

# Visual–Inertial Combined Odometry System for Aerial Vehicles

Ji Zhang and Sanjiv Singh

*The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213*

Received 10 October 2015; accepted 10 October 2015

The requirement to operate aircraft in GPS-denied environments can be met by using visual odometry. Aiming at a full-scale aircraft equipped with a high-accuracy inertial navigation system (INS), the proposed method combines vision and the INS for odometry estimation. With such an INS, the aircraft orientation is accurate with low drift, but it contains high-frequency noise that can affect the vehicle motion estimation, causing position estimation to drift. Our method takes the INS orientation as input and estimates translation. During motion estimation, the method virtually rotates the camera by reparametrizing features with their depth direction perpendicular to the ground. This partially eliminates error accumulation in motion estimation caused by the INS high-frequency noise, resulting in a slow drift. We experiment on two hardware configurations in the acquisition of depth for the visual features: 1) the height of the aircraft above the ground is measured by an altimeter assuming that the imaged ground is a local planar patch, and 2) the depth map of the ground is registered with a two-dimensional laser in a push-broom configuration. The method is tested with data collected from a full-scale helicopter. The accumulative flying distance for the overall tests is approximately 78 km. We observe slightly better accuracy with the push-broom laser than the altimeter. © 2015 Wiley Periodicals, Inc.

## 1. INTRODUCTION

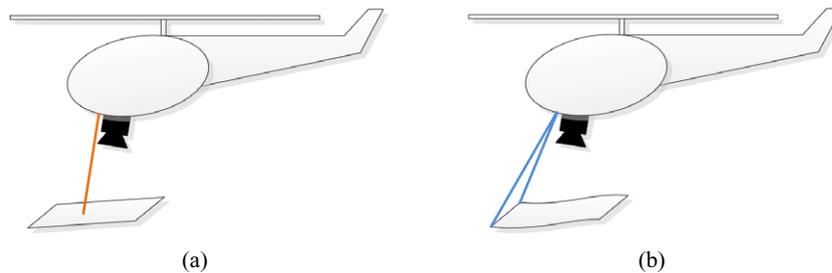
This paper addresses the problem of vision-based motion estimation for an aerial vehicle. Typically, vision-based methods are useful in the case in which GPS is unavailable or insufficiently accurate. On aerial vehicles, continuously accurate GPS positioning can be hard to insure, especially when the vehicle flies at a high velocity. Herein we present a method that combines vision and an inertial navigation system (INS). The method is designed for a full-scale aircraft equipped with a high-accuracy INS. For such an INS, the roll and pitch angles are measured by integration of gyro and accelerometer readings. The drift in these angles is corrected by the gravity vector. The yaw angle is from integration of gyro measurements only and is subject to drift. However, the drift is only at a small rate ( $< 0.02^\circ/\text{km}$ ) and is much lower than what can be estimated by most vision systems. With this INS, we consider the orientation to be reliable. However, the translation from integration of the accelerometer readings is less accurate ( $\approx 2\%$  of distance traveled). Our goal, therefore, is to improve/estimate translation with a vision system given that a high-accuracy INS is available. The method directly takes orientation from the INS as input.

Even with a high-accuracy INS, high-frequency noise is still present due to such factors as time synchronization imprecision and nonrigidity of the mounting hardware. The noise is further exaggerated by high-frequency vibration of the aircraft engine and rotors (we observe  $1^\circ\text{--}2^\circ$  high-frequency noise in INS orientation). The noise can affect

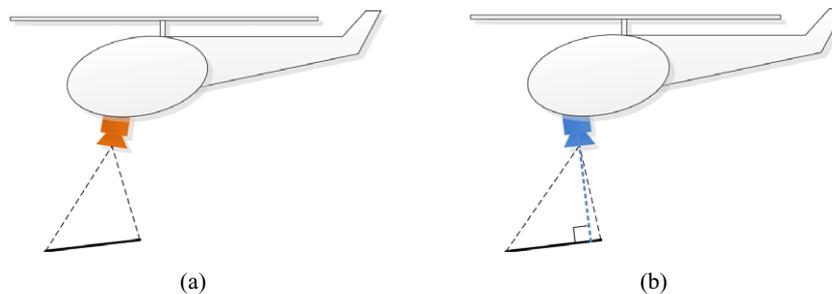
motion estimation causing errors to accumulate and become drift over time. To deal with the noise in the INS orientation, we propose to reparametrize features with their depth direction perpendicular to the ground. This is equivalent to rotating the camera virtually to be perpendicular to the ground, as shown in Figure 2. By doing this, we can partially eliminate the accumulation of the translation estimation error—the accumulated translation error introduced by roll and pitch angle noise from the INS partially cancels itself overtime, especially when the vehicle flies at a constant altitude. Our analysis also shows that it is hard to prevent propagation of the yaw angle noise from the INS and the altimeter noise in motion estimation, but we can only reduce the noise amount from the error sources.

Common methods of visual odometry applications employ two cameras forming a stereo pair (Cheng, Maimone, & Matthies, 2005; Maimone, Cheng, & Matthies, 2007; Nister, Naroditsky, & Bergen, 2006). Stereo cameras fixed on the aircraft can be used to recover full six-degrees-of-freedom (6-DOF) motion, but this requires the baseline between the cameras to be at least a nontrivial fraction of the vehicle altitude 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 cannot be insured.

In this study, a monocular camera was used looking downward toward the ground. The depth of the image



**Figure 1.** Two sensor configurations involved in this study. In (a), an altimeter measures the distance of the vehicle above the ground assuming that the ground is a locally flat patch with two rotation DOFs. In (b), a 2D laser in a push-broom configuration registers a depth map of the ground using estimated motion of the visual odometry.



**Figure 2.** Illustration of feature reparametrization. Part (a) shows the real orientation of the camera. During motion estimation, features are reparametrized with their depth direction perpendicular to the ground. This is equivalent to rotating the camera to a virtual point perpendicular to the ground, as in (b). We will show that using features associated with the virtual camera for motion estimation can decelerate the accumulation of the position error.

features is obtained by additional sensors in two hardware configurations: 1) [Figure 1(a)] an altimeter measures the distance of the vehicle above the ground assuming that the ground is a locally flat patch with two rotation DOFs (roll and pitch), and 2) [Figure 1(b)] a two-dimensional (2D) laser in a push-broom configuration registers a depth map using the estimated motion of the visual odometry. In the altimeter configuration, the method models the imaged ground to be a local planar patch, and concurrently estimates translation and inclination angles of the ground patch. Even though the imaged ground is not exactly planar, for example due to vegetation and hilly terrain, the method eliminates outlier features with large tracking errors during motion estimating, mostly those on top of trees or hills, keeping the assumption valid in practice. In the push-broom laser configuration, the method uses the registered depth map to associate depth information to image features in order to solve for motion. Our experiments show that the push-broom laser configuration conducts slightly better accuracy than the altimeter configuration due to more accurate depth information.

The contributions of the paper are 1) a method that improves estimation of translation for a high-accuracy INS, 2) a feature reparametrization technique to partially remove odometry drift, and 3) experimentation of two hardware configurations in obtaining depth for image features. The

method is tested with data collected from flying tests with 78 km of travel overall.

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

## 2. RELATED WORK

Vision-based methods are now common for vehicle state estimation (Amidi, Kanade, & Fujita, 1999; Kelly, Saripalli, & Kukhatme, 2008; Weiss, Scaramuzza, & Siegwart, 2011). Typically, the problem solves 6-DOF camera motion in an arbitrary environment. When stereo cameras are used (H. Badino & Kanade, 2013; Warren & Upcroft, 2013), the relative poses of the two cameras function as a constraint that helps to solve the motion estimation problem, typically through bundle adjustment (Achtelik, Bachrach, He, 2009; Howard, 2008). For example, Konolige et al.'s stereo visual odometry recovers the motion of stereo cameras on a ground vehicle (Konolige, Agrawal, & Sol, 2011). The method is integrated with an inertial measurement unit (IMU) that

handles orientation drift in long-distance navigation. For a monocular camera (Guizilini & Ramos, 2011; Paz, Pinies, & Tardos, 2008; Roger-Verdeguer, 2012), if the camera motion is unconstrained and no prior knowledge is assumed about the environment, the scale ambiguity is generally unsolvable (Engel, Sturm, & Cremers, 2013; Forster, Pizzoli, & Scaramuzza, 2014; Newcombe, Lovegrove, & Davison, 2011). The camera motion can be recovered up to similarity using the five-point algorithms (Li & Hartley, 2006; Nister, 2004). Further, Klein and Murray develop a visual simultaneous localization and mapping (SLAM) method by parallel tracking and mapping of monocular imagery (Klein & Murray, 2007). The method is improved by Weiss and modified to be a visual odometry method (Weiss et al., 2013).

When using a monocular camera, if the camera motion follows certain constraints, the scale ambiguity can be solved in constrained cases. For example, Scaramuzza et al.'s visual odometry method uses an omnidirectional camera mounted on a vehicle that follows a nonholonomic motion constraint (Scaramuzza, 2011). The method can be used to solve the vehicle motion with a low computation cost, and it shows significantly improved accuracy compared to the unconstrained case. Also shown by Scaramuzza, it is possible to recover the scale ambiguity with the nonholonomic constraint if the camera is mounted with an offset to the vehicle rotation center (Scaramuzza, 2009). On the other hand, if depth information for the environment is available, it can also help to solve the motion estimation problem. A number of visual odometry methods are developed using RGB-D cameras (Engelhard, Endres, Hess, Sturm, & Burgard, 2011; Henry, Krainin, Herbst, Ren, & Fox, 2012; Huang et al., 2011; Kerl, Sturm, & Cremers, 2013), which can provide visual images and the associated depth for each pixel. When using imagery with a known depth, bundle adjustment becomes less necessary because an important role of bundle adjustment is to recover the motion of the camera and the depth of the features simultaneously.

To solve scale, another option is to attach an IMU to the camera (Kneip, Chli, & Siegwart, 2011; Martinelli, 2012). These methods solve for motion with IMU measurements as the motion prior, where the acceleration readings help to determine scale. However, if the vehicle travels at a constant velocity, scale starts to drift. This problem is especially true for full-scale aircrafts, which do not oscillate as often as micro aerial vehicles. In our system, the height above the ground is measured by a range sensor, ensuring the scale is accurate without drift. Most of the methods in this category use filters to integrate visual odometry with inertial measurements (Hesch, Kottas, Bowman, & Roumeliotis, 2013; Jones & Soatto, 2011; Li & Mourikis, 2013). For example, Weiss et al. (2013) construct a visual-internal system that integrates output from parallel tracking and mapping (PTAM) visual odometry (Klein & Murray, 2007) with an IMU. The system also simultaneously estimates the bias of

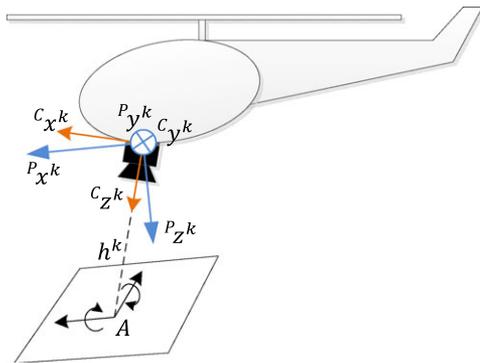
the IMU and extrinsic parameters between the camera and the IMU. In essence, these methods use the IMU as the backbone, but the accuracy in 6-DOF motion estimation depends on image features. Our method is designed to work with a high-accuracy INS and in a different category. We fully trust the INS orientation and focus only on the translation. Our effort is devolved to dealing with high-frequency noise in the INS orientation in order to improve translation accuracy.

When visual odometry is used on an aircraft that flies at a relatively high altitude, a certain assumption about the ground, such as an assumption that it is locally flat, can help to solve the motion. For example, Conte and Doherty's visual navigation system considers the ground to be flat and level (Conte & Doherty, 2009). The vehicle motion is solved by planar homography between ground images. The method also uses georeferenced aerial images to correct for visual odometry drift. Caballero et al.'s visual odometry also assumes flat ground and uses planar homography (Caballero, Merino, Ferruz, & Ollero, 2009). However, the method does not require the ground to be level, and its orientation is recovered online with respect to the vehicle. The scale is solved by the vehicle elevation above the ground measured by a range sensor.

Our method with the altimeter configuration [Figure 1(a)] is similar to the method of Caballero et al. (2009) in the sense that both assume the imaged ground is locally flat but not necessarily level. The difference is that our method does not rely on planar homography. The orientation from the INS is directly used in solving for translation in a tightly coupled fashion. The result is that it solves a problem with a lower DOF. Further, our method with the push-broom laser configuration [Figure 1(b)] is similar to the method of Huang et al. (2011) in that both methods utilize features associated with depth, without using bundle adjustment. However, since the 2D laser does not provide depth for each image pixel, our method registers a depth map from which a depth is acquired for the image features. This paper is an extended version of our conference paper (Zhang & Singh, 2014). It extends the method from using an altimeter to a push-broom laser with more experiments conducted.

### 3. ASSUMPTIONS AND COORDINATE SYSTEMS

The visual odometry problem addressed in this paper is to estimate the motion of an aerial vehicle using monocular vision, an INS, and an altimeter or a push-broom laser. We assume that the camera is well-modeled as a pinhole camera (Hartley & Zisserman, 2004). The camera's intrinsic parameters are known from precalibration, and the lens distortion is removed. We also assume that a high-accuracy INS is available on the aircraft, which is calibrated and whose extrinsic parameters with respect to the camera are known. The vision system has to work when the aircraft altitude above the ground is over certain level such that image



**Figure 3.** 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. In the altimeter configuration [Figure 1(a)], the distance from  $A$  to the origin of  $\{C^k\}$ ,  $h^k$ , is measured by an altimeter, and the ground patch is modeled with roll and pitch DOF around  $A$ .

features can reliably track. Toward that end, we assume that a GPS single is available upon taking off. The GPS single helps initialize the INS in yaw orientation and guides the aircraft to fly above a certain altitude before the proposed method takes over. As 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  $\mathcal{I}$  to denote the set of feature points. We define two coordinate systems.

- The 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  $i$ ,  $i \in \mathcal{I}$ , in  $\{I^k\}$  is denoted as  ${}^I x_i^k$ .
- The camera coordinate system  $\{C\}$  is a 3D coordinate system. As shown in Fig. 3, 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  $i$ ,  $i \in \mathcal{I}$ , in  $\{C^k\}$  is denoted as  ${}^C X_i^k$ .

To work with the altimeter configuration [Figure 1(a)], we model the imaged ground as a locally flat patch, as

shown in Figure 3. 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 a roll and pitch DOF around  $A$ . Here, we define another coordinate system.

- Parallel to the ground coordinate system  $\{P\}$  is a 3D coordinate system. The origin of  $\{P\}$  coincides with the origin of  $\{C\}$ , and the  $x$ - $y$  plane is parallel to the ground with the  $x$ -axis pointing to the forward direction of the vehicle. The  $z$ -axis points perpendicularly to the ground. A point  $i$ ,  $i \in \mathcal{I}$ , in  $\{P^k\}$  is denoted as  ${}^P X_i^k$ .

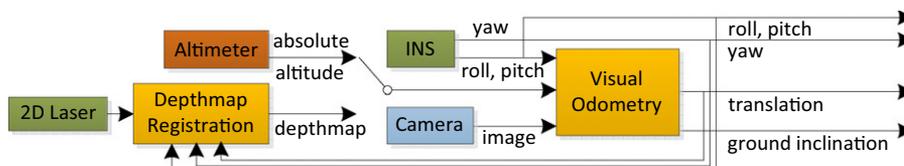
#### 4. SOFTWARE SYSTEM DIAGRAM

The system diagram of the visual odometry software is shown in Figure 4. The system takes camera images and orientation from the INS as input, and it computes the translation. The method has the option of using the absolute altitude measured by an altimeter or the depth map provided by a 2D laser. In the first case, the method assumes the imaged ground to be a local planar patch. It estimates the inclination of the planar patch online in order to recover the translation. In the second case, the method uses the estimated motion to register the laser points on a depth map, from which it associates depth information to the tracked visual features to estimate motion.

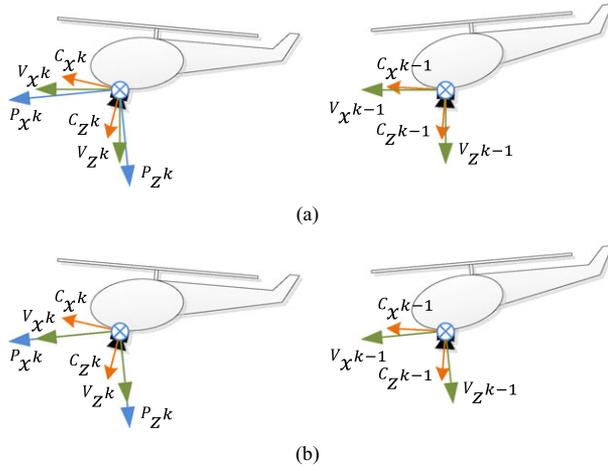
#### 5. VISUAL ODOMETRY METHOD

##### 5.1 Method Intuition

Figure 5 presents the key idea of 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 let  $\{V^k\}$  be a coordinate system with its origin coinciding with that of  $\{C^k\}$ . Initially,  $\{V^{k-1}\}$  and  $\{V^k\}$  are rotated to the horizontal orientation as shown in Figure 5(a), using orientation from the INS. Then, through motion estimation,  $\{V^{k-1}\}$  and  $\{V^k\}$  are rotated to be parallel to  $\{P^k\}$ , the coordinate system parallel to the ground at frame  $k$ , as shown in Figure 5(b). During motion estimation,  $\{V^{k-1}\}$  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.



**Figure 4.** Visual odometry software system diagram.



**Figure 5.** Definition of  $\{V_{k-1}\}$  and  $\{V_k\}$ . 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 to align with the horizontal orientation as illustrated in (a), using orientation from the INS. Then, through motion estimation,  $\{V_{k-1}\}$  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 DOF with respect to  $\{C_{k-1}\}$  and  $\{C_k\}$ .

## 5.2 Mathematical Derivation

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

$$\alpha^l \mathbf{X}_i^l = \mathbf{K}^C \mathbf{X}_i^l, \quad (1)$$

where  $\alpha$  is a scale factor, and  $\mathbf{K}$  is the camera's intrinsic matrix, which is known from precalibration (Hartley & Zisserman, 2004).

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

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

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

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

$${}^V \tilde{\mathbf{X}}_i^l = {}^V \mathbf{X}_i^l / {}^V z_i^l, \quad (3)$$

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

Let  $\Delta_x^k$ ,  $\Delta_y^k$ , and  $\Delta_z^k$  be the vehicle translation in the  $x$ -,  $y$ - and  $z$ -directions between frames  $k-1$  and  $k$ , and let  $\Delta_\phi^k$

be the corresponding yaw rotation between the two frames. From the vehicle motion, we can establish a relationship between  $\{V^{k-1}\}$  and  $\{V^k\}$ ,

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

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

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

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

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

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

where  ${}^V \tilde{x}_i^l$  and  ${}^V \tilde{y}_i^l$ ,  $l \in \{k-1, k\}$ , are the first and second entries of  ${}^V \tilde{\mathbf{X}}_i^l$ , respectively,  ${}^V z_i^k$  is the third entry of  ${}^V \mathbf{X}_i^k$ , and  $s$  is a scale factor. Combining Eqs. (5) and (6) to eliminate  $s$ , we have

$$\begin{aligned} \frac{{}^V \tilde{y}_i^{k-1}}{{}^V z_i^k} \Delta_x^k + \frac{{}^V \tilde{x}_i^{k-1}}{{}^V z_i^k} \Delta_y^k - ({}^V \tilde{x}_i^{k-1} {}^V \tilde{x}_i^k + {}^V \tilde{y}_i^{k-1} {}^V \tilde{y}_i^k) \Delta_\phi^k \\ + {}^V \tilde{x}_i^{k-1} {}^V \tilde{y}_i^k + {}^V \tilde{y}_i^{k-1} {}^V \tilde{x}_i^k = 0. \end{aligned} \quad (8)$$

In the push-broom laser configuration, the depth of the feature point  ${}^V z_i^k$  can be associated from the depth map. Equation (8) contains three unknowns,  $\Delta_x^k$ ,  $\Delta_y^k$ ,  $\Delta_\phi^k$ , as a linear function. In the altimeter configuration, the depth is obtained from the locally planar ground assumption, discussed as follows.

Recall that  $h^k$  is the distance from the vehicle to the ground patch, and the ground patch has a roll and pitch DOF around point  $A$  in Figure 3. Let  ${}^V_p \theta^k$  and  ${}^V_p \psi^k$  be the roll and pitch angles from  $\{P^k\}$  to  $\{V^k\}$ . The depth  ${}^V z_i^k$  can be computed from a simple geometry relationship. This is a linear approximation, but it works well in practice.

$${}^V z_i^k = h^k [1 - ({}^V \tilde{x}_i^k - {}^C_p \psi^k) {}^V_p \theta^k - ({}^V \tilde{y}_i^k - {}^C_p \theta^k) {}^V_p \psi^k]. \quad (9)$$

Combining Eqs. (8) and (9), we have

$$\frac{a \Delta_x^k + b \Delta_y^k}{c + d {}^V_p \psi^k + e {}^V_p \theta^k} + f \Delta_\psi^k + g = 0, \quad (10)$$

where

$$a = {}^V \tilde{y}_i^{k-1}, \quad b = -{}^V \tilde{x}_i^{k-1}, \quad c = h^k, \quad d = -h^k ({}^V \tilde{x}_i^k - {}^C_p \psi^k), \quad (11)$$

$$e = -h^k ({}^V \tilde{y}_{(k,i)}^k - {}^C_p \theta^k), \quad f = -{}^V \tilde{x}_i^k {}^V \tilde{x}_i^{k-1} - {}^V \tilde{y}_i^k {}^V \tilde{y}_i^{k-1}, \quad (12)$$

$$g = {}^V \tilde{x}_i^k {}^V \tilde{y}_i^{k-1} - {}^V \tilde{y}_i^k {}^V \tilde{x}_i^{k-1}. \quad (13)$$

In Eq. (10), we have a total of five unknowns:  $\Delta_x^k$ ,  $\Delta_y^k$ ,  $\Delta_\phi^k$ ,  ${}^V_p \theta^k$ ,  ${}^V_p \psi^k$ . 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 Eq. (10) can be solved linearly with three or more feature

points. Next, we will give a linear and a nonlinear method to solve the function. Both methods will be useful for the visual odometry algorithm in the next section.

### 5.2.1 Linear Method

Consider  $\Delta_x^k, \Delta_y^k, \Delta_\phi^k$  as the unknowns. For  $m, m \geq 3$ , feature points, stack Eq. (10) for each feature. This gives us a linear function in the form of

$$\mathbf{A}\mathbf{X}_L = \mathbf{b}, \quad (14)$$

where  $\mathbf{A}$  is an  $m \times 3$  matrix,  $\mathbf{b}$  is an  $m \times 1$  vector, and  $\mathbf{X}_L$  contains the unknowns,  $\mathbf{X}_L = [\Delta_x^k, \Delta_y^k, \Delta_\phi^k]^T$ . Solving Eq. (14) with the singular value decomposition method (Hartley & Zisserman, 2004), we can recover  $\mathbf{X}_L$ .

### 5.2.2 Nonlinear Method

For  $m, m \geq 5$ , feature points, stack Eq. (10) for each feature and reorganize the function into the following form:

$$\mathbf{f}(\mathbf{X}_N) = \mathbf{b}, \quad (15)$$

where  $\mathbf{f}$  is a nonlinear function with five inputs and  $m$  outputs.  $\mathbf{b}$  is a  $m \times 1$  vector, and  $\mathbf{X}_N$  contains the unknowns,  $\mathbf{X}_N = [\Delta_x^k, \Delta_y^k, \Delta_\phi^k, \theta^k, \psi^k]^T$ . Compute the Jacobian matrix of  $\mathbf{f}$  with respect to  $\mathbf{X}_N$ , denoted as  $\mathbf{J}$ , where  $\mathbf{J} = \partial \mathbf{f} / \partial \mathbf{X}_N$ . Equation (15) can be solved through nonlinear iterations using the Levenberg-Marquardt method (Hartley & Zisserman, 2004),

$$\mathbf{X}_N \leftarrow \mathbf{X}_N + [\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})]^{-1} \mathbf{J}^T [\mathbf{b} - \mathbf{f}(\mathbf{X}_N)], \quad (16)$$

where  $\lambda$  is a scale factor determined by the Levenberg-Marquardt method.

## Algorithm

### Algorithm 1: Translation Estimation for the Altimeter Configuration

**input** :  $\mathbf{x}_i^{k-1}, \mathbf{x}_i^k, i \in \mathcal{S}, \theta_{INS}^{k-1}, \psi_{INS}^{k-1}, \theta_{INS}^k, \psi_{INS}^k, h^k$

**output** :  $\Delta_x^k, \Delta_y^k, \Delta_z^k$

**begin**

Rotate  $\{V^l\}$  to the horizontal orientation by  $\theta_V^l \leftarrow \theta_{INS}^l, \psi_V^l \leftarrow \psi_{INS}^l, l \in \{k-1, k\}$ ;

Compute  ${}^V \tilde{\mathbf{x}}_i^{k-1}, {}^V \tilde{\mathbf{x}}_i^k$  for  $i \in \mathcal{S}$  based on (1-3);

Use  $i \in \mathcal{S}$  to compute  $\Delta_x^k, \Delta_y^k, \Delta_\phi^k$  linearly by setting  $\theta_p^k, \psi_p^k \leftarrow 0$  based on (14);

**for** a number of iterations **do**

    Compute image reprojection error (IRE) for  $i \in \mathcal{S}$ , then compute a weight for  $i \in \mathcal{S}$  using the IREs;

**for** a number of iterations **do**

        Use  $i \in \mathcal{S}$  to update  $\Delta_x^k, \Delta_y^k, \Delta_\phi^k, \theta_p^k, \psi_p^k$  for one iteration based on (16);

        Rotate  $\{V^l\}$  by  $\theta_V^l \leftarrow \theta_V^l + \theta_p^k$  and  $\psi_V^l \leftarrow \psi_V^l + \psi_p^k, l \in \{k-1, k\}$ , then  $\theta_p^k, \psi_p^k \leftarrow 0$ ;

        Project  ${}^V \tilde{\mathbf{x}}_i^{k-1}, {}^V \tilde{\mathbf{x}}_i^k, i \in \mathcal{S}, \Delta_x^k, \Delta_y^k, \Delta_\phi^k$  to the newly rotated  $\{V^{k-1}\}$  and  $\{V^k\}$ ;

**if** the nonlinear optimization converges **then**

            Break;

**if** the robust fitting converges **then**

        Break;

Compute  $\Delta_z^k$  based on (7) as the weighted average of the features;

Return  $\Delta_x^k, \Delta_y^k, \Delta_z^k$ ;

## 5.3 Algorithm

Algorithm 1 presents the visual odometry algorithm for the altimeter configuration. The algorithm first initializes using readings from the INS. Let  $\theta_{INS}^{k-1}$  and  $\psi_{INS}^{k-1}$  be the roll and pitch angles of the vehicle at frame  $k-1$ , measured by the INS, and let  $\theta_{INS}^k$  and  $\psi_{INS}^k$  be the corresponding angles at frame  $k$ . We rotate  $\{V^{k-1}\}$  and  $\{V^k\}$  to be horizontal 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, we set  $\theta_p^k, \psi_p^k \leftarrow 0$  and compute  $\Delta_x^k, \Delta_y^k, \Delta_\phi^k$  linearly. The result is used as initialization for the nonlinear optimization that solves the problem iteratively. At each iteration, the five unknowns  $\Delta_x^k, \Delta_y^k, \Delta_\phi^k, \theta_p^k, \psi_p^k$  are updated, then  $\{V^{k-1}\}$  and  $\{V^k\}$  are rotated to the newly updated orientation and the features are projected 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 (Andersen, 2008) framework to ensure robustness against large feature tracking errors. The algorithm assigns a weight for each feature based on the image reprojection error (IRE). Features with larger IREs are assigned with smaller weights, while features with 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,  $\Delta_x^k, \Delta_y^k$ . To obtain the  $z$ -translation,  $\Delta_z^k$ , we use Eq. (7) with the selected inlier features.  $\Delta_z^k$  is computed as a weighted average of the inlier features using the same weights generated by the robust fitting.

The visual odometry algorithm for the push-broom laser configuration is a simpler version of Algorithm 1. The

depth map is fitted into a plane, which gives us the inclination of the ground. Instead of rotating  $\{V^{k-1}\}$  and  $\{V^k\}$  gradually during the robust fitting, we only reparametrize the features once, and  $\{V^{k-1}\}$  and  $\{V^k\}$  are perpendicular to the ground. The nonlinear optimization is unnecessary. Only a linear method is needed to solve  $\Delta_x^k, \Delta_y^k, \Delta_\phi^k$  by stacking Eq. (8). We use the same robust fitting as in Algorithm 1 to assign weights to the features.

### 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 orientation because vertical orientation drift can be largely corrected by readings of the altimeter or the push-broom laser. We will derive the upper bound of the accumulated position drift using the altimeter configuration as an example. Error propagation for the push-broom laser configuration can be analyzed similarly.

We start with the INS roll and pitch angles. Recall that  $\theta_{INS}^l$  and  $\psi_{INS}^l, l \in \{k-1, k\}$ , are the roll and pitch inclination angles of the vehicle measured by the INS at frame  $l$ . Let us define  $\hat{\theta}_{INS}^l$  and  $\hat{\psi}_{INS}^l$  as their measurement values containing errors. Letting  $e_\theta^l$  and  $e_\psi^l$  be the corresponding errors, we have  $e_\theta^l = \hat{\theta}_{INS}^l - \theta_{INS}^l$  and  $e_\psi^l = \hat{\psi}_{INS}^l - \psi_{INS}^l$ . By examining each step in Algorithm 1, we find that  $e_\theta^l$  and  $e_\psi^l$  are introduced into the algorithm at the initialization step. With the INS measurements, the coordinate systems  $\{V^{k-1}\}$  and  $\{V^k\}$  are intended to be rotated to the horizontal orientation. However, because of  $e_\psi^l$  and  $e_\theta^l, \{V^{k-1}\}$  and  $\{V^k\}$  are not exactly aligned with the horizontal orientation. The roll and pitch differences between  $\{V^{k-1}\}$  and  $\{V^k\}$  are  $e_\theta^{k-1} - e_\theta^k$  and  $e_\psi^{k-1} - e_\psi^k$ , respectively. The angular differences are kept through the algorithm since the two coordinate systems are rotated simultaneously by the same angle. Here, note that in the altimeter configuration, the coordinate systems are rotated gradually through nonlinear iterations, while in the push-broom laser configuration they are rotated only once. However, the same rotation is always applied to both coordinate systems. 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  $\{P^k\}$ . Letting  ${}^C_V\hat{\theta}^{k-1}$  and  ${}^C_V\hat{\psi}^{k-1}$  be the measurement values of the roll and pitch angles from  $\{V^{k-1}\}$  to  $\{C^{k-1}\}, {}^C_V\hat{\theta}^{k-1}$  and  ${}^C_V\hat{\psi}^{k-1}$ , we can compute

$${}^C_V\hat{\theta}^{k-1} = {}^C_V\theta^{k-1} + e_\theta^{k-1} - e_\theta^k, \quad {}^C_V\hat{\psi}^{k-1} = {}^C_V\psi^{k-1} + e_\psi^{k-1} - e_\psi^k. \quad (17)$$

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

$${}^C X_i^{k-1} = \mathbf{R}_x({}^C_V\hat{\theta}^{k-1} + e_\theta^{k-1} - e_\theta^k) \mathbf{R}_y({}^C_V\hat{\psi}^{k-1} + e_\psi^{k-1} - e_\psi^k) {}^V X_i^{k-1}. \quad (18)$$

Correspondingly, we derive Eq. (10) again containing the errors,

$$\begin{aligned} & a \left( \frac{\Delta_x^k}{c + d_V^V \psi^k + e_\psi^V \theta^k} + e_\psi^{k-1} - e_\psi^k \right) \\ & + b \left( \frac{\Delta_y^k}{c + d_V^V \psi^k + e_\psi^V \theta^k} + e_\theta^{k-1} - e_\theta^k \right) \\ & + f \Delta_\phi^k + g = 0, \end{aligned} \quad (19)$$

where  $a, b, c, d, e, f,$  and  $g$  are defined in Eqs. (11)–(13).

Now, we compare Eq. (19) with Eq. (10). Note that after convergence of the nonlinear optimization in Algorithm 1, we have  ${}^V_P \psi^k, i_k^{PV} \rightarrow 0$ . In this condition, if we define  $\hat{\Delta}_x^k = \Delta_x^k + (e_\psi^{k-1} - e_\psi^k)h^k$  and  $\hat{\Delta}_y^k = \Delta_y^k + (e_\theta^{k-1} - e_\theta^k)h^k$ , and if we substitute the terms into Eq. (19), it becomes essentially the same as Eq. (10) except that  $\Delta_x^k$  and  $\Delta_y^k$  are replaced by  $\hat{\Delta}_x^k$  and  $\hat{\Delta}_y^k$ . Examining the expressions of  $\hat{\Delta}_x^k$  and  $\hat{\Delta}_y^k$ , we find that the terms are invariant with respect to different features. This indicates that if we use  $\hat{\Delta}_x^k$  and  $\hat{\Delta}_y^k$  as the measurement values of  $\Delta_x^k$  and  $\Delta_y^k$  for the case containing errors, Eq. (15) is satisfied for each of its  $m$  rows. Defining  $e_x^k$  and  $e_y^k$  as the estimation errors corresponding to  $\Delta_x^k$  and  $\Delta_y^k$ , we have

$$\begin{aligned} e_x^k &= \hat{\Delta}_x^k - \Delta_x^k = (e_\psi^{k-1} - e_\psi^k)h^k, \quad e_y^k = \hat{\Delta}_y^k - \Delta_y^k \\ &= (e_\theta^{k-1} - e_\theta^k)h^k. \end{aligned} \quad (20)$$

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  $n, n \in \mathbb{Z}^+$ ,

$$e_x = \sum_{k=1}^n e_x^k, \quad e_y = \sum_{k=1}^n e_y^k. \quad (21)$$

We want to find the upper bounds of  $|e_x|$  and  $|e_y|$ . Let us define  $E_\theta$  and  $E_\psi$  as the upper bounds of the roll and pitch errors from the INS, where  $|e_\theta^k| \leq E_\theta$  and  $|e_\psi^k| \leq E_\psi, k \in \{1, 2, \dots, n\}$ . Substituting Eq. (20) into Eq. (21), we can derive

$$\begin{aligned} e_x &= \sum_{k=2}^n (e_\psi^{k-1} - e_\psi^k)h^k = \sum_{k=2}^n (e_\psi^{k-1} - e_\psi^k)h^{(2)} \\ &+ \sum_{k=3}^n (e_\psi^{k-1} - e_\psi^k)(h^k - h^{(2)}) \\ &= \sum_{k=2}^n (e_\psi^{k-1} - e_\psi^k)h^{(2)} + \sum_{j=3}^n \sum_{k=j}^n (e_\psi^{k-1} - e_\psi^k)(h^j - h^{j-1}) \\ &= (e_\psi^1 - e_\psi^n)h^{(2)} + \sum_{j=3}^n (e_\psi^{j-1} - e_\psi^j)(h^j - h^{j-1}). \end{aligned} \quad (22)$$

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

$$|e_x| \leq 2E_\psi \left( h^{(2)} + \sum_{j=3}^n |h^j - h^{j-1}| \right). \quad (23)$$

Similarly, we can derive the upper bound of  $|e_y|$  as

$$|e_y| \leq 2E_\theta \left( h^{(2)} + \sum_{j=3}^n |h^j - h^{j-1}| \right). \quad (24)$$

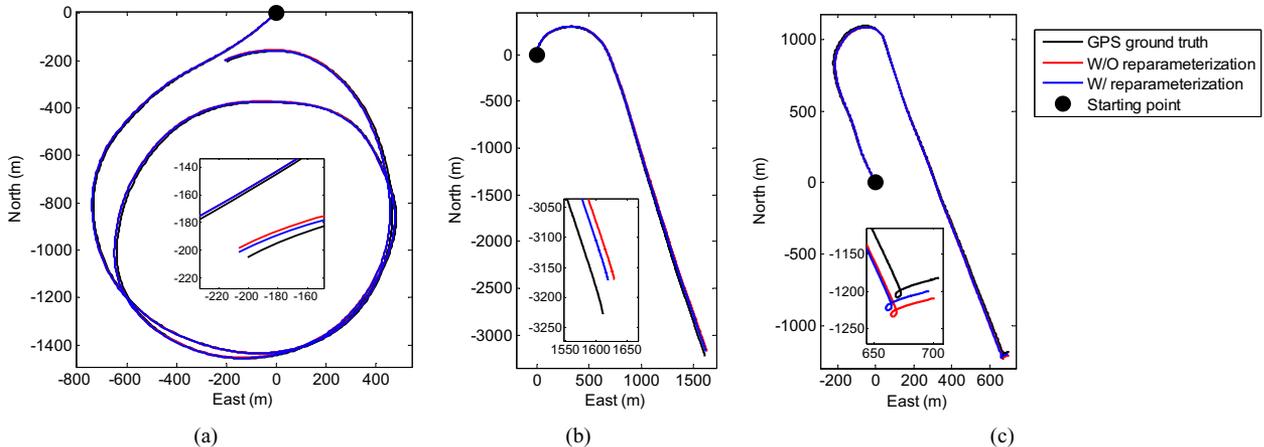
Equations (23) and (24) indicate that the accumulated translation error introduced by the roll and pitch noise from the INS is only relevant to the altitude change of the vehicle, regardless of the flying distance. In a special case in which the vehicle keeps a constant height above the ground during 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 in which 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  $|e_y| < 2E_\theta \sum_{j=3}^n |h^j - h^{j-1}|$ . For the push-broom laser configuration, the above procedure can be conducted with Eq. (8), which draws a similar conclusion.

Further, we find that the upper bound of the position drift introduced by the yaw angle noise from the INS and altimeter/push-broom laser noise is proportional to the flying distance. Without mathematical proof, the conclusion can be explained intuitively in that if the yaw angle is off by a constant value, the position estimation will constantly drift to the left or right side. Similarly, noise in the altimeter or push-broom laser readings will result in an under- or overestimated translation scale. In the next section, we use a high-quality laser altimeter/2D laser to reduce the features' depth error and therefore the translation estimation error.



**Figure 6.** (a) Helicopter used in the experiments. A downward pointing camera is mounted to the front of the helicopter. (b) Tracked features. A total of 450 feature points are tracked between image frames. The red dots are outlier features assigned with zero weights in Algorithm 1. The blue dots are inlier features used in the motion estimation.



**Figure 7.** Visual odometry outputs with feature reparameterization (blue) and without feature reparameterization (red) compared to GPS/INS ground truth (black) for three tests using the altimeter configuration. In each figure, we show a zoomed-in view of the trajectory ends. The subfigures from left to right correspond to Tests 1–3 in Table I. The average position error of the three tests is 0.57% of the flying distance with the feature reparameterization and 0.76% without the feature reparameterization, calculated using 3D coordinates and errors at the trajectory ends.

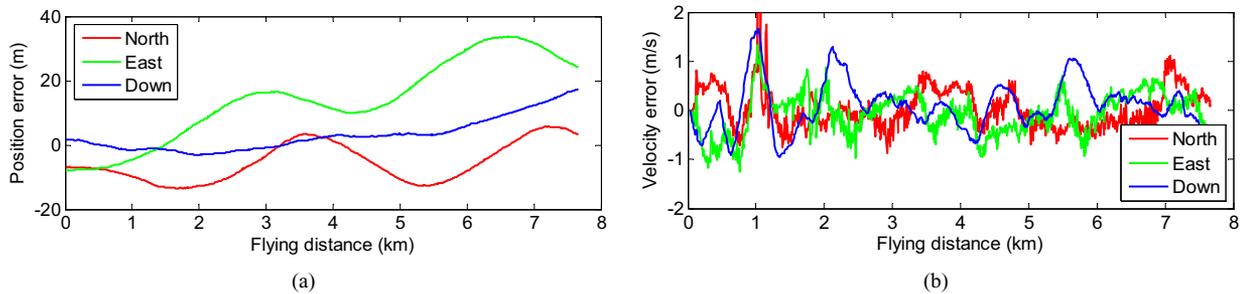
**Table I.** Configuration and accuracy of the three tests in Figure 7 (from left to right).

Test No.	Flying Distance (km)	Altitude (m)	Flying Speed (m/s)	Accuracy	
				W/O Reparam	W/Reparam
1	7.8	≈300	≈30	0.50%	0.39%
2	3.7	≈150	≈20	0.78%	0.73%
3	4.5	≈200	≈20	1.12%	0.78%

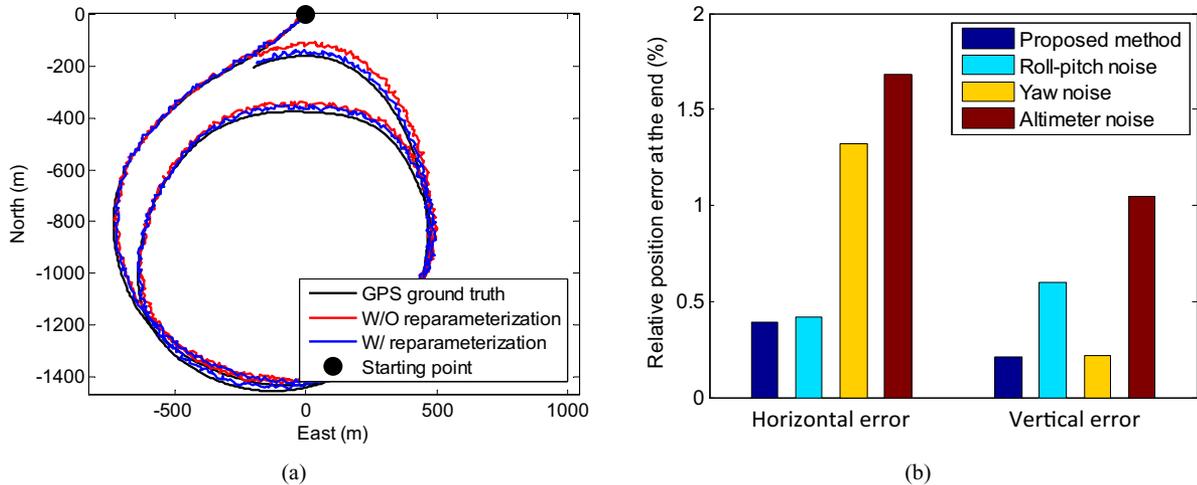
**7. EXPERIMENTS**

We obtain image sequences from a downward pointing camera mounted to a full-scale helicopter [Figure 6(a)]. The camera resolution is  $612 \times 512$  pixels with a horizontal field of view of  $75^\circ$ . The camera frame rate is set at 14 Hz. The helicopter is also equipped with a 2D laser and a high-accuracy GPS/INS. When testing in the altimeter configuration, the laser is rotated to point vertically down and used as a laser

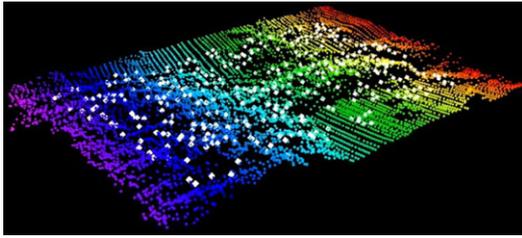
altimeter (only one laser beam pointing to the image center is used to obtain the absolute altitude). The orientation from integration of the IMU measurements is used by the visual odometry. Upon taking off, a GPS signal is available to the IMU, which initializes the yaw angle. The visual odometry is configured to start when the helicopter is over 100 m above the ground. At this time, the initialization is complete and only integration of the IMU measurements is



**Figure 8.** (a) Accumulated position drift and (b) velocity errors in Test 1 [Figure 7(a)].



**Figure 9.** (a) Blue: visual odometry output with artificial add-in noise, with feature reparameterization. The noise is added to the roll and pitch angles from the INS. Red: visual odometry output without feature reparameterization. Black: GPS/INS ground truth. (b) Relative errors with respect to different add-in noise. The noise is added to the roll and pitch angles, yaw angle from the INS, and altimeter reading, respectively. The angle noise follows a  $\sigma = 3^\circ$  Gaussian distribution, and the altimeter noise is 3% ( $\sigma$  value) of the altitude with a Gaussian distribution.



**Figure 10.** The depth map corresponding to Figure 6(b). The colored points are perceived by the push-broom laser and registered by the visual odometry estimation. The white points are features projected onto the ground.

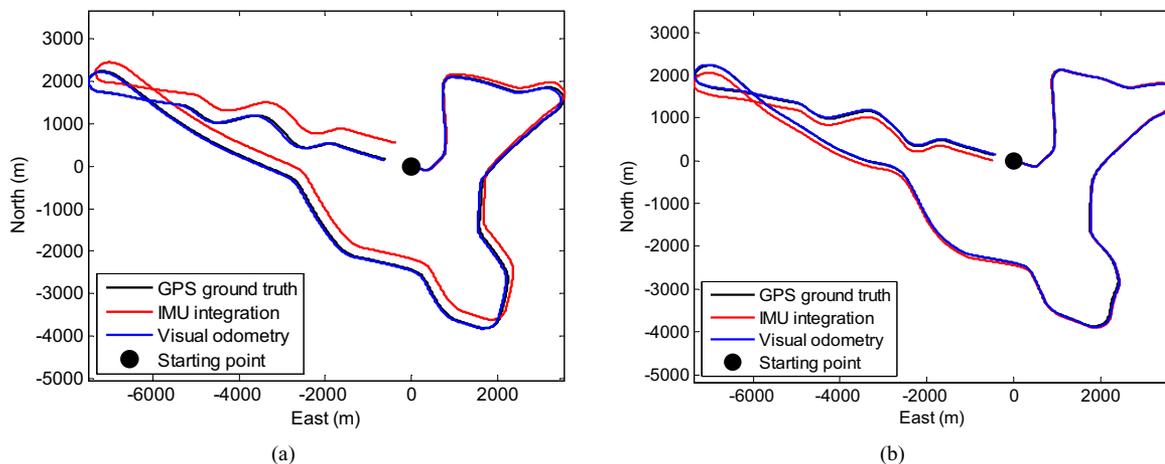
used by the visual odometry. During the tests, the position and orientation coupled with GPS are used as ground truth for comparison purposes.

The visual odometry program runs on a laptop computer with 2.5 GHz quad cores and 6 Gib memory in Linux, taking three-quarters of a single core. The feature tracking takes half of a core, and motion estimation and depth map registration together take another quarter of a core. The program selects a maximum of 450 Harris corners (Hartley & Zisserman, 2004) using the openCV library, and it tracks the feature points using the Kanade-Lucas-Tomasi method (Lucas & Kanade, 1981). To evenly distribute the feature points in the images, we separate the images into 9 ( $3 \times 3$ ) identical subregions. Each subregion provides up to 50 features. Figure 6 shows an example of the tracked features. The red dots are outliers assigned with zero weights in Algorithm 1, and the blue dots are inliers used in the motion estimation.

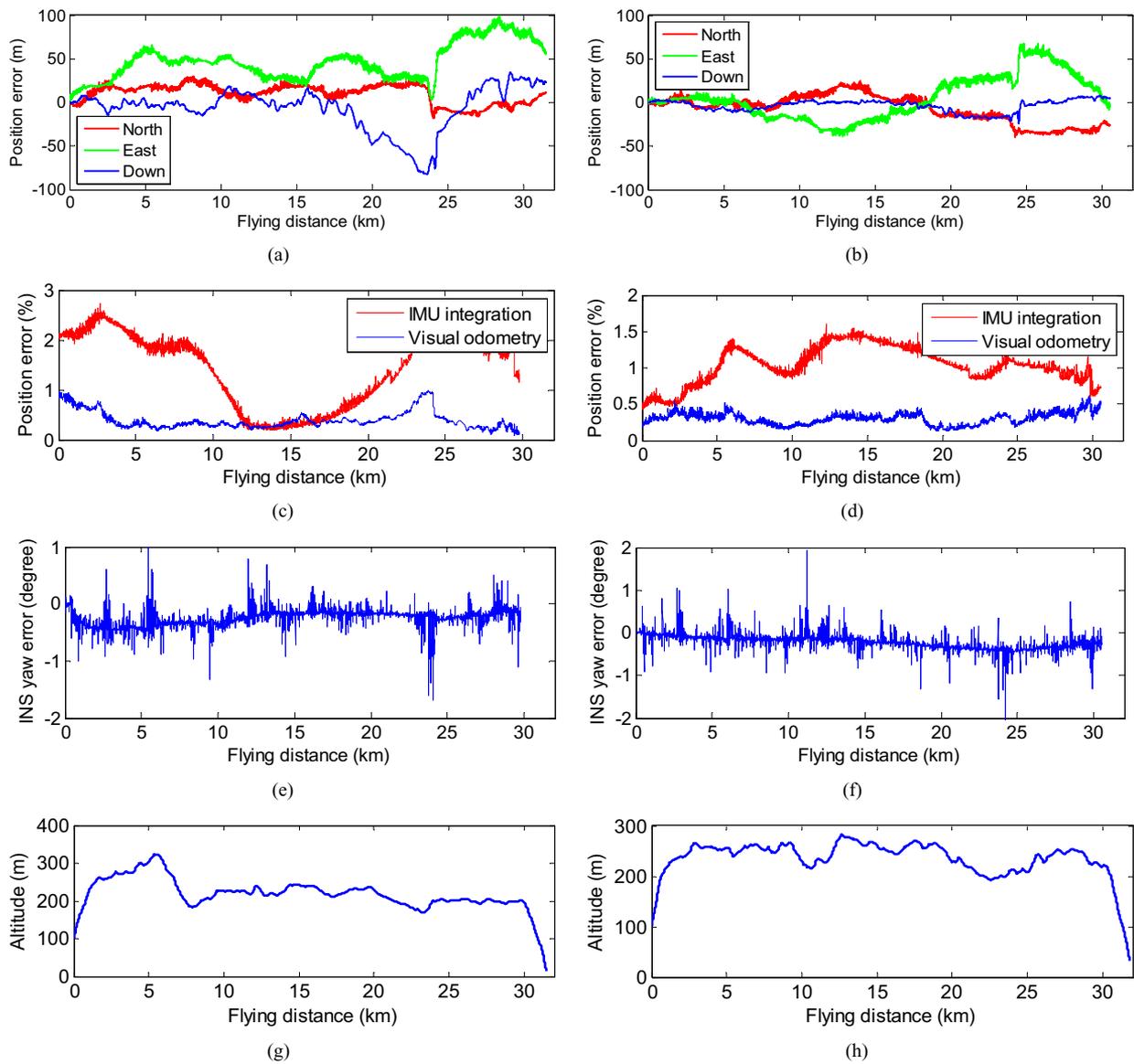
Figure 7 shows the results of the proposed method in three flight tests using the altimeter configuration. The blue curves are visual odometry outputs with feature reparametrization, the red curves are outputs with feature reparametrization removed, and the black curves are ground truth provided by the GPS/INS. More detailed configurations and an accuracy comparison of the three tests are shown in Table I. Tests 1–3 correspond to the subfigures in Figure 7 from left to right. Due to the short flying distance of the three tests, we only use errors at the trajectory ends to evaluate the accuracy. The average position error of the three tests is 0.57% of the flying distance with the feature reparametrization, and 0.76% without the feature reparametrization. The accuracy is calculated based on 3D coordinates.

We select Test 1 (the left subfigure in Figure 7) to show more details. Figure 8 presents position and velocity errors for Test 1. Figure 8(a) shows the accumulated position drift through the test. Figure 8(b) shows the absolute velocity errors. Most of the velocity errors are smaller than 1 m/s, while the average speed of the helicopter is about 30 m/s.

To inspect how sensor noise affects the motion estimation, we add artificial noise to the INS and altimeter readings. In Figure 9(a),  $\sigma = 3^\circ$ ,  $\mu = 1^\circ$ , nonwhite Gaussian noise is added to the roll and pitch angles from the INS. The corresponding visual odometry output becomes locally noisy but drifts little in global scale with feature reparametrization (blue curve). In comparison, the output without feature reparametrization drifts more (red curve). This confirms the theory proposed in this paper that the motion estimation is insensitive to the roll and pitch angle noise. Here, note that even though we simulate with Gaussian noise, the theory is not limited to a particular noise type. Figure 9(b)



**Figure 11.** Vehicle trajectories using the altimeter configuration (a) and the push-broom laser configuration (b). In both subfigures, the blue curves are the visual odometry outputs, the red curves are from the IMU integration, and the black curves are GPS/INS ground truth. The helicopter flies on the same route twice with the altimeter and the push-broom laser configurations, respectively. Each flight is over 30 km in length and lasts for 15 min.



**Figure 12.** Parts (a) and (b) are accumulated position drift corresponding to Figures 11(a) and 11(b), respectively. Using the push-broom laser configuration gives a slightly smaller drift than the altimeter configuration. Parts (c) and (d) show the relative position errors compared to the flying distance. This is calculated using the metric proposed by the KITTI odometry benchmark (Geiger, Lenz, & Urtasun, 2012; Geiger, Lenz, Stiller, & Urtasun, 2013) by averaging relative position errors using segmented trajectories at lengths of 1, 2, . . . , 10 km. Parts (e) and (f) show the INS yaw drift compared to that coupled with GPS. Except for the high-frequency noise, the yaw angle drifts less than  $0.5^\circ$  during the tests. In (g) and (h), we show the absolute altitude of the helicopter. The visual odometry is started when the helicopter is over 100 m above the ground.

presents a more complete comparison with respect to different add-in noise. With roll and pitch angle noise, we only prove the upper bound of the position drift in the horizontal direction but not in the vertical direction. The light-blue bars indicate that position drift accumulates in the vertical direction. A possible solution of fixing the drift is using

the altitude of the vehicle measured by the altimeter or the push-broom laser. As expected, the position drift from yaw angle noise accumulates in the horizontal direction (yellow bars), and the position drift from altimeter noise accumulates in both the horizontal and vertical directions (brown bars).

Further, we conduct long-distance tests to evaluate both the altimeter and the push-broom laser configurations. Because we only have one laser scanner on the helicopter, the same route is followed twice, one with the laser pointing vertically down to function as an altimeter, the other with the laser in a push-broom configuration. Figure 10 shows an example of the depth map. The colored points are registered laser points where the color codes the elevation, and the white points are image features projected onto the depth map. When associating depth to the features, we represent each point on the depth map using spherical coordinates with a distance and two angles. The points are stored in a 2D KD-tree (de Berg, van Kreveld, Overmars, & Schwarzkopf, 2008) using the two angular coordinates. For each feature point, the three closest points on the depth map are found, and the depth of the feature is calculated by projecting a ray through the plane formed by the three points.

The length of each flight is over 30 km. Figures 11(a) and 11(b) show the trajectories from the visual odometry (blue) and the estimation from integration of the IMU measurements (red) compared to the GPS/INS ground truth (black). We see that the translation drifts when using the INS only. With orientation from the INS and translation estimated by the proposed method, the position drift is significantly reduced.

We show more details of the test in Figure 12. Here, the left column corresponds to Figure 11(a) with the altimeter configuration, and the right column corresponds to Figure 11(b) with the push-broom laser configuration. In the first row, we show the accumulated position errors in three orthogonal directions. We observe that the position errors with the push-broom laser configuration [Figure 12(b)] are slightly smaller than those with the altimeter configuration [Figure 12(a)]. In the second row, we show the relative position errors compared to the distance traveled. Here, we adopt the metric used by the KITTI odometry benchmark (Geiger et al., 2012, 2013). The accuracy is calculated by averaging relative position errors using segmented trajectories at lengths of 1, 2, ..., 10 km. The proposed method estimates translation more accurately than the IMU integration. Also, the relative position errors at the trajectory ends are 0.20% and 0.09% of the distance traveled, respectively, for Figures 11(a) and 11(b).

In the third row of Figure 12, we show the drift of the INS yaw angle. The error is calculated by comparison to the GPS/INS output, where the IMU integration is coupled with GPS in a filter. During the 15 min duration of each test, the yaw angle drift is less than  $0.5^\circ$  except for the high-frequency noise. Our hypothesis is that the high-frequency noise is mostly caused by the vibration of the helicopter. Further, in the fourth row, we show the absolute altitude of the helicopter. As mentioned before, the visual odometry is started when the helicopter's absolute altitude is over 100 m.

## 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 position estimates to drift. The proposed method reduces the accumulation of the position estimation error by reparametrizing features with their depth direction perpendicular to the ground. In this way, the position error resulting from the INS orientation noise cancels itself partially, resulting in a slow drift. The method is tested on a full-scale helicopter for 78 km of overall flying experiments, with two different laser configurations. The results indicate most relative position errors to be less than 1% of the distance traveled. Additional work with the proposed method involves integration of the visual odometry estimation with position measurements from a GPS, and using the visual odometry estimation in a closed control loop to guide an autonomous helicopter.

## REFERENCES

- Achtelik, M., Bachrach, A., He, R., & et al. (2009). Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments. *Proceedings of SPIE*.
- Amidi, O., Kanade, T., & Fujita, K. (1999). A visual odometer for autonomous helicopter flight. *Robotics and Autonomous Systems*, 28(2–3), 185–193.
- Andersen, R. (2008). *Modern methods for robust regression*. Sage University Paper Series on Quantitative Applications in the Social Sciences.
- Caballero, F., Merino, L., Ferruz, J., & Ollero, A. (2009). Vision-based odometry and SLAM for medium and high altitude flying UAVs. *Journal of Intelligent and Robotic Systems*, 54(1–3), 137–161.
- Cheng, Y., Maimone, M., & Matthies, L. (2005). Visual odometry on the Mars exploration rovers. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* (pp. 903–910). Big Island, HI.
- Conte, G., & Doherty, P. (2009). Vision-based unmanned aerial vehicle navigation using geo-referenced information. *EURASIP Journal on Advances in Signal Processing*, 10.
- de Berg, M., van Kreveld, M., Overmars, M., & Schwarzkopf, O. (2008). *Computation geometry: Algorithms and applications*, 3rd ed. Springer.
- Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense visual odometry for a monocular camera. In *2013 IEEE International Conference on Computer Vision (ICCV)* (pp. 1449–1456). IEEE.
- Engelhard, N., Endres, F., Hess, J., Sturm, J., & Burgard, W. (2011). Real-time 3D visual SLAM with a hand-held RGB-D camera. In *RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*.
- Forster, C., Pizzoli, M., & Scaramuzza, D. (2014). SVO: Fast semi-direct monocular visual odometry. In *Proceedings of*

- the IEEE International Conference on Robotics and Automation.
- Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32, 1229–1235.
- Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3354–3361).
- Guizilini, V., & Ramos, F. (2011). Visual odometry learning for unmanned aerial vehicles. In *IEEE International Conference on Robotics and Automation*. Shanghai.
- Badino, H. A. Y., & Kanade, T. (2013). Visual odometry by multi-frame feature integration. In *Workshop on Computer Vision for Autonomous Driving (Collocated with ICCV 2013)*, Sydney.
- Hartley, R., & Zisserman, A. (2004). *Multiple view geometry in computer vision*. New York: Cambridge University Press.
- Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2012). RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research*, 31(5), 647–663.
- Hesch, J. A., Kottas, D. G., Bowman, S. L., & Roumeliotis, S. I. (2013). Towards consistent vision-aided inertial navigation. In *Algorithmic Foundations of Robotics X* (pp. 559–574).
- Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. In *IEEE International Conference on Intelligent Robots and Systems*, Nice, France.
- Huang, A., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., & Roy, N. (2011). Visual odometry and mapping for autonomous flight using an RGB-D camera. In *International Symposium on Robotics Research (ISRR)*.
- Jones, E. S., & Soatto, S. (2011). Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4), 407–430.
- Kelly, J., Saripalli, S., & Kukhatme, G. (2008). *Combined visual and inertial navigation for an unmanned aerial vehicle*. Springer Tracts in Advanced Robotics (p. 42).
- Kerl, C., Sturm, J., & Cremers, D. (2013). Robust odometry estimation for RGB-D cameras. In *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany.
- Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality* (pp. 1–10), Nara, Japan.
- Kneip, L., Chli, M., & Siegwart, R. (2011). Robust real-time visual odometry with a single camera and an imu. In *British Machine Vision Conference*.
- Konolige, K., Agrawal, M., & Sol, J. (2011). Large-scale visual odometry for rough terrain. *Robotics Research*, 66, 201–212.
- Li, H., & Hartley, R. (2006). Five-point motion estimation made easy. In *The 18th International Conference on Pattern Recognition* (pp. 630–633).
- Li, M., & Mourikis, A. I. (2013). High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6), 690–711.
- Lucas, B., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop* (pp. 121–130).
- Maimone, M., Cheng, Y., & Matthies, L. (2007). Two years of visual odometry on the Mars exploration rovers. *Journal of Field Robotics*, 24(2), 169–186.
- Martinelli, A. (2012). Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1), 44–60.
- Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 2320–2327).
- Nister, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6), 756–770.
- Nister, D., Naroditsky, O., & Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 3–20.
- Paz, L., Pinies, P., & Tardos, J. (2008). Large-scale 6-DOF SLAM with stereo-in-hand. *IEEE Transactions on Robotics*, 24(5), 946–957.
- Roger-Verdeguer, J. (2012). Visual odometry with failure detection for the aegeia uav. In *IEEE International Conference on Imaging Systems and Techniques*, Manchester, UK.
- Scaramuzza, D. (2009). Absolute scale in structure from motion from a single vehicle mounted camera by exploiting non-holonomic constraints. In *IEEE International Conference on Computer Vision*, Kyoto, Japan.
- Scaramuzza, D. (2011). 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision*, 95, 74–85.
- Warren, M., & Upcroft, B. (2013). High altitude stereo visual odometry. *Proceedings of Robotics: Science and Systems IX*.
- Weiss, S., Achtelik, M., Lynen, S., Achtelik, M., Kneip, L., Chli, M., & Siegwart, R. (2013). Monocular vision for long-term micro aerial vehicle state estimation: A compendium. *Journal of Field Robotics*, 30(5), 803–831.
- Weiss, S., Scaramuzza, D., & Siegwart, R. (2011). Monocular-SLAM-based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6), 854–874.
- Zhang, J., & Singh, S. (2014). Ins assisted monocular visual odometry for aerial vehicles. In *The 9th International Conference on Field and Service Robots (FSR)*.