\hat{\theta}^{k-1}$ and ${}^C_V\hat{\psi}^{k-1}$ propagate through (2). With the errors introduced, we rewrite the equation as follows,

$$
{}^C\boldsymbol{X}^{k-1}_i = \mathbf{R}_x({}^C_V\theta^{k-1} + e^{k-1}_\theta - e^k_\theta)\mathbf{R}_y({}^C_V\psi^{k-1} + e^{k-1}_\psi - e^k_\psi)\ {}^V\boldsymbol{X}^{k-1}_i. \tag{17}
$$

Correspondingly, we derive (9) again containing the errors,

$$
a\Big(\frac{\Delta^k_x}{c + d\,{}^V_P\psi^k + e\,{}^V_P\theta^k} + e^{k-1}_\psi - e^k_\psi\Big) + b\Big(\frac{\Delta^k_y}{c + d\,{}^V_P\psi^k + e\,{}^V_P\theta^k} + e^{k-1}_\theta - e^k_\theta\Big) + f\Delta^k_\phi + g = 0, \tag{18}
$$

where a, b, c, d, e, f, and g are defined in (10)-(12).

Now, we compare (18) with (9). Note that after the nonlinear optimization in Algorithm 1 converges, we have ${}^V_P\psi^k, t^{PV}_k \to 0$. Under this condition, if we define $\hat{\Delta}^k_x = \Delta^k_x + (e^{k-1}_\psi - e^k_\psi)h^k$ and $\hat{\Delta}^k_y = \Delta^k_y + (e^{k-1}_\theta - e^k_\theta)h^k$, and substitute the terms into (18), (18) becomes essentially the same as (9) except that Δ^k_x and Δ^k_y are replaced by $\hat{\Delta}^k_x$ and $\hat{\Delta}^k_y$. Examining the expressions of $\hat{\Delta}^k_x$ and $\hat{\Delta}^k_y$, we find that the terms are invariant with respect to different features. This indicates that if we use $\hat{\Delta}^k_x$ and $\hat{\Delta}^k_y$ as the measurement values of Δ^k_x and Δ^k_y for the case that contains the errors, (14) is satisfied for each of its m rows. Define e^k_x and e^k_y as the estimation errors corresponding to Δ_x^k and Δ_y^k, we have

$$e_x^k = \hat{\Delta}_x^k - \Delta_x^k = (e_\psi^{k-1} - e_\psi^k)h^k, \; e_y^k = \hat{\Delta}_y^k - \Delta_y^k = (e_\theta^{k-1} - e_\theta^k)h^k. \tag{19}$$

We want to analyze how the errors accumulate over time. Let us define e_x and e_y as the accumulated errors of e_x^k and e_y^k respectively, from frames 1 to n, $n \in \mathbb{Z}^+$,

$$e_x = \sum_{k=1}^{n} e_x^k, \; e_y = \sum_{k=1}^{n} e_y^k. \tag{20}$$

We want to find the upper bounds of $|e_x|$ and $|e_y|$. Let us define E_θ and E_ψ as the upper bounds of the roll and pitch errors from the INS, where $|e_\theta^k| \leq E_\theta$ and $|e_\psi^k| \leq E_\psi$, $k \in \{1, 2, ..., n\}$. Substituting (19) into (20), we can derive

$$\begin{aligned} e_x &= \sum_{k=2}^{n} (e_\psi^{k-1} - e_\psi^k)h^k = \sum_{k=2}^{n} (e_\psi^{k-1} - e_\psi^k)h^{(2)} + \sum_{k=3}^{n} (e_\psi^{k-1} - e_\psi^k)(h^k - h^{(2)}) \\ &= \sum_{k=2}^{n} (e_\psi^{k-1} - e_\psi^k)h^{(2)} + \sum_{j=3}^{n} \sum_{k=j}^{n} (e_\psi^{k-1} - e_\psi^k)(h^j - h^{j-1}) \\ &= (e_\psi^1 - e_\psi^n)h^{(2)} + \sum_{j=3}^{n} (e_\psi^{j-1} - e_\psi^n)(h^j - h^{j-1}). \end{aligned} \tag{21}$$

Here, since $|e_j^p - e_n^p| \leq |e_j^p| + |e_n^p| \leq 2E_\psi$, $j \in \{1, 2, ..., n\}$, we can find the upper bound of $|e_x|$ as

$$|e_x| \leq 2E_\psi(h^{(2)} + \sum_{j=3}^{n} |h^j - h^{j-1}|). \tag{22}$$

Similarly, we can derive the upper bound of $|e^y|$ as

$$|e_y| \leq 2E_\theta(h^{(2)} + \sum_{j=3}^{n} |h^j - h^{j-1}|). \tag{23}$$

Eq. (22) and (23) indicate that the accumulated translation error introduced by the roll and pitch noise from the INS is only related to the altitude change of the vehicle, regardless of the flying distance. In a special case that the vehicle keeps a constant height above the ground during a flight, $|e_x|$ and $|e_y|$ are bounded by two constants, $|e_x| < 2E_\psi h$ and $|e_y| < 2E_\theta h$, where h is the constant height of the flight. In another case that the vehicle takes off from the ground, h^k starts from zero. The upper bounds of $|e_x|$ and $|e_y|$ are proportional to the accumulated altitude change during the flight, $|e_x| < 2E_\psi \sum_{j=3}^{n} |h^j - h^{j-1}|$ and $|e_y| < 2E_\theta \sum_{j=3}^{n} |h^j - h^{j-1}|$.

Further, we find that the upper bound of the position drift introduced by the yaw angle and altimeter noise is proportional to the flying distance. For space issue, we eliminate the proof. The conclusion can be explained intuitively that if the yaw angle is off, the position estimate will constantly drift to the left or right side. Similarly, noise in the altimeter reading will result in under or overly estimated translation scale. Note that the proposed visual odometry also estimates the yaw angle of the aircraft. This is particularly proposed and allows us to integrate the yaw angle from the INS and visual odometry in a Kalman filter. The integrated yaw angle has a lower amount of noise and is used to register the translation in the world. Also, we use a high quality laser altimeter to reduce the drift in scale.

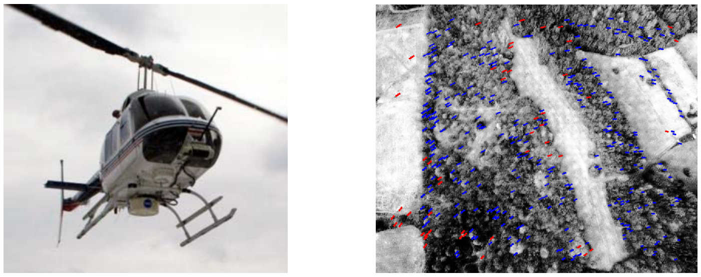
7 Experiments

We obtain image sequences from a downward pointing camera mounted to a full-scale helicopter (Fig. 5(a)). The camera resolution is 612×512 pixels with the horizontal field of view of 75°. The camera frame rate is set at 14Hz. The helicopter is also equipped with a laser altimeter and a GPS/INS. The orientation measurement from the GPS/INS is used by the visual odometry, while the position reading is used as the ground truth for comparison purposes.

The algorithm selects a number of 450 Harris corners [13] using the openCV library, and tracks the feature points between image frames using the Kanade Lucas Tomasi method [15]. To evenly distribute the feature points in the images, we separate the images into 9 (3×3) identical subregions. Each subregion provides 50 features. Fig. 5 shows an example of the tracked features. The red colored segments are outliers assigned with zero weights in Algorithm 1, and the blue colored segments are inliers used in the motion estimation.

Fig. 6 shows results of the proposed method in three flight tests. The blue colored curves are visual odometry outputs, the red colored curves are ground truth provided by the GPS/INS, and the black colored dots are starting points. More detailed configurations and accuracy comparison of the three tests are in Table 1. Tests 1-3 correspond to the subfigures in Fig. 6 from left to right. The overall flying distance is 16km, and the average error at the end is 0.57% of the flying distance.

Fig. 7 presents position and velocity errors for Test 1 (the left subfigure in Fig. 6). Fig. 7(a) shows the accumulated position drift through the test. Fig. 7(b) gives the


(a)

(b)

Fig. 5 (a) Helicopter used in the visual odometry tests. A downward pointing camera is mounted to the front of the helicopter. (b) Tracked features. A number of 450 feature points are tracked between image frames. The red colored segments are outlier features assigned with zero weights in Algorithm 1. The blue colored segments are inlier features used in the motion estimation.

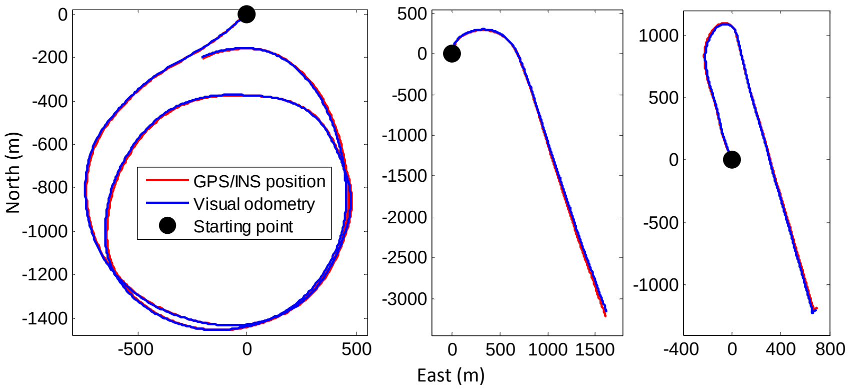
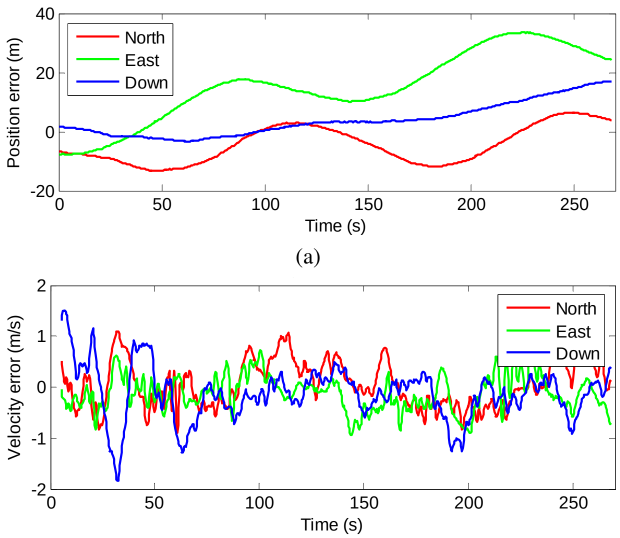


Fig. 6 Visual odometry outputs (blue) compared to GPS/INS ground truth (red) for three tests. The subfigures from left to right correspond to Test 1-3 in Table 1. The overall distance of the three tests is 16km, and the average error at the end is 0.57% of the flying distance.

Table 1 Configuration and accuracy of the three tests in Fig. 6 (from left to right).

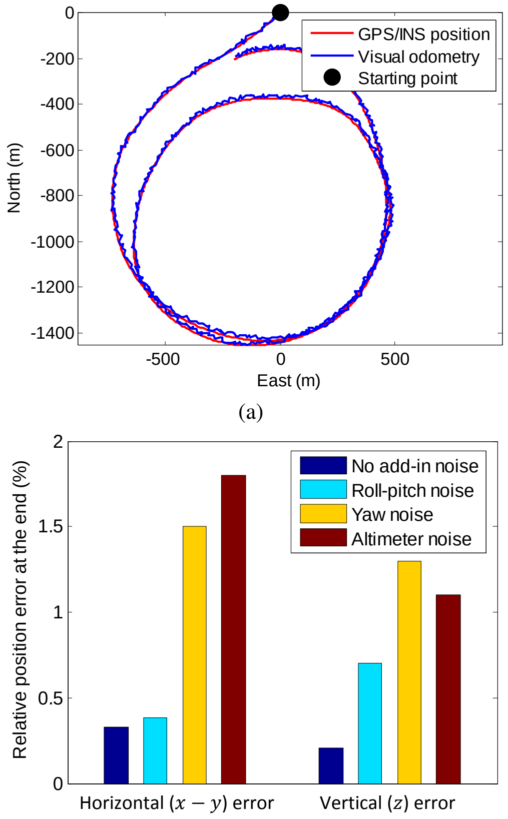
Test No.	Flying Distance	Altitude	Flying Speed	Accuracy
1	7800m	300m	30m/s	0.39%
2	3700m	150m	20m/s	0.73%
3	4500m	200m	20m/s	0.78%



(a)

(b)

Fig. 7 (a) Accumulated position drift and (b) velocity errors in Test 1 (the left subfigure in Fig. 6).



(a)

(b)

Fig. 8 (a) Blue: visual odometry output with artificial add-in noise. The noise is added to the roll and pitch angles from the INS. Red: GPS/INS ground truth (b) Relative errors with respect to different add-in noise. The noise is added to the roll and pitch angles from the INS, yaw angle, and altimeter reading, respectively. The angle noise follows $\sigma = 3°$ Gaussian distribution, and the noise for the altimeter is 3% (σ value) of the elevation with Gaussian distribution.

absolute velocity errors. Most of the velocity errors are smaller than 1m/s, while the average speed of the helicopter is 30m/s during the test.

To inspect how sensor noise affects the motion estimation, we add artificial noise to the INS and altimeter readings. In Fig. 8(a), $\sigma = 3°$ Gaussian noise is added to the roll and pitch angles from the INS. The corresponding visual odometry output becomes locally noisy but little drift happens in global scale. This confirms to the theory proposed in this paper that the motion estimation is insensitive to the roll and pitch angle noise. Fig. 8(b) presents a more complete comparison with respect to different add-in noise. Note that with roll and pitch angle noise, we only prove upper bound of the position drift on horizontal plane but not in vertical direction. The light blue colored bars indicate that position drift does accumulate in vertical direction. A possible solution of fixing the drift is using elevation of the vehicle measured by the altimeter. As expected, the position drift from yaw angle noise and altimeter noise accumulates overtime (yellow and brown colored bars).

8 Conclusion and Future Work

When using INS orientation readings in solving a visual odometry problem, the noise contained in the INS measurements can affect the vehicle motion estimation, causing the position estimate to drift. The proposed method reduces the accumulation of the position estimation error in two ways. First, we assume the imaged ground is locally flat and online estimate the inclination angles, and second, we reproject features with their depth direction perpendicular to the ground. This way, the translation error from the INS orientation noise cancels itself partially, resulting in a slow position drift. The method is tested on a full-scale helicopter for 16km of flying experiments. The results indicate a relative error of less then 1%.

References

1. D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry for ground vechicle applications,” *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.
2. M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the mars exploration rovers,” *Journal of Field Robotics*, vol. 24, no. 2, pp. 169–186, 2007.
3. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, The MIT Press, 2005.
4. O. Amidi, T. Kanade, and K. Fujita, “A visual odometer for autonomous helicopter flight,” *Robotics and Autonomous Systems*, vol. 28, no. 2–3, pp. 185–193, 1999.
5. S. Nuske, J. Roberts, and G. Wyeth, “Robust outdoor visual localization using a three-dimensional-edge map,” *Journal of Field Robotics*, vol. 26, no. 9, pp. 728–756, 2009.
6. J. Kelly, S. Saripalli, and G. Kukhatme, “Combined visual and inertial navigation for an unmanned aerial vehicle,” *Spriger Tracts in Advanced Robotics*, vol. 42, 2008.
7. K. Konolige, M. Agrawal, and J. Sol, “Large-scale visual odometry for rough terrain,” *Robotics Research*, vol. 66, p. 201212, 2011.
8. G. Klein and D. Murray, “Parallel tracking amd mapping for small AR workspaces,” in *Proc. of the International Symposium on Mixed and Augmented Reality*, Nara, Japan, Nov. 2007, pp. 1–10.
9. S. Weiss, “Vision based navigation for micro helicopters,” Ph.D. dissertation, ETH Zurich, 2012.
10. J. Artieda, J. Sebastian, P. Campoy, and et al, “Viusal 3-d SLAM from UAVs,” *Journal of Intelligent and Robotic Systems*, vol. 55, 2009.
11. G. Conte and P. Doherty, “Vision-based unmanned aerial vehicle navigation using georeferenced information,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, p. 10, 2009.
12. F. Caballero, L. Merino, J. Ferruz, and A. Ollero, “Vision-based odometry and SLAM for medium and high altitude flying UAVs,” *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1-3, pp. 137–161, 2009.
13. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York, Cambridge University Press, 2004.
14. R. Andersen, “Modern methods for robust regression.” *Sage University Paper Series on Quantitative Applications in the Social Sciences*, 2008.
15. B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of Imaging Understanding Workshop*, 1981, pp. 121–130.