Micromachines **2015**, *6*, 487-522; doi:10.3390/mi6040487

OPEN ACCESS

micromachines

ISSN 2072-666X

www.mdpi.com/journal/micromachines

Article

Invariant Observer-Based State Estimation for Micro-Aerial Vehicles in GPS-Denied Indoor Environments Using an RGB-D Camera and MEMS Inertial Sensors

Dachuan Li [1,*], Qing Li [1], Liangwen Tang [2], Sheng Yang [1], Nong Cheng [1] and Jingyan Song [1]

[1] Department of Automation, Tsinghua University, Beijing 100084, China;
E-Mails: liqing@tsinghua.edu.cn (Q.L.); taogashi08@gmail.com (S.Y.);
ncheng@tsinghua.edu.cn (N.C.); jysong@tsinghua.edu.cn (J.S.)

[2] National Key Laboratory on Flight Vehicle Control Integrated Technology, Flight Automatic Control Research Institute, Xi'an 710065, China; E-Mail: tangliangwen2014@foxmail.com

* Author to whom correspondence should be addressed; E-Mail: dachuanLi86@gmail.com.

Academic Editor: Aboelmagd Noureldin

Received: 27 December 2014 / Accepted: 17 April 2015 / Published: 22 April 2015

Abstract: This paper presents a non-linear state observer-based integrated navigation scheme for estimating the attitude, position and velocity of micro aerial vehicles (MAV) operating in GPS-denied indoor environments, using the measurements from low-cost MEMS (micro electro-mechanical systems) inertial sensors and an RGB-D camera. A robust RGB-D visual odometry (VO) approach was developed to estimate the MAV's relative motion by extracting and matching features captured by the RGB-D camera from the environment. The state observer of the RGB-D visual-aided inertial navigation was then designed based on the invariant observer theory for systems possessing symmetries. The motion estimates from the RGB-D VO were fused with inertial and magnetic measurements from the onboard MEMS sensors via the state observer, providing the MAV with accurate estimates of its full six degree-of-freedom states. Implementations on a quadrotor MAV and indoor flight test results demonstrate that the resulting state observer is effective in estimating the MAV's states without relying on external navigation aids such as GPS. The properties of computational efficiency and simplicity in gain tuning make the proposed invariant observer-based navigation scheme appealing for actual MAV applications in indoor environments.

Keywords: integrated navigation; micro aerial vehicles (MAVs); state observer; visual odometry (VO); RGB-D cameras

1. Introduction

Micro aerial vehicles (MAV) are playing an increasingly important role in both civil and military applications. The recent development of MAV technologies has provide such vehicles with capabilities to accomplish a variety of tasks such as search and rescue, disaster relief, surveillance of hostile territory, as well as inspection of hazardous environments such as indoor fire monitoring, investigation of hazardous or hostile buildings, disaster inspection of enclosed infrastructures (collapsed buildings, underground mines and subway stations after earthquakes or terrorist attacks, *etc.*) [1–4]. MAVs with autonomous capabilities are ideal candidates for these tasks since such environments are highly risky for human beings and inaccessible by ground robots. In recent years, the development of MAV navigation technologies in indoor or enclosed environments has been an active area of research. However, the development of such autonomous MAVs poses a number of technical challenges in the field of navigation and control. One of the key problems is the state estimation of MAVs since the control and other decision-making functions rely on reliable and accurate knowledge of the MAV's position, attitude and velocity. This requires the design and development of lightweight navigation systems to provide reliable state estimation of the vehicle. Due to onboard payload limitations, the navigation of MAVs generally relies on lightweight, low-cost MEMS (micro electro-mechanical systems) based IMU (inertial measurement unit) sensors which typically consist of rate gyroscopes, accelerometers and magnetometers. Although the angular velocity and attitude of the vehicle can be estimated using inertial measurements, the accurate translational velocity and position cannot be obtained by simply integrating inertial data due to the unbounded bias of low-cost MEMS sensors. For conventional outdoor robot applications, a GPS is commonly used as an aid to provide absolute position measurement for the vehicle. However, a GPS does not function effectively in urban canyons and it is even unavailable in indoor and enclosed environments. As a result, exteroceptive sensors such as laser rangefinders, cameras and RGB-D sensors are used as aids to inertial systems, and measurements of exteroceptive sensors can be fused with inertial measurements to bound the sensor bias and provide more accurate state estimates.

In recent years, there has been considerable research on the development of indoor MAV systems using exteroceptive aided inertial navigation approaches. Laser rangefinders have been successfully implemented in previous MAV systems for indoor exploration and mapping [5–10]. However, laser rangefinders can only provide distance measurements inside the 2D sensing plane of the sensor, thus their effectiveness is restricted to environments characterized with vertical structures. Moreover, they are unable to fully utilize the information in a 3D environment. In addition, both monocular [11–13] and stereo vision [14–16] technologies based on onboard cameras have also been employed to provide relative estimates for indoor navigation of MAVs. Although appealing in many applications, the monocular and stereo vision based approaches have several drawbacks in addressing indoor MAV navigation problems: Conventional RGB cameras do not directly provide distance information of environments, thus monocular/stereo vision based approaches must calculate the depth data using image features, which is a computationally intensive process.

The recent development of low-cost commercial RGB-D devices has led to increased capabilities for MAV applications. RGB-D devices are based on structured light technologies and can provide depth data even in poorly textured environments. Taking advantage of RGB-D devices, many researchers have achieved successful results in the field of indoor MAV navigation, such as state estimation, control and indoor mapping [17,18]. Despite these advances achieved in this domain, there is still significant progress to be made in developing more robust and computationally efficient visual odometry approaches for MAVs in complex environments, using lightweight and low-cost RGB-D devices and MEMS sensors.

The primary contributions in this paper are as follows: firstly, this paper presents a novel robust RGB-D based visual odometry (VO) approach that estimates the MAV's relative motion by extracting and matching features of successive frames captured by the RGB-D camera. We made several extensions in the primary aspects of RGB-D visual odometry and proposed a robust featuring detection and matching strategy (termed OFC-ORB, optical flow-constrained ORB), as well as a robust inlier detection and relative motion estimation framework (termed Consistency-RANSAC, Consistency- random sample consensus). The properties of the proposed robust RGB-D VO ensure the algorithm's computational efficiency and improve its robustness for complex environments with various texture conditions, as well as images blurs and partitions caused by the MAV's maneuvers, making the approach particularly suitable for MAVs operating in complex environments. The second contribution is the design of a nonlinear observer-based state estimation framework that fuses the inertial data with aiding measurements from the RGB-D VO, which is built upon the invariant observer theory. The system dynamics and observation model for the RGB-D visual-aided inertial navigation problem is formulated, followed by the verification of system symmetries, and the invariant state observer is finally derived based on the system model. Using the resulting RGB-D aided-inertial navigation framework, the drift of inertial sensors can be corrected, yielding a more accurate estimate of the MAV's states. Taking advantage of the invariant observer's tuning and computational simplicity, the resulting state estimation approach is well-suited to implementations on MAVs with constrained payloads and computation capabilities. Finally, the paper presents the implementations of the proposed approaches and design on a quadrotor MAV platform, along with validations through indoor flight experiments. Experimental results demonstrate that the proposed RGB-D visual aided-inertial navigation framework can provide accurate and reliable state estimates for MAVs without relying on external navigation aids such as GPS.

The remaining contents of this paper are organized as follows. A brief review of related work is presented in Section 2. Section 3 provides an overview of the RGB-D visual/inertial navigation scheme. Details of the robust RGB-D relative motion estimation are described in Section 4, followed by the design of invariant observer-based state estimation method in Section 5. After the implementations and experimental validations of the resulting system design in Section 6, the paper is finally concluded in Section 7.

2. Related Work

Most types of MAV have non-linear dynamics and the aided inertial navigation systems are formulated as nonlinear models. Therefore, the state estimation of such systems is a typical non-linear state observer design problem, where the most widely used approach by far is the extended Kalman filter (EKF) and its variants. In recent research on MAVs, EKF-based methods have been applied in the vast majority of MAV state estimation applications and have demonstrated appealing performance. Rather than directly accounting for the nonlinearity, the EKFs linearize the system dynamics and observation model about the current best estimate, and apply the Kalman filter based on the linear approximation of the system. There are several variants and implementations of the EKFs, depending on the formulation of the system dynamics and observation model, as well as the representation of attitude and estimation errors. Typical examples of the conventional EKF-based state estimation scheme proposed by Bachrach *et al.* [5–7], Chowdhary *et al.* [8,9] and Sobers *et al.* [10] utilize laser scan-matching algorithms to provide position and heading measurements of the MAV, and fuse these measurements with inertial information via EKFs. The systems developed by Bachrach *et al.* [5–7] employ two groups of measurements, where the IMU readings are treated as measurements of the attitude and accelerations, and laser scan-matching outputs are incorporated as position and heading measurements. In contrast, the state estimation scheme in [8–10] utilizes the gyroscope and accelerometer measurements as noisy inputs of the state propagation model, while the attitude error is estimated as part of the MAV states and is used to correct the final altitude estimates. Similarly, EKF-based approaches have also been applied to visual-aided state estimation schemes of MAVs. In [11], a single camera is leveraged in an inertial-optical flow framework to obtain a metric velocity estimate, which is then treated as a measurement to a real-time EKF scheme. The proposed navigation scheme is capable of operating onboard a processor and enables real-time control of the MAV. A navigation and mapping system developed by Wu [12,13] consists of two EKF estimators, where the MAV's position, velocity and attitude are estimated via a EKF scheme by fusing IMU measurements and observations of landmark features provided by monocular-vision. A separate mapping EKF estimator approximates the landmark feature positions when the MAV's state estimates are available. Acgtelik *et al.* [14,15] use an EKF estimator to provide state estimates for the autonomous navigation and exploration of MAVs. The EKF sensor fusion filter combines relative motion estimates from the stereo visual odometry with IMU measurements, and periodically incorporates position corrections from a SLAM module to bound the drift. More recently, Voigt *et al.* [16] proposed an EKF-based visual-inertial ego-motion estimation method that combines stereo vision and IMU measurements in a tightly coupled manner. The resulting scheme utilizes IMU state and covariance propagation information to aid the feature matching of the stereo vision, leading to increased efficiency and robustness of MAV state estimation in complex industrial environments. In addition, EKF based methods have also proven useful for RGB-D vision-aided navigation of MAVs, and relevant examples can be found in [17,18]. Among the variants of the EKFs, multiplicative extended Kalman filters (MEKF) are especially useful for MAV attitude estimation applications. In order to employ a non-singular attitude representation in the estimator, MEKFs formulate the attitude as a multiplication of an estimated attitude quaternion and an error quaternion representing the deviation between the above estimates and true attitude. Due to the advantages of the non-singularity representation, the MEKF method has been applied to the design of micro attitude and heading reference system (AHRS) [19], as well as the MAV velocity and attitude estimation problem [20]. Although EKFs have been successfully applied in a number of applications, they have several disadvantages in addressing the navigation problem of MAVs. Since EKFs relies on the linearization of the system, the accuracy of state estimation may degrade significantly in cases that involve high degree of nonlinearities. In addition, the tuning and identification of EKFs' parameters such as noise covariance and initial estimate parameters require extensive experiments, which may reduce the suitability for actual MAV applications.

The sigma-point unscented Kalman filter (UKF) is an effective alternative to the conventional EKFs when the system dynamics or observation model is highly non-linear, or the states are highly uncertain. To cope with high non-linearity and uncertainty, the UKF employs a high-order stochastic linear approximation of non-linear systems using weighted sigma-points. A UKF-based monocular vision-IMU system is proposed in [21] to perform state estimation, mapping, as well as self-calibration of the transform parameters between the camera and IMU. Moreover, Van der Merwe *et al.* [22] apply a UKF estimator to the design of a loosely-coupled GPS/INS (inertial navigation system) integration navigation system on an autonomous unmanned helicopter. However, the UKF also operates under the assumption of a Gaussian system, and it is therefore less effective for applications with non-Gaussian models. In contrast, the particle filter (PF) does not necessarily require the assumption that the process and measurement noise are Gaussian, and it operates by approximating the posterior distribution of the states using sampled, weighted particles. This property makes the PF more suitable for non-Gaussian estimation problems. A typical example of the PF-based state estimation method can be found in [23], where a Gaussian PF-based filter is employed to compute pseudo-measurements from laser data, and these laser measurements are integrated with IMU data to yield a full estimate. The resulting framework is implemented onboard a fixed wing MAV which is capable of performing aggressive flight in indoor environments. Due to the high computational cost, the PFs have not yet found wide use in actual MAV applications.

Alternatively to the aforementioned optimal estimators, several nonlinear observer approaches have been introduced into MAV state estimation applications in the past few years. Unlike optimal estimation approaches that propagate the posterior conditional probability distribution of the state using sequences of observations, nonlinear observer-based approaches are generally designed directly on the nonlinear geometry of the systems. Nonlinear observers are especially attractive since they are usually accompanied by global stability proofs of observer error dynamics, *i.e.*, global convergence of the estimation error for all initial conditions and system trajectories [24]. In [25], a Luenberger observer-based fusion filter is designed and implemented onboard a quadrotor MAV. The proposed filter combines the IMU measurements and absolute position estimates provided by a monocular visual SLAM (simultaneous localization and mapping) module, featuring a high update rate of position and velocity estimates to enable fast position control. Similarly, Boutayeb *et al.* [26] propose a time-varying reduced-order Luenberger-like observer for the velocity estimation for a MAV using linear acceleration measurements. Recent work has also focused on applying sliding-mode observers and adaptive observers to the MAV state estimation problems. Benallegue *et al.* [27,28] utilize a sliding-mode observer to estimate the MAV's velocity, as well as model uncertainties and disturbances such as winds. Taking advantage of this observer, a feedback linearization controller [27] and a back-stepping controller [28] are employed to achieve MAV position tracking control in the presence of external disturbances. Nonlinear adaptive observer techniques are employed in [29] for velocity estimation of a quadrotor MAV, using noisy acceleration and angular measurements from IMUs, and the resulting observer demonstrates robustness to measurement noise in simulations. However, the proposed adaptive observer may be computationally demanding for onboard implementations since the cascade observer contains high-order terms. It is also worth mentioning the recent work on the observer based simultaneous state estimation and sensor fault diagnosis for MAV. In [30], a group of reduced-order time varying observers are designed to diagnose and isolate accelerometers faults, and to simultaneously estimate the MAV velocity from acceleration measurements. Despite the aforementioned research on observer-based approaches, it remains to be seen whether the non-linear observer can be more generally useful for actual MAVs.

In particular, an important development that came from previous research on system symmetries theory is the symmetry-preserving observer. Bonnabel *et al.* [31,32] propose the invariant observer which is built on the invariant properties (symmetry geometry) of such systems. In [31,32], it is proved that when the system is invariant by a transformation Lie-group, one can design a nonlinear invariant observer that possesses the same symmetry properties as the original system. A relevant approach that is closely related to invariant observer is the complementary filter [33], which is designed directly on the matrix representation of the special orthogonal group $SO(3)$. The invariant observer method is applied to the design of low-cost AHRS in [34–36], and is further used in the GPS-aided inertial navigation system for outdoor MAV applications in [37–39]. In addition, examples of complementary filter-based visual/inertial state estimation of a helicopter MAV can be found in [40].

The invariant observer provides a systematic approach of designing non-linear state observer for a class of systems symmetry properties. Instead of linearizing the system model as in the EKF based approach, the invariant observer takes advantage of the symmetry geometry of the system, yielding an invariant state estimation error dynamics, and therefore the calculation of observer gains can be simplified. Moreover, the gain matrices of the observer are constant on permanent trajectories sets rather than equilibrium points. Due to its properties of computational efficiency and simplicity in parameters tuning, the invariant observer is well suitable for the state estimation of actual MAV platforms with limited onboard computational resources. Motivated by previous research, we seek to adapt the invariant observer approach to the design of a RGB-D visual/inertial navigation scheme, in order to provide state estimates for indoor MAV systems without relying on external navigation aids.

3. Overview of the RGB-D Visual/Inertial Navigation System Framework

The overall RGB-D visual/inertial navigation scheme is depicted in Figure 1. A typical onboard sensor set mounted on a MAV consists of a RGB-D camera, a MEMS IMU and a magnetometer (In this paper, a sonar altimeter is also utilized to provide altitude measurements). The RGB-D camera captures the RGB image and depth data (depth image) of the surrounding environment structures that fall within its sensing range. A MEMS IMU typically integrates tri-axial gyroscopes and tri-axial accelerometers, providing tri-axial angular rates and tri-axial linear acceleration measurements of the MAV (both expressed in the MAV body-fixed frame), respectively. The magnetometer measures the local magnetic field vector expressed in the MAV body-fixed frame.

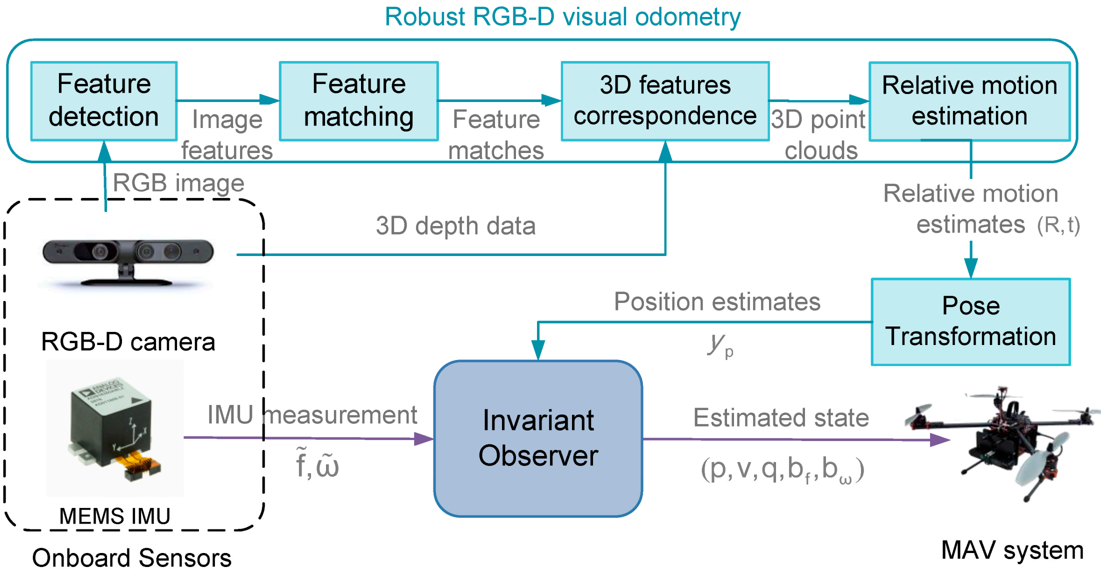


Figure 1. Block diagram of the RGB-D visual/inertial navigation scheme. MEMS, micro-electro-mechanical system; IMU, inertial measurement unit; MAV, micro-aerial vehicle.

As illustrated in Figure 1, the RGB-D visual odometry (cyan blocks in Figure 1) algorithm utilizes the RGB image and depth data captured by the RGB-D camera, and estimates the relative motion sequence of the MAV by extracting and matching features from consecutive RGB-D images. The above motion estimate sequence is then combined and transformed to obtain an estimate of the MAV's pose in a global frame. The RGB-D VO motion estimates and IMU measurements are finally fused by the invariant observer, yielding a full estimate of the MAV's 6-DOF (degree of freedom) states, as well as inertial sensor biases.

4. Robust RGB-D Visual Odometry

As described previously, RGB-D visual odometry refers to the process of estimating the relative motion of the MAV between successive time steps, using environmental features from consecutive images captured by the RGB-D camera. The sequence of motion estimates can then be integrated as position observations into the data fusion algorithm. A contribution from this paper is the development of a robust RGB-D VO approach which extends existing algorithms in VO operations including feature detection and matching, as well as relative motion calculation. The overall process flow of the robust RGB-D VO is illustrated in Figure 2. Details on the proposed robust RGB-D VO steps will be specified in the following subsections.

4.1. Robust Feature Detection and Matching

The robust feature detection and matching strategy proposed in this paper is built on the ORB detector (oriented FAST (features from accelerated segment test) and rotated BRIEF; ORB is first proposed in [41]) as well as the optical flow method (Figure 2). The RGB image captured by the RGB-D camera is converted to grayscale first, and a scale pyramid scheme of the image is then established using Gaussian kernels with different scale factors, such that features can be extracted from each level to enhance robustness. The ORB detector is utilized in our system due to its computational efficiency and robustness in feature detection. The ORB algorithm utilizes the FAST (features from accelerated segment test) detector [42] to extract feature points from each level of the image scale pyramid. The basic idea of the FAST detector is to determine a feature point by comparing the intensity threshold to the grayscale gradient between a center pixel and pixels in its circular neighborhood [42]. Although FAST detector is computationally efficient in finding corner features, it generates large responses along edges that provide rare useful information. Therefore, the ORB algorithm employs a Harris response measure to filter features extracted by FAST. For each FAST feature, a Harris response is computed using its grayscale gradient and all the features are then sorted in a descending order of the Harris responses. The *n* features with the highest Harris responses are selected to eliminate edge points (where *n* is a pre-defined number). In addition, the ORB also incorporates an orientation component calculated using the intensity centroid method to improve the rotation invariance of feature detection. In the feature detection step of our approach, the corresponding depth data is also extracted from the depth image, and features without corresponding depth are pruned out to eliminate fault detections. After that, a bit string-based descriptor consisting of a fixed-length vector is computed for each feature to uniquely describe the feature. The descriptor employed in the ORB algorithm is based on an extension of the BRIEF (binary robust independent elementary features) [43], termed the rBRIEF, which calculates feature's descriptor vector by performing a series of grayscale intensity binary tests in an image pixel patch around the feature. In order to enhance the BRIEF's invariance to planar rotation, the rBRIEF computes the descriptor by steering the pixel patch according to the orientation of the feature. Through the above operations, the feature detection step generates a set of features from an image, each with a BRIEF descriptor represented by a vector of length *n* (*n* = 256 for the ORB).

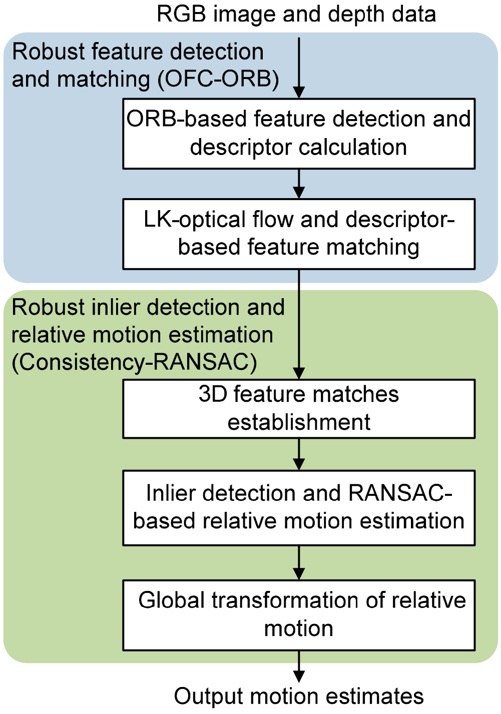

RGB image and depth data
Robust feature detection and matching (OFC-ORB)
ORB-based feature detection and descriptor calculation
LK-optical flow and descriptor-based feature matching
Robust inlier detection and relative motion estimation (Consistency-RANSAC)
3D feature matches establishment
Inlier detection and RANSAC-based relative motion estimation
Global transformation of relative motion
Output motion estimates

Figure 2. Process flow of the robust RGB-D visual odometry. ORB, oriented FAST and rotated BRIEF algorithm; OFC-ORB, optical flow-constrained ORB; RANSAC, random sample consensus algorithm.

Once the features are extracted from consecutive images, the feature matching procedure is performed to generate feature correspondence. In order to increase the robustness of the feature matching procedure to complex task scenarios that involve varied feature quality, lighting conditions and motion blur, we propose a robust feature-matching strategy called OFC-ORB (optical flow-constrained ORB) that combines optical-flow tracking and ORB feature descriptor-based matching. The basic idea of the OFC-ORB strategy is to constrain the ORB descriptor-based feature matching using a confidence sub-window region predicted by the optical-flow tracking. An inverse-check procedure based on descriptor-based matching and optical flow estimation is also employed to remove false matching. This strategy significantly reduces outliers and increases robustness of the feature matching in complex environments.

The OFC-ORB algorithm proposed in this paper employs the LK-optical-flow approach [44] to compute the optical-flow disparities between successive images. Let I_m and I_{m+1} be two consecutive grayscale images captured at time t_m and $t_{m+1} = t_m + \Delta t$ by the RGB-D camera. The functions $I_m(\mathbf{u})$ and $I_{m+1}(\mathbf{u})$ provide the respective grayscale intensity values of pixel $\mathbf{u}$ at the location $\mathbf{u} = [x, y]^{\mathrm{T}}$ in two images. Given a point $\mathbf{p} = [x_p, y_p]^{\mathbf{T}}$ in image I_m, the objective of the optical flow-based tracking is to find the corresponding point $\mathbf{q} = [x_q, y_q]^{\mathbf{T}}$ in I_{m+1}, with the optical-flow disparity $\mathbf{d} = [d_x, d_y]^{\mathrm{T}}$ that minimizes the following error function:

$$\varepsilon(\mathbf{d}) = \sum_{x=x_p-w_x}^{x_p+w_x} \sum_{y=y_p-w_y}^{y_p+w_y} \left[I_m(x,y) - I_{m+1}(x+d_x, y+d_y) \right]^2 \tag{1}$$

where w_x, w_v denote the size of the pixel window around $\mathbf{p}$ and $\mathbf{q}$. The Lucas–Kanade (LK)-optical-flow approach employs the first-order linear approximation of the error function given by Equation (1), and solves the above problem using a Newton-Raphson iteration method. Let $\mathbf{d}_k$ be the optical-flow calculated from the kth iteration step and $\mathbf{d}_0 = 0$. The $k + 1$th iteration calculates $\boldsymbol{\delta}_k$ that minimizes the following error function:

$$\varepsilon_k(\boldsymbol{\delta}_k) = \sum_{x=x_p-w_x}^{x_p+w_x} \sum_{y=y_p-w_y}^{y_p+w_y} \left[I_m(\mathbf{x}) - I_{m+1}(\mathbf{x} + \mathbf{d}_k + \boldsymbol{\delta}_k) \right]^2 \tag{2}$$

Given $\boldsymbol{\delta}_k$, the optical-flow from the $k + 1$th iteration step can be obtained by:

$$\mathbf{d}_{k+1} = \mathbf{d}_k + \boldsymbol{\delta}_k \tag{3}$$

The above calculations repeat until k exceeds the maximum iteration number, or $\boldsymbol{\delta}_k < \Delta\boldsymbol{\delta}$ ($\Delta\boldsymbol{\delta}$ is a pre-defined threshold), and the estimates of $\mathbf{d}_k$ converge to optimal $\mathbf{d}$ ideally.

In order to achieve a tradeoff between local accuracy and robustness, a pyramid representation [45] of the image is established in the OFC-ORB algorithm, and the optical-flow tracking is performed recursively through each level of the pyramid to obtain a more accurate optical flow estimate. Let I be the original image (*i.e.*, the image intensity function) of size $n_x \times n_y$, and it is considered as the zeroth level of the pyramid, *i.e.*, $I^0 = I$, the pyramid representation of the image I can be established as follows: Define $l = 0, 1, 2, \ldots, l_n$ as levels of the pyramid, and let I^{l-1} be the image of the $l - 1$th level of size n_x^{l-1}, n_y^{l-1}, the lth level image I^l is given by:

$$I^{l} = \frac{1}{4}I^{l-1}(2x,2y) + \frac{1}{8}\left(I^{l-1}(2x-1,2y) + I^{l-1}(2x+1,2y) + I^{l-1}(2x,2y-1) + I^{l-1}(2x,2y+1)\right) + \frac{1}{16}\left(I^{l-1}(2x-1,2y-1) + I^{l-1}(2x+1,2y+1) + I^{l-1}(2x-1,2y+1) + I^{l-1}(2x+1,2y-1)\right) \tag{4}$$

where x, y denotes the coordinates of pixels in the image, and the size of the $l - 1$th level image (n_x^l, n_y^l) must satisfy the following constraint:

$$\begin{aligned} n_x^l &\le \left(n_x^{l-1}+1\right)/2 \\ n_y^l &\le \left(n_y^{l-1}+1\right)/2 \end{aligned} \tag{5}$$

After the construction of the image pyramid, the optical-flow estimation is performed recursively at each level. The optical-flow estimation procedure starts from the top level (l_n) of the pyramid, and the estimate results from the previous level are used as initial parameters of the computations in the next level to obtain a refined estimate. This procedure traverses the image pyramid until the original level image is reached (l_0).

The pyramid optical-flow tracking procedure is illustrated in Figure 3. Define I_{m-1}^0 and I_m^0 as the original two consecutive images, and denote the corresponding point in the kth level of $\mathbf{p}$ as $\mathbf{p}^k$, $\mathbf{p}^k = [x_p^k, y_p^k]^{\mathrm{T}}$. Following Equations (4) and (5), the relationship of $\mathbf{p}^k$ and $\mathbf{p}$ is given by: $\mathbf{p}^k = \mathbf{p}/2^k$. Assume that $\mathbf{g}^k = [\mathbf{g}_x^k, \mathbf{g}_y^k]^{\mathrm{T}}$ is the optical-flow estimate calculated from level l_n to level l_{k+1}, and set the initial optical-flow estimate at level l_n to zero (*i.e.*, $\mathbf{g}^k = [0\ 0]^{\mathrm{T}}$). The level l_k images can be centered-compensated using the initial estimate $\mathbf{g}^k$:

$$\begin{aligned} I_{m-1,c}^k(\mathbf{x}) &= I_{m-1}^k(\mathbf{x}+\mathbf{p}^k), \\ I_{m,c}^k(\mathbf{x}) &= I_m^k(\mathbf{x}+\mathbf{g}^k+\mathbf{p}^k) \end{aligned} \tag{6}$$

The disparity $\mathbf{d}^k$ between the compensated images $I_{m-1,c}^k$ and $I_{m,c}^k$ can be estimated following the aforementioned LK-optical-flow method, which operates by minimizing the following error function:

$$\varepsilon(\mathbf{d}^k) = \sum_{x=x_p-w_x}^{x_p+w_x} \sum_{y=y_p-w_y}^{y_p+w_y} [I_{m-1,c}^k(\mathbf{x}) - I_{m,c}^k(\mathbf{x}+\mathbf{d}^k)]^2 \tag{7}$$

The initial optical-flow estimate $\mathbf{g}^k$ can then be corrected using $\mathbf{d}^k$:

$$\mathbf{g}^{k-1} = 2(\mathbf{g}^k + \mathbf{d}^k) \tag{8}$$

$\mathbf{g}^{k-1}$ can be used as the initial estimate of the computation in level l_{k-1}. The above procedure goes on through the pyramid to the original level l_0, and the final estimate of the optical-flow $\mathbf{d}$ is given by:

$$\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0 \tag{9}$$

The above pyramid implementation enables the estimation of optical-flow to handle large-scale motions, while maintaining local sub-pixel accuracy.

Taking advantage of the ORB detector and the optical-flow tracking strategy, the OFC-ORB proposed in this paper improves the descriptor-based feature matching by employing the optical-flow information for predicting confidence regions and checking false matching. The overall process flow of the OFC-ORB strategy is depicted in Figure 4, and the primary steps of the OFC-ORB strategy are presented in Algorithm 1. For consecutive images, the OFC-ORB employs the optical-flow tracking strategy to predict a confidence sub-window in the previous image of each feature in the current image (lines 5 and 6). Therefore, the descriptor-based search of feature matches can then be constrained to this sub-window around the expected feature point as predicted by the optical flow algorithm. After finding the best matches from the predicted region in the previous image (line 7), an inverse matching procedure is performed in the current image to check existing matches, and so that false matches can be removed (line 8). Finally, the optical-flow tracking is introduced again to compute the optical-flow disparities between the current feature matches (line 10). These disparities are compared to the optical-flow parameters obtained in the previous sub-window prediction step to further eliminate outliers (lines 11–13).

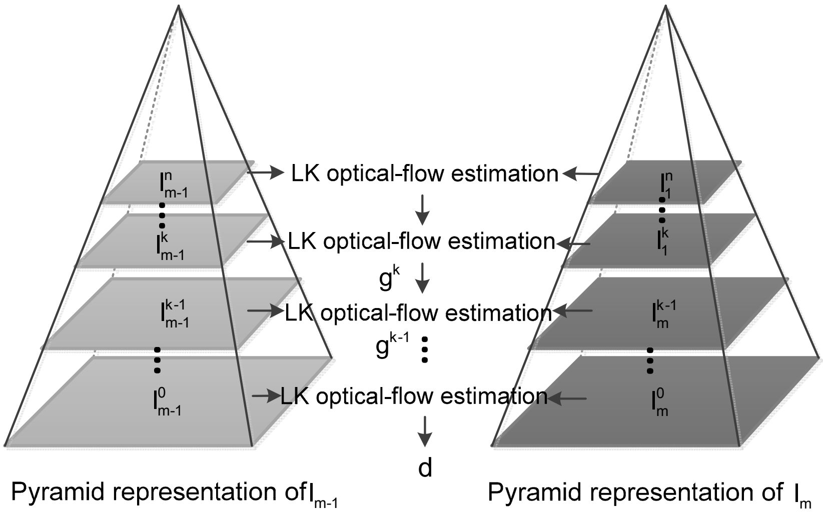

I^n_{m-1}
I^k_{m-1}
I^{k-1}_{m-1}
I^0_{m-1}
LK optical-flow estimation
LK optical-flow estimation
g^k
LK optical-flow estimation
g^{k-1}
LK optical-flow estimation
d
I^n_1
I^k_1
I^{k-1}_m
I^0_m
Pyramid representation of I_{m-1}
Pyramid representation of I_m

Figure 3. Pyramid implementation of the optical-flow tracking operation. LK optical-flow: Lucas–Kanade optical-flow algorithm.

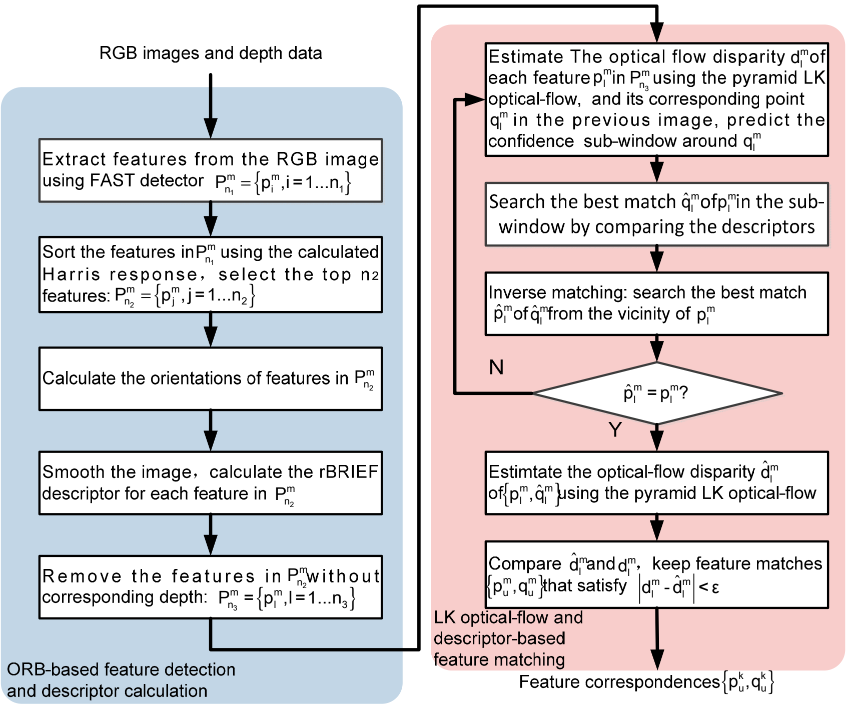

RGB images and depth data
Extract features from the RGB image using FAST detector $P^m_{n_1} = \{p^m_i, i = 1...n_1\}$
Sort the features in $P^m_{n_1}$ using the calculated Harris response，select the top n2 features: $P^m_{n_2} = \{p^m_j, j = 1...n_2\}$
Calculate the orientations of features in $P^m_{n_2}$
Smooth the image，calculate the rBRIEF descriptor for each feature in $P^m_{n_2}$
Remove the features in $P^m_{n_2}$ without corresponding depth: $P^m_{n_3} = \{p^m_l, l = 1...n_3\}$
ORB-based feature detection and descriptor calculation
Estimate The optical flow disparity d^m_l of each feature p^m_l in $P^m_{n_3}$ using the pyramid LK optical-flow, and its corresponding point q^m_l in the previous image, predict the confidence sub-window around q^m_l
Search the best match $\hat{q}^m_l$ of p^m_l in the sub-window by comparing the descriptors
Inverse matching: search the best match $\hat{p}^m_l$ of $\hat{q}^m_l$ from the vicinity of p^m_l
N
$\hat{p}^m_l = p^m_l$?
Y
Estimtate the optical-flow disparity $\hat{d}^m_l$ of $\{p^m_l, \hat{q}^m_l\}$ using the pyramid LK optical-flow
Compare $\hat{d}^m_l$ and d^m_l，keep feature matches $\{p^m_u, q^m_u\}$ that satisfy $|d^m_l - \hat{d}^m_l| < \varepsilon$
LK optical-flow and descriptor-based feature matching
Feature correspondences $\{p^k_u, q^k_u\}$

Figure 4. Process flow of the OFC-ORB feature detection and matching strategy.

Algorithm 1. OFC-ORB feature detection and matching.

```
Input   Two consecutive images $\{I^{m-1}, D^{m-1}\}, \{I^m, D^m\}$ captured by the RGB-D camera
Output  Feature correspondence set $S$ of $I^{m-1}, I^m$
Extract features $\mathbf{P}_{n_1}^m = \{\mathbf{p}_i^m, i = 1 \ldots n_1\} \mathbf{Q}_{n_1'}^{m-1} = \{\mathbf{q}_j^{m-1}, j = 1 \ldots n_1'\}$ from $I^m, I^{m-1}$ respectively, using the ORB detector, compute the descriptor $v(\mathbf{p}_i^m), v(\mathbf{q}_j^{m-1})$ for each feature of $\mathbf{P}_{n_1}^m, \mathbf{Q}_{n_1'}^{m-1}$
Extract the depth data for each feature of $\mathbf{P}_{n_1}^m, \mathbf{Q}_{n_1'}^{m-1}$, discard features that do not have corresponding depth, sort features in $\mathbf{P}_{n_1}^m, \mathbf{Q}_{n_1'}^{m-1}$ in an ascending order of the x-coordinates, obtain $\mathbf{P}_{n_2}^m, \mathbf{Q}_{n_2'}^{m-1}$
Initialize the feature correspondence set $S \leftarrow \{\varnothing\}$
for each $\mathbf{p}_i^m \in \mathbf{P}_{n_2}^m$
    Estimate the optical-flow vector $\mathbf{d}_i^m$ of $\mathbf{p}_i^m$ using the pyramid LK-optical-flow algorithm, calculate the corresponding point $\mathbf{q}_i^{m-1}$ in $I^{m-1}$: $\mathbf{q}_i^m = \mathbf{p}_i^{m-1} + \mathbf{d}_i^m$.
    Build the confidence sub-window $w_{\mathbf{q}}^{m-1}$ of size $0.1n_x \times 0.1n_y$ centered at $\mathbf{q}_i^{m-1}$
    Find the best match $\hat{\mathbf{q}}_i^m$ of $\mathbf{p}_i^m$ from $w_{\mathbf{q}}^{m-1}$, by searching for $\mathbf{q}_r \in w_{\mathbf{q}}^{m-1}$ that satisfies: $r = \arg\min \|v(\mathbf{p}_i) - v(\mathbf{q}_r)\|$
    Build a sub-window $w_{\mathbf{p}}^m$ centered at $\mathbf{p}_i^m$ in $I^m$, find the best match $\hat{\mathbf{p}}_i^m$ of $\hat{\mathbf{q}}_i^m$ from $w_{\mathbf{p}}^m$
    if $\hat{\mathbf{p}}_i^k = \mathbf{p}_i^k$
        Estimate the optical-flow vector $\hat{\mathbf{d}}_i^m$ between $\{\mathbf{p}_i^m, \hat{\mathbf{q}}_i^m\}$
        if $|\mathbf{d}_i^m - \hat{\mathbf{d}}_i^m| < \varepsilon$
            $S \leftarrow S \cup \{\mathbf{p}_i^m, \hat{\mathbf{q}}_i^m\}$
        end if
    end if
end for
return $S$
```

The primary advantage of the OFC-ORB feature detection and matching strategy is that it achieves a balance between robustness and computational efficiency. The prediction of the sub-window based on optical-flow tracking can restrict the search of feature correspondences to a small confidence region, which reduces false matches and increases the robustness to complex environments and the impacts caused by MAV motion. In addition, the overall computations can be reduced because of the constrained search region of feature matching, as well as the increased number of inliers for the relative motion estimation procedure.

4.2. Robust Inlier Detection and Relative Motion Estimation

Once the 2D image feature matches are extracted by the OFC-ORB procedure, these feature matches are corresponded to 3D-space using their corresponding depth data from the depth image, yielding two sets of 3D point clouds with known correspondences, denoted as $\mathbf{P}_{m-1}$, $\mathbf{Q}_m$ ($\mathbf{P}_{m-1} = \{\mathbf{p}_i\}$, $\mathbf{Q}_m = \{\mathbf{q}_i\}$, $i = 1, \ldots, n$, $\mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^3$). Given these consecutive 3D feature correspondences captured at different time steps, the MAV's relative motion from the prior to the subsequent time step can be estimated based on the transformation of the two 3D point clouds. The concept of relative motion estimation is illustrated in Figure 5.

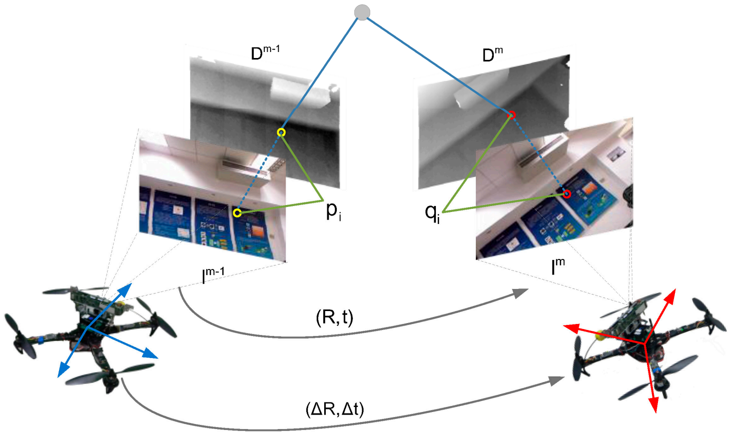


Figure 5. Relative motion estimation of MAV based on the RGB-D visual odometry.

The transformation of two 3D point clouds consists of the rotation **R** and translation **t**, and the relationship of the 3D feature correspondences can then be given by:

$$\mathbf{p}_i = \mathbf{R}\mathbf{q}_i + \mathbf{t} + \mathbf{v}_i, \quad (\mathbf{p}_i \in \mathbf{P}_{m-1}, \mathbf{q}_i \in \mathbf{Q}_m, i = 1 \ldots n) \tag{10}$$

where $\mathbf{v}_i \in \mathbb{R}^3$ denotes the error vector. The optimal solution of **R** and **t** in a least-square sense can be obtained by minimizing the following error function:

$$\varepsilon(\mathbf{R}, \mathbf{t}) = \sum_i^n \left\| \mathbf{p}_i - (\mathbf{R}\mathbf{q}_i + \mathbf{t}) \right\|_2^2 \tag{11}$$

In the system designed in this paper, the above problem is solved using the SVD (singular value decomposition) approach [46], where the basic idea is built on the property that rigid motion does not change the relative distances between points and their centroid of a rigid body. Assume $\hat{\mathbf{R}}, \hat{\mathbf{t}}$ to be the optimal solution that minimizes the error function in Equation (11), and let: $\mathbf{P}'_{m-1}$ = $\{\mathbf{p}'_i\}$ be the 3D point cloud obtained by applying the optimal transformation $\hat{\mathbf{R}}, \hat{\mathbf{t}}$ to $\mathbf{Q}_m$

$$\mathbf{p}_i^{'} \triangleq \hat{\mathbf{R}}\mathbf{q}_i + \hat{\mathbf{t}} \quad i = 1 \ldots n \tag{12}$$

According to the distance-preserving property of rigid body motion, the centroid of the transformed 3D point cloud $\mathbf{P}'_{m-1}$ must coincide with that of the actual point cloud $\mathbf{P}_{m-1}$:

$$\overline{\mathbf{p}}^{'} = \overline{\mathbf{p}} \tag{13}$$

where:

$$\begin{aligned} \overline{\mathbf{p}} &= \frac{1}{n}\sum_i^n \mathbf{p}_i \\ \overline{\mathbf{p}}^{'} &= \frac{1}{n}\sum_i^n \mathbf{p}_i^{'} = \hat{\mathbf{R}}\frac{1}{n}\sum_i^n \mathbf{q}_i + \hat{\mathbf{t}} = \hat{\mathbf{R}}\overline{\mathbf{q}} + \hat{\mathbf{t}} \end{aligned} \tag{14}$$

$\overline{\mathbf{q}}$ denotes the centroid of point cloud $\mathbf{Q}_m$. Following Equations (13) and (14), we have:

$$\overline{\mathbf{p}} = \hat{\mathbf{R}}\overline{\mathbf{q}} + \hat{\mathbf{t}} \tag{15}$$

The relative distance between each point and the centroid can be given by:

$$\Delta\mathbf{p}_i \triangleq \mathbf{p}_i - \overline{\mathbf{p}}, \quad \Delta\mathbf{q}_i \triangleq \mathbf{q}_i - \overline{\mathbf{q}} \quad i = 1 \ldots n \tag{16}$$

Following Equations (15) and (16), the error function in Equation (11) can be rewritten as:

$$\begin{aligned}\varepsilon(\mathbf{R},\mathbf{t}) &= \sum_i^n \left\|\bar{\mathbf{p}} + \Delta\mathbf{p}_i - (\mathbf{R}(\bar{\mathbf{q}} + \Delta\mathbf{q}_i) + \mathbf{t})\right\|_2^2 = \sum_i^n \left\|\Delta\mathbf{p}_i - \mathbf{R}\Delta\mathbf{q}_i + \bar{\mathbf{p}} - \mathbf{R}\bar{\mathbf{q}} - \mathbf{t}\right\|_2^2 \\ &= \sum_i^n \left\|\Delta\mathbf{p}_i - \mathbf{R}\Delta\mathbf{q}_i\right\|_2^2\end{aligned} \tag{17}$$

As can be seen from Equation (17), the transformed error function contains only the transformation component **R**. Therefore, the above simplified least-square problem can be solved through two primary steps: (1) Find the optimal $\hat{\mathbf{R}}$ that minimizes the error function in Equation (17); (2) Calculate $\hat{\mathbf{t}}$ according to Equation (15).

The detailed procedures of the SVD-based algorithm are as follows. The centroids ($\bar{\mathbf{p}},\bar{\mathbf{q}}$) of two point clouds are computed first, and the relative distances ($\Delta \boldsymbol{p}_i$, $\Delta \boldsymbol{q}_i$) are then calculated according to Equation (16). Given $\Delta \boldsymbol{p}_i$, $\Delta \boldsymbol{q}_i$, the algorithm computes the following matrix:

$$\mathbf{H} \triangleq \frac{1}{n}\sum_i^n \Delta\mathbf{q}_i \Delta\mathbf{p}_i^{\mathrm{T}} \tag{18}$$

Apply SVD to matrix **H**:

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \tag{19}$$

The optimal estimate of $\hat{\mathbf{R}}$ and can then be obtained by:

$$\begin{aligned}\hat{\mathbf{R}} &= \mathbf{V}\mathbf{U}^{\mathrm{T}} \\ \hat{\mathbf{t}} &= \bar{\mathbf{p}} - \hat{\mathbf{R}}\bar{\mathbf{q}}\end{aligned} \tag{20}$$

A detailed mathematical proof of the SVD-based approach can be found in [46]. The above procedure yields the optimal estimate of the relative motions between consecutive time steps given two groups of 3D feature correspondences.

It can be concluded from the above calculation procedure that the performance of relative motion estimation relies heavily on the quality of extracted feature correspondences, and it is thus highly sensitive to outliers. Although the optical-flow-constrained feature matching strategy described in Section 4.1 can significantly reduce the ratio of false matches, a robust strategy is necessary to further eliminate outliers. In order to ensure robustness and computational efficiency, we propose the Consistency-RANSAC, a refinement of the conventional RANSAC, which employs feature consistency check in the inlier detection procedure. The distance-preserving constraint of rigid motion is used again in the design of the consistency check strategy: ideally, the relative Euclidean distance between two points belonging to a rigid body should be identical to thier distance after rigid motions. For the visual odometry problem, this property indicates that the distances between different 3D features do not change substantially from one time step to the subsequent time step (*i.e.*, consistency). Recall the two 3D point clouds $\mathbf{P}_{m-1}$, $\mathbf{Q}_m$, let $\mathbf{p}_i$, $\mathbf{p}_j \in \mathbf{P}_{m-1}$ be two arbitrary 3D features from the 3D point clouds $\mathbf{P}_{m-1}$ at time $m - 1$, and $\mathbf{q}_i$, $\mathbf{q}_j \in \mathbf{Q}_m$ denote their corresponding matches in $\mathbf{Q}_m$ at time m, respectively. The consistency constraints of $\{\mathbf{p}_i, \mathbf{p}_j\}$ and $\{\mathbf{q}_i, \mathbf{q}_j\}$ is then given as:

$$\left|\left\|\mathbf{p}_i - \mathbf{p}_j\right\| - \left\|\mathbf{q}_i - \mathbf{q}_j\right\|\right| < \delta \tag{21}$$

where δ represents the threshold, and $\mathbf{p}_i$, $\mathbf{p}_j$, $\mathbf{q}_i$, $\mathbf{q}_j$ are expressed in the same global coordination. By performing the consistency check to the 3D feature correspondences, the algorithm can prune out incorrect matches that do not satisfy the consistency constraints in Equation (21) (*i.e.*, The deviations of relative distances exceed the threshold δ). Using the consistency check results, finding the largest set of inliers can be transformed into the problem of determining the maximum clique on a graph. This problem can then be solved by iteratively adding feature matches with the greatest degree (*i.e.*, feature matches with the largest number of consistent matches), which is consistent with all the feature matches in the existing consistent set. After finding the set of inliers, the final relative motion is estimated using an improved RANSAC procedure which operates by selecting consensus feature matches based on their similarity to a hypothesis, and progressively refining the relative motion hypothesis based on the selected inliers [47]. Since the consistency check-based inlier detection procedure has already increased the ratio of inliers, the RANSAC procedure can generate a good estimate through only a very limited number of iterations. This significantly reduces the overall computation cost of the algorithm.

The overall Consistency-RANSAC inlier detection and relative motion estimation procedure is illustrated in Algorithm 2. The algorithm starts by checking the consistency of the given 3D feature matches. Unlike the conventional consistency check strategy that directly utilizes the relative distances between different feature points, we first calculate the centroid of each 3D point cloud ($\bar{\mathbf{p}}, \bar{\mathbf{q}}$, line 1), and then compute the relative distances (($\Delta \boldsymbol{p}_i$, $\Delta \boldsymbol{q}_i$) from each 3D feature point to its corresponding centroid (line 3). These relative distances-to-centroid are then used to check the consistency of each pair of feature matches: the algorithm computes the errors between the distances-to-centroid of feature points and those of their corresponding matches (line 4), and sort all the 3D feature points according to these consistency errors (line 6). The last ξ% ($\xi = (n - n_1)/n$) feature match pairs with the largest consistency errors in the sorted feature point set are discarded (line 7). Using the distances-to-centroid information, the above refined strategy exempts the consistency check from the conventional time-consuming maximum-clique search procedure, and can ensure a high ratio of inliers.

Based on the detected inlier feature point set (Q), a refined RANSAC procedure is performed to further prune out outliers and compute the final motion estimate. In each iteration, λ pairs of feature matches are drawn randomly from the inlier set Q (line 11) to compute an initial hypothesis of the relative motion ($\mathbf{R}_0$, $\mathbf{t}_0$ line 13), using the SVD-based least-square approach described previously. The rest of the feature match pairs in Q are then checked for their compatibility with this initial hypothesis, by computing the transformation error using $\mathbf{R}_0$, $\mathbf{t}_0$, and comparing this error to the consensus error bound ε (line 15), and all compatible feature match pairs are added to a consensus set S_c (line 16). This operation serves to prune out outliers from Q. Once we get a sufficient number of feature match pairs in the consensus set (line 16), these data in S_c are used to compute a refined motion estimate ($\mathbf{R}_1$, $\mathbf{t}_1$). To adjust the consensus threshold, the average transformation error of feature matches in S_c is calculated using ($\mathbf{R}_1$, $\mathbf{t}_1$) (lines 21–24), and this error is used as the updated consensus error bound ε in the next iteration cycle if it is lower than the original ε (line 26). The above procedure continues until the probability of finding a better solution becomes sufficiently low (e.g., the difference between current average transformation error and prior consensus error is negligibly small), and the current solution is then accepted as the final motion estimate of ($\mathbf{R}$, $\mathbf{t}$) (line 28).

Algorithm 2. Consistency-RANSAC inlier detection and relative motion estimation.

```
Input   Two 3D point clouds with correspondences: $\mathbf{P}_{m-1}=\{\mathbf{p}_i\},\mathbf{Q}_m=\{\mathbf{q}_i\}, i=1...n$
Output  Relative motion $(\mathbf{R},\mathbf{t})$ of $\mathbf{P}_{m-1},\mathbf{Q}_m$
Calculate the centroid $\overline{\mathbf{p}},\overline{\mathbf{q}}$ of $\mathbf{P}_{m-1},\mathbf{Q}_m$
for $i = 1$ to $n$
    $\Delta\mathbf{p}_i \leftarrow \|\mathbf{p}_i-\overline{\mathbf{p}}\|,\ \ \Delta\mathbf{q}_i \leftarrow \|\mathbf{q}_i-\overline{\mathbf{q}}\|$
    $d_i \leftarrow |\Delta\mathbf{p}_i-\Delta\mathbf{q}_i|$
end for
Sort the point set $M=\{(\mathbf{p}_i,\mathbf{q}_i), i=1...n\}$ in the ascending order of $d_i$: $M_{\text{sorted}} \leftarrow M$
Select the top $n_1$ pairs of feature matches: $Q \leftarrow \{(\mathbf{p}_i,\mathbf{q}_i)\,|\,(\mathbf{p}_i,\mathbf{q}_i)\in M_{\text{sorted}}, i=1...n_1\}$
RANSAC initialization: $j\leftarrow 1$, $(\mathbf{R},\mathbf{t})\leftarrow(\mathbf{I},\mathbf{0})$, $\varepsilon\leftarrow\infty$
while $j <$ MaxIteration do
    Initialize the sample set and the $j$th consensus set: $S^j \leftarrow \varnothing, S_c^j \leftarrow \varnothing$, $\varepsilon' \leftarrow 0$
    Randomly select $\lambda$ pairs of feature matches from $Q$: $S^j \leftarrow \{(\mathbf{p}_i,\mathbf{q}_i)\,|\,(\mathbf{p}_i,\mathbf{q}_i)\in Q, i=1...\lambda\}$
    $S_c^j \leftarrow S^j$
    Estimate $(\mathbf{R}_0^j,\mathbf{t}_0^j)$ that minimizes the error function given in Equation (17) based on SVD approach, using features in $S^j$
    for each $(\mathbf{p}_i,\mathbf{q}_i)\in Q$ AND $(\mathbf{p}_i,\mathbf{q}_i)\notin S^j$ do
        if $\|\mathbf{p}_i-(\mathbf{R}_0^j\mathbf{q}_i+\mathbf{t}_0^j)\|<\varepsilon$
            $S_c^j \leftarrow S_c^j \cup \{\mathbf{p}_i,\mathbf{q}_i\}$
        end if
    end for
    if size$(S_c^j) > \eta n_1$
        Re-estimate $(\mathbf{R}_1^j,\mathbf{t}_1^j)$ that minimizes the error function based on SVD approach, using features in the consensus set $S_c^j$
        for each $(\mathbf{p}_i,\mathbf{q}_i)\in S_c^j$ do
            $\varepsilon' \leftarrow \varepsilon' + \|\mathbf{p}_i-(\mathbf{R}_1^j\mathbf{q}_i+\mathbf{t}_1^j)\|$
        end for
        $\varepsilon' \leftarrow \varepsilon'/\text{size}(S_c^j)$
        if $\varepsilon' < \varepsilon$
            $\varepsilon \leftarrow \varepsilon'$, $(\mathbf{R},\mathbf{t}) \leftarrow (\mathbf{R}_1^j,\mathbf{t}_1^j)$
        end if
        if $|\varepsilon-\varepsilon'| < \sigma$
            break
        end if
    end if
    $j \leftarrow j+1$
end while
return $(\mathbf{R}\mathbf{t})$
```

Taking advantage of the reduced ratio of outliers generated by consistency check, the number of iterations in RANSAC can be constrained effectively. The maximum iteration number (MaxIteration in line 9) can be set to a small integer between 5 and 10. The consistency check strategy along with RANSAC framework can significantly reduce outliers and increase robustness in motion estimation, while ensuring computational efficiency for real-time implementations.

4.3. Global Transformation of Relative Motions

The motion estimation procedure described in Section 4.2 provides a sequence of estimated transformations (**R**, **t**) of consecutive 3D feature point clouds at different time steps. Note that in actual applications, the features are actually fixed in the environment while the MAV moves around the features. Therefore, given the estimated feature transformation **T** of feature points, the corresponding motion (Δ**T**) of the MAV's current body frame with respect to the prior frame can be obtained by:

$$\Delta\mathbf{T} = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{t} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R}^{T} & -\mathbf{R}^{T}\mathbf{t} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \tag{22}$$

where Δ**R**, Δ**t** denote the rotation and translation component, respectively. We can derive a global representation of the MAV's pose with respect to an initial pose $\mathbf{T}_0$ using the following sequence of homogenous transformations:

$$\tilde{\mathbf{T}}_t = \Delta\mathbf{T} \cdot_t \Delta\mathbf{T}_{t-1} \cdot \ldots \cdot \Delta\mathbf{T}_{t-n+1} \cdot \mathbf{T}_0 \tag{23}$$

where Δ**T** denotes the relative transformation of the MAV's pose at different time steps. This global motion estimate can then be used as a measurement of the MAV's state in the data fusion scheme.

5. Invariant Observer Based State Estimation

5.1. Review of Invariant Observer Theory

The state estimation based on aided-inertial navigation systems is a typical nonlinear state observer design problem, where few general approaches exist for such problem. However, when the system possesses the geometry with symmetries under a transformation group, its state observer can be designed using a systematic approach, namely the invariant observer, which is originally proposed by Bonnabel *et al.* [31,32]. The primary feature of the invariant observer is that it is built upon the system's symmetry geometry and yields an invariant form of the state estimation error, which significantly simplifies the derivation of the observer gains and convergence analysis, making the observer particularly suitable for MAVs with computationally constrained onboard embedded systems. In this section, the theoretical foundations of the invariant observer are reviewed, which will be used to design the state observer of the RGB-D/inertial navigation system in following sections.

Definition 1. (Transformation Lie roup) Define G as a Lie group with identity e, and let M be a manifold. The transformation group $\phi_{g\epsilon G}$ acting on the manifold M can be defined as a smooth map $(g,\mu)\in G\times M \mapsto \phi_g(\mu)\in M$ and:

$$\begin{aligned} &\phi_{g_1}\circ\phi_{g_2} = \phi_{g_1 g_2}(\mu), \forall g_1, g_2 \in \Sigma, \forall\mu\in\Sigma \\ &\phi_e(\mu) = \mu, \forall\mu\in\Sigma \end{aligned} \tag{24}$$

The inverse group action $\phi_{g^{-1}}$ is also a smooth map, this makes $\phi_{g\in G}$ a diffeomorphism.

Consider a system of the following form:

$$\begin{aligned}\dot{\mathbf{x}} &= f(\mathbf{x},\mathbf{u})\\ \mathbf{y} &= h(\mathbf{x},\mathbf{u})\end{aligned} \tag{25}$$

where both *f*, *h* are smooth maps, and (**x**, **u**, **y**) ϵ *X* × *U* × *Y*. For the state estimation problem of systems modeled as the above formulation, (**x**, **u**, **y**) represent the state, system input and output, respectively, where $X\in\mathbb{R}^n, U\in\mathbb{R}^m, Y\in\mathbb{R}^r$ are all smooth manifolds (*i.e.*, the state manifold, the input manifold and the output manifold, respectively). Assuming that *B* = *X* × *U* is the trivial fiber bundle over the state manifold *X*, let φ_g: *G* × *X* → *X*, ψ_g: *G* × *U* → *U* and ρ_g: *G* × *Y* → *Y* be the smooth Lie group actions on the system's state, input and output manifold, respectively, where is *G* the system's Lie group with the property described in Definition 1. The invariance of the system in Equation (25) by the transformation group *G* can be defined as:

Definition 2. (G-invariance and G-equivariance) The system dynamics in Equation (25) is G-invariant if:

$$f(\varphi_g(\mathbf{x}),\psi_g(\mathbf{u})) = D\varphi_g(\mathbf{x})\cdot f(\mathbf{x},\mathbf{u}),\ \forall g\in G, \forall \mathbf{x}\in X, \forall \mathbf{u}\in U \tag{26}$$

and for the output map $\forall g\in G, \forall\mathbf{x}\in X, \forall\mathbf{u}\in U$, the system is G-equivariant if:

$$h(\varphi_g(\mathbf{x}),\psi_g(\mathbf{u})) = \rho_g(h(\mathbf{x},\mathbf{u})),\ \forall g\in G, \forall \mathbf{x}\in X, \forall \mathbf{u}\in U \tag{27}$$

The property in Equations (26) and (27) can also be expressed as: *X* = *f*(*X*, *U*), *Y* = *h*(*X*, *U*), *i.e.*, the system dynamics and outputs is invariant by the transformation group *G*. In coordinates, Equation (26) reads:

$$\frac{d}{dt}(\varphi_g(\mathbf{x})) = f(\varphi_g(\mathbf{x}),\psi_g(\mathbf{u})),\ \forall g\in G, \forall\mathbf{x}\in X, \forall\mathbf{u}\in U \tag{28}$$

Similarly, Equation (27) can be rewritten as:

$$\rho_g(\mathbf{y}) = \rho_g(h(\mathbf{x},\mathbf{u})),\ \forall g\in G, \forall\mathbf{x}\in X, \forall\mathbf{u}\in U \tag{29}$$

If the invariant system of Equation (25) verifies the properties described in Definition 2, the existence of its invariant observer can be verified by the following theorem [31]:

Theorem 1. For a G-invariant and G-equivariant system $\dot{\mathbf{x}} = f(\mathbf{x},\mathbf{u})$, **y** = *h*(**x**, **u**), there exists an invariant observer $\dot{\hat{\mathbf{x}}} = F(\hat{\mathbf{x}},\mathbf{u},\mathbf{y})$ that verifies the following properties:

(a) $F(\mathbf{x},\mathbf{u},h(\mathbf{x},\mathbf{u})) = f(\mathbf{x},\mathbf{u})$;

(b) $(\varphi_g)\cdot F(\hat{\mathbf{x}},\mathbf{u},\mathbf{y}) = F(\varphi_g(\hat{\mathbf{x}}),\psi_g(\mathbf{u}),\rho_g(\mathbf{y}))$, *i.e.*, the observer is invariant by the transformation group.

and the invariant observer $\dot{\hat{\mathbf{x}}} = F(\hat{\mathbf{x}},\mathbf{u},\mathbf{y})$ associated with the system reads:

$$F(\hat{\mathbf{x}},\mathbf{u},\mathbf{y}) = f(\hat{\mathbf{x}},\mathbf{u}) + \sum_{i=1}^{n} L_i(\mathbf{I}(\hat{\mathbf{x}},\mathbf{u}),\varepsilon(\hat{\mathbf{x}},\mathbf{u},\mathbf{y}))\varepsilon(\hat{\mathbf{x}},\mathbf{u},\mathbf{y})w_i \tag{30}$$

where the terms in Equation (30) are as follows:

(1) w_i is the invariant frame. A vector field *w*: *TX* → *X* is G-invariant if it verifies:

$$\varphi_g\cdot w(\mathbf{x}) = w(\varphi_g(\mathbf{x})), \forall g\in G \tag{31}$$

The invariant frame is defined as the invariant vector fields that form a global frame for *TX*. Therefore, $(w_1(\mathbf{x}), w_2(\mathbf{x})...w_n(\mathbf{x}))$ forms a basis for T_xX. An invariant frame can be calculated by:

$$w_i(\mathbf{x}) = (\varphi_{\gamma(x)^{-1}}) \cdot \partial / \partial x_i = \frac{d}{d\tau}\left(\varphi_{\gamma(x)^{-1}}(\upsilon_i \tau)\right) \tag{32}$$

where $\upsilon_i \in T_eX$ is a basis of $\upsilon_i \in T_eX$, and $\gamma(x)$ is the moving frame. Following the Cartan moving frame method, the moving frame $\gamma(x)$ can be derived by solving $\varphi_g(x) = c$ for $g = \gamma(x)$, where c is a constant. In particular, one can choose $c = e$ such that $\gamma(x) = x^{-1}$.

(2) $\varepsilon(\hat{\mathbf{x}}, \mathbf{u}, \mathbf{y})$ denotes the invariant output error, which is defined as follows:

Definition 3. (Invariant output error) The smooth map $\varepsilon : (\hat{\mathbf{x}}, \mathbf{u}, \mathbf{y}) \mapsto \varepsilon(\hat{\mathbf{x}}, \mathbf{u}, \mathbf{y})$ is an invariant error which verifies the following properties:

(a) For any $\hat{\mathbf{x}}, \mathbf{u}$, $\varepsilon : (\hat{\mathbf{x}}, \mathbf{u}, \mathbf{y}) \mapsto \varepsilon(\hat{\mathbf{x}}, \mathbf{u}, \mathbf{y})$ is invertible;

(b) For any $\hat{\mathbf{x}}, \mathbf{u}$, $\varepsilon(\hat{\mathbf{x}}, \mathbf{u}, h(\hat{\mathbf{x}}, \mathbf{u})) = 0$;

(c) For any $\hat{\mathbf{x}}, \mathbf{u}$, $\varepsilon(\varphi_g(\hat{\mathbf{x}}), \psi_g(\mathbf{u}), \rho_g(\mathbf{y})) = \varepsilon(\hat{\mathbf{x}}, \mathbf{u}, \mathbf{y})$;

According to the moving frame method, the invariant output error can be given by:

$$\varepsilon(\hat{\mathbf{x}}, \mathbf{u}, \mathbf{y}) = \rho_{\gamma(\hat{\mathbf{x}})}(h(\hat{\mathbf{x}}, \mathbf{u})) - \rho_{\gamma(\hat{\mathbf{x}})}(\mathbf{y}) \tag{33}$$

(3) $\mathbf{I}(\hat{\mathbf{x}}, \mathbf{u})$ is the invariant of *G*, which verifies:

$$\mathbf{I}(\varphi_g(\mathbf{x})) = \mathbf{I}(\mathbf{x}) \tag{34}$$

Following the moving frame method, the invariant $\mathbf{I}(\hat{\mathbf{x}}, \mathbf{u})$ is obtained by:

$$\mathbf{I}(\hat{\mathbf{x}}, \mathbf{u}) := \varphi_{\gamma(\hat{\mathbf{x}})}(\hat{\mathbf{x}}) \times \psi_{\gamma(\hat{\mathbf{x}})}(\mathbf{u}) \tag{35}$$

where $\gamma(x)$ is the moving frame.

(4) L_i is a 1 × r observer gain matrix that depends on **I** and ε, such that:

$$L_i(\mathbf{I}(\hat{\mathbf{x}}, \mathbf{u}), 0) = 0\,, \ \forall \hat{\mathbf{x}} \in \mathbf{X} \tag{36}$$

The observer gains can be obtained using the invariant state estimation error, which is defined as:

Definition 4. (Invariant state estimation error) The smooth map $(\mathbf{x}, \hat{\mathbf{x}}) \mapsto \eta(\hat{\mathbf{x}}, \mathbf{x})$ is an invariant state estimation error if it satisfies the following properties:

(a) $\eta(\hat{\mathbf{x}}, \mathbf{x})$ is a diffeomorphism on $X \times X$;

(b) For any $\mathbf{x} \in X$, $\eta(\mathbf{x}, \mathbf{x}) = 0$;

(c) $\eta(\varphi_g(\hat{\mathbf{x}}), \varphi_g(\mathbf{x})) = \eta(\hat{\mathbf{x}}, \mathbf{x})$.

Theorem 2 [31]. For a G-invariant and G-equivariant system given by Equation (25), the state estimation error of its invariant state observer is given as:

$$\eta(\hat{\mathbf{x}}, \mathbf{x}) = \varphi_{\gamma(x)}(\hat{\mathbf{x}}) - \varphi_{\gamma(x)}(\mathbf{x}) \tag{37}$$

with the following dynamics:

$$\dot{\eta} = \Upsilon(\eta, \mathbf{I}(\hat{\mathbf{x}}, \mathbf{u})) \tag{38}$$

where $\mathbf{I}(\hat{\mathbf{x}}, \mathbf{u})$ are the invariants that take the form as in Equation (35).

As can be seen from Equation (38), in contrast to the state error dynamics of general nonlinear observer which depends on the trajectory of the system (($\mathbf{x}(t)$, $\mathbf{u}(t)$)), the dynamics of the invariant state error depends only on the estimated invariants $\mathbf{I}(\hat{\mathbf{x}},\mathbf{u})$. This significantly simplifies the stability analysis and the selection of observer gains.

A comprehensive description of the mathematical foundation and proof of the invariant observer theory can be found in [31,32].

5.2. Sensor Measurement Models

As described previously, the onboard sensors equipped on the MAV consist of two primary parts: the MEMS IMU module and the RGB-D camera. The MEMS IMU module integrates three types of tri-axial sensors, generating tri-axial scalar measurements expressed in the MAV's body-fixed coordinate: a tri-axial gyroscope module that provides measurements of the angular rate ω_m, a tri-axial accelerometer module that provides measurements of the acceleration f_m, as well as a tri-axial magnetometer module that measures the local magnetic field vector expressed in the body-fixed frame: y_m, where the magnetic field vector can be considered as constant over a small-scale operating environment.

All measurements provided by the above inertial sensor are corrupted by sensor bias and measurement noises. It is a common practice to assume that the imperfections of sensor measurements include two components: a constant additive bias term and a Gaussian noise term with mean zero. Therefore, the gyroscope signals ω_m can be modeled as:

$$\omega_m = \omega + b_\omega + \nu_\omega \tag{39}$$

where ω is the actual angular rate, b_ω denotes the constant gyroscope bias, and ν_ω is Gaussian noise with zero mean. Similarly, signals of the accelerometer model can be modeled as:

$$f_m = f + b_f + v_f \tag{40}$$

where f is the actual acceleration, b_f represents the constant accelerometer bias, and ν_f is also a Gaussian noise vector with zero mean.

The local magnetic field in the earth-fixed can be expressed frame as $m = [m_x, 0, m_z]^\mathrm{T}$. Since the magnetometer is fixed with the MAV body, the magnetic readings provided by the magnetometer are measured in the body-fixed frame, which also contain sensor noises. Denoting q as the quaternion that represents the orientation of the MAV's body fixed frame with respect to the ground-fixed frame, the magnetometer model can be given as:

$$y_m = q^{-1} \times m \times q + v_m \tag{41}$$

where y_m is the readings of the magnetometer, and m denotes the Gaussian noise with zero mean.

The RGB-D visual odometry described in Section 4 can provide estimates of the MAV's relative motion, which form observations of the MAV's state and can be used as an aid to the inertial measurements (note that estimates of MAV's rotations are not used as observations of the MAV's attitude, since it is more desirable to utilize the inertial measurements for attitude estimation). In addition, an ultrasonic altimeter is employed in our system to measure the MAV's altitude relative to the ground. For our system, the translation estimates of the RGB-D visual odometry are transformed into the MAV's positions in the global frame, this yields the following output model of the RGB-D VO and altimeter:

$$\begin{aligned} y_p &= p_{x,y} + v_p \\ y_s &= p_z + v_s \end{aligned} \tag{42}$$

where y_p and y_s are measurements of the RGB-D visual odometry and altimeter, respectively. $p_{x,y}$ and p_z denote the MAV's planar positions and altitude in the global frame. Both y_p and v_s are the Gaussian white-noise of measurements.

5.3. RGB-D Visual/Inertial Navigation System Model

In our system, the MAV's orientation is represented in the quaternion formulation since the quaternion parameterization is nonsingular and well-suited for implementation on computer systems. Using the aforementioned measurement model of gyroscope and accelerometer sensors, as well as the kinematics of a rigid body, the quaternion-based dynamics model of the MAV can be formulated as:

$$\begin{aligned} \dot{q} &= \frac{1}{2} q \times (\omega_m - b_\omega - v_\omega) \\ \dot{p} &= v \\ \dot{v} &= q \times (f_m - b_f - v_\omega) \times q^{-1} + \mathbf{g} \\ \dot{b}_\omega &= 0 \\ \dot{b}_f &= 0 \end{aligned} \tag{43}$$

where q is the unit attitude quaternion $(q \in H, \|q\| = 1)$ representing the orientation of the body-fixed frame with respect to the global frame; $p \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$ represent the MAV's position and velocity in the global frame, respectively; $\mathbf{g} = [0\ 0\ g]^T$ is the local gravity vector in the ground-fixed frame. The state vector chosen for observer design is $\mathbf{x} = [q\ p\ v\ b_\omega\ b_f]^T$, along with the system input $\mathbf{u} = [\omega_m\ f_m\ \mathbf{g}]^T$.

The observations of the system consist of two parts: the magnetic measurements y_m in the body-fixed frame and measurements of the MAV's position $y_p\ y_s$ provided by RGB-D VO and altimeter, both expressed in the global frame. Using the measurement model given in Equations (41) and (42), the system output is written as:

$$\begin{bmatrix} y_m \\ y_p \\ y_s \end{bmatrix} = \begin{bmatrix} q^{-1} \times m \times q + v_m \\ p_{x,y} + v_p \\ p_z + v_s \end{bmatrix} \tag{44}$$

5.4. Observer Design of the RGB-D Visual/Inertial Navigation System

In order to verify the invariant properties of the RGB-D/inertial navigation system, the system dynamics model in Equation (43) is rewritten by ignoring the noise terms:

$$\begin{aligned} \dot{q} &= \frac{1}{2} q \times (\omega_m - b_\omega) \\ \dot{p} &= v \\ \dot{v} &= q \times (f_m - b_f) \times q^{-1} + \mathbf{g} \\ \dot{b}_\omega &= 0 \\ \dot{b}_f &= 0 \end{aligned} \tag{45}$$

Similarly, the system output model in Equation (44) is now given as:

$$\begin{aligned} y_m &= q^{-1} \times m \times q \\ y_p &= p_{x,y} \\ y_s &= p_z \end{aligned} \tag{46}$$

Following the definitions and theorems described in Section 5.1, we can verify the invariant properties of the system dynamics and output model, by defining the transformation Lie group *G* that acts on the state manifold *X* through the following actions:

$$\varphi_g(\mathbf{x}) = \varphi_{(q_0,p_0,b_{\omega,0},b_{f,0})}\begin{pmatrix} q \\ p \\ v \\ b_\omega \\ b_f \end{pmatrix} = \begin{pmatrix} q_0 \times q \\ q_0 \times (p + p_0) \times q_0^{-1} \\ q_0 \times v \times q_0^{-1} \\ b_\omega + b_{\omega,0} \\ b_f + b_{f,0} \end{pmatrix} \tag{47}$$

where $g = (q_0, p_0, b_{\omega,0}, b_{f,0})$ denotes the group action of *G* with the following physical meaning: q_0 and p_0 represent the constant rotations and translations in the global frame, and $b_{\omega,0}$, $b_{f,0} \in R^3$ denote constant translations on the bias of gyroscopes and accelerometers, respectively.

Similarly, the group actions on the system input and output manifold (*U*, *Y*) can be defined as:

$$\psi_g(\mathbf{u}) = \psi_{(q_0,p_0,b_{\omega,0},b_{f,0})}\begin{pmatrix} \omega_m \\ f_m \\ \mathbf{g} \\ m \end{pmatrix} = \begin{pmatrix} \omega_m + b_{\omega,0} \\ f_m + b_{f,0} \\ q_0 \times \mathbf{g} \times q_0^{-1} \\ q_0 \times m \times q_0^{-1} \end{pmatrix} \tag{48}$$

$$\rho_g(\mathbf{y}) = \rho_{(q_0,p_0,b_{\omega,0},b_{f,0})}\begin{pmatrix} q^{-1} \times m \times q \\ \begin{pmatrix} p_{x,y} \\ p_z \end{pmatrix} \end{pmatrix} = \begin{pmatrix} q^{-1} \times m \times q \\ q_0 \times \left(\begin{pmatrix} p_{x,y} \\ p_z \end{pmatrix} + p_0 \right) \times q_0^{-1} \end{pmatrix} \tag{49}$$

Following Equation (28), we have:

$$\begin{aligned} &\overbrace{q_0 \times q}^{\cdot} = q_0 \times \dot{q} = \frac{1}{2} q_0 \times (\omega_m - b_\omega) \\ &\overbrace{q_0 \times (p + p_0) \times q_0^{-1}}^{\cdot} = q_0 \times \dot{p} \times q_0^{-1} = q_0 \times v \times q_0^{-1} \\ &\overbrace{q_0 \times v \times q_0^{-1}}^{\cdot} = q_0 \times \dot{v} \times q_0^{-1} = q_0 \times (q \times (f_m - b_f) \times q^{-1} + \mathbf{g}) \times q_0^{-1} \\ &\qquad = (q_0 \times q) \times (f_m - b_f) \times (q_0 \times q)^{-1} + q_0 \times \mathbf{g} \times q_0^{-1} \\ &\overbrace{b_\omega + b_{\omega,0}}^{\cdot} = \dot{b}_\omega = 0 \\ &\overbrace{b_f + b_{f,0}}^{\cdot} = \dot{b}_f = 0 \end{aligned} \tag{50}$$

Therefore, the dynamics model in Equation (45) verifies $D\varphi_g(\mathbf{x}) \cdot f(\mathbf{x},\mathbf{u}) = f(\varphi_g(\mathbf{x}), \psi_g(\mathbf{u}))$. Following Definition 2, it can be concluded that the RGB-D visual/inertial navigation system is G-invariant.

Similarly, using the group actions ψg(**u**) and ρg(**y**), we can directly verify:

$$h(\varphi_g(\mathbf{x}),\psi_g(\mathbf{u}))=\begin{pmatrix}(q_0\times q)^{-1}\times q_0\times m\times q_0^{-1}\times(q_0\times q)\\ q_0\times\left(\begin{pmatrix}p_{x,y}\\ p_z\end{pmatrix}+p_0\right)\times q_0^{-1}\end{pmatrix}=\begin{pmatrix}q^{-1}\times m\times q\\ q_0\times\left(\begin{pmatrix}p_{x,y}\\ p_z\end{pmatrix}+p_0\right)\times q_0^{-1}\end{pmatrix}=\rho_g(\mathbf{y}) \quad (51)$$

Following Definition 2, the system output model given in Equation (46) is G-equivariant under the group actions φ*g*, ψ*g* and ρ*g*. As a result, the existence of the invariant observer $\dot{\hat{\mathbf{x}}}=F(\hat{\mathbf{x}},\mathbf{u},\mathbf{y})$ for the system in Equations (45) and (46) can be guaranteed by Theorem 1. Using the verified invariant properties of the system dynamics and output, we can now design the invariant state observer for the RGB-D visual/inertial navigation system by following the systematic steps described in Section 5.1.

As mentioned previously in Section 5.1, the moving frame γ(*x*) of the invariant observer can be obtained by solving φ*g*(*x*) = *c*. Let *c* be the unity (*i.e.*, *c* = *e*), φ*g*(*x*) can be given as:

$$\begin{aligned}&q_0\times q=1\\&q_0\times(p+p_0)\times q_0^{-1}=0\\&b_\omega+b_{\omega,0}=0\\&b_f+b_{f,0}=0\end{aligned} \quad (52)$$

Solving the above equations, the moving frame can be given by:

$$\gamma(x)=g=\begin{pmatrix}q^{-1} & -p & -b_\omega & -b_f\end{pmatrix}^{\mathrm{T}} \quad (53)$$

Therefore, the invariants of group *G* can be obtained by:

$$I(\hat{\mathbf{x}},\mathbf{u})=\begin{pmatrix}\varphi^b_{\gamma(\hat{x})}(\hat{v})\\ \psi_{\gamma(\hat{x})}\begin{pmatrix}\omega_m\\ f_m\\ \mathbf{g}\\ m\end{pmatrix}\end{pmatrix}=\begin{pmatrix}\hat{q}^{-1}\times v\times\hat{q}\\ \omega_m-\hat{b}_\omega\\ f_m-\hat{b}_f\\ \hat{q}^{-1}\times\mathbf{g}\times\hat{q}\\ \hat{q}^{-1}\times m\times\hat{q}\end{pmatrix}=\begin{pmatrix}I_v\\ I_{\omega_m}\\ I_{f_m}\\ I_g\\ I_m\end{pmatrix} \quad (54)$$

$$J(\hat{\mathbf{x}},\mathbf{y})=\rho_{\gamma(\hat{x})}(\mathbf{y})=\begin{pmatrix}q^{-1}\times m\times q\\ \hat{q}^{-1}\times\left(\begin{pmatrix}p_{x,y}\\ p_z\end{pmatrix}-\hat{p}\right)\times\hat{q}\end{pmatrix} \quad (55)$$

where $J(\hat{\mathbf{x}},\mathbf{y})$ is the complete set of invariants of *G* which depends on the system output **y** = *h*(**x**, **u**).

Using the invariants $I(\hat{\mathbf{x}},\mathbf{u})$ and $J(\hat{\mathbf{x}},\mathbf{y})$, the invariant output error can be obtained by following Definition 3:

$$\begin{aligned}\varepsilon(\hat{\mathbf{x}},\mathbf{u},\mathbf{y})&=\rho_{\gamma(\hat{\mathbf{x}})}(h(\hat{\mathbf{x}},\mathbf{u}))-\rho_{\gamma(\hat{\mathbf{x}})}(\mathbf{y})=J_{\gamma(\hat{\mathbf{x}})}(h(\hat{\mathbf{x}},\mathbf{u}))-J_{\gamma(\hat{\mathbf{x}})}(\mathbf{y})\\&=\begin{pmatrix}\hat{q}^{-1}\times m\times\hat{q}\\ \hat{q}^{-1}\times\left(\begin{pmatrix}\hat{p}_{x,y}\\ \hat{p}_z\end{pmatrix}-\hat{p}\right)\times\hat{q}\end{pmatrix}-\begin{pmatrix}q^{-1}\times m\times q\\ \hat{q}^{-1}\times\left(\begin{pmatrix}p_{x,y}\\ p_z\end{pmatrix}-\hat{p}\right)\times\hat{q}\end{pmatrix}=\begin{pmatrix}\hat{q}^{-1}\times m\times\hat{q}-q^{-1}\times m\times q\\ \hat{q}^{-1}\times\left(\hat{p}-\begin{pmatrix}p_{x,y}\\ p_z\end{pmatrix}\right)\times\hat{q}\end{pmatrix}\end{aligned} \quad (56)$$

Denoting υ_i ($\upsilon_i^q\times\upsilon_i^p\times\upsilon_i^v\times\upsilon_i^{b_\omega}\times\upsilon_i^{b_f}\in T_eX$) as the basis vectors of the tangent space *TeX* over the state manifold $X=S^3\times R^3\times R^3\times R^3\times R^3$, the invariant frame *wi*(**x**) is calculated by:

$$\frac{d}{d\tau}\left(\varphi_{\gamma(x)^{-1}}(\upsilon_i\tau)\right)\Big|_{\tau=0}=\frac{d}{d\tau}\begin{pmatrix} q\times e_i\tau \\ q\times(e_i\tau+p)\times q^{-1} \\ q\times e_i\tau\times q^{-1} \\ e_i\tau+b_\omega \\ e_i\tau+b_f \end{pmatrix}_{\tau=0}=\begin{pmatrix} q\times e_i \\ q\times e_i\times q^{-1} \\ q\times e_i\times q^{-1} \\ e_i \\ e_i \end{pmatrix}=\begin{pmatrix} w_i^q \\ w_i^p \\ w_i^v \\ w_i^{b_\omega} \\ w_i^{b_f} \end{pmatrix} \tag{57}$$

where $\gamma(x)^{-1}=\begin{pmatrix} q & p & b_\omega & b_f \end{pmatrix}^{\mathrm{T}}$ according to Equation (53).

Following Theorem 1 and Equation (57), the invariant observer of system in Equations (45) and (46) can be given by:

$$\begin{aligned} \dot{\hat{q}}&=\frac{1}{2}\hat{q}\times(\omega_m-\hat{b}_\omega)+\sum_{i=1}^{3}(L_i^qE)\hat{q}\times e_i \\ \dot{\hat{p}}&=\hat{v}+\sum_{i=1}^{3}(L_i^pE)\hat{q}\times e_i\times\hat{q}^{-1} \\ \dot{\hat{v}}&=\hat{q}\times(f_m-\hat{b}_f)\times\hat{q}^{-1}+\mathbf{g}+\sum_{i=1}^{3}(L_i^vE)\hat{q}\times e_i\times\hat{q}^{-1} \\ \dot{\hat{b}}_\omega&=\sum_{i=1}^{3}(L_i^{b_\omega}E)e_i \\ \dot{\hat{b}}_f&=\sum_{i=1}^{3}(L_i^{b_f}E)e_i \end{aligned} \tag{58}$$

where L_i are 1 × 3 gain matrices of the observer. Notice that:

$$\sum_{i=1}^{3}(L_i^qE)\hat{q}\times e_i=\hat{q}\times\left(\sum_{i=1}^{3}(L_i^qE)e_i\right)=\hat{q}\times\left(\begin{pmatrix} L_1^q \\ L_2^q \\ L_3^q \end{pmatrix}E\right)=\hat{q}\times\left(L^qE\right) \tag{59}$$

where L^q is a 3 × 6 matrix, and the rows of L^q are from row matrix L_i^q. Other terms in Equation (58) that associated with the observer gains can also be transformed in the same manner, leading to:

$$\begin{aligned} \dot{\hat{q}}&=\frac{1}{2}\hat{q}\times(\omega_m-\hat{b}_\omega)+\hat{q}\times\left(L^qE\right) \\ \dot{\hat{p}}&=\hat{v}+\hat{q}\times(L^pE)\times\hat{q}^{-1} \\ \dot{\hat{v}}&=\hat{q}\times(f_m-\hat{b}_f)\times\hat{q}^{-1}+\mathbf{g}+\hat{q}\times(L^vE)\times\hat{q}^{-1} \\ \dot{\hat{b}}_\omega&=L^{b_\omega}E \\ \dot{\hat{b}}_f&=L^{b_f}E \end{aligned} \tag{60}$$

Following Equation (37), the invariant state estimation error η of the observer can be obtained by:

$$\begin{pmatrix} \eta_q \\ \eta_p \\ \eta_v \\ \eta_{b_\omega} \\ \eta_{b_f} \end{pmatrix}=\varphi_{\gamma(x)}\begin{pmatrix} \hat{q} \\ \hat{p} \\ \hat{v} \\ \hat{b}_\omega \\ \hat{b}_f \end{pmatrix}-\varphi_{\gamma(x)}\begin{pmatrix} q \\ p \\ v \\ b_\omega \\ b_f \end{pmatrix}=\begin{pmatrix} q^{-1}\times\hat{q}-1 \\ q^{-1}\times(\hat{p}-p)\times q \\ q^{-1}\times\hat{v}\times q \\ \hat{b}_\omega-b_\omega \\ \hat{b}_f-b_f \end{pmatrix} \tag{61}$$

Therefore, the dynamics of the state estimation error η can be directly obtained by calculating $(d/dt)\eta$ and bringing in the invariant terms $I(\hat{\mathbf{x}},\mathbf{u})$:

$$\begin{aligned}
\dot{\eta}_q &= q^{-1}\times\dot{\hat{q}} - q^{-1}\times\dot{q}\times q^{-1}\times\hat{q} \\
&= q^{-1}\times\left(\frac{1}{2}\hat{q}\times(\omega_m-\hat{b}_\omega)+\hat{q}\times\left(L^q E\right)\right) - q^{-1}\times\left(\frac{1}{2}q\times(\omega_m-b_\omega)\right)\times(q^{-1}\times\hat{q}) \\
&= -\frac{1}{2}\eta_{b_\omega}\times(\eta_q+1)+(\eta_q+1)\times I_{\omega_m}+(\eta_q+1)\times L^q E \\
\dot{\eta}_p &= -q^{-1}\times\dot{q}\times q^{-1}\times(\hat{p}-p)\times q+\dot{q}^{-1}\times(\dot{\hat{p}}-\dot{p})\times q+q^{-1}\times(\hat{p}-p)\times\dot{q} \\
&= \eta_p\times(I_{\omega_m}+\eta_{b_\omega})+\eta_v+(\eta_q+1)\times L^p E\times(\eta_q+1)^{-1} \\
\dot{\eta}_v &= \eta_v\times(I_{\omega_m}+\eta_{b_\omega})+(\eta_q+1)\times I_{f_m}\times(\eta_q+1)^{-1}+(\eta_q+1)\times I_g\times(\eta_q+1)^{-1} \\
&\quad +(\eta_q+1)\times L^v E\times(\eta_q+1)^{-1}-I_{f_m}-\eta_{b_f} \\
\dot{\eta}_{b_\omega} &= \dot{\hat{b}}_\omega-\dot{b}_\omega = L^{b_\omega}E \\
\dot{\eta}_{b_f} &= \dot{\hat{b}}_f-\dot{b}_f = L^{b_f}E
\end{aligned} \tag{62}$$

As can be seen from Equation (62), the dynamics of the invariant state error depends on the estimated state only through the invariant terms $I(\hat{\mathbf{x}},\mathbf{u})$ rather than the trajectory of the system, which simplifies the calculation of the observer gains. Using the invariant state error in Equation (61), the invariant output error is rewritten as:

$$\begin{aligned}
E=\varepsilon(\hat{\mathbf{x}},\mathbf{u},\mathbf{y}) &= \begin{pmatrix} \hat{q}^{-1}\times m\times\hat{q}-q^{-1}\times m\times q \\ \hat{q}^{-1}\times\left(\hat{p}-\begin{pmatrix} p_{x,y} \\ p_z \end{pmatrix}\right)\times\hat{q} \end{pmatrix} \\
&= \begin{pmatrix} I_m-q^{-1}\times\hat{q}\times\hat{q}^{-1}\times m\times\hat{q}\times\hat{q}^{-1}\times q \\ \hat{q}^{-1}\times q\times q^{-1}\times(\hat{p}-p)\times q\times q^{-1}\times\hat{q} \end{pmatrix} = \begin{pmatrix} I_m-(\eta_q+1)\times I_m\times(\eta_q+1)^{-1} \\ (\eta_q+1)^{-1}\times\eta_p\times(\eta_q+1) \end{pmatrix}
\end{aligned} \tag{63}$$

5.5. Calculation of Observer Gains Based on Invariant-EKF

In order to calculate the gain matrices (L) of the invariant observer in Equation (60), we adopted a systematic approach based on the invariant-EKF (IEKF) [37]. The basic idea of the IEKF is to linearize the dynamics of the invariant state estimation error η about the current estimated state, and implement a Kalman filter on the linearized error dynamics to obtain the optimal observer gains. Details of the IEKF-based observer gains calculation will be specified in the parts of this subsection.

To better illustrate the IEKF method, we first recall the standard EKF approach that operates by linearizing the system dynamics. Consider the following nonlinear system described by:

$$\begin{aligned}
\dot{\mathbf{x}} &= f(\mathbf{x},\mathbf{u})+\mathbf{B}w \\
\mathbf{y} &= h(\mathbf{x},\mathbf{u})+\mathbf{D}v
\end{aligned} \tag{64}$$

where *w*, *v* denote the mutually-independent process and measurement Guassian white-noise with covariances $E\left\langle ww^T\right\rangle=\mathbf{Q}_w$, $E\left\langle vv^T\right\rangle=\mathbf{Q}_v$, respectively. **B**, **D** are the input matrices of the noise vectors. The optimal state estimate $\hat{\mathbf{x}}$ that minimizes the estimation error $\varepsilon=\mathbf{x}-\hat{\mathbf{x}}$ can be obtained using the conventional continuous-time EKF procedure given by:

$$\begin{aligned} &\dot{\hat{\mathbf{x}}} = f(\hat{\mathbf{x}},\mathbf{u}) + \mathbf{K}(\mathbf{y} - h(\hat{\mathbf{x}},\mathbf{u})) \\ &\mathbf{K} = \mathbf{P}\mathbf{C}^T(\mathbf{D}\mathbf{Q}_v\mathbf{D}^T)^{-1} \\ &\dot{\mathbf{P}} = \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T + \mathbf{B}\mathbf{Q}_w\mathbf{B}^T - \mathbf{P}\mathbf{C}^T(\mathbf{D}\mathbf{Q}_v{}^T\mathbf{D})^{-1}\mathbf{C}\mathbf{P} \end{aligned} \tag{65}$$

where **K** denotes the Kalman gain, and **A**, **C** are the Jacobian of the process model *f* and measurement model *h* with respect to the current estimated state ($\mathbf{A} = \partial f(\mathbf{x},\mathbf{u})/\partial\mathbf{x}\,|_{\hat{x}}$, $\mathbf{C} = \partial h(\mathbf{x},\mathbf{u})/\partial\mathbf{x}\,|_{\hat{x}}$), respectively. According to the EKF method, we linearize the system dynamics and output model given by Equation (64) about the latest estimated state using Taylor expansion:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x},\mathbf{u})\big|_{\hat{\mathbf{x}}} + \left.\frac{\partial f(\mathbf{x},\mathbf{u})}{\partial\mathbf{x}}\right|_{\hat{\mathbf{x}}}(\mathbf{x}-\hat{\mathbf{x}}) + Bw = f(\hat{\mathbf{x}},\mathbf{u}) + \mathbf{A}(\mathbf{x}-\hat{\mathbf{x}}) + \mathbf{B}w \\ \mathbf{y} &= h(\mathbf{x},\mathbf{u})\big|_{\hat{\mathbf{x}}} + \left.\frac{\partial h(\mathbf{x},\mathbf{u})}{\partial\mathbf{x}}\right|_{\hat{\mathbf{x}}}(\mathbf{x}-\hat{\mathbf{x}}) + Dv = h(\hat{\mathbf{x}},\mathbf{u}) + \mathbf{C}(\mathbf{x}-\hat{\mathbf{x}}) + \mathbf{D}v \end{aligned} \tag{66}$$

Equation (65) can be rewritten as:

$$\dot{\hat{\mathbf{x}}} = f(\hat{\mathbf{x}},\mathbf{u}) + \mathbf{K}[\mathbf{C}(\mathbf{x}-\hat{\mathbf{x}}) + \mathbf{B}v] \tag{67}$$

Following Equations (66) and (67), the dynamics of the EKF state estimation error is given by:

$$\dot{\varepsilon} = \dot{\hat{\mathbf{x}}} - \dot{\mathbf{x}} = (\mathbf{A} - \mathbf{K}\mathbf{C})\varepsilon - \mathbf{B}w + \mathbf{K}\mathbf{D}v \tag{68}$$

As described in Section 5.1, the invariant observer $F(\hat{\mathbf{x}},\ \mathbf{u},\ \mathbf{y})$ of the RGB-D visual/inertial navigation system was initially designed without considering the process and measurement noises. Introducing the noise terms *w*, *v*, the system dynamics and the associated invariant observer are rewritten as:

$$\dot{\mathbf{x}} = f(\mathbf{x},\tilde{\mathbf{u}}\,\text{-}\,w),\quad \dot{\hat{\mathbf{x}}} = F(\hat{\mathbf{x}},\tilde{\mathbf{u}},\mathbf{y}+v) \tag{69}$$

Recall the invariant state estimation error given by Equation (37): $\eta(\hat{\mathbf{x}},\mathbf{x}) = \varphi_{\gamma(x)}(\hat{\mathbf{x}}) - \varphi_{\gamma(x)}(\mathbf{x})$. From Equation (69), we can find that the time derivative of the estimation error $\dot{\eta}$ will also contain the noise terms *w*, *v*. According to the IEKF, we can now linearize $\dot{\eta}$ about the latest estimated state. Denote $\bar{\eta}$ as the actual state estimation error, and $\bar{\eta}$ should be close to the group identity *e* (*i.e.*, $\bar{\eta} = e$) when $\hat{\mathbf{x}} = \mathbf{x}$. Let $\delta\eta = \eta - \bar{\eta}$, it is proved in [48] that by linearizing $\dot{\eta}$ about $\eta = \bar{\eta}, w = 0, v = 0$ the dynamics of $\delta\eta$ reads:

$$\delta\dot{\eta} = (\mathbf{A} - \mathbf{K}\mathbf{C})\delta\eta - \mathbf{B}w + \mathbf{K}\mathbf{D}v \tag{70}$$

which takes the same form as in the conventional EKF case (Equation (68). Therefore, the observer gains *L* can be obtained from **K**, which is calculated following the procedure given by Equation (65).

As discussed in subsection 5.3, the RGB-D visual/inertial navigation system model with noise terms is given as in Equations (43) and (44) (Note that $p_{x,y}$, p_z and v_p, v_s are now merged and denoted as p and v_p):

$$\begin{aligned} &\dot{q} = \frac{1}{2}q \times (\omega_m - b_\omega - \nu_\omega) \\ &\dot{p} = v \\ &\dot{v} = q \times (f_m - b_f - \nu_f) \times q^{-1} + \mathbf{g} \\ &\dot{b}_\omega = 0 \\ &\dot{b}_f = 0 \\ &\tilde{y}_m = q^{-1} \times m \times q + v_m \\ &\tilde{y}_p = p + v_p \end{aligned} \tag{71}$$

The invariant observer with noise terms is now written as:

$$\begin{aligned}
\dot{\hat{q}} &= \frac{1}{2}\hat{q}\times(\tilde{\omega}_m-\hat{b}_\omega)+\hat{q}\times\left(L^q\tilde{E}\right)\\
\dot{\hat{p}} &= \hat{v}+\hat{q}\times(L^p\tilde{E})\times\hat{q}^{-1}\\
\dot{\hat{v}} &= \hat{q}\times(\tilde{f}_m-\hat{b}_f)\times\hat{q}^{-1}+\mathbf{g}+\hat{q}\times(L^v\tilde{E})\times\hat{q}^{-1}\\
\dot{\hat{b}}_\omega &= L^{b_\omega}\tilde{E}\\
\dot{\hat{b}}_f &= L^{b_f}\tilde{E}
\end{aligned} \tag{72}$$

where the invariant output error $\tilde{E}$ is given as:

$$\begin{aligned}
\tilde{E} &= \begin{pmatrix} \hat{q}^{-1}\times m\times\hat{q}-\tilde{y}_m \\ \hat{q}^{-1}\times\left(\hat{p}-\tilde{y}_p\right)\times\hat{q} \end{pmatrix} = \begin{pmatrix} \hat{q}^{-1}\times m\times\hat{q}-q^{-1}\times m\times q-v_m \\ \hat{q}^{-1}\times\left(\hat{p}-p-v_p\right)\times\hat{q} \end{pmatrix}\\
&= \begin{pmatrix} I_m-(\eta_q+1)\times I_m\times(\eta_q+1)^{-1}-v_m \\ (\eta_q+1)^{-1}\times\eta_p\times(\eta_q+1)-\hat{q}^{-1}\times v_p\times\hat{q} \end{pmatrix}
\end{aligned} \tag{73}$$

The derivation of the invariant state estimation error dynamics is the same as described in Subsection 5.4. Computing the time derivative of η, we have:

$$\begin{aligned}
\dot{\eta}_q &= q^{-1}\times\dot{\hat{q}}-q^{-1}\times\dot{q}\times q^{-1}\times\hat{q}\\
&= q^{-1}\times\left(\frac{1}{2}\hat{q}\times(\tilde{\omega}_m-\hat{b}_\omega)+\hat{q}\times\left(L^q\tilde{E}\right)\right)-q^{-1}\times\left(\frac{1}{2}q\times(\omega_m-b_\omega-v_m)\right)\times(q^{-1}\times\hat{q})\\
&= -\frac{1}{2}\eta_{b_\omega}\times(\eta_q+1)+(\eta_q+1)\times\tilde{I}_{\omega_m}+(\eta_q+1)\times L^q\tilde{E}+\frac{1}{2}v_\omega\times(\eta_q+1)\\
\dot{\eta}_p &= -q^{-1}\times\dot{q}\times q^{-1}\times(\hat{p}-p)\times q+q^{-1}\times(\dot{\hat{p}}-\dot{p})\times q+q^{-1}\times(\hat{p}-p)\times\dot{q}\\
&= \eta_p\times(\tilde{I}_{\omega_m}+\eta_{b_\omega})+\eta_v+(\eta_q+1)\times L^p\tilde{E}\times(\eta_q+1)^{-1}+v_m\times\eta_p\\
\dot{\eta}_v &= \eta_v\times(\tilde{I}_{\omega_m}+\eta_{b_\omega})+(\eta_q+1)\times\tilde{I}_{f_m}\times(\eta_q+1)^{-1}+(\eta_q+1)\times I_g\times(\eta_q+1)^{-1}\\
&\quad +(\eta_q+1)\times L^v\tilde{E}\times(\eta_q+1)^{-1}-\tilde{I}_{f_m}-\eta_{b_f}-\eta_v\times v_\omega+v_f\\
\dot{\eta}_{b_\omega} &= \dot{\hat{b}}_\omega-\dot{b}_\omega=L^{b_\omega}\tilde{E}\\
\dot{\eta}_{b_f} &= \dot{\hat{b}}_f-\dot{b}_f=L^{b_f}\tilde{E}
\end{aligned} \tag{74}$$

where $\tilde{I}_{\omega_m}=\omega_m-\hat{b}_\omega$, and $\tilde{I}_{f_m}=f_m-\hat{b}_f$.

Linearizing the output error $\tilde{E}$ given by Equation (73) about $\eta=\bar{\eta}, w=0, v=0$ and omitting the high-order terms ($Q(\delta\eta,\delta\eta)$, $Q(v,\delta\eta)$), we can obtain:

$$\delta\tilde{E}=\tilde{E}-\bar{\tilde{E}}=\begin{pmatrix} 2I_m\times\delta\eta_q-v_m \\ \delta\eta_p-\hat{q}^{-1}\times v_p\times\hat{q} \end{pmatrix} \tag{75}$$

Similarly, denoting $\delta\eta=\eta-\bar{\eta}$, the linearized $\delta\eta$ is given by:

$$\begin{aligned}
\delta\dot{\eta}_q &= \dot{\eta}_q - \dot{\bar{\eta}}_q = -\frac{1}{2}\delta\eta_{b_\omega} - \tilde{I}_{\omega_m} \times \delta\eta_q + L^q\delta\tilde{E} + \frac{1}{2}v_\omega \\
\delta\dot{\eta}_p &= \dot{\eta}_p - \dot{\bar{\eta}}_p = -\tilde{I}_{\omega_m} \times \delta\eta_p + \delta\eta_v + L^p\delta\tilde{E} \\
\delta\dot{\eta}_v &= \dot{\eta}_v - \dot{\bar{\eta}}_v = -\tilde{I}_{\omega_m} \times \delta\eta_v - 2\tilde{I}_{f_m} \times \delta\eta_q + L^v\delta\tilde{E} - \delta\eta_{b_f} + v_f \\
\delta\dot{\eta}_{b_\omega} &= \dot{\eta}_{b_\omega} - \dot{\bar{\eta}}_{b_\omega} = \dot{\hat{b}}_\omega - \dot{b}_\omega = L^{b_\omega}\delta\tilde{E} \\
\delta\dot{\eta}_{b_f} &= \dot{\eta}_{b_f} - \dot{\bar{\eta}}_{b_f} = \dot{\hat{b}}_f - \dot{b}_f = L^{b_f}\delta\tilde{E}
\end{aligned} \tag{76}$$

Denoting $w = \begin{bmatrix} v_f & v_\omega \end{bmatrix}^{\mathrm{T}}$ and $v = \begin{bmatrix} v_m & v_p \end{bmatrix}^{\mathrm{T}}$, Equation (76) takes the following form:

$$\delta\dot{\eta} = (\mathbf{A} - \mathbf{KC})\delta\eta - \mathbf{B}w + \mathbf{KD}v \tag{77}$$

where:

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} -S(\tilde{I}_{\omega_m}) & 0 & 0 & -\frac{1}{2}\mathbf{I} & 0 \\ 0 & -S(\tilde{I}_{\omega_m}) & \mathbf{I} & 0 & 0 \\ -2S(\tilde{I}_{f_m}) & 0 & -S(\tilde{I}_{\omega_m}) & 0 & -\mathbf{I} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} -\frac{1}{2}I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\mathbf{I} & 0 \end{bmatrix}^{\mathrm{T}} \\
\mathbf{C} &= \begin{bmatrix} -2S(I_m) & 0 & 0 & 0 & 0 \\ 0 & -\mathbf{I} & 0 & 0 & 0 \end{bmatrix} \\
\mathbf{D} &= \begin{bmatrix} -\mathbf{I} & 0 \\ 0 & -R(\hat{q}) \end{bmatrix} \\
\mathbf{K} &= \begin{bmatrix} L_m^q & L_m^p & L_m^v & L_m^{b_\omega} & L_m^{b_f} \\ L_p^q & L_p^p & L_p^v & L_p^{b_\omega} & L_p^{b_f} \end{bmatrix}^{\mathrm{T}}
\end{aligned} \tag{78}$$

As can be found from Equation (78), the matrix **K** is composed by the gains of the invariant observer in Equation (60). Therefore, observer gains L can be extracted from matrix **K**, which is updated via the procedure given by Equation (65) using matrices **A**, **B**, **C**, **D**.

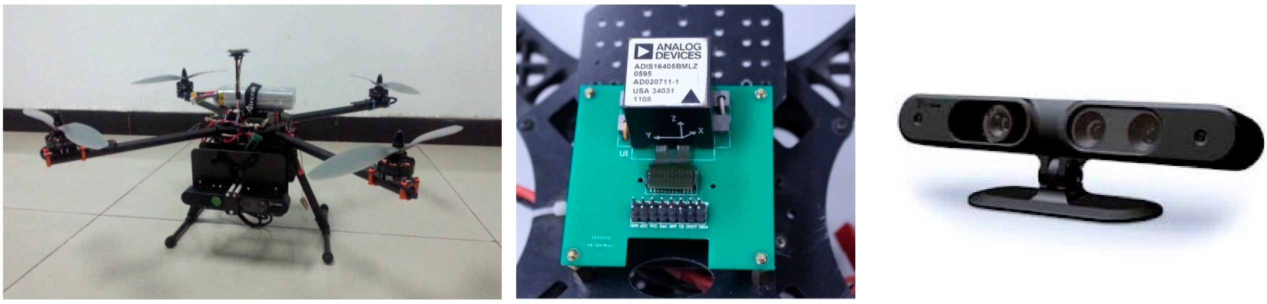
6. Implementation and Experimental Results

6.1. Implementation Details and Experimental Scenarios

In order to validate the effectiveness of the proposed RGB-D visual/inertial navigation scheme via flight test, the robust RGB-D VO and invariant observer were implemented on a prototype quadrotor MAV system shown in Figure 6a. The quadrotor was equipped with an onboard low-cost MEMS IMU (ADIS16405, produced by Analog Devices Inc., Norwood, MA, USA, Figure 6b) and an RGB-D camera (PrimeSense Carmine 1.08, produced by PrimeSense Inc., Tel-Aviv, Israel, Figure 6c).

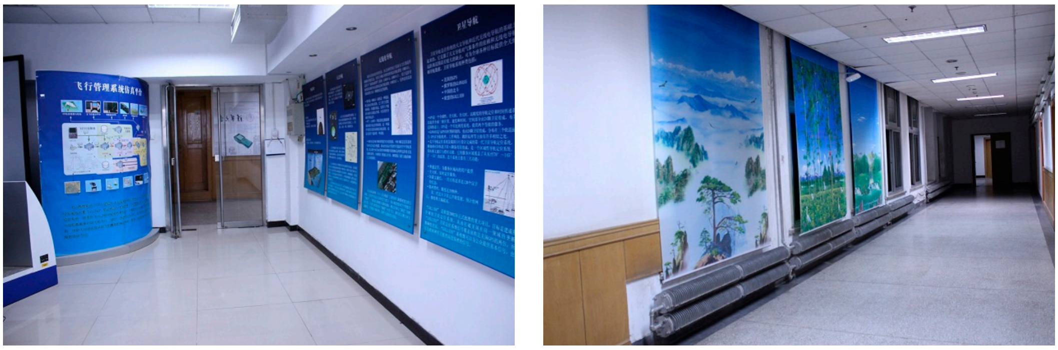
The ADIS16405 IMU consists of tri-axial gyroscope, accelerometer and magnetometer and provided inertial and magnetic measurement data at a rate of 100 Hz. The PrimeSense Carmine camera outputs RGB image and depth data with a resolution of 640 × 480 (pixels), at a rate of 30 Hz. The robust RGB-D VO algorithm described in Section 4 is implemented based on C++ and the OpenCV library [49], and runs at a same rate of 30 Hz. The overall invariant observer is implemented using an Euler numerical integration method and a complementary update scheme: the rough state estimate $\hat{\mathbf{x}}^-$ is propagated at 100 Hz, using the system dynamics $f(\hat{\mathbf{x}},\mathbf{u})$ and inertial measurements ($\mathbf{u}$), while the full estimate $\hat{\mathbf{x}}$ is updated at a rate of 30 Hz, when the measurement from the RGB-D VO is available.

Using the prototype quadrotor MAV system, a series of flight tests were carried out in two typical indoor scenarios shown in Figure 7. Scenario 1 represents the indoor environment inside a laboratory, and the MAV is controlled to follow a smooth 3D rectangular trajectory in the laboratory. The environment of Scenario 2 was a corridor inside a building, and after taking off from one end, the MAV is guided to traverse the corridor to land at the other end. For all flight experiments, the environments are without access to GPS signal and the MAV must rely only on the RGB-D visual/inertial navigation scheme to obtain state estimates.



(**a**) (**b**) (**c**)

Figure 6. Experiment system: a prototype quadrotor MAV equipped with a RGB-D camera and a low-cost MEMS IMU module. (**a**) Prototype quadrotor MAV system; (**b**) MEMS IMU: ADIS16405; **(c)** RGB-D Camera: PrimeSense Carmine 1.08.



(**a**) (**b**)

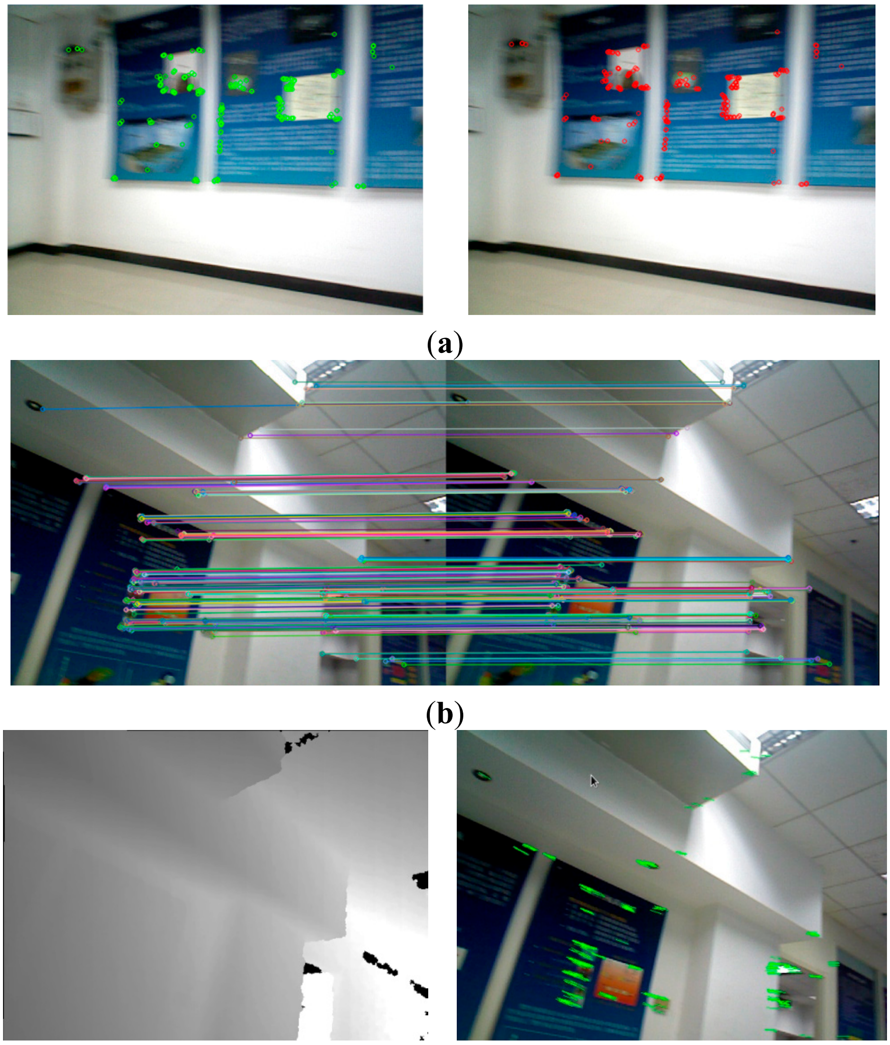
Figure 7. Indoor flight test scenarios. (**a**) Scenario 1: an actual laboratory; (**b**) Scenario 2: a corridor inside a building. Both environments are without access to GPS signal.

6.2. Indoor Flight Test Results

6.2.1. RGB-D Visual Odometry Test Results

We first conducted a number of experiments to evaluate the performance of the robust RGB-D VO described in Section 3, using the RGB image and depth data provided by the onboard PrimeSense Carmine

RGB-D camera. Figure 8 shows the feature detection and matching results in Scenario 1. Figure 8a illustrates the features (drawn in green and blue) extracted by the ORB detector from two consecutive images. Examples of the feature correspondences found by the OFC-ORB strategy from the detected features are show in Figure 8b, and the lines connecting the pairs of points represent the correspondence relationships of features. Figure 8c shows the corresponding depth image and the optical-flow disparity (drawn in green) between consecutive images. The experimental results demonstrate the effectiveness of the OFC-ORB strategy, as well as its robustness to image noise and motion blur.



(**a**)

(**b**)

(**c**)

Figure 8. Experimental results of feature detection and matching. (**a**) Features extracted from images captured at time step $m - 1$ (**left**) and m (**right**); (**b**) feature correspondences; (**c**) depth image captured by the RGB-D camera (**left**) and optical-flow between consecutive images (**right**).

To further evaluate the computational efficiency of the proposed strategy, the performance of OFC-ORB and various existing methods (Harris Corner, SIFT (Scale-invariant feature transform), SURF (speeded up robust features)) were compared in terms of time spent on finding feature correspondences. We applied different methods to the same pairs of 640 × 480 images captured by the RGB-D camera and recorded the overall computation time. For each method, the algorithm ran on the same computer platform (a single-board computer mounted on the MAV), the experiment was repeated 30 times and the average computation time was calculated over these experiments. The performance comparisons of various methods are shown in Table 1. As can be seen from Table 1, the OFC-ORB feature detection and matching strategy outperforms other existing in terms of computation efficiency. This feature makes it suitable for implementation on computationally constrained onboard platforms.

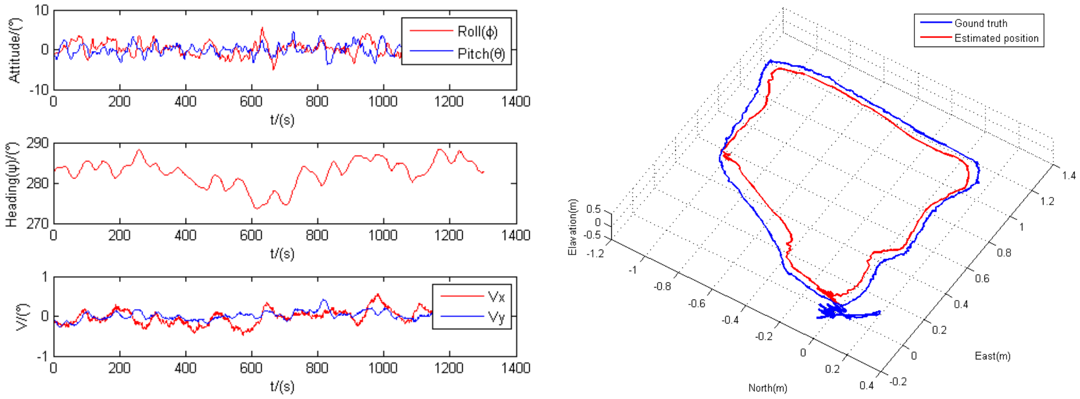
Table 1. Performance comparisons of various algorithms.

Algorithms	Average Time (ms)
Harris Corner	16.6
SIFT	6290.1
SURF	320.5
OFC-ORB	**13.2**

6.2.2. State Estimation Results

The state estimation results from indoor flight tests are plotted in Figure 9. Figure 9a,c depicts examples of attitude and velocity estimates of Scenario 1 and Scenario 2 from the invariant observer, respectively. The estimated position of Scenario 1 and Scenario 2 are shown in Figure 9b,d. These results indicate that the drifts of the low-cost MEMS IMU sensor can be effectively bounded through data fusion of RGB-D VO estimates and inertial measurements using the proposed invariant observer.

For comparison purposes, an external motion capture camera is employed in the flight tests of Scenario 1 to record the actual flight data, which is used as the ground truth trajectory. Both the estimated position and the ground truth 3-D trajectory derived from the external motion capture camera are plotted in Figure 9b. The results indicate that the estimated position closely matches the ground truth trajectory, except for a slightly larger deviation in the *x*-direction (east) of approximately 7 cm in maximum, which is likely due to a decreased number of environmental features along that direction. Despite these occasional deviations, the flight test results prove that the overall performance and accuracy of state estimation are satisfactory, and the proposed navigation scheme can provide reliable and accurate state estimates for stabilization and control of the MAV. In addition, the overall RGB-D visual/inertial navigation scheme can operate effectively in indoor environments, without relying on external navigation aids such as GPS.



(**a**) (**b**)

Figure 9. *Cont.*

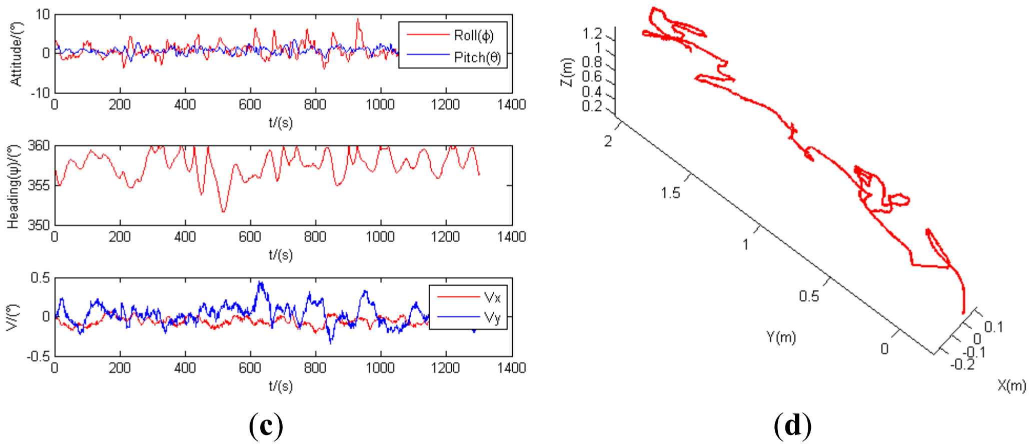


Figure 9. State estimation results. (**a**) Attitude and velocity estimates of Scenario 1; (**b**) estimated position *vs.* ground truth trajectory of Scenario 1; (**c**) attitude and velocity estimates of Scenario 2; (**d**) estimated position of Scenario 2.

7. Conclusions and Future Work

This paper presents an integrated RGB-D visual/inertial navigation scheme for the state estimation of MAVs operating in GPS-denied indoor environments. A robust RGB-D visual odometry approach was developed to estimate the relative motions of the MAV using consecutive image and depth data provided by the RGB-D camera. The motion estimates from the RGB-D VO are fused with MEMS IMU measurements through the invariant observer, which is designed based on the symmetry-preserving observer theory. The proposed navigation scheme and corresponding algorithms were implemented on a quadrotor MAV, and experimental results from indoor flight test demonstrate the efficiency and robustness of the RGB-D VO, as well as the effectiveness of the invariant observer-based estimation approach. Future work will focus on evaluating the system in more challenging, actual indoor environments with disturbances, and comparing the invariant observer-based approach with other existing filters.

Acknowledgments

This work was supported in part by the China High-Tech 863 Program under Grants 2001AA415340 and 2007AA04Z1A6, and in part by the National Natural Science Foundation of China under Grant 61174168, as well as the Aviation Science Foundation of China under Grants 20100758002 and 20128058006. The authors appreciate the technical support in the experiments provided by Zhen Tian.

Author Contributions

Dachuan Li proposed the integrated RGB-D visual/inertial navigation scheme, designed and developed the invariant observer-based state estimation approach, and wrote the paper. Liangwen Tang developed the RGB-D visual odometry algorithm, and Sheng Yang developed the quadrotor MAV prototype system and conducted the flight experiments. Nong Cheng provided advices on the derivation of the navigation system model. Qing Li and Jingyan Song supervised the work.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Bouabdallah, S.; Bermes, C.; Grzonka, S.; Gimkiewicz, C.; Brenzikofer, A.; Hahn, R.; Schafroth, D.; Grisetti, G.; Burgard, W.; Siegwart, R. Towards palm-size autonomous helicopters. *J. Intell. Robot. Syst.* **2011**, *61*, 445–471.
2. Goodrich, M.A.; Cooper, J.L.; Adams, J.A.; Humphrey, C.; Zeeman, R.; Buss, B.G. Using a mini-UAV to support wilderness search and rescue: Practices for human–robot teaming. In Proceedings of 2007 IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR 2007), Rome, Italy, 27–29 September 2007.
3. Tomic, T.; Schmid, K.; Lutz, P.; Domel, A.; Kassecker, M.; Mair, E.; Grixa, I.L.; Ruess, F.; Suppa, M.; Burschka, D. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE Robot. Autom. Mag.* **2012**, *19*, 46–56.
4. Lin, L.; Roscheck, M.; Goodrich, M.; Morse, B. Supporting wilderness search and rescue with integrated intelligence: autonomy and information at the right time and the right place. In Proceedings of 24th AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010; pp. 1542–1547.
5. Bachrach, A.; He, R.; Roy, N. Autonomous flight in unknown indoor environments. *Int. J. Micro Air Veh.* **2009**, *4*, 277–298.
6. Bachrach, A.; He, R.; Roy, N. Autonomous flight in unstructured and unknown indoor environments. In Proceedings of European Conference on Micro Aerial Vehicles (EMAV 2009), Delft, The Netherlands, 14–17 September 2009.
7. Bachrach, A.; Prentice, S.; He, R.; Roy, N. RANGE: Robust autonomous navigation in GPS-denied environments. *J. Field Robot.* **2011**, *28*, 644–666.
8. Chowdhary, G.; Sobers, D.M., Jr.; Pravitra, C.; Christmann, C.; Wu, A.; Hashimoto, H.; Ong, C.; Kalghatgi, R.; Johnson, E.N. Self-contained autonomous indoor flight with ranging sensor navigation. *J. Guid. Control Dyn.* **2012**, *29*, 1843–1854.
9. Chowdhary, G.; Sobers, D.M.; Pravitra, C.; Christmann, C.; Wu, A.; Hashimoto, H.; Ong, C.; Kalghatgi, R.; Johnson, E.N. Integrated guidance navigation and control for a fully autonomous indoor UAS. In Proceedings of AIAA Guidance Navigation and Control Conference, Portland, OR, USA, 8–11 August 2011.
10. Sobers, D.M.; Yamaura, S.; Johnson, E.N. Laser-aided inertial navigation for self-contained autonomous indoor flight. In Proceedings of AIAA Guidance Navigation and Control Conference, Toronto, Canada, 2–5 August 2010.
11. Weiss, S.; Achtelik, M.W.; Lynen, S.; Chli, M.; Siegwart, R. Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In Proceedings of 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 957–964.

12. Wu, A.D.; Johnson, E.N.; Kaess, M.; Dellaert, F.; Chowdhary, G. Autonomous flight in GPS-denied environments using monocular vision and inertial sensors. *J. Aerosp. Comput. Inf. Commun.* **2013**, *10*, 172–186.
13. Wu, A.D.; Johnson, E.N. Methods for localization and mapping using vision and inertial sensors. In Proceedings of AIAA Guidance, Navigation, and Control Conference, Honolulu, HI, USA, 18–21 August 2008.
14. Acgtelik, M.; Bachrach, A.; He, R.; Prentice, S.; Roy, N. Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. *Proc. SPIE* **2009**, *7332*, 733219.
15. Achtelik, M.; Roy, N.; Bachrach, A.; He, R.; Prentice, S.; Roy, N. Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments. In Proceedings of the 1st Symposium on Indoor Flight, International Aerial Robotics Competition, Mayagüez, Puerto Rico, 21 July 2009.
16. Voigt, R.; Nikolic, J.; Hurzeler, C.; Weiss, S.; Kneip, L.; Siegwart, R. Robust embedded egomotion estimation. In Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September; pp. 2694–2699.
17. Bachrach, A.; Prentice, S.; He, R.; Henry, P.; Huang, A.S.; Krainin, M.; Maturana, D.; Fox, D.; Roy, N. Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments. *Int. J. Robot. Res.* **2012**, *31*, 1320–1343.
18. Leishman, R.; Macdonald, J.; McLain, T.; Beard, R. Relative navigation and control of a hexacopter. In Proceedings of 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 4937–4942.
19. Guerrero-Castellanos, J.F.; Madrigal-Sastre, H.; Durand, S.; Marchand, N.; Guerrero-Sanchez, W.F.; Salmeron, B.B. Design and implementation of an attitude and heading reference system (AHRS). In Proceedings of 2011 8th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE), Merida, Mexico, 26–28 October 2011.
20. Bonnabel, S.; Martin, P.; Salaün, E. Invariant extended Kalman filter: Theory and application to a velocity-aided attitude estimation problem. In Proceedings of Joint 48th IEEE Conference on Decision and Control and 2009 28th Chinese Control Conference (CDC/CCC 2009), Shanghai, China, 15–18 December 2009; pp. 1297–1304.
21. Kelly, J.; Sukhatme, G.S. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *Int. J. Robot. Res.* **2011**, *30*, 56–79.
22. Van der Merwe, R.; Wan, E. Sigma-point Kalman filters for integrated navigation. In Proceedings of 60th Annual Meeting of the Institute of Navigation (ION), Dayton, OH, USA, 7–9 June 2004; pp. 641–654.
23. Bry, A.; Bachrach, A.; Roy, N. State estimation for aggressive flight in GPS-denied environments using onboard sensing. In Proceedings of 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 1–8.
24. Crassidis, J.L.; Markley, F.L.; Cheng, Y. Survey of nonlinear attitude estimation methods. *J. Guid. Control Dyn.* **2007**, *30*, 12–28.

25. Achtelik, M.; Achtelik, M.; Weiss, S.; Siegwart, R. Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. In Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 3056–3063.
26. Boutayeb, M.; Richard, E.; Rafaralahy, H.; Souley Ali, H.; Zaloylo, G. A simple time-varying observer for speed estimation of UAV. In Proceedings of 17th IFAC World Congress, Seoul, Korea, 6–11 July 2008; pp. 1760–1765.
27. Benallegue, A.; Mokhtari, A.; Fridman, L. High-order sliding-mode observer for a quadrotor UAV. *Int. J. Robust Nonlinear Control* **2008**, *18*, 427–440.
28. Madani, T.; Benallegue, A. Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles. In Proceedings of 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 5887–5892.
29. Benzemrane, K.; Santosuosso, G.L.; Damm, G. Unmanned aerial vehicle speed estimation via nonlinear adaptive observers. In Proceedings of 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 985–990
30. Rafaralahy, H.; Richard, E.; Boutayeb, M.; Zasadzinski, M. Simultaneous observer based sensor diagnosis and speed estimation of unmanned aerial vehicle. In Proceedings of 47th IEEE Conference on Decision and Control (CDC 2008), Cancun, Mexico, 9–11 December 2008; pp. 2938–2943.
31. Bonnabel, S.; Martin, P.; Rouchon, P. Symmetry-preserving observers. *IEEE Trans. Autom. Control* **2008**, *53*, 2514–2526.
32. Bonnabel, S.; Martin, P.; Rouchon, P. Non-linear symmetry-preserving observers on lie groups. *IEEE Trans. Autom. Control* **2009**, 54, 709–1713.
33. Mahony, R.; Hamel, T.; Pflmlin, J.-M. Nonlinear complementary filters on the special orthogonal group. *IEEE Trans. Autom. Control* **2008**, *53*, 1203–1218.
34. Martin, P.; Salaun, E. Invariant observers for attitude and heading estimation from low-cost inertial and magnetic sensors. In Proceedings of 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 1039–1045.
35. Martin, P.; Salaun, E. Design and implementation of a low-cost attitude and heading nonlinear estimator. In Proceedings of Fifth International Conference on Informatics in Control, Automation and Robotics, Signal Processing, Systems Modeling and Control, Funchal, Portugal, 11–15 May 2008; pp. 53–61.
36. Martin, P.; Salaün, E. Design and implementation of a low-cost observer-based attitude and heading reference system. *Control Eng. Pract.* **2010**, *18*, 712–722.
37. Bonnabel, S. Left-invariant extended Kalman filter and attitude estimation. In Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 1027–1032.
38. Barczyk, M.; Lynch, A.F. Invariant extended Kalman filter design for a magnetometer-plus-GPS aided inertial navigation system. In Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, 12–15 December 2011; pp. 5389–5394.

39. Barczyk, M.; Lynch, A.F. Invariant observer design for a helicopter UAV aided inertial navigation system. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 791–806.
40. Cheviron, T.; Hamel, T.; Mahony, R.; Baldwin, G. Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of UAV. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 2010–2016.
41. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
42. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In Proceedings of Computer Vision–ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin, Germany, 2006; Volume 1, pp. 430–443.
43. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In Proceedings of Computer Vision–ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; Springer: Berlin, Germany, 2010; pp. 778–792.
44. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. *Proc. IJCAI* **1981**, *81*, 674–679.
45. Bouguet, J.Y. Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker Description of the Algorithm. Available online: http://robots.stanford.edu/cs223b04/algo_affine_tracking.pdf (accessed on 20 April 2015).
46. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *5*, 698–700.
47. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395.
48. Nistér, D. Preemptive RANSAC for live structure and motion estimation. *Mach. Vis. Appl.* **2005**, *16*, 321–329.
49. OpenCV. Available online: http://opencv.org/ (accessed on 27 December 2014).