# Obtaining Liftoff Indoors

## *Autonomous Navigation in Confined Indoor Environments*

By Shaojie Shen, Nathan Michael, and Vijay Kumar



In this article, we consider the problem of autonomous navigation with a microaerial vehicle (MAV) in three-dimensional (3-D) confined indoor environments with multiple floors. We present experimental results with ground truth comparisons and performance analysis. We also highlight field experiments in multiple environments.

### Autonomous Navigation in Confined Indoor Environments with a Quadrotor

We are interested in the problem of surveilling and exploring environments that include both indoor and outdoor settings. Aerial vehicles offer mobility and perspective advantages over ground platforms, and MAVs are particularly applicable to confined buildings with multiple floors, where stairwells or clutter prohibit the motion of ground vehicles. A challenge when operating in indoor environments is the lack of an external source of localization, such as a global positioning system. Therefore, we focus on autonomous navigation in buildings with multiple floors without an external source of localization or prior knowledge of the environment. To ensure that the robot is fully autonomous, we require all computation to occur on the robot without relying on external infrastructure, communication, or human interaction beyond high-level commands. Therefore, we pursue a system design and methodology capable of autonomous navigation with real-time performance on a mobile processor using only onboard sensors (Figure 1). In this article, the concept of autonomous navigation includes robust pose estimation, localization, multifloor 3-D mapping with loop closure, planning, and control.

The actuation of quadrotors introduces another challenge in confined indoor environments. Quadrotors generate lift via rotating propellers. However, this lift varies depending on the airflow over the propellers, which in turn is a function of the nominal airflow in the environment. In confined environments, the induced airflow results in changes in the lift

*Digital Object Identifier 10.1109/MRA.2013.2253172*
*Date of publication: 6 December 2013*



1070-9932/13/$31.00©2013IEEE

and, consequentially, in the motion of the vehicle [2]. Thus, when the robot operates in confined environments, we must be able to estimate not only the vehicle state but also the aerodynamic state that affects the dynamic model of the robot.

This article consists of three parts: 1) a system overview that details our approach and places it into context with existing methodologies, 2) extensions necessary to permit operation onboard the robot in confined multifloor environments and to compensate for external aerodynamic effects, and 3) experimental results that characterize system performance and accuracy and demonstrate application in confined multifloor environments.

We note that the topic of autonomous navigation with an MAV has been addressed by others in the community with some similarities in approach and methodology and with results leaning toward online autonomous navigation and exploration with an aerial vehicle. A survey by Kendoul [3] provides a thorough review of ongoing research in the domain of autonomous navigation approaches for rotorcraft platforms. Particularly relevant to our article is the work of Bachrach et al. [4] and Grzonka et al. [5], where laser scanners serve as the primary source of sensory information. Similarly, Dryanovski et al. [6] presented an open-source implementation of a laser-based localization system for an MAV. Vision-based approaches that rely on monocular, stereo, and depth cameras as the primary information source for state estimation are also proposed in the literature [7]–[9]. A challenge with vision-based approaches is that they require significant computational power to process imagery in real time and thus often require external processing options in addition to the computation on the vehicle. Bills et al. [10] proposed a mapless navigation algorithm for use with an MAV that circumvents this challenge but at the cost of not building a global representation of the environment.

The major points of differentiation between existing results and our work are twofold. First, all of the processing is onboard, requiring algorithms that lend themselves to real-time computation on a small processor. Second, we design adaptive controllers to compensate for external aerodynamic effects that would otherwise prohibit operation in constrained environments. By developing a system capable of autonomous navigation without the need for external infrastructure, we ensure that the system is able to robustly operate in challenging multifloor environments where reliance on external processing or sustained communication with the vehicle is infeasible. We present results from field experiments in multiple environments and a live demonstration that requires the system to operate without failure over an extended period of time.

## Methodology and Related Literature

To achieve our goal of real-time autonomous navigation in confined multifloor indoor environments using an aerial vehicle with pragmatic constraints on available computational resources (speed and memory), we must address the problems of mapping, localization, planning, and control, given these system requirements. Each of these topics covers a breadth of literature and, as such, we focus here only on research that directly impacts our system design and methodology. We evaluated many strategies in the development of the system and will motivate algorithm selection based on this evaluation, but we will restrict any quantitative analysis to only those methods used in the experiments (see the section "Experimental Results"). The discussion follows the logical flow of the system design (Figure 2).

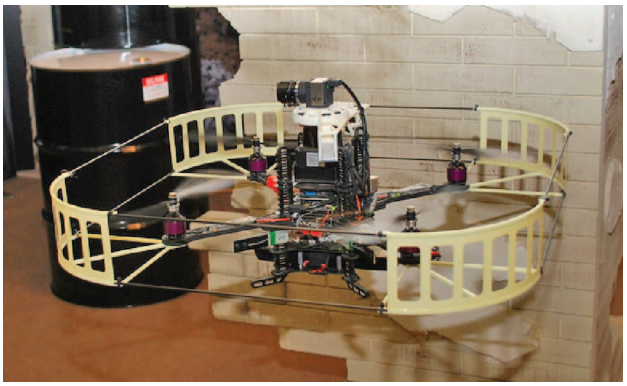

**Figure 1.** The experimental platform with onboard computation (1.6-GHz atom processor) and sensing [laser, camera, and inertial measurement unit (IMU)].

**We require all computation to occur on the robot without relying on external infrastructure, communication, or human interaction.**

### *Notation*

The world frame, $\mathcal{W}$, is defined by axes $x_W$, $y_W$, and $z_W$, with $z_W$ pointing upward. The body frame, $\mathcal{B}$, is attached to the center of mass of the quadrotor with $x_B$ coinciding with the preferred forward direction and $z_B$ perpendicular to the plane of the rotors pointing vertically up during perfect hover. We use ZYX Euler angles to model the rotation of the quadrotor in the world frame. To get from $\mathcal{W}$ to $\mathcal{B}$, we first rotate about the $z_W$ axis by the yaw angle, $\psi$, then rotate about the intermediate $y$ axis by the pitch angle, $\theta$, and finally rotate about the $x_B$ axis by the roll angle, $\phi$, The rotation matrix that transforms coordinates from $\mathcal{B}$ to $\mathcal{W}$ is given by

$$R = R_\psi \, R_\theta \, R_\phi,$$

where $R_\psi$, $R_\theta$, and $R_\psi$ are elementary rotation matrices with respect to the $x$, $y$, and $z$ axes. The six degrees-of-freedom (6-DoF) pose of the center of mass in the world frame is defined by $\{x, y, z, \phi, \theta, \psi\}$.

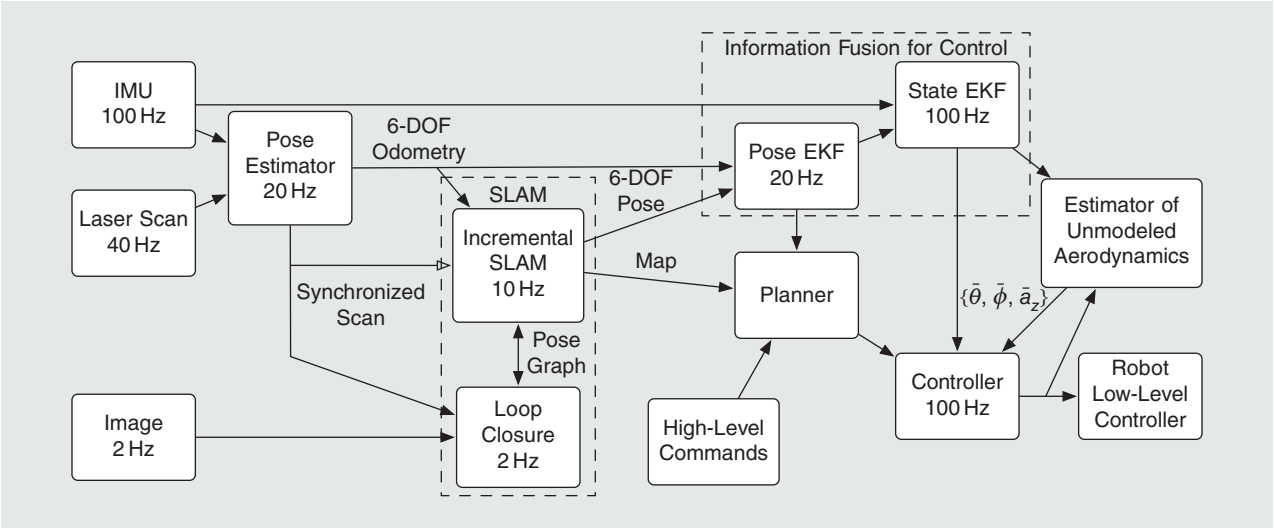


**Figure 2.** An architecture diagram showing the important software modules with update rates.

### *Pose Estimation*

A horizontally mounted scanning laser range sensor retrofitted with mirrors for beam redirection to the floor and ceiling serves as a primary source of information for position and yaw estimation. We evaluated several laser-based methods for pose estimation, such as exhaustive search [11] and feature-based approaches [12]. However, as the vehicle dynamics require pose estimates with update rates of 20 Hz and our onboard computational resources are limited, we chose the iterative closest point (ICP) algorithm, which yields a robust and inexpensive continuous pose estimate. We make use of a grid-based search [13] to speed up the computationally expensive closest point search in ICP. The result of the ICP algorithm is an estimate of $\{x, y, \psi\}$. The algorithm implementation is able to run at 20 Hz and requires ~20% of the total CPU time of the onboard processor.

**MAVs are particularly applicable to confined buildings with multiple floors.**

A Kalman filter (KF), similar to [5], fuses internal measurement unit (IMU) data with redirected laser scans to provide altitude estimates (see the section "Altitude Measurement"). The remaining state variables $\{\phi, \theta\}$ are estimated using the onboard IMU.

### *Simultaneous Localization and Mapping*

We address the problems of mapping and drift compensation via an incremental simultaneous localization and mapping (SLAM) algorithm. Both the particle filter-based occupancy grid [14] and feature-based [15] methods perform well in practice. However, after evaluating these approaches, we found the computational requirements to be unacceptably high for the onboard processor. Therefore, we pursue a simplified occupancy grid-based incremental SLAM algorithm.

Given the incremental motion of the robot provided by ICP-based scan matching and the IMU, we correct the error in $\{x, y, \psi\}$ by aligning incoming laser scans against the existing map using a windowed exhaustive grid search. The cost map generated from the obstacles in the existing map is approximated by an image distance transform. If a stable floor transition is detected by the pose estimator, we create a new layer in a multilayered occupancy grid. The incremental SLAM algorithm runs at 10 Hz and consumes <30% of the total CPU time.

Unlike particle filter-based approaches, our incremental SLAM algorithm does not embed loop closure functionality. To correct the global inconsistency caused by incremental SLAM, we employ vision-based techniques to enable robust loop closure detection that does not depend on the actual pose estimation error [16]. A fixed-size visual vocabulary is constructed offline by clustering a large number of speeded up robust features (SURF) features collected from different environments [17]. The detected SURF features of each incoming image are converted into the vocabulary format and matched against previous images via histogram voting. Any matched loop closure candidates are further verified by scan matching.

Loop closures add constraints to a pose graph, where each node in the graph is a sparse sample of the robot poses and their associated sensor data. Akin to the methods of [18], we apply an optimization based on an iterative extended KF (IEKF) to create a globally consistent pose graph configuration. The optimization occurs in the full 6-DOF pose space with the assumption that closure only happens at the same floor level. From this optimization, we obtain a globally consistent multifloor map and pose graph. Approximations that further speed up the optimization are detailed in the section "Incremental, Approximate Loop Closure."

### Information Fusion for Control

To ensure high-rate, accurate, and drift-free pose estimates for feedback control, we use two separate extended KFs (EKFs) to fuse and boost the pose estimate to 100 Hz. The first EKF combines the 20-Hz pose estimate and the 10-Hz SLAM pose correction. The information from the pose estimator and SLAM is not independent, but the major role of this EKF is to provide smooth compensation for possible delay in the SLAM estimate introduced by large-scale loop closure and map corrections. We additionally employ a delayed measurement update scheme to match the timing of the different sources of information driving the pose estimate.

The second EKF combines the 20-Hz pose estimate from the first EKF and the 100-Hz IMU data to provide 100-Hz pose and linear velocity estimates in the world frame. The final estimation output has an average delay of 0.01 s and feeds directly into the feedback control loop of the robot for position and velocity control.

### Environment Representation

Given the estimated global pose of the robot, we can transform laser scans accordingly and produce a dense 3-D environment representation for planning and obstacle avoidance. However, the amount of onboard memory is limited in mobile processors like ours. As we are primarily interested in indoor environments, the majority of obstacles take the form of vertical walls. Therefore, we use a modified multivolume occupancy grid map similar to [19] to create a compact occupied space representation by merging contiguous occupied cells into common vertical regions. For general indoor environments, the resulting map typically has a memory cost on the same order as a two-dimensional (2-D) occupancy grid map.

### Planning and Control

Since the focus of this article is primarily the mapping and localization required for autonomous navigation, we note here only a general overview of our approach. Given the current pose estimate of the robot and a map of the environment, we employ incremental sampling-based planning (RRT*) [20], where the robot is treated as a point-model kinematic system with orientation (about $z_W$). We employ proportional-derivative feedback control laws to transform these kinematic inputs to appropriate dynamic control inputs [21]. Tuning of control gains is accomplished via optimal control methods (LQR) based on a linearized system model.

## Extensions

In the previous section, we described the system design and discussed algorithm selection based on performance requirements and processor/sensor constraints. We now detail extensions to these algorithms that permit real-time operation across multiple floors. Many of these extensions are based on approximate solutions that are necessary to enable real-time implementation. Additionally, we discuss the online estimation and compensation of aerodynamic effects due to wind or propeller backwash in small corridors or near walls.

### Environment Assumptions

As the domain of interest is indoor environments and periphery, we assume 2.5-dimensional (2.5-D) environment models formed by collections of vertical walls and horizontal ground planes, all assumed to be piecewise constant. Let $[x_s, y_s, z_s]^T$ be the laser scan endpoints in the body frame. We can project the laser scans to a horizontal plane by

$$[x_g, y_g, z_g]^T = R_\theta R_\phi [x_s, y_s, z_s]^T.$$

We eliminate the scans that hit the floor or ceiling, which are generally not useful for scan matching, by comparing $z_g$ with the redirected laser scans pointing upward and downward.

Although this approach largely simplifies the challenges of full 3-D scan matching using only 2-D laser range finders, the 2.5-D environment assumption is violated in cluttered and outdoor environments with natural structure. In our experiments, we see that partial structure in the environment satisfies this assumption, and the overall performance is acceptable.

**MAVs offer a unique capability for mapping and exploration in complex environments.**

### Altitude Measurement

We measure the variation in altitude with scans pointing to the floor and use this value to approximate the variance used in the measurement update of the KF. If the variation is too high, generally due to uneven surfaces or during a floor level change, we discard the current laser measurement and defer the measurement update until a stable measurement is obtained. An additional mirror deflects scans vertically upward toward the ceiling. These measurements provide additional robustness should there be no stable downward measurements available. When no upward or downward laser measurements are available, we use an onboard pressure sensor for short-term measurement updates and force the robot to lower its height. Corrections of accumulated errors caused by frequent floor transitions are resolved through loop closure.

### Incremental, Approximate Loop Closure

Loop closure for large loops is typically computationally expensive with significant memory requirements. Although several methods exist for reducing the computational burden [18], [22], we found that these methods are too demanding for a 1.6-GHz processor. With the understanding that we are more concerned with global consistency than absolute accuracy, we propose a mild extension that results in an incremental but approximate approach.

As shown in Figure 3, the general idea of this approach is as follows: A robot travels from $P_1$ to $P_1$. Loop closure is detected between $P_1$ and $P_5$; we contract this loop to $P_{1:5}$. We

do not discard nodes in the pose graph as they are required for closure detection. However, we do ignore future iterations on this loop. Therefore, if the robot travels from $P_5$ to $P_9$, where a loop closure is detected between $P_3$ and $P_9$, we do not correct any pose graph configuration inside the loop $P_{1:5}$. In this way, we only close the "new" loops.

Although the proposed method maintains a globally consistent map, it does not improve any loop structure that is already corrected and is therefore approximate. Closing inner loops, ignoring this approximation, yields solutions consistent with traditional approaches. This will lead to locally distorted maps (i.e., the walls bend slightly), but the overall loop structure is correct. In practice, however, we find that a slightly distorted map is still sufficient to localize the robot even in environments with multiple loops.

### Estimating and Compensating for Unmodeled Aerodynamics

Disturbances due to propeller backwash and ambient air flow in indoor environments introduce uncertainty in the system actuation. We estimate and compensate for these disturbances via a KF with an augmented state representation that includes external forces applied to the vehicle body and uncertain plant parameters. The resulting disturbance estimates are introduced as compensation terms in the feedback control. The details of this approach are provided in [1].

## Experimental Results

### Experiment Design and Implementation Details

Four experiments are presented: 1) a study of the estimator performance compared to ground truth data, 2) navigation in confined multifloor indoor environments, 3) large-scale mapping in a three-story building, and 4) an extended live public demonstration. The motivation for each study is stated in the respective discussions.

The robot platform is sold by Ascending Technologies, GmbH (http://www.asctec.de) and is equipped with an IMU (accelerometer, gyroscope), magnetometer, and pressure sensor. We developed custom firmware to run at the embedded level to address feedback control and estimation requirements. The other onboard computation unit is a 1.6-GHz Atom processor with 1 GB of RAM. The sensors on the robot include a Hokuyo UTM-30LX (laser) and a uEye 1220SE with a Kowa 3.5-mm f1.4 lens (camera). A custom 3-D printed mount is attached to the laser. This mount houses mirrors that redirect some of the laser beams upward and downward (along $z_B$). The mass of the vehicle is 1.0 kg with a sensor and processor payload of 0.75 kg. Communication with the robot for monitoring experiment progress is via 802.11n networking. All algorithm development is in C++ using ROS (http://www.ros.org) as the interfacing robotics middleware. The flight time of our platform is ~10 min, but this value varies greatly based on the environment. We limit the maximum speed of the vehicle in the experiments to 0.5 m/s.

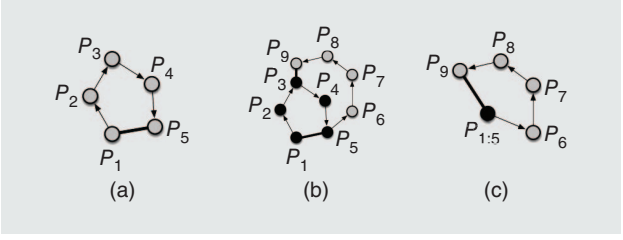


**Figure 3.** The loop closure optimization of the section "Incremental, Approximate Loop Closure." (a) A robot transitions through pose graph nodes $P_1$–$P_5$ with loop closure detected between $P_1$ and $P_5$. (b) After continuing, loop closure occurs again between $P_3$ and $P_9$, (c) but any future optimization updates view the loop as previously closed and only consider $P_{1:5}$–$P_9$.

### Evaluating Estimator Performance

We wish to study the performance of the onboard estimator as compared to ground truth, where ground truth is defined by a submillimeter accurate Vicon motion tracking system (http://www.vicon.com). Two studies are presented. The first study considers the accuracy of the onboard estimate compared with the Vicon estimate while the robot flies along a specified trajectory (Figure 4). In this case, the Vicon and onboard estimate compare well, with a standard deviation of $\{\sigma_x, \sigma_y, \sigma_z\} = \{2.47\text{ cm}, 3.23\text{ cm}, 0.70\text{ cm}\}$ and $\sigma_\psi = 0.55°$. The CPU usage required for the autonomous flight is shown in Figure 5.

The second study compares feedback control to maintain a single position using the pose and velocity feedback from the onboard estimator (Figure 6). The standard deviations of the hover performance are $\{\sigma_x, \sigma_y, \sigma_z\} = \{4.42\text{ cm}, 2.79\text{ cm}, 1.76\text{ cm}\}$.

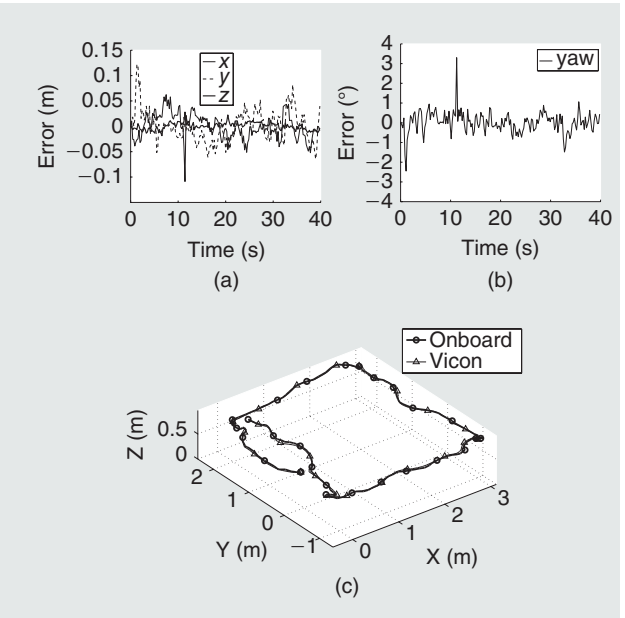


**Figure 4.** The (a) position and (b) yaw error between estimates from Vicon and the onboard estimator while (c) the robot controls along a specified trajectory.

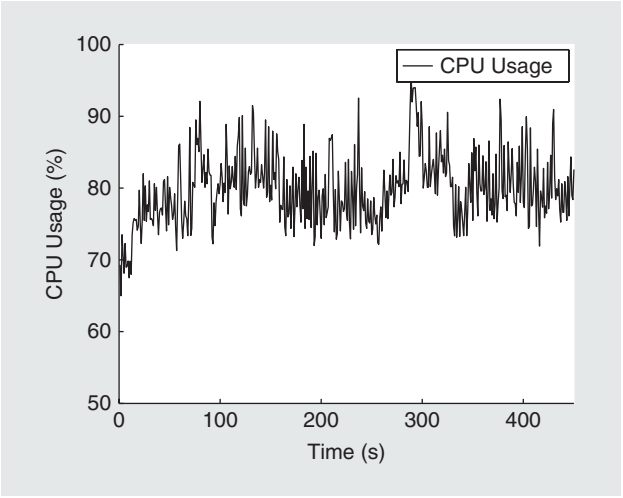


**Figure 5.** The percentage of CPU time used by the system (including host OS) while following a trajectory similar to that in Figure 4(c).

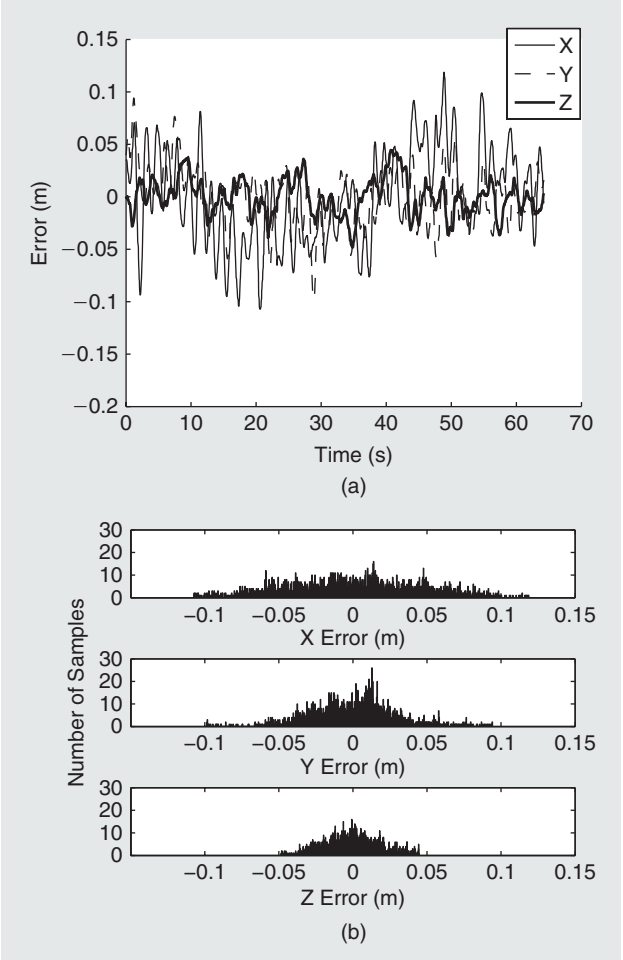


**Figure 6.** The robot is commanded to hover based on feedback from the onboard estimator. The (a) resulting regulation error from the closed-loop system and (b) its distribution are shown.

### *Navigation in Confined Multifloor Indoor Environments*

We now consider autonomous navigation in confined indoor environments with single or multiple floors. To verify map consistency, environments with small loops are considered. While the quadrotor is autonomous, the planned trajectory is determined by a user specifying goal locations with respect to the map.

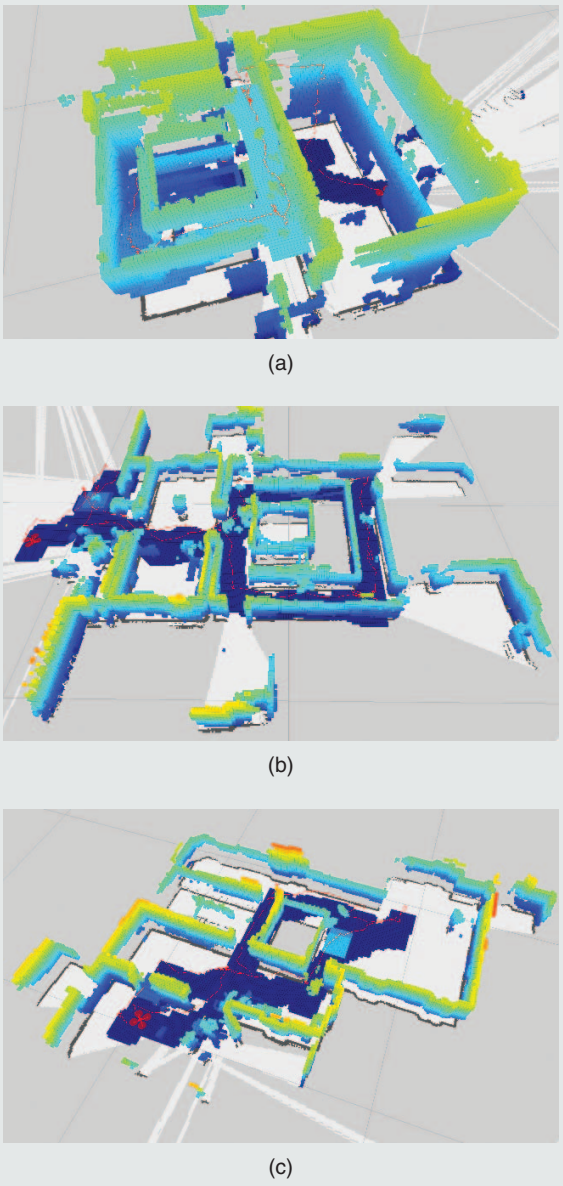


**Figure 7.** (a) Maps generated during autonomous navigation across multiple floors and (b) and (c) through confined environments with small loops. The vehicle and its trajectory are depicted as a red mesh and line, respectively.

Figure 7(a) shows the map generated by the robot navigating through an indoor two-floor environment. The robot starts on the first floor and transitions to the second floor via a stairwell [Figure 8(a)]. The vehicle returns to the starting location on the first floor by flying through a window into an open lobby [Figure 8(b)]. The vehicle side clearance when passing through the window is ~5 cm. Although we do not have ground truth for this environment, we observe that there is minimal error or drift in the map. The robot is

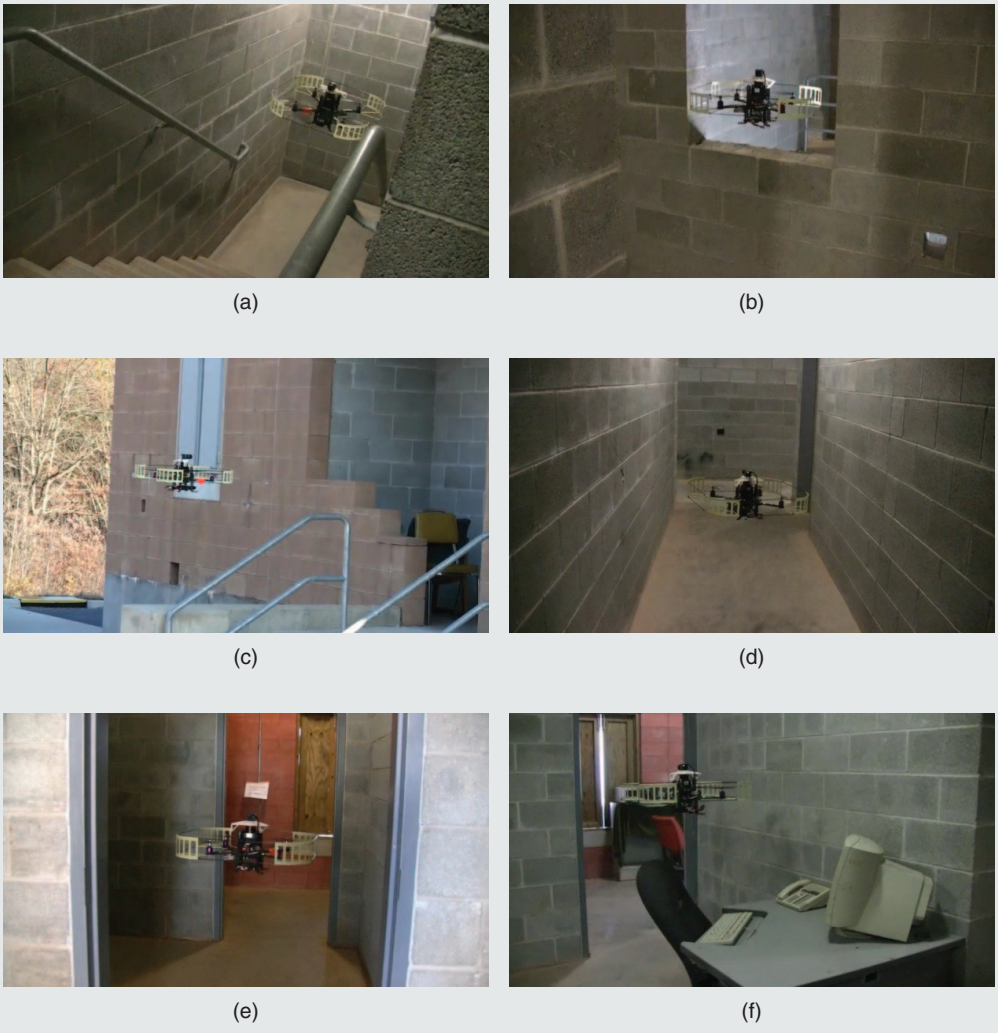


**Figure 8.** The images of the vehicle autonomously navigating (a) and (b) between multiple floors and (c)–(f) in confined indoor environments. A video of the experiments is available at http://mrsl.grasp.upenn.edu/shaojie/ram2012.m4v.

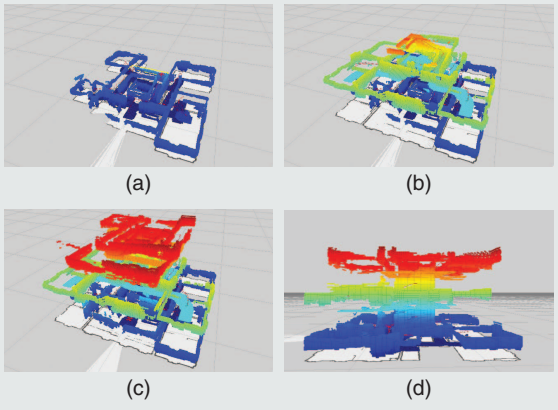


**Figure 9.** The mapping process (a)–(c) across three floors of an indoor environment and (d) a side view of the three-story building.

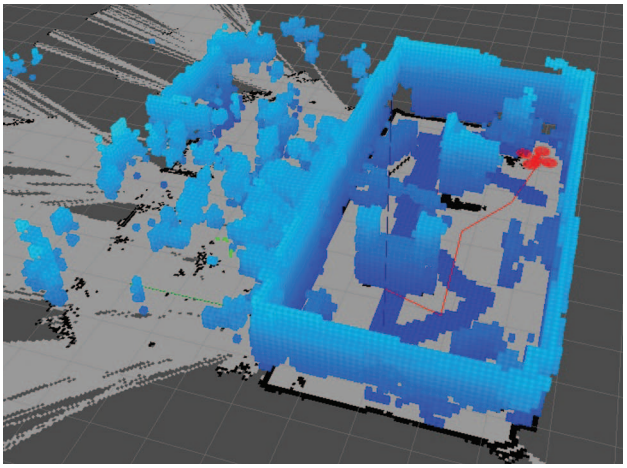

**Figure 10.** The 3-D map of the demonstration site generated during an autonomous flight.

commanded to return and hover at its starting location. The error was observed to be on the order of a couple of centimeters.

To evaluate the performance of loop closure, we consider the two buildings in Figure 7(b) and (c). In both environments, the vehicle starts outside and navigates through doorways, hallways, and windows to generate 3-D maps. During the experiments, the vehicle experienced nontrivial external disturbances due to wind gusts through open doors and windows. We see in Figure 8(e)–(f) that the vehicle successfully navigates through multiple doorways and around furniture and other environmental obstacles. In both cases, loops in the final maps are successfully recovered and empirically appear to be consistent. The trajectory length and duration for each experiment is noted in Figure 7.

### *Large-Scale Mapping Across Multiple Floors*

We now consider the performance of the onboard SLAM and loop-closure algorithms when mapping across several floors. To pursue a large-scale experiment with a length that exceeds feasible flight time, we carry the vehicle such that it emulates flight. Note that the map is consistent throughout the experiment, and the robot is able to close the loop when it returns to its starting location after mapping a three-story building. Maps at multiple stages of the experiment are shown in Figure 9. It is worth noting that, even when mapping large-scale environments, the onboard CPU usage remained on par with that shown for the trajectory-following case in Figure 5.

### *Live Public Demonstration*

As a final discussion on the performance of the autonomous navigation system, we briefly report results from a live demonstration held at the 27th Army Science Conference in Orlando, Florida, 29 November–2 December 2010. We report these results to highlight the repeatable performance of the system.

The demonstration considered autonomous navigation through a mazelike indoor environment following the size and layout shown in Figure 10. During each demonstration (trial), the vehicle began with no prior information and concurrently built a 3-D map while navigating through the mazelike environment. Dynamic obstacles (people) were introduced into the environment with the intention of disrupting the planned trajectories of the vehicle. The goal of the demonstration was to show that the autonomous system was able to effectively navigate the environment while avoiding collisions with static and dynamic obstacles (Figure 11). The trajectory of the vehicle was defined by waypoints provided by an external operator.

Over 50 trials were conducted during the live demonstration without any system failures. The vehicle successfully navigated the environment without issue and avoided all obstacles (static and dynamic).

## Conclusions and Future Work

In this article, we considered the problem of autonomous navigation with an MAV in indoor environments. In particular, we discussed autonomous navigation in confined buildings with multiple floors. To ensure that the robot is fully autonomous, we require all computation to occur on the robot without the

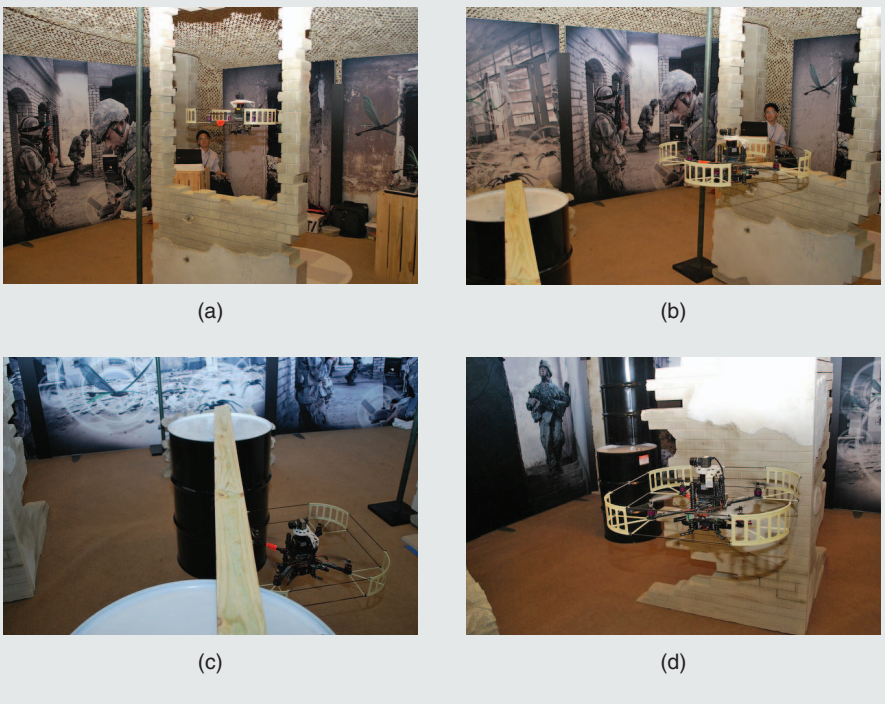


**Figure 11.** The demonstration environment with (a) and (b) windows, (c) various types of clutter, and (d) walls.

use of external infrastructure, communication, or human interaction beyond high-level commands. We presented a system design and methodology that enables autonomous navigation with real-time performance on a mobile processor using only onboard sensors. We reviewed experimental results with performance analysis and addressed the adaptation to changing aerodynamic conditions induced by wind disturbances or changes in airflow in constrained environments.

In a recent article, we considered algorithmic approaches to autonomous exploration with an MAV [23]. As a follow-up, we expanded our research to consider operation in indoor and outdoor environments [24]. We also tested the system in a multistory earthquake-damaged building [25]. We believe that MAVs offer a unique capability for mapping and exploration in complex environments for applications such as search-and-rescue operations and building inspection. However, several challenges must still be addressed before such systems are capable of handling the challenging environments of these application domains. A key limitation of our approach is the 2.5-D environment assumption. We are actively considering strategies for addressing this assumption, given our vehicle's computational constraints, using vision-based methods. Preliminary results in this direction appear in a forthcoming publication [26].

## References


[1] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proc. IEEE Int. Conf. Robotics Automation*, China, May 2011, pp. 20–25.

[2] M. Bangura and R. Mahony, "Nonlinear dynamic modeling for high performance control of a quadrotor," in *Proc. Australasian Conf. Robotics Automation*, Wellington, NZ, Dec. 2012.

[3] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *J. Field Robot.*, vol. 29, no. 2, pp. 315–378, 2012.

[4] A. Bachrach, S. Prentice, R. He, and N. Roy, "RANGE—Robust autonomous navigation in GPS-denied environments," *J. Field Robot.*, vol. 28, no. 5, pp. 644–666, 2011.

[5] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 90–100, Feb. 2012.

[6] I. Dryanovski, W. Morris, and J. Xiao, "An open-source pose estimation system for micro-air vehicles," in *Proc. IEEE Int. Conf. Robotics Automation,* Shanghai, China, May 2011, pp. 4449–4454.

[7] M. W. Achtelik, M. C. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments," in *Proc. IEEE Int. Conf. Robotics Automation*, China, May 2011, pp. 3056–3063.

[8] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," in *Proc. IEEE Int. Conf. Robotics Automation*, China, May 2011, pp. 2992–2997.

[9] A. 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. Int. Symp. Robotics Research*, Flagstaff, AZ, Aug. 2011.

[10] C. Bills, J. Chen, and A. Saxena, "Autonomous MAV flight in indoor environments using single image perspective cues," in *Proc. IEEE Int. Conf. Robotics Automation,* China, May 2011, pp. 5776–5783.

[11] E. B. Olson, "Robust and efficient robotic mapping," Ph.D. dissertation, Dept. Elect. Eng., Massachusetts Inst. Technol., Cambridge, MA, June 2008.

[12] G. D. Tipaldi and K. O. Arras, "FLIRT—Interest regions for 2D range data," in *Proc. IEEE Int. Conf. Robotics Automation*, Anchorage, AK, May 2010, pp. 3616–3622.

[13] D. Chetverikov, "Fast neighborhood search in planar point sets," *Pattern Recognit. Lett.*, vol. 12, no. 7, pp. 409–412, July 1991.

[14] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE Int. Conf. Robotics Automation,* Barcelona, Spain, Apr. 2005, pp. 2432–2437.

[15] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.

[16] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing," in *Proc. IEEE Int. Conf. Robotics Automation*, Rome, Italy, Apr. 2007, pp. 2042–2048.

[17] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. European Conf. Computer Vision*, Graz, Austria, May 2006.

[18] C. Estrada, J. Neira, and J. D. Tardos, "Hierarchical SLAM: Real-time accurate mapping of large environments," *IEEE Trans. Robot.,* vol. 21, no. 4, pp. 588–596, Aug. 2005.

[19] I. Dryanovski, W. Morris, and J. Xiao, "Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots System,* Taipei, Taiwan, Oct. 2010, pp. 1553–1559.

[20] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *Proc. Robotics: Science System,* Zaragoza, Spain, June 2010.

[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] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *IEEE Trans. Intell. Transp. Syst.,* vol. 10, no. 3, pp. 428–439, Sept. 2009.

[23] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3D exploration with a micro-aerial vehicle," in *Proc. IEEE Int. Conf. Robotics Automation,* Saint Paul, MN, May 2012, pp. 9–15.

[24] S. Shen and N. Michael, "State estimation for indoor and outdoor operation with a micro-aerial vehicle," in *Proc. Int. Symp. Exp. Robot.,* Quebec, QC, Canada, June 2012.

[25] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," in *Proc. Int. Conf. Field Service Robotics*, Matsushima, Japan, July 2012.

[26] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation for autonomous rotorcraft MAVs in complex environments," in *Proc. IEEE Int. Conf. Robotics Automation,* Karlsruhe, Germany, May 2013.



***Shaojie Shen***, GRASP Laboratory, University of Pennsylvania, Philadelphia. E-mail: shaojie@grasp.upenn.edu.

***Nathan Michael***, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania. E-mail: nmichael@cmu.edu.

***Vijay Kumar***, GRASP Laboratory, University of Pennsylvania, Philadelphia. E-mail: kumar@grasp.upenn.edu.