3D Mapping and Navigation for Autonomous Quadrotor Aircraft

Sajad Saeedi, Amr Nagaty, Carl Thibault, Michael Trentini, and Howard Li

***Abstract*—Autonomous navigation is a challenging problem in GPS-denied environments. For small flying robots and quadrotors, inherent limitations such as limited sensor payload and small system time-constant add another layer of challenge to the problem. This paper presents a solution for autonomous navigation of a quadrotor by performing autonomous behaviors, such as exploration, returning home, or following waypoints. The solution relies on accurate localization, mapping, and navigation between waypoints.**

A 3D mapping algorithm is proposed that enables the fast moving quadrotor to make a reliable 3D model. One of the contributions of the paper is the 3D mapping algorithm which is built upon a 2D mapping algorithm and IMU information. Compared with state-of-the-art algorithms, the proposed 3D mapping allows the quadrotor to generate more accurate models with low-bandwidth channels. Additionally, the algorithm allows the quadrotor to have aggressive changes in rotation and translation that occur frequently due to the fast dynamics of the quadrotor. Also 2D and 3D maps are displayed on multiple remote devices, such as tablets. Multiple tests were performed in simulated and real-world environments to show the effectiveness of the proposed solution.

I. Introduction

This paper studies the possibility of using state-of-the-art sensing technologies to perform perception in GPS-denied environments with a quadrotor rotorcraft and achieve autonomy with minimum human intervention. The paper explains the requirements of 3D mapping for an autonomous quadrotor aircraft. The paper is an extension to the previous work [1] where the autonomy for the quadrotor was achieved using a custom-designed sensor suite including a laser ranger and an IMU. In the extended paper, new features such as 3D mapping, target localization, and remote mapping have been introduced. Due to fast dynamics of quadrotors and also possible loss of image frames, SLAM based on visual odometry alone often fails to generate a consistent map. Fig. 1-(top) shows the map of a classroom built by a quadrotor using only visual odometry and bundle adjustment [2]. Fig. 1-(bottom) shows the consistent map of the same classroom using the proposed algorithm. The visual odometry is initialized with the laser odometry as proposed in this paper.

The contributions of this work is an integrated autonomy solution, which includes 1) behaviour-based mission planning, 2) 3D SLAM by a novel model-based mapping initialized using reliable 2D SLAM, 4) modified path planning and exploration, and 5) remote map displaying on multiple tablets. Our 3D mapping algorithm is based on accurate 2D maps and inertial information; therefore, the generated 3D models are more accurate and reliable. Also the algorithm allows the quadrotor to build 3D maps while the quadrotor is moving fast which is an important requirement in real-world applications. The combination of 2D and 3D maping algorithms enables the quadrotor to perform mapping with low-bandwidth channels.

The rest of the work is organized as follows: Section II presents background information to quadrotor autonomous navigation. Section III introduces the proposed solution for perception and navigation of an autonomous quadrotor. Section IV presents experimental results in simulated and real-world environments. Finally, Section V summarizes the work.

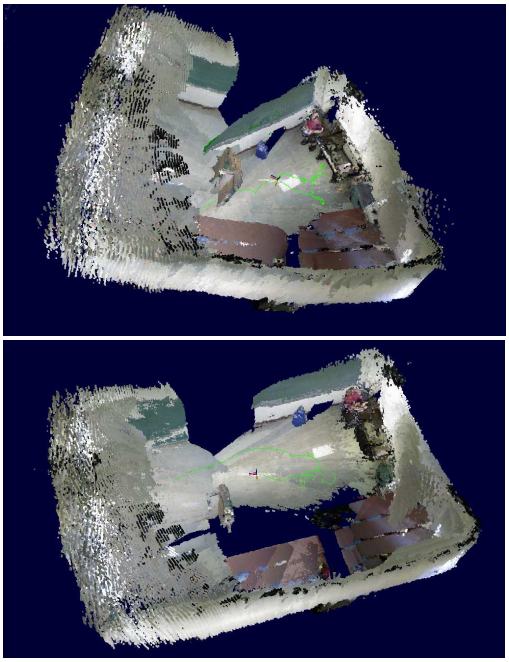

Figure 1. Due to fast dynamics of quadrotors and also loss of image frames, SLAM based on visual odometry alone often fails to generate a consistent map. (top) Map of a classroom built by a quadrotor using only visual odometry. The poses are optimized using a bundle adjustment algorithm [2]. (bottom) Consistent map of the same classroom. The visual odometry is initialized with the laser odometry as proposed in the this paper.

II. Background and Literature Review

This section presents the background for rotorcraft perception including control, state estimation, path planning, localization, mapping, and high level mission planning.

A. Control and State Estimation

Compared to other types of rotorcraft, quadrotors are mechanically simpler and easier to control. However, controlling a quadrotor is still a challenging problem due to its system nonlinearities, cross couplings of the gyroscopic moments and underactuation [3], [4]. For more information on controlling a quadrotor, see [5] and [6].

Any control algorithm needs to have access to the real state of the system. Typically, the state of a quadrotor includes its orientation, position, and velocities. In 2006, S. Thrun et al. [7] were among the first researchers who did real-time state estimation using a flying robot. They performed state estimation with a helicopter in an outdoor environment. In 2009, Grzonka et al. [8] and Bachrach et al. [9] performed state estimation for a quadrotor, independently. Since then, researchers have investigated different configurations for state estimation, but one thing which is common in most of the solutions is the use of Kalman filtering.

B. Map Learning

Deploying an autonomous robot in a real-world scenario requires learning maps. To learn maps, a number of key problems should be addressed, including a) mapping, b) localization, and c) path planning. The combination of mapping, localization, and path planning is often referred to as simultaneous planning, localization, and mapping (SPLAM); integrated autonomy solutions; or autonomy packages [10]. Complete information on different methods for components of map learning in relation to mapping and localization can be found in [11], and for path planning in [12].

Various classical path planning algorithms were modified and applied to UAVs [13], [14] for different purposes such as coverage [15], [16] and mapping and localization [9]. SLAM and path planning for indoor quadrotors were first performed in 2009 by Grzonka et al. [8] and Bachrach et al. [9], [17], independently. Later, other researchers continued the same trend to improve on the previous results. In 2010, Dryanovski et al. [18] presented their approach for quadrotor SLAM. It was similar to the work proposed by Grzonka et al. [8]. In 2011, Kohlbrecher et al. [19] proposed their fast approach for quadrotor SLAM. The latter two solutions had no path planning involved. All these solutions are using scanning laser rangers as a key perception sensor to perform SLAM. Other researchers [20], [21], [22], [23], [24], [25] published similar results for SLAM or path planning with the quadrotor.

In the literature, there are various 3D mapping algorithms for UAVs [26], [27], [28] and other robots [29], [30], [31], [32]; however, most of them have limitations in real-world quadrotor applications. For instance, the Kinect Fusion algorithm [33] is a fast and reliable 3D mapping algorithm, but it operates in relatively small environments and needs very fast processing units. It is based on the iterative closest point matching and the main contribution of the paper is representing the map of the environment using signed distance function (SDF) [34] instead of the traditional occupancy grid map.

The 3D dense tracking and mapping algorithm [35] is also based on the SDF. This algorithm is very sensitive to the loss of data, and in the reported experiments, the captured depth and color images are transmitted to the ground control station using a wired link. This is a major limitation for quadrotors.

In the fast visual odometry and mapping [36], a 3D mapping algorithm is used to develop real-time 3D maps. The mapping algorithm uses extracted features from consecutive images. It develops a model of the world using the extracted features. Once a new frame is received, it is matched with the model. Compared with other mapping algorithms which are based on matching images frame-by-frame, the model-based mapping accumulates significantly less error; however, the change in the pose of the camera should be small; otherwise the mapping algorithm fails to produce consistent maps.

C. Mission Planning

Mission planning for an autonomous flying robot is defined as a set of actions which should be taken at different times. It is part of the navigation stack and like a high level decision maker, it tells the robot what to do.

A solution for this task of navigation is to define a set of navigational behaviors. Each behavior is a process or control law that achieves and/or maintains a goal [37], [38]. As an example, *Obstacle avoidance* behavior maintains the goal of preventing collisions, and *follow me* behavior achieves the goal of following the position of a person. Some behaviors are prerequisites for other behaviors. For instance, if a robot wants to perform the *return-home* behavior, knowledge of the current position of the robot is required. Therefore, *localization* behavior is required for following a person.

III. Proposed Perception and Autonomy Solution

An overview of the proposed autonomy system is shown in Fig. 3. The proposed solution is composed of two main modules: *Mission Planner* and *Autonomy*. Each module consists of several blocks. *Mission Planner* takes care of sequencing the autonomous behaviors. The *Autonomy* module accepts behaviors from *Mission Planner* and takes actions to achieve or maintain the goal of the behavior. Each module and its blocks are explained in this section.

The ideal sensor suite includes an IMU, a scanning laser rangefinder with a horizontal scan, a second scanning laser rangefinder with a vertical scan, an RGB-D camera, an altimeter, and a GPS. Sensors connected by a dashed line to the autonomy blocks are optional sensors. Fig. 2 shows the sensors of the quadrotor. The sensor suite includes a CH-Robotics IMU, an ASUS Xtion Pro RGB-D camera, a Hokuyo UTM-30LX scanning laser rangefinder with a horizontal scan, and a reflective mirror mounted on the laser ranger to reflect a few beams to the ground and measure the altitude of the quadrotor. An Atom processor is used to interface the sensors and communicate with the ground control station.

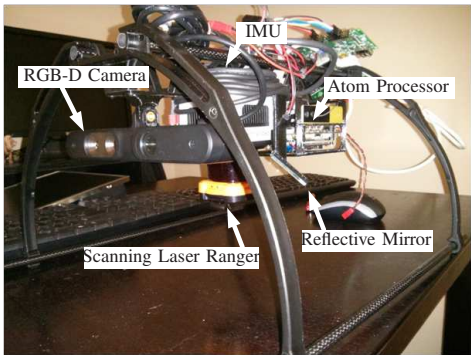

Figure 2. The sensor suite of the quadrotor includes an IMU (not visible in the figure), an RGB-D camera, a laser ranger, and a reflective mirror. An Atom processor is used to interface the sensors and communicate with the ground control station.

A. Autonomous Behaviors

Autonomous behaviors are a set of behaviors designed to navigate the robot efficiently. These behaviors rely on exploration, path planning, and obstacle avoidance behaviors. Behaviors such as *follow-me*, *go-home*, *move-to-goal* and *return-to-me* are based on the *path planning* behaviour between

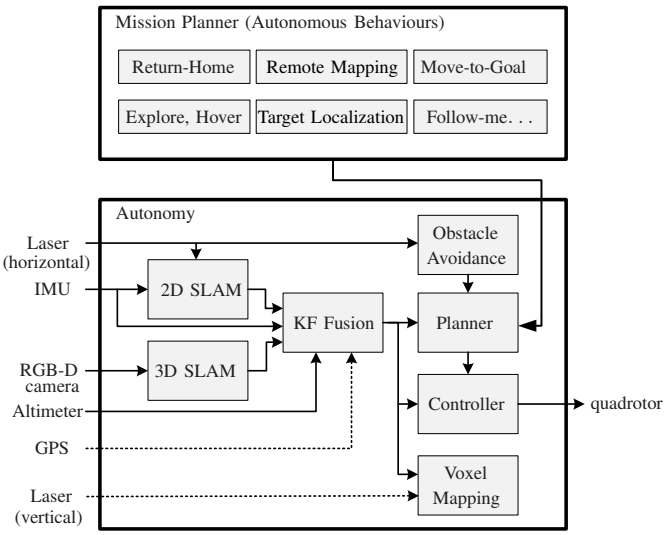

Figure 3. Proposed autonomous navigation in GPS-denied environments is composed of two main modules: *Mission Planner* and *Autonomy*. *Mission Planner* takes care of organizing and sequencing the autonomous behaviours. The *Autonomy* block accepts these behaviours and takes proper actions.

two given points and the *follow-path* behaviour. *Obstacle avoidance*, a behavior with the highest priority, is performed at two levels. First, it is performed at the map level where occupied cells are dilated and an optimal path is designed. Second, it is performed using the laser ranger and keeping a safe distance from instantaneous detected obstacles which are closer than a pre-specified threshold.

The *target localization* behavior is another behaviour that identifies specific targets, and marks them in the 3D or 2D map of the environment. A target localization algorithm is used to localize targets according to their colors [39]. The localized targets are marked in the developed map with an arrow. Two attached videos of two experiments demonstrate this capability. The algorithm can identify multiple targets of the same color or different colors, as defined in the configuration file. The targets are identified in the image frame, but since the depth information is also available, the 3D coordinates of the targets are easily calculated. Given that the pose of the camera is known, the global coordinates of the targets are calculated by transforming the coordinates from the camera frame to the global frame.

Remote mapping is another useful behavior for field applications. It is often required to monitor the perception of the quadrotor on small, portable, and remote stations such as tablets. For instance, if a quadrotor is exploring a building where humans cannot enter, having access to camera view of the quadrotor and also 2D and 3D maps on multiple tablets will help the mission planner or quadrotor users to manage the mission efficiently.

To show the maps on multiple remote tablets, two different approaches were taken and tested. The first one was developing a web-based mapping application. In the web-based application, map processing is performed on a server, and the results are displayed as web contents on a remote client computer. This approach requires webGL which works well on Windows and Android systems with fast CPUs; however, the application is not stable on Android tablets due to memory and processing limitations. For example, it works well on a Core2Duo laptop with Windows, but on a Galaxy Tab 3, of 50% of the tests, the browser fails.

The other approach taken to handle this problem is through web data transmission. Fig. 4 shows the diagram of this approach. The 3D mapping application runs on the ground station. A jpg-server captures jpeg images from the map visualizer. The captured images are streamed to the tablets by the motion-jpeg server which is based on the motion-jpeg standard. Each tablet specifies the type of the image that should be streamed, and the motion-jpeg server on the workstation subscribes to the requested image and sends it to the tablet. Camera view can also be transmitted to the tablets.

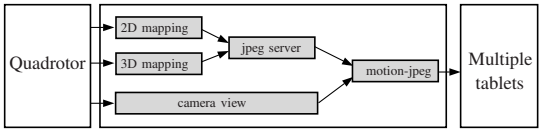

Figure 4. Remote mapping on multiple tablets.

Any complicated mission can be represented as a set of behaviors. A sample mission composed of the aforementioned behaviors is presented in Fig. 5. In this mission (mapping, localization, and path planning are not shown for clarity), first the robot explores the environment (*explore* behavior). Once exploration is complete, it moves to a given goal point (*move-to-goal* behavior). Once it reaches the destination, it returns to the start point where the mission was started (*go-home* behavior). Two behaviors, obstacle avoidance and hold (as an emergency stop), have a higher priority than others.

The transition between behaviours is determined by their priorities and also by their state of accomplishment. For behaviors such as *go-home*, *move-to-goal*, and *follow-path*, the accomplishment criteria is the distance of the robot to the goal. If the distance is less than a threshold, the behaviour is assumed to be done and the next behavior starts. A behavior can also be called while another behavior is running, depending on the priority of the behaviors. For example, if the robot is exploring an environment, but suddenly a dynamic object appears in the field of view, then the new behavior is *avoiding obstacle*. Once there is no risk of collision, the robot retrieves its previous behaviour, and the exploration behaviour continues until there is nothing left to explore.

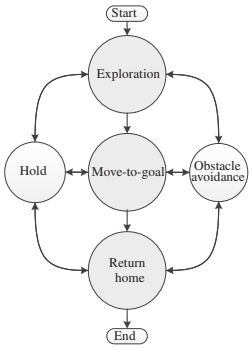

Figure 5. A sample mission composed of basic navigation behaviours. First, the robot explores an unknown environment, then it moves to a given goal. Once it arrives at the goal, it returns home. While performing these behaviours, obstacle avoidance and hold (as an emergency stop) are running at a higher priority than others.

B. 2D SLAM

The quadrotor can fly at a fixed altitude; therefore, a 2D grid map is generated and used for navigation. The horizontal scanning laser ranger is used to perform 2D SLAM and generate an occupancy grid map. The method used for 2D SLAM is adopted from [19] which is accurate and fast. In this method the scans are transformed into the stabilized local frame given the roll and pitch angles. These angles are estimated through an attitude and heading reference system (AHRS). Utilizing bilinear filtering, scans are matched against the current map at the rate of 40 Hz.

The mapping process seeks to calculate a transformation between a current scan and the map. Assume the transformation is represented by $\delta = (x, y, \psi)^T$. For the i^{th} beam of the scan, if the transformed end point of the scan is represented by $s_i = S_i(\delta)$, then the map value at the end point is shown by $M(S_i(\delta))$. The transformation should minimize the difference between the current map and the new transformed scan:

$$\delta^* = \underset{\delta}{\operatorname{argmin}} \sum_{i=1}^{n} (1 - M(S_i(\delta)))^2, \tag{1}$$

where n is the number of scan beams and δ^* is the desired outcome. This problem is formulated and solved in [19] using the gradient ascent approach. To do this, first order Taylor expansion is applied to equation (1) and the resulting Gauss-Newton equation is solved.

C. 3D SLAM

To perform 3D SLAM with a flying robot such as a quadrotor, three major problems should be addressed. These problems are computational demand, communication issues, and fast dynamics. In the next few paragraphs, each of these issues is presented in detail.

Most 3D SLAM algorithms which are based on the RGB-D cameras rely on very fast processing units. Unfortunately, such units are rarely available on quadrotors. One solution to handle this problem is to transmit image and depth data to a ground control station, perform 3D mapping on the workstation, and send the results to the quadrotor. This solution while effective creates a major limitation which is the need for reliable and high bandwidth communication channels. Compared with other sensors, such as the scanning laser ranger, the data rate of an RGB-D camera is very high. For instance, for QVGA quality, the size of each depth image is 320×240 pixels and the size of each color image is $3 \times 320 \times 240$ pixels. The camera operates at 30 Hz. Each color pixel occupies one byte, and each depth pixel occupies two bytes; therefore, the transmission rate of the camera is $(2 \times 320 \times 240 + 3 \times 320 \times 240) \times 30 = 11.52$ mega bytes per second, or approximately 11 MB/s. In contrast, a scanning laser ranger which operates at 40 Hz and has 1082 beams (each beams produces a float64 and needs four bytes) has the transmission rate of $1082 \times 40 \times 4 = 0.17$ MB/s. This rate is approximately 68 times less than the data rate of the camera. In such high rate transmission, the rate of the package loss also increases. Additionally, some frames arrive out-of-sequence which either should be discarded or reprocessed with some previous frames.

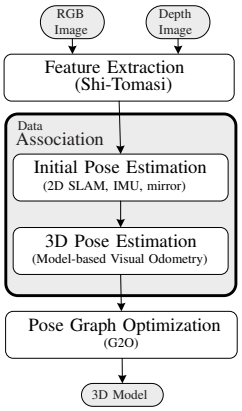

Figure 6. Flow chart of 3D SLAM. In the proposed 3D SLAM, first Shi-Tomasi features are extracted. Then an initial estimate is determined using 2D SLAM, IMU, and the reflective mirror. Then the extracted features are matched with a model of the environment. Finally a G2O optimization is performed over the poses.

Another major problem in 3D feature-based SLAM with quadrotors is its fast dynamics. If the quadrotor has a large rotation or translation, the amount of the overlap between two consecutive frames of the onboard camera decreases. Therefore, matching features of the frames, will produce invalid estimates.

Most 3D SLAM algorithms with RGB-D sensors use a wired link connected to a high speed processing unit to avoid communication issues and high computational demand. For instance, [33] and [36] use a wired connection. [35] also develops 3D SLAM on a quadrotor while the camera is connected to a ground control station with a USB link. Additionally, these algorithms suffer from fast dynamics of the quadrotor.

To deal with these limitations, the information from the IMU and the laser ranger is used to produce an initial and accurate estimate of the 3D pose of the quadrotor. This initial estimate, which includes the 3D coordinates and 3D orientation of the robot, is refined by using the information from the RGB-D camera. There is no need to have a wired link. Also, the motion of the camera can be very aggressive, and sudden changes, like the changes we expect from a quadrotor, are dealt with easily.

Figure 6 shows the required processing blocks of the proposed 3D SLAM algorithm. Inputs of 3D SLAM are color and depth images, captured simultaneously. These two images are processed to extract the pose of the camera and build up the map of the environment. The first three blocks, *Feature Extraction*, *Initial Pose Estimation*, and *3D Pose Estimation*, provide accurate 3D visual odometry. The last block, *Pose Graph Optimization*, provides global consistency of transformations and detects loop closures. In the visual odometry, first Shi-Tomasi features are extracted from the RGB image. Then the results of 2D SLAM and roll and pitch angles are used to provide an initial estimate of the change-in-pose of the robot. This is an initial estimate and needs to be refined by the model-based pose estimation algorithm. Here each block is explained in detail.

1) Feature Extraction.: To extract features, Shi-Tomasi, which is a corner detector algorithm, is used. Shi-Tomasi descriptor is rotation invariant and offers a good tradeoff between computational demand and robustness [40].

2) Initial Pose Estimation.: when the motion of camera involves a large rotation or translation, any visual odometry

algorithm will fail to provide accurate pose estimates. Having a large rotation or translation is very common in robots with fast dynamics, such as quadrotors. Therefore, it is very important to have an initial estimate of the pose of the robot.

For ground robots, direct odometry can be used as an approximate initial estimate, but such direct odometry is not available for quadrotors. As mentioned previously, the proposed 2D SLAM block provides x, y, and heading of the robot. Based on the extensive experiments, the 2D pose of the quadrotor is very accurate and reliable. Additionally, the roll and pitch angles, which are directly used from the EKF implementation of the IMU, have less than one degree error. And the z component can easily be calculated using the reflective mirror mounted on the laser. 10 beams are reflected to the ground, and the results are averaged to produce an average distance. If the average distance of the beams is d, the z component of the 3D pose is calculated as follows:

$$z = d\cos\phi\cos\theta, \tag{2}$$

where ϕ and θ are roll and pitch angles. The 2D SLAM, IMU, and reflective mirror can provide a reliable initial estimate of the 3D pose of the robot. The initial estimate is refined in the next block.

3) 3D Pose Estimation.: Data association is the key element of localization and mapping. This becomes more important when SLAM relies on features only. Correct localization relies on finding correct correspondences between observations and the available map. Incorrect associations may cause pose estimates to rapidly diverge.

In most visual odometry algorithms, features of two consecutive frames are extracted and matched. While this frame-based feature matching looks efficient, it operates locally which means the correlation of the features in the current frame with two or three frames earlier or very past frames is ignored. To address this issue, instead of frame-based feature matching, a model-based feature matching paradigm is used.

To perform model-based visual odometry, first a model of the world using the extracted features is developed. Once a new frame is received, it is matched with the model. The new features which are not available in the model are added to the model. This process is repeated with each incoming frame. The model-based mapping is a reliable and robust algorithm. Compared with other mapping algorithms which are based on matching images frame-by-frame, the model-based mapping accumulates significantly less error [36]. Fig. 7 shows a developed map using this algorithm.

4) Pose Graph Optimization.: The global and consistent mapping of the SLAM algorithm is based on the G2O algorithm which is a bundle adjustment algorithm [2]. Generally, in graphSLAM, poses of the robot are represented as nodes in a graph. The edges connecting nodes are modeled with motion and observation constraints. These constraints are extracted and calculated by the SLAM front-end. Next, these constraints need to be optimized to calculate the spatial distribution of the nodes and their uncertainties [11]. This is performed by the SLAM back-end [41]. Usually the optimization by the back-end takes more time and memory than extracting the constraints; therefore, it runs at a lower frequency.

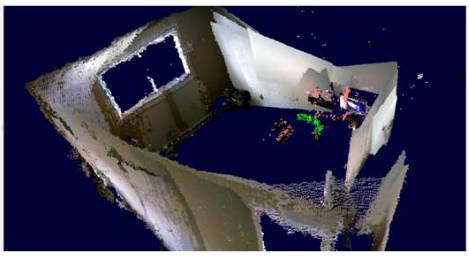

Figure 7. A 3D map, developed using the model based mapping algorithm.

Algorithm 1 summarizes the proposed 3D SLAM algorithm, explained in this section. In line 1, features are extracted, in line 2, the extracted features are transformed using the accurate 2D pose, roll and pitch angles, and the altitude. In line 4, the transformed features are matched with the existing model, build by the features up to time $t-1$. The matching is based on the ICP algorithm. At the end of the algorithm, a bundle adjustment algorithm [2] is applied to the model.

Algorithm 1 3D SLAM for fast dynamics.

Input: M_{t-1} (3D pose and model at time $t-1$), F_t (RGBD frames), p_t (2D pose), θ_t, ϕ_t (roll and pitch), z_t (altitude)
Output: M_t: 3D pose and model at time t

1: $m_t \leftarrow \text{getFeatures}(F_t)$
2: $m'_t \leftarrow \text{transform}(m_t, p_t, \theta_t, \phi_t, z_t)$.
3: $M_t \leftarrow \text{match}(m'_t, M_{t-1})$.
4: $M_{1:t} \leftarrow \text{bundleAdjustment}(M_{1:t})$.

D. State Estimation

Results from SLAM are fused with the information from the IMU by a Kalman filter to provide state estimation which is used in the controller block. This estimate is used by the planner and controller blocks. Prior to using IMU measurements in the Kalman filtering, they are filtered by an AHRS system; therefore, IMU measurements are not present in the state vector of the system. The state of the system is defined as

$$\mathbf{x} = \begin{bmatrix}\mathbf{r^T} & \mathbf{v^T}\end{bmatrix}^\mathbf{T}, \tag{3}$$

where $\mathbf{r}$ is the position of the robot and $\mathbf{v}$ is the velocity of the robot.

$$\mathbf{r} = \begin{bmatrix}x & y & z\end{bmatrix}^T, \tag{4}$$

$$\mathbf{v} = \begin{bmatrix}\dot{x} & \dot{y} & \dot{z}\end{bmatrix}^T. \tag{5}$$

The state prediction is performed using the accelerometers of the IMU, $\mathbf{a} = \begin{bmatrix}a_x & a_y & a_z\end{bmatrix}^\mathbf{T}$, given the following system equations

$$\dot{\mathbf{r}} = \mathbf{v} \tag{6}$$

$$\dot{\mathbf{v}} = \mathbf{Ra} + \mathbf{g}, \tag{7}$$

where $\mathbf{g}$ is the constant gravity vector and $\mathbf{R}$ is the direction cosine matrix. Euler angles from the AHRS are used in the direction cosine matrix.

In the update step, the state of the system is updated using the other sensors. The altimeter, 2D SLAM, and 3D SLAM are

used to update the state. For instance, a measurement from 2D SLAM is used to update x and y coordinates and the heading angle, since the horizontal laser ranger provides only the 2D coordinates. An altimeter updates only the z coordinate.

E. Controller

The controller block accepts waypoints generated by the planner block and adjusts rotor speeds. The controller block is not a part of this work. Nagaty et al. [3] developed a cascaded controller for a quadrotor which is used here. The controller is composed of inner loop and outer loop PID controllers. The inner loop controller stabilizes roll, pitch, yaw and altitude while the outer loop controller is responsible for position tracking or waypoint following.

F. Planner-Exploration of Unknown Space and Path Planning

To explore an unknown environment, the frontier algorithm is used. This algorithm uses unknown cells located in the boundaries of the known cells as *frontiers* waypoints and moves towards them [42]. Once a waypoint for the exploration is known, a path planning algorithm is used to guide the robot to the new waypoint.

To navigate between two given points, the wavefront algorithm is used. This is performed using the map generated while exploring the environment. The wavefront produces an optimal solution and no local minima are generated [12]. To improve the algorithm and avoid getting close to the obstacles, occupied cells are dilated such that the planner generates a path with a safe distance from obstacles.

G. Obstacle Avoidance

Obstacle avoidance is achieved using laser beams directly to avoid any unexpected collisions caused by disturbance or dynamic objects. Laser beams are smoothed by a sliding window to remove noisy measurements. Then beams are divided into b bins (for example, for the Hokuyo UTM-30LX laser ranger, 54 bins are used, each covering $5°$). The range of a bin is defined as the average range of laser beams in that bin. A collision bin is one which identifies an obstacle within a given range using the range measurements within the bin. If the range of a bin is smaller than a safety threshold, the bin has collision risks. A free bin has a range greater than a threshold. The free bin is the bin used to avoid the collision. If a bin is identified as a collision bin, then the current waypoint is changed such that the new waypoint has $180°$ offset from the collision bin.

IV. Experiment

To test the proposed perception and navigation solution, the mission shown in Fig. 5 was implemented. Additionally, two more experiments, demonstrating other behaviors, are also presented. 2D maps (at 40 Hz) and 3D maps (at 15 Hz) are built in real-time and on-the-fly. The exploration, path planning, and mapping tasks are performed on the ground control station station and the results are sent to the quadrotor. with all the mentioned sensors, the quadrotor can fly up to six minutes.

A. Case 1: Quadrotor in an Indoor Environment

This real-world experiment was performed with the custom-built COBRA quadrotor, shown and explained in Fig. 2.

1) Description: The test environment is approximately 94.7 m^2 and two boxes are placed in the environment as obstacles. Fig. 8 shows the quadrotor in the test environment. According to the mission plan, shown in Fig. 5 and starting from the point shown by a black circle in Fig. 8-a, first, the quadrotor explores the environment and maps all unknown places such as behind boxes. Then it moves to a goal point, shown by a black square. Finally, it returns home, where it started the mission.

2) Results: The whole mission took about $320sec$, with an average speed of $0.25m/s$. Fig 8-b shows the robot at approximately 280 seconds into the test, where the robot is returning home. The green curve shows the path generated by the planner. The red curve shows the trajectory of the robot. The arrow shows the final destination of the robot which is the home. The cyan sphere shows a waypoint along the path, generated by the planner for the quadrotor. (The path, the trajectory and the waypoint are projected to the ground level for clarity of the figure.) A video of the experiment is available in [43]. The robot successfully completed the mission by exploring, mapping, and navigating through obstacles. The maximum error of achieving to the goal points is 0.16 meters.

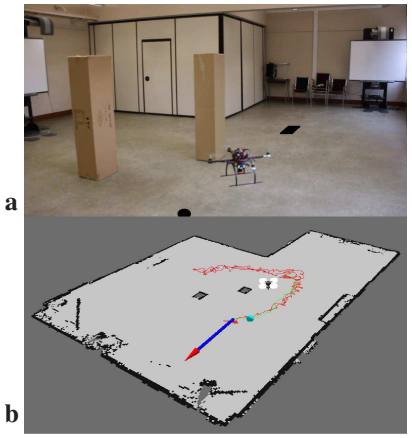

Figure 8. Experiment with COBRA quadrotor. This experiment, performs the three-stage mission depicted in Fig. 5: *exploration*, *move-to-goal*, and *return-home*. **a)** This figure shows the test environment. **b)** A snapshot of the developed map and trajectory of the robot.

B. Case 2: Autonomous Entry and Exit

This is a simulated and real-world experiment performed with COBRA quadrotor. The purpose of this test is to demonstrate the capability of the quadrotor in transition from outdoors to indoors. Details of the experiment including flight time, the length of the path, and the error of achieving the goal points are not reported for the sake of brevity.

1) Description: For the simulated experiment, performed in Gazebo, a building was simulated which is unknown for the robot. It starts the mission outside of the building and the mission is to go inside, map the building, and return to the start point. Notice that in this simple scenario, exploration is not involved. The robot chooses a traversable goal point inside

the building, flies to the point according to the planned path, and returns. The same scenario is implemented for the real-world experiment. These experiments demonstrate the ability of the robot to move from outdoors to indoors which is an important task in most autonomy solutions.

2) Results: Fig. 9 shows the simulated experiment. The figure represents the moment that the robot is returning to the start point. In Fig. 9-a, a snapshot of the simulated world is shown. The quadrotor is inside of the building and is not visible. The green disc is the start point. Fig. 9-b shows the developed 3D map, built by the vertical laser rangefinder. Fig. 9-c shows the camera view. Finally, Fig. 9-d demonstrates the 2D map, developed by the horizontal laser rangefinder, designed path, and trajectory of the robot.
Fig. 10 shows the same scenario in a real-world experiment. Fig. 10-a, shows the test environment. Fig. 10-b shows the developed map, trajectory of the robot, and designed path to return to the start point. A video of the experiment, including both simulated and real-world experiments, can be found in [43].

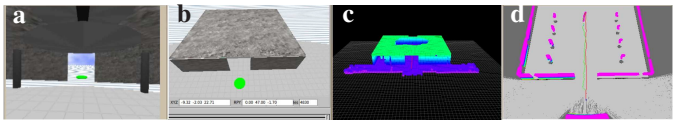

Figure 9. Autonomous entry and exit in simulation. **a)** Simulated world in Gazebo. **b)** The developed 3D map **c)** Camera view. **d)** The developed 2D map, path, and trajectory.

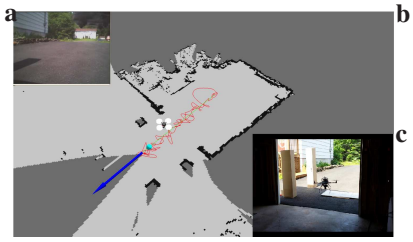

Figure 10. Autonomous entry and exit in a real-world experiment. This figure shows the experiment at the moment that the quadrotor returns home. **a)** Onboard camera view **b)** The developed 2D map, path, and trajectory. **c)** Ground view of the quadrotor, exiting through the door.

C. Case 3: Outdoor Unstructured Environment

This outdoor experiment demonstrates the ability of the quadrotor in mapping and navigating in unstructured environments, such as woods and cliffs. The experiment is performed at UNB Woodlot in Fredericton, New Brunswick. This experiment demonstrates the usefulness of the quadrotor and the proposed solution in different applications such as forest and vegetation control, terrain monitoring and inspection in untraversable and catastrophic environments, and saving human lives trapped in dangerous spots.

1) Results: Fig. 11-a shows the experiment site. As the figure shows, the environment is not traversable by ground robots. Fig. 11-b shows the 2D developed map and trajectory of the robot. Trees and cliffs are mapped consistently.

D. Case 4: 3D mapping

The setup of the quadrotor is similar to Case 1. The test environment is approximately 60 m^2. Similar to Case 1, first

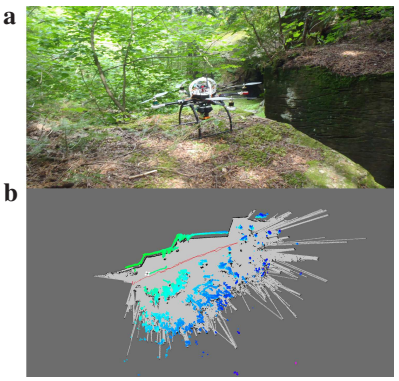

Figure 11. The experiment in an outdoor unstructured environment. **a)** The test environment. **b)** The developed map and trajectory of the robot. Laser measurements are color-coded.

the quadrotor explores the room, then it returns home.

1) Results: The mission took about $50sec$. Fig 12-(top) shows the 3D map of the environment. The green curve is the trajectory of the quadrotor. Fig 12-(bottom) shows the 2D map of the same environment, developed at the same time that the 3D map was built. A video of the experiment is available in [43]. In the video, it has been demonstrated that a blue target has been identified and marked in the 3D map. The robot completed the mission by exploring, mapping, and navigating through obstacles successfully. In comparison with other 3D

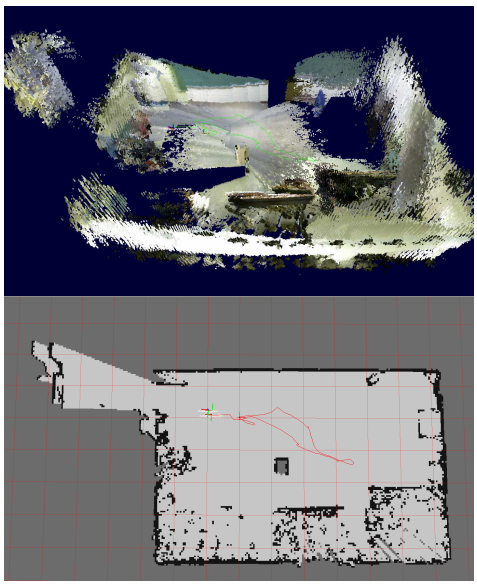

Figure 12. (top) 3D map of an indoor test environment, (bottom) 2D map of the same environment

mapping algorithms such as [36], as shown in Fig. 1, our algorithm is able to generate consistent models despite the fast dynamics and loss of the frames in the communication channels. This advantage is achieved by initializing the 3D frame matching with laser and inertial information which allows the optimization process to avoid the local minima.

V. Conclusion and Future Work

In this work, an autonomy solution for an unmanned rotorcraft was proposed and implemented. The proposed solution

tackles a few key requirements for autonomous navigation: mapping, localization, and path planning. A reliable 3D mapping was proposed that handles aggressive motion of the quadrotor efficiently. Also, the developed 2D and 3D maps were displayed on multiple remote tablets. Moreover, a behavior based mission control was proposed to plan, organize, and sequence different flight behaviors. The proposed system was tested extensively in various environments.

In the future, it would be desirable to extend the work to multiple-robots: cooperatively exploring, mapping, localizing, and performing the mission.

REFERENCES

[1] S. Saeedi, A. Nagaty, C. Thibault, M. Trentini, and H. Li, "Perception and navigation for autonomous rotorcraft," in *International Conference on Intelligent Unmanned Systems (ICIUS)*, 2014.

[2] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3607–3613.

[3] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, and H. Li, "Control and navigation framework for quadrotor helicopters," *Journal of Intelligent and Robotic Systems*, vol. 70, no. 1-4, pp. 1–12, 2013.

[4] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004, pp. 2451–2456.

[5] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[6] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV testbed," *IEEE Robotics and Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[7] S. Thrun, M. Diel, and D. Hahnel, "Scan alignment and 3-D surface modeling with a helicopter platform," *Field and Service Robotics*, vol. 24, pp. 287–297, 2006.

[8] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2009, pp. 2878–2883.

[9] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *International Journal of Micro Air Vehicles*, vol. 1, no. 4, pp. 217–228, 2009.

[10] C. Stachniss, *Robotic Mapping And Exploration*. Springer Tracts in Advanced Robotics, 2009, vol. 55.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: The MIT press, 2005.

[12] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[13] L. Babel, "Flight path planning for unmanned aerial vehicles with landmark-based visual navigation," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 142 – 150, 2014.

[14] E. Frazzoli, M. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1077–1091, 2005.

[15] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," *Autonomous Robots*, vol. 36, no. 4, pp. 365–381, 2014.

[16] D. E. Soltero, M. Schwager, and D. Rus, "Decentralized path planning for coverage tasks using gradient descent adaptive control," *The International Journal of Robotics Research*, vol. 33, no. 3, pp. 401–425, 2013.

[17] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments," pp. 733 219–733 219–10, 2009.

[18] I. Dryanovski, W. Morris, and J. Xiao, "An open-source pose estimation system for micro-air vehicles," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2011, pp. 4449–4454.

[19] S. Kohlbrecher, O. v. Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 2011, pp. 155–160.

[20] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, 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," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[21] S. Shen, N. Michael, and V. Kumar, "Obtaining liftoff indoors: Autonomous navigation in confined indoor environments," *IEEE Robotics Automation Magazine*, vol. 20, no. 4, pp. 40–48, 2013.

[22] M. Blosch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *International Conference on Robotics and Automation (ICRA)*, 2010, pp. 21–28.

[23] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular SLAM based navigation for autonomous micro helicopters in GPS-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[24] Y. Li, Y. Wang, W. Lu, Y. Zhang, H. Zhou, and B. Yan, "Tsinghua team entry for the 2012 AUVSI international aerial robotics competition."

[25] B. Guerreiro, P. Batista, C. Silvestre, and P. Oliveira, "Globally asymptotically stable sensor-based simultaneous localization and mapping," *Robotics, IEEE Transactions on*, vol. 29, no. 6, pp. 1380–1395, 2013.

[26] K. Schmid, T. Tomic, F. Ruess, H. Hirschmuller, and M. Suppa, "Stereo vision based indoor/outdoor navigation for flying robots," in *International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 3955–3962.

[27] F. Fraundorfer, L. Heng, D. Honegger, G. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4557–4564.

[28] A. Wendel, A. Irschara, and H. Bischof, "Natural landmark-based monocular localization for MAVs," in *International Conference on Robotics and Automation (ICRA)*, 2011, pp. 5792–5799.

[29] K. Pathak, A. Birk, N. Vaskkevicius, and J. Poppinga, "Fast registration based on noisy planes with unknown correspondences for 3-D mapping," *Robotics, IEEE Transactions on*, vol. 26, no. 3, pp. 424–441, 2010.

[30] D. Borrmann, J. Elseberg, K. Lingemann, A. Nchter, and J. Hertzberg, "Globally consistent 3d mapping with scan matching," *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 130 – 142, 2008.

[31] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM: 3D mapping outdoor environments," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 699–722, 2007.

[32] J. Weingarten and R. Siegwart, "EKF-based 3D SLAM for structured environment reconstruction," in *International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 3834–3839.

[33] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[34] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Annual Conference on Computer Graphics and Interactive Techniques*, 1996, pp. 303–312.

[35] J. Sturm, E. Bylow, F. Kahl, and D. Cremers, "Dense tracking and mapping with a quadrocopter," in *Unmanned Aerial Vehicle in Geomatics (UAV-g)*, 2013.

[36] I. Dryanovski, R. Valenti, and J. Xiao, "Fast visual odometry and mapping from RGB-D data," in *International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2305–2310.

[37] R. A. Brooks, "Elephants don't play chess," *Robotics and Autonomous Systems*, vol. 6, pp. 3–15, 1990.

[38] M. J. Mataric, "Behavior-based control: Examples from navigation, learning, and group behavior," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 9, pp. 323–336, 1997.

[39] J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," in *International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2000, pp. 2061–2066.

[40] F. Fraundorfer and D. Scaramuzza, "Visual odometry : Part II: Matching, robustness, optimization, and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[41] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2010.

[42] B. Yamauchi, A. Schultz, and W. Adams, "Integrating exploration and localization for mobile robots," *Autonomous Robots*, 1999.

[43] 2016, retrieved Feb 3, 2016, http://www.ece.unb.ca/COBRA/quadrotor.htm.