Vision-Based State Estimation for Autonomous Rotorcraft MAVs in Complex Environments

Shaojie Shen, Yash Mulgaonkar, Nathan Michael, and Vijay Kumar

Abstract— **In this paper, we consider the development of a rotorcraft micro aerial vehicle (MAV) system capable of vision-based state estimation in complex environments. We pursue a systems solution for the hardware and software to enable autonomous flight with a small rotorcraft in complex indoor and outdoor environments using only onboard vision and inertial sensors. As rotorcrafts frequently operate in hover or near-hover conditions, we propose a vision-based state estimation approach that does not drift when the vehicle remains stationary. The vision-based estimation approach combines the advantages of monocular vision (range, faster processing) with that of stereo vision (availability of scale and depth information), while overcoming several disadvantages of both. Specifically, our system relies on fisheye camera images at 25 Hz and imagery from a second camera at a much lower frequency for metric scale initialization and failure recovery. This estimate is fused with IMU information to yield state estimates at 100 Hz for feedback control. We show indoor experimental results with performance benchmarking and illustrate the autonomous operation of the system in challenging indoor and outdoor environments.**

I. Introduction

Rotorcrafts, and quadrotors in particular, are able to adeptly fly in 3-D environments with the ability to hover in place and quickly maneuver around clutter and obstacles. For this reason, rotorcrafts are an attractive platform for search-and-rescue and first-response applications. However, one of the key challenges for autonomous flight is the lack of low-power and lightweight sensor solutions for state estimation. While 3-D lidars are used in many settings, their mass exceeds most MAV payload capacities. Lightweight laser-based solutions are available but are mostly suitable for 2-D and 2.5-D indoor environments. Indeed, in our previous work [1], we developed a state estimation method based on laser and IMU sensors to enable a MAV to fly autonomously in a structured indoor environment. Our assumption of a rectilinear model of the environment in this previous work necessarily limited the capabilities of the MAV and prevented operation in more complex unstructured environments. Stereo-based solutions can potentially be scaled down in terms of weight and power but the resulting smaller baseline can make them unsuitable for large spaces [2].

In this paper, we focus on the problem of estimating the state of a micro aerial vehicle using only onboard cameras and an IMU, with a modest netbook processor whose power budget is less than 5% of the power required to fly. The problem of monocular visual-inertial (VINS) state estimation is well studied in the literature [3]–[5]. A nonlinear observability analysis of the estimation problem shows the presence of unobservable modes that can only be eliminated through motions that involve non-zero linear accelerations [4, 5]. Indeed, this motion is required to estimate the metric information that is otherwise not available. Thus it may be difficult to directly use state-of-the-art, tightly-coupled VINS systems such as the ones described in [6] on rotorcrafts, when the missions may require the rotorcraft to hover in place for extended periods of time. Moreover, these approaches often have complexity that scales with respect to both the number of features and the number of observations of a specific feature, making them difficult to use with low-power and lightweight processors at high frame rates.

Fig. 1. The experimental platform with onboard computation (Intel Core 2 Duo 1.86 GHz processor) and sensing (right: primary fisheye camera, center: IMU, left: secondary camera).

To our knowledge, the only functional monocular vision-based autonomous rotorcraft MAV is proposed in [7] and uses a modified version of the Parallel Tracking and Mapping (PTAM) software [8] for pose estimation using a downward-facing camera. It is clear that while PTAM offers a powerful framework for real-time visual tracking and SLAM, it was originally designed for augmented reality applications and as such may not be the best approach for MAV state estimation. We believe better performance and computational efficiency can be achieved by leveraging additional sensors onboard the vehicle. Further, this approach relies on using a downward-facing camera which may not be sufficient for low-altitude operation with potentially large changes in depth (height) and rapid changes in the observed environment as propeller downwash interacts with vegetation and debris. Finally, a downward-facing camera severely limits the application of vision-based obstacle detection for planning and control purposes.

Stereo vision-based state estimation approaches for autonomous MAVs such as those proposed in [2, 9, 10] do not suffer from the problem scale drift as seen in monocular

S. Shen, Y. Mulgaonkar, and V. Kumar are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA. {shaojie, yashm, kumar}@grasp.upenn.edu

N. Michael is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. nmichael@cmu.edu

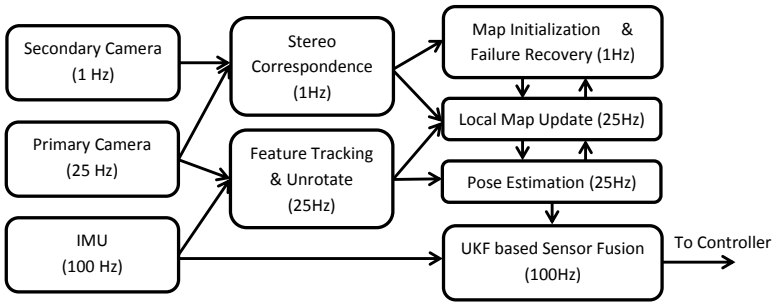


Fig. 2. The vision-based state estimation system design with update frequency.

systems or limit the observable camera motion. However, due to a limited baseline, stereo vision systems are often unable to make use of the information from distant features.

At the other end of the spectrum, there are many successful attempts at integrating vision for autonomous flight which do not rely on estimates of metric state estimation [11, 12]. While sensors that yield, for example, only optical flow or visual odometry can be used for low-level control, they fundamentally limit the flight capabilities of the MAV when operating in complex environments.

The main contribution of this paper is a systems solution (hardware, software, and algorithms) to enable vision-based state estimation onboard a rotorcraft MAV and permit autonomous flight in complex unstructured environments. Our solution combines the advantages of monocular vision (longer range and faster processing) with that of stereo vision (availability of scale and depth information), while overcoming the disadvantages of monocular vision (unobservability and drift) and stereo (small baseline and shorter range). Thus our approach relies on the fusion of information from a high frame-rate forward facing fisheye camera, a low frame-rate second camera, and an IMU. We choose to decouple the problems of attitude and position estimating and develop a constant time complexity vision-based state estimator. Our approach is particularly apt for rotorcraft state estimation as it does not exhibit pose estimate drift while the vehicle hovers. Second, we design and implement our approach on a quadrotor platform using only onboard computation and processing. We benchmark the system performance and present several experimental studies of autonomous operation in challenging indoor and outdoor environments.

II. Approach

For this work, we focus on addressing the problem of developing a high-rate state estimation methodology that permits autonomous rotorcraft MAV flight. We do not address the full vision-based SLAM problem [13, 14] as these approaches are too computationally expensive to operate in real-time on the MAV onboard computer. Given our prior remarks regarding the observability requirement of motion during flight for monocular vision-based methods, we choose to use two cameras in this work, a primary forward facing fisheye camera that operates at a high-rate for pose estimation and local mapping, and a secondary camera that operates at a low-rate and compensates for the limitations of monocular vision-based approaches. Note that although we have a stereo camera arrangement on our MAV, we do not rely on stereo correspondences to perform motion estimation. We now detail our methodology following the logical ordering suggested by the system diagram (Fig. 2).

A. Camera Model, Feature Detection and Tracking

Both cameras in the system are modeled with a spherical camera model and calibrated using the Omnidirection Calibration Toolbox [15]. For the primary camera that runs at 25 Hz, we detect and track Shi-Tomasi corners [16] using the KLT tracker [17]. It is worth noting that due to the limited motion that occurs between image frames, we are able to perform the feature detection and tracking calculations on the distorted fisheye camera image, reducing the overall computational burden of this step. Using the camera calibration parameters, all features are transformed into unit length feature observation vectors $\mathbf{u}_{ij}^C$ for further processing. Here we denote $\mathbf{u}_{ij}^C$ as an observation of the i^{th} feature in the j^{th} image in the camera body frame.

B. Pose Estimation via 2D-3D Correspondences

We begin by assuming a known 3-D local feature map and detail the initialization and maintenance of this map later in this section. Given observations of a local map consisting of known 3-D features, the 3-D position of the camera can be estimated by minimizing the sum-of-square reprojection error of the observed features:

$$\mathbf{r}_j = \underset{\mathbf{r}_j}{\operatorname{argmin}} \sum_{i \in \mathcal{I}} \left\| \frac{\mathbf{r}_j - \mathbf{p}_i}{\|\mathbf{r}_j - \mathbf{p}_i\|} \times \mathbf{u}_{ij} \right\|^2 \tag{1}$$

where, as shown in Fig. 3, $\mathbf{r}_j$ is the 3-D position of the camera in the world frame when the j^{th} image is captured, $\mathcal{I}$ represents the set of features observed in the j^{th} image, and $\mathbf{p}_i$ is the 3-D position of the i^{th} feature in the world frame. $\mathbf{u}_{ij}$ is a unit length vector in the unrotated camera frame such that

$$\mathbf{u}_{ij} = R_I^W R_C^I \mathbf{u}_{ij}^C \tag{2}$$

where R_I^W and R_C^I are rotation matrices representing the orientation of the IMU in the world frame and the orientation of the camera in the IMU frame, respectively. R_I^W is obtained from the IMU and R_C^I is calibrated offline.

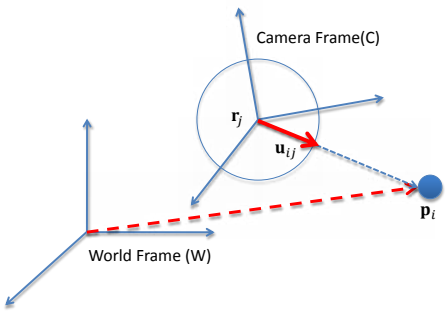


Fig. 3. Notation. $\mathbf{r}_j$ is the position vector of the j^{th} camera pose in the world frame and $\mathbf{p}_i$ is the position vector of the i^{th} feature in the world frame. $\mathbf{u}_{ij}$ is a unit length vector in the unrotated camera frame.

Note that (1) is nonlinear, however, if we assume that the translative motion of the camera between two consecutive images is small, we can approximate (1) as:

$$\mathbf{r}_j = \underset{\mathbf{r}_j}{\operatorname{argmin}} \sum_{i \in \mathcal{I}} \frac{\|(\mathbf{r}_j - \mathbf{p}_i) \times \mathbf{u}_{ij}\|^2}{d_i} \tag{3}$$

where $d_i = \|\mathbf{r}_{j-1} - \mathbf{p}_i\|$ are known quantities. By taking the derivative of (3) and setting it to zero, we obtain a linear system with the optimal camera position $\mathbf{r}_j$ as the solution:

$$\left(\sum_{i \in \mathcal{I}} \frac{\mathbb{I}_3 - \mathbf{u}_{ij}\mathbf{u}_{ij}^{\mathrm{T}}}{d_i} \right) \mathbf{r}_j = \sum_{i \in \mathcal{I}} \frac{\mathbb{I}_3 - \mathbf{u}_{ij}\mathbf{u}_{ij}^{\mathrm{T}}}{d_i} \mathbf{p}_i \tag{4}$$

Equation (4) always represents three equations in three unknowns, regardless of the number of observed features. Therefore, the position estimation problem can be solved efficiently in constant time.

A minimum of two feature correspondences are required to solve this linear system. As such, an efficient 2-point RANSAC can be applied for outlier rejection. This leads to significant reduction of the required computation load compared to the traditional 5-point algorithm [18].

C. Monocular Vision-Based Local Mapping

As stated in Sect. II-B, a map consisting of 3-D features is required to estimate the position of the camera. We approach the local mapping problem as an iterative procedure where the position of the camera is assumed to be a *noiseless* quantity. We do not perform optimizations for the position of both the camera and the features at the same time (following traditional SLAM approaches) due to CPU limitations.

We define the *local map* as the set of currently tracked 3-D features and cull lost features from the map. New features are added to the local map if the current number of features is smaller than the maximum allowable feature count. Given a set of $\mathcal{J}$ observations of the i^{th} feature at different camera positions, we can formulate the 3-D feature location $\mathbf{p}_i$ via triangulation as:

$$\mathbf{p}_i = \underset{\mathbf{p}_i}{\operatorname{argmin}} \sum_{j \in \mathcal{J}} \|(\mathbf{p}_i - \mathbf{r}_j) \times \mathbf{u}_{ij}\|^2 \tag{5}$$

and solve for it via the following linear system:

$$\left(\sum_{j \in \mathcal{J}} \mathbb{I}_3 - \mathbf{u}_{ij}\mathbf{u}_{ij}^{\mathrm{T}} \right) \mathbf{p}_i = \sum_{j \in \mathcal{J}} \left(\mathbb{I}_3 - \mathbf{u}_{ij}\mathbf{u}_{ij}^{\mathrm{T}} \right) \mathbf{r}_j \tag{6}$$

Again, it can be seen that regardless of the number of observations of a specific feature, the dimensionality of (6) is always three. This enables multi-view triangulation with constant computation complexity. Moreover, the condition number or the ratio between the minimum and maximum eigenvalues of the matrix $\left(\sum_{j \in \mathcal{J}} \mathbb{I}_3 - \mathbf{u}_{ij}\mathbf{u}_{ij}^{\mathrm{T}} \right)$ gives us information about the quality of the solution of the linear system. We evaluate every feature based on this ratio $(\lambda_{min}/\lambda_{max})$ and reject specific features as unsuitable for feature triangulation based on a predefined threshold.

D. Low Frame Rate Stereo Correspondence

The secondary camera (running at 1 Hz) tracks Shi-Tomasi corners augmented with BRIEF [19] descriptors in order to perform stereo correspondence with the features observed and tracked by the primary camera. With a known extrinsic camera calibration, we eliminate outlier features using the epipolar geometry constraint. Features are triangulated using the associated stereo correspondence and added to the local map. Additionally, we initialize the estimation approach detailed in Sect. II-B using a local 3-D feature map generated using stereo observations.

The stereo-based feature update runs alongside the monocular mapping thread. For each feature, if both stereo and monocular-based location estimates are available, the stereo-based location is preferred for camera position estimation.

While, this low frame rate stereo correspondence does not provide sufficient information for camera position estimation, it does enable the initialization of the local map and failure detection and recovery. By reducing the update rate of the stereo vision subsystem, we greatly reduce the computational overhead while benefitting from the improved state estimate via the stereo pair correspondance.

E. Failure Detection and Recovery

We now introduce two heuristics to detect the failure of the monocular vision algorithm to accurately track features based on stereo correspondance. Such estimation failures are fatal for the MAV and can occur when the camera undergoes a large rotation with negligible translation, resulting in the sudden lost of features.

The first heuristic seeks to cross-validate the expectation of the observed feature positions in the camera frame based on features from both the monocular and stereo camera subsystems. If we combine both features sets into a common set $\mathcal{K}$, this ratio is stated as:

$$\gamma = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \frac{\|\mathbf{p}_k - \mathbf{r}_j\|}{\|\mathbf{p}_k^s - \mathbf{r}_j\|} \tag{7}$$

where $\mathbf{p}_k^s$ is the location of feature k obtained by stereo correspondence. The two subsystems are consistent when $\gamma \approxeq 1$ and inconsistent otherwise.

Alternatively, we can compute the error between state estimates according to the calculations based on the fully decoupled monocular and stereo vision-based features. We note a failure if the error in state estimates grows beyond a threshold.

During failure recovery, we cull the local map of all feature positions obtained via the monocular-vision subsystem and reinitialize the map based on stereo-vision triangulated features ($\mathbf{p}^s$). We also purge the history of all tracked features, consequently reinitializing the scale estimation based on $\mathbf{p}^s$.

F. UKF-Based Sensor Fusion

The 25 Hz pose estimate from the vision system alone is not sufficient to control the robot. We therefore employ a UKF (Unscented Kalman filter) framework with delayed measurement compensation to estimate the pose and velocity of the robot at 100 Hz [20]. The system state is defined as:

$$\mathbf{x} = \left[\mathbf{s}, \dot{\mathbf{s}}, \mathbf{\Phi}, \mathbf{a}_b, \mathbf{\Psi}_b\right]^{\mathrm{T}} \tag{8}$$

where $\mathbf{s} = [x,\, y,\, z]^{\mathrm{T}}$ is the position of the robot; $\mathbf{\Phi} = [\phi,\, \theta,\, \psi]^{\mathrm{T}}$ is the roll, pitch, and yaw Euler angles that represent the 3-D orientation of the robot; $\mathbf{a}_b = \left[a_{b_x},\, a_{b_y},\, a_{b_z}\right]^{\mathrm{T}}$ is the bias of the 3D accelerometer in the body frame; and $\mathbf{\Psi}_b = [\phi_b,\, \theta_b]^{\mathrm{T}}$ is the bias of the roll and pitch estimate from the IMU. We avoid the need to estimate the metric scale in the filter (as in [7]) through the stereo-based reinitialization noted above.

1) Process Model: We consider an IMU-based process model:

$$\mathbf{u} = [\omega,\, \mathbf{a}]^{\mathrm{T}} = [\omega_x,\, \omega_y,\, \omega_z,\, a_x,\, a_y,\, a_z]^{\mathrm{T}} \tag{9}$$

$$\mathbf{v} = [\mathbf{v}_\omega,\, \mathbf{v}_a,\, \mathbf{v}_{a_b}]^{\mathrm{T}} \tag{10}$$

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t,\, \mathbf{u}_t,\, \mathbf{v}_t) \tag{11}$$

where $\mathbf{u}$ is the body frame angular velocities and linear accelerations from the IMU. $\mathbf{v}$ represents additive Gaussian noise associated with the gyroscope, accelerometer, accelerometer bias, and IMU attitude bias.

2) Measurement Model: The position estimate from the vision system is first transformed to the IMU frame and then assembled with the orientation estimate from the IMU to form a 6-DOF pose measurement:

$$\mathbf{z} = \begin{bmatrix} \mathbf{r} + R_I^W R_C^I \mathbf{d}_C^I \\ \mathbf{\Phi} \end{bmatrix} \tag{12}$$

where $\mathbf{d}_C^I$ is the translation between the camera and the IMU. The measurement model is linear and may be written as:

$$\mathbf{z} = H\mathbf{x} + \mathbf{n} \tag{13}$$

where H extracts the 6-DOF pose in the state and $\mathbf{n}$ is additive Gaussian noise. The measurement model is linear and the measurement update can be performed via a KF update step.

G. Control

Although the focus of this work is primarily the state estimation considerations required for autonomous navigation, we wish to build a complete autonomous MAV system that uses the onboard state estimate for feedback control. As such, we implemented an LQR controller based on a linearized system model [21]. During the experiments, the operator either directly commands the velocity of the robot or sends high-level waypoint commands. When no command is given, the robot hovers in place. Details of the control subsystem are provided in our prior work [1].

III. Experimental Results

A. Experiment Design and Implementation Details

The robot platform is the Pelican quadrotor sold by Ascending Technologies, GmbH [22] and equipped with an IMU (accelerometer, gyroscope) and and AutoPilot board with two ARM7 processors. We developed custom firmware to run on the embedded processor to address attitude stabilization requirements. The other computation unit onboard is an Intel Core 2 Duo 1.86 GHz processor with 2 GB of RAM installed on a baseboard developed by the PIXHAWK team [23] with a 120 GB SSD hard drive. The sensors on the robot include two uEye 1220SE cameras that capture images with 752×480 resolution. A Fujinon 1.4 mm f1.4 fisheye lens with 185 degrees field of view is installed on the primary camera that runs at 25 Hz. A Kowa 3.5 mm f1.4 lens is installed on the secondary camera that runs at 1 Hz. We also added a MicroStrain 3DM-GX3-25 IMU for enhanced attitude estimation performance. Communication with the robot for monitoring experiment progress is via 802.11g networking. The total mass of the platform is 1.983 kg.

All algorithm development is in C++ using ROS [24] as the interfacing robotics middleware. We utilize the OpenCV library [25] for corner and descriptor extraction. The maximum number of features is set to be 1000. We determine the threshold for monocular-based feature triangulation ($\lambda_{min}/\lambda_{max}$, Sect. II-C) by performing triangulation of a checkerboard with known size. We pick the threshold when the triangulation error is less than 5 %. The threshold that triggers failure of the monocular-based algorithm (γ, Sect. II-E) is set to be 0.9. We experimentally verified that the quality of the roll and pitch estimate from the IMU is sufficiently good and does not drift over time. Therefore, during actual implementation, we do not include the bias in attitude estimate $\mathbf{\Psi}_b$ in the UKF state vector (Sect. II-F). All computation is done onboard the robot. The whole system consumes approximately 50% of the CPU power.

The experiment environment includes a laboratory space (Fig. 4) equipped with a sub-millimeter accurate Vicon motion tracking system [26], and a courtyard in the School of Engineering and Applied Science at the University of Pennsylvania. Three experiments are presented: (1) a study of the estimator performance compared to ground truth data; (2) autonomous hovering using the onboard state estimation for

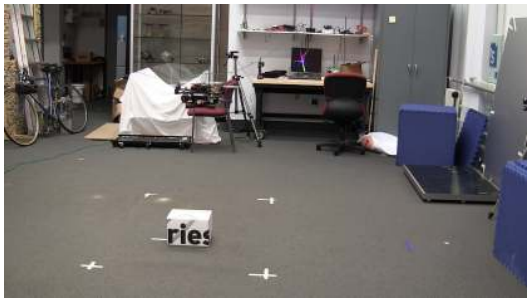


Fig. 4. The laboratory space that the indoor experiments are performed. There is no special modification to the lab. No artificial features are used.

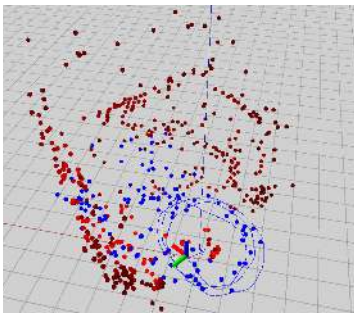


Fig. 5. The local map maintained by the vision system during the experiment (Sect. III-B) in a laboratory environment (Fig. 4). The structure of the lab is clearly visible. Red dots are features that are triangulated via the monocular vision-based mapping (Sect. II-C). The intensity level of the red dots indicates the quality of the triangulation. Blue dots represent features that are triangulated via stereo correspondence (Sect. II-D). All features in the map are visible in a single image.

feedback control; and (3) autonomous navigation in complex environments.

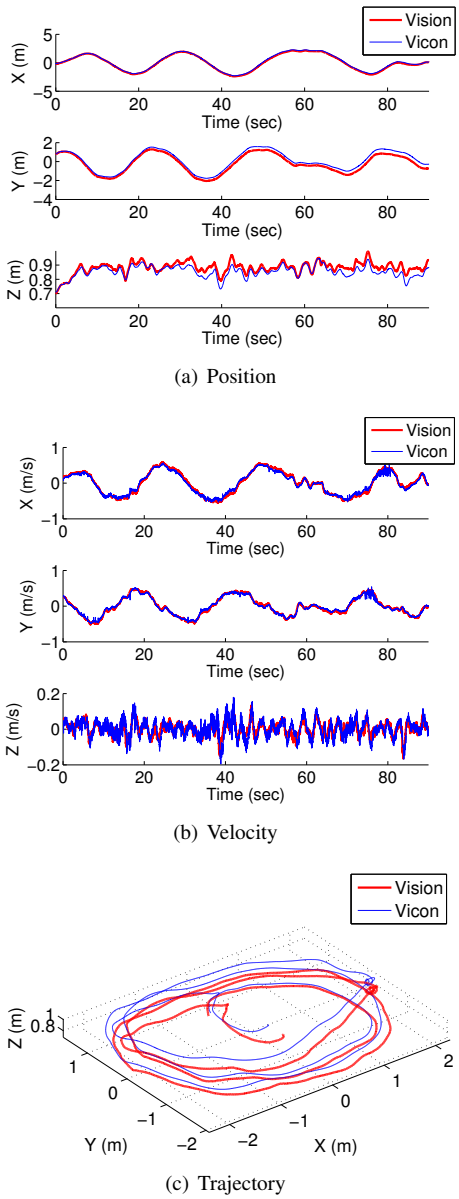
B. Evaluating Estimator Performance

The first study considers the accuracy of the onboard estimate compared to the Vicon estimate (Fig. 6) while the robot is commanded via operator velocity inputs. The local map maintained by the vision system is shown in Fig. 5. Since our approach does not enforce global consistency, we expect that a slow drift may occur in the position estimate (Figs. 6(a) and 6(c)). However, we can empirically verify that the scale and the shape of the position estimate matches with the ground truth. On the other hand, the onboard velocity estimate

TABLE I

AVERAGE NUMBER OF MONOCULAR-BASED AND STEREO-BASED FEATURES WITH CHANGING YAW RATE.

Avg. Yaw Rate (deg/s)	Avg. Mono. Feat.	Avg. Stereo Feat.
0.14	355.16	56.60
6.43	125.92	36.76
11.04	67.79	18.69
33.01	6.00	18.09



(a) Position

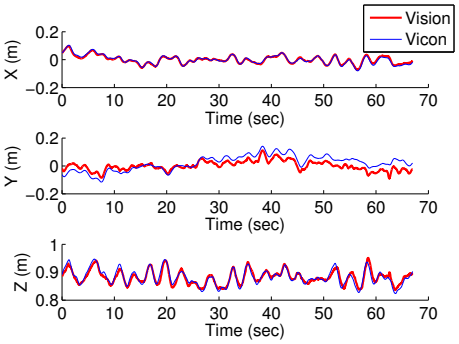
(b) Velocity

(c) Trajectory

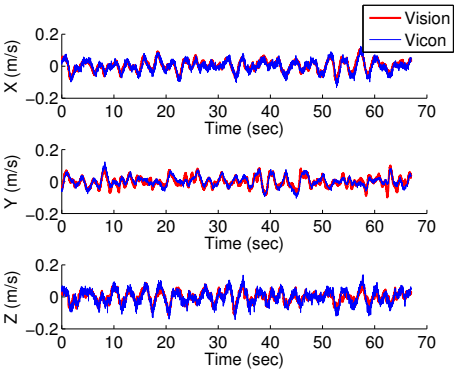
Fig. 6. Error between the estimates according to the Vicon and onboard estimator while the robot flies in a laboratory environment.

compares well with the Vicon estimates with standard deviations of $\{\sigma_{v_x}, \sigma_{v_y}, \sigma_{v_z}\} = \{0.0512, 0.0383, 0.0317\}$ (m/s). Note that the Vicon velocity estimate is obtained by a one-step numerical derivative of the position and in fact more stochastic than the onboard velocity estimate. Therefore, it is likely that the actual velocity estimation error are smaller than the values reported above.

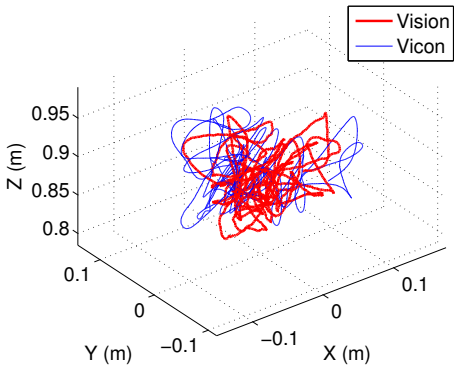
We conducted further experiments to evaluate the effects of yaw rate on estimator performance by evaluating the number of tracked features as the robot flies in a trajectories that are similar to Fig. 6(c) in shape, but with different yaw rate. In Table I, we report the average number of features that are triangulated from monocular and stereo observations in each trial. As the yaw rate increases, the number of valid monocular-based features drops rapidly due to insufficient



(a) Position



(b) Velocity



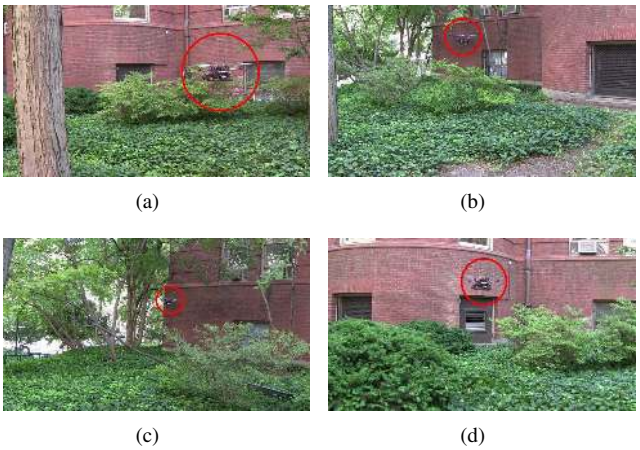
(c) Trajectory

Fig. 7. The robot is commanded to hover at $\{0, 0, 0.9\}$ based only on feedback from the onboard estimator.

translation while the feature is observable. The stereo-based also degrade, but less severely, permitting state estimation and normal operation. Clearly, these results support the argument that stereo observations, even at a low rate, improve the robustness of the state estimation algorithm enough to motivate the inclusion of an additional camera despite the added payload cost.

C. Autonomous Hovering

In the second study, the robot is commanded to hover about a fixed point using the onboard state estimate for feedback control (Fig. 7). As expected and



(a) (b) (c) (d)

Fig. 8. Autonomous navigation in a complex environment that contains buildings, trees, shrub, and grass. We highlight the position of the robot with a red circle. Videos of the experiments are available at `http://www.seas.upenn.edu/~shaojie/ICRA2013.mp4`.

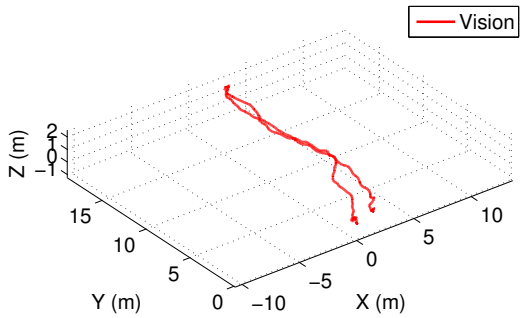


Fig. 9. The trajectory of the robot in the autonomous navigation experiment.

by design, in the hover case, the feature set does not change significantly and as such, there is very little drift in the onboard position estimate (Figs. 7(a) and 7(c)). The standard deviations of the error of the onboard estimate in position and velocity are $\{\sigma_x, \sigma_y, \sigma_z\} = \{0.0059, 0.0341, 0.0099\}$ (m) and $\{\sigma_{v_x}, \sigma_{v_y}, \sigma_{v_z}\} = \{0.0170, 0.0176, 0.0251\}$ (m/s). The hover performance during closed loop control based on the onboard state estimate is $\{\sigma_x, \sigma_y, \sigma_z\} = \{0.0313, 0.0529, 0.0272\}$ (m).

D. Autonomous Navigation in Complex Environments

We now consider a study of autonomous navigation in complex 3-D environments. The experimental environment contains a mixture of buildings, trees, shrub, and grass. The robot is commanded to autonomously follow a sequence of waypoints with a maximum speed of 0.5 m/s. The duration of this experiment is 160.2 seconds and the total flight distance of the robot is 52.3 m. Images of the robot flying is shown in Fig. 8. The robot returns to the starting position after the flight. The trajectory of the robot is shown in Fig. 9. As expected, there is a small amount drift in the position due to the lack of global consistency enforcement. However, this

slow drifting does not affect the flight performance. Note that during the experiment, the downwash generated by the propellers causes significant movement of shrub and grass that are directly under the robot (Fig. 8(b)). Such changes in the environment may impact the performance of systems requiring downward facing cameras. Empirically, we observe that during the experiment the vehicle maintains consistent and stable operation.

IV. Conclusion and Future Work

In this paper, we detail a systems solution (hardware, software, and algorithms) to enable vision-based state estimation onboard a rotorcraft MAV and permit autonomous flight in complex unstructured environments. Our solution balances the advantages and disadvantages of monocular and stereo vision based approaches to yield a robust real-time solution that operates on a netbook class processor. Our approach is particularly apt for rotorcraft state estimation as it does not exhibit estimate drift while the vehicle hovers. We detail the design and implementation our approach on a quadrotor platform using only onboard computation and processing. We benchmark the system performance and present several experimental studies of autonomous operation in challenging indoor and outdoor environments.

We are interested in several research directions building upon this work. First, we must address the global consistency problem which we do not consider in this work. To this end, we are currently investigating appearance-based loop closure detection methods such as those in [13, 14]. Additionally, we are interested in addressing the problem of vehicle path planning given local (or global) maps which represent the spatially projected locations of observed features. These maps clearly pose a challenge when considering path planning as they lack or indirectly represent the necessary free-space information required to ensure collision free plans.

References

[1] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011, pp. 20–25.

[2] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, , and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor MAV," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Vilamoura, Algarve, Portugal, Oct. 2012.

[3] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Roma, Italy, Apr. 2007, pp. 3565–3572.

[4] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Intl. J. Robot. Research*, vol. 30, no. 1, pp. 56–79, Jan. 2011.

[5] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *Intl. J. Robot. Research*, vol. 30, no. 4, pp. 407–430, Apr. 2011.

[6] D. G. Kottas, J. A. Hesch, S. L. Bowman, and S. I. Roumeliotis, "On the consistency of vision-aided inertial navigation," in *Proc. of the Intl. Sym. on Exp. Robot.*, Quebec, Canada, June 2012.

[7] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Saint Paul, MN, May 2012, pp. 957–964.

[8] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, Nov. 2007.

[9] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments," in *Proceedings of the SPIE Unmanned Systems Technology XI*, vol. 7332, Orlando, FL, 2009.

[10] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Proc. of the Intl. Sym. of Robot. Research*, Flagstaff, AZ, Aug. 2011.

[11] C. Bills, J. Chen, and A. Saxena, "Autonomous MAV flight in indoor environments using single image perspective cues," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011, pp. 5776–5783.

[12] G. de Croon, C. D. Wagterb, B. Remesb, and R. Ruijsinkb, "Sub-sampling: Real-time vision for micro air vehicles," *Robot. and Autom. Syst.*, vol. 60, no. 2, pp. 167–181, Feb. 2012.

[13] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," in *Proc. of Robot.: Sci. and Syst.*, Zaragoza, Spain, June 2010.

[14] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *Intl. J. Robot. Research*, vol. 30, no. 9, pp. 1100–1123, Aug. 2011.

[15] D. Scaramuzza, A. Martinelli, and R. R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proc. of IEEE Intl. Conf. of Vision Systems*, New York, NY, Jan. 2006.

[16] J. Shi and C. Tomasi, "Good features to track," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, June 1994, pp. 593–600.

[17] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the Intl. Joint Conf. on Artificial Intelligence*, Vancouver, Canada, Aug. 1981, pp. 24–28.

[18] D. Nister, "An efficient solution to the five-point relative pose problem," in *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 2, Madison, WI, June 2003, pp. 195–202.

[19] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Proc. of the 11th European Conference on Computer Vision*, Heraklion, Crete, Greece, Sept. 2010, pp. 778–792.

[20] R. V. D. Merwe, E. A. Wan, and S. I. Julier, "Sigma-point Kalman filters for nonlinear estimation: Applications to integrated navigation," in *Proc. of AIAA Guidance, Navigation, and Controls Conf.*, Providence, RI, Aug. 2004.

[21] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro UAV testbed," *IEEE Robot. Autom. Mag.*, vol. 17, no. 3, pp. 56–65, Sept. 2010.

[22] "Ascending Technologies, GmbH," http://www.asctec.de/.

[23] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011, pp. 2992–2997.

[24] "Robot Operating System," http://http://www.ros.org/.

[25] "The OpenCV library," http://http://opencv.willowgarage.com/.

[26] "Motion capture systems from Vicon," http://www.vicon.com/.