An Effective Method for Autonomous Localization and Navigation in Unknown Indoor Environment using MAV

Swee King Phang*, Fei Wang, Kangli Wang, Shupeng Lai and Ben M. Chen
Department of Electrical and Computer Engineering, National University of Singapore, Singapore, 117576

ABSTRACT

Localization in GPS-denied indoor environment is one of the biggest challenge in robotics field especially in aerial robots. Limited payload of micro air vehicles (MAVs) makes indoor navigation even harder as compared to ground robots. In this manuscript, we explore an effective method for indoor localization using two 2-D laser scanners for map and obstacles detection, that can be processed onboard. Key idea and algorithm of the proposed method will be discussed in detail, starting from laser data feature extraction to rotational tracking, then to feature association, followed by position tracking. The localization method is implemented to a quadrotor MAV of dimension lesser than 90 cm and weight less than 3 kg. This method is proven to be effective and reliable throughout flight tests in different indoor scenario, including the indoor mission of the IMAV 2014 Competition that was held in Delft, the Netherlands.

1 INTRODUCTION

Today, as the electronics are getting more powerful while its sizes are getting smaller, the concept of miniaturize unmanned aerial vehicles (UAVs) has aroused among the researchers around the world. The term micro air vehicle (MAV) is now widely used to describe small scale UAVs. The shrinking of the size of MAVs has enabled them to navigate indoor, which opens up a wide range of application not only in military field but also in civilian market [1]. As a result, GPS-denied localization and navigation methods are being explored and utilized to realize autonomous indoor flight.

In most indoor scenario, GPS signal is weak or unavailable in an indoor environment and prior information about the indoor structure is unknown. This boils down to the famous problem of simultaneous localization and mapping (SLAM) in robotics field. Many theoretical works and practical implementations of SLAM on ground robots [2] [3], and on UAV platforms [4] [5] [6] [7] have been published in literature. However, few of them have considered the computation limitation on MAVs, and they usually exploit the unlimited payload on ground robots or rely on high-bandwidth communication to the ground control station (GCS) where a powerful computer is running a computationally intensive algorithm. In consequence, some of them only work in controlled lab environments to guarantee good communication link between the MAVs and GCS.

*Email address: king@u.nus.edu

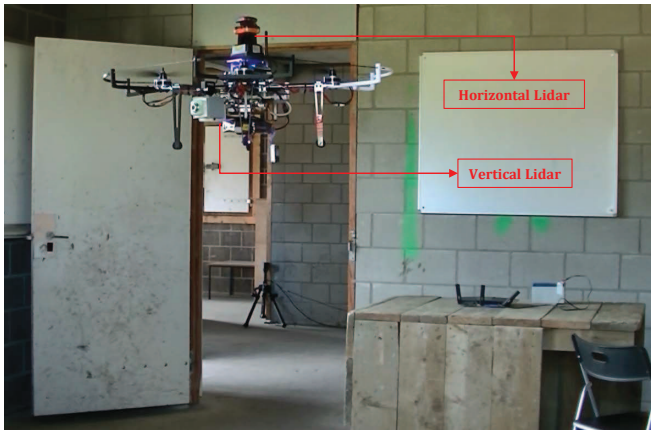

Figure 1: MAV platform autonomous room exploration in IMAV 2014 indoor mission

In more practical applications, localization and navigation of MAVs should not heavily depend on communication link. In this manuscript, an effective method for indoor localization using two 2-D laser scanners for map and obstacles detection is proposed. The localization algorithm should be processed onboard with limited computational power. Key idea and algorithm of the proposed method will be discussed in detail, starting from laser data feature extraction to rotational tracking, then to feature association, followed by position tracking. The localization method is implemented to a quadrotor MAV of dimension lesser than 90 cm and weight less than 3 kg.

The manuscript is divided into six sections. Section 1 gives brief introduction to the current development of indoor localization trend. A system overview which include electronics used in the MAV will be discussed in Section 2. Section 3 details the proposed algorithm for effective localization and navigation using onboard processors, followed by controller design and implementation on MAV to realize autonomous navigation in Section 4. Results of the proposed localization and navigation are shown in Section 5. Finally, concluding remarks will be made in Section 6.

2 HARDWARE OVERVIEW

In order to solve real-life complicated scenarios, such as IMAV 2014 indoor mission, the MAV platform (see Fig. 1) is built to meet the following requirements:

1. The maximum dimension of the MAV must be lesser than 90 cm in order to navigate pass windows and doorways;
2. A long range 2-D laser scanner is needed to scan horizontal plane for localization purpose;
3. A short range 2-D laser scanner is needed to scan vertical plane to detect ground level in obstacle rich environment;
4. Able to receive live images from an attached camera on GCS.

To meet the above-mentioned requirements, three processors are used for flight control, command generation and measurement calculation (see Fig. 2). Pixhawk Autopilot board is used to stabilize the MAV as a low level flight controller. Gumstix Overo Fire is used to generate position and heading command to Pixhawk given the reference and measurement data obtained from Mastermind computer of Ascending Technologies. The proposed localization method will be implemented to the Mastermind computer with the data from two laser scanners. Camera images and important flight data will be sent to the GCS via the wireless link of the Gumstix and GCS. Map exploration algorithm which provides trajectory reference to the MAV will also be implemented in the Mastermind computer. However, it is not in the focus of this paper, thus will not be mentioned in the manuscript.

3 LOCALIZATION AND NAVIGATION ALGORITHM

For most practical tasks involved building exploration, The MAV is expected to inspect the interior of a building with no prior knowledge of the building structure. Throughout the flight, an overview of the indoor map should be slowly built up and displayed on the GCS.

To achieve this, two innovative yet reasonable assumptions on the indoor environment are made to effectively lower the computational cost needed, such that the navigation algorithm can run onboard on the MAV in real time:

1. The environment can be described by sparse features, which include corners and straight line segments;
2. The line features are orthogonal to each other or off-set by multiples of a constant angle displacement, such as 30° or 45°.

These two assumptions are fulfilled for most of modern indoor environments. It should also be highlighted that the proposed algorithm will work properly as long as the majority of corner and line features in the indoor environment meets

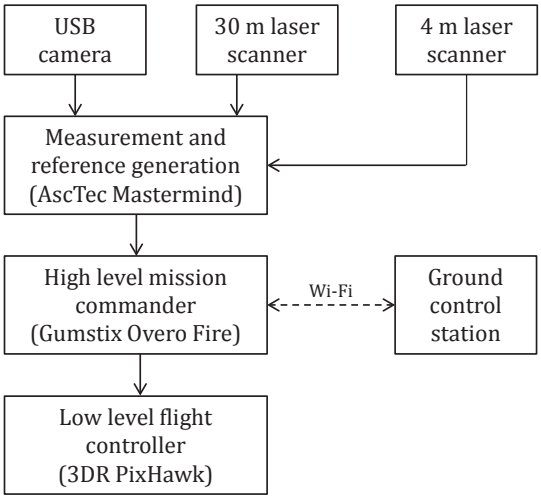

Figure 2: Hardware configuration of indoor MAV

the assumptions. The algorithm is able to detect and filter common outliers, and thus the localization performance will not be affected.

The MAV's pose in the map frame can be represented by its 3-D coordinates x, y, z and heading angle ψ. They are divided into two groups: the planar pose (x, y, ψ) and the altitude z. The former can be estimated by the 30 meters laser scanner which scans horizontally, while the later can be computed by the 4 meters laser scanner which scans vertically.

The localization algorithm via the first laser scanner contains the fundamental ideas that make the whole navigation algorithm robust and efficient. With the first assumption, the conventional point cloud matching algorithm can be avoided, leaving the number of point matching pairs to be thousands of times lesser than the conventional method. With the second assumption, the estimation of rotational motion of MAV can be done by comparing the difference between line gradients instead of relying on point feature matching, thus making it to be decoupled from translational motion. It is beneficial as the rotational motion usually results in inconsistent point matching results, especially when the feature points are far. The planar localization algorithm include five distinct steps, namely feature extraction, rotation tracking, point feature association, line feature association and position tracking. They will be discussed in the following subsections.

3.1 Feature Extraction

The 30 meters laser scanner, Hokuyo UTM-30LX is utilized as the source of measurement for the proposed localization algorithm. For each frame of scanned data, there are 1081 data representing the measured distances in the frontal 270 degrees span. The 2-D data is then going through *split-and-merge* algorithm so that they are grouped into clusters of

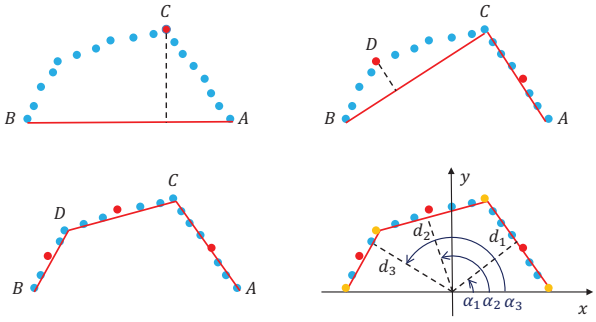

Figure 3: The *split-and-merge* and line extraction algorithm.

straight line segment. The main steps of the *split-and-merge* algorithm are briefly summarized in the following points with Fig. 3 giving a graphical illustration [8]:

1. The first point A and the last point B of the input data is connected by a straight line;
2. point C is found among all data points that has the longest perpendicular distance to the line AB;
3. A cluster is created with points in between A and B if this longest distance is within a preset threshold;
4. The input points will be split into two subgroups, A-C and C-B, otherwise. For each group, the *split-and-merge* algorithm will be applied recursively.

Once the clusters of points are obtained, least-mean-square fitting is done to extract the line feature parameters. Here, the end points of the line segments are chosen to be point features. By convention, each line feature can be represented by two parameters, that is the line's normal direction α_k and its perpendicular distance to the center of laser scanner d_k (see Fig. 3 for reference). Each point feature can be simply represented by its 2-D coordinates.

3.2 Rotation tracking

The second assumption is utilized in an innovative way to keep track of the MAV's heading direction ψ. Without loss of generality, x-axis of the map frame is align with the longest observable wall in the indoor environment at initial position. With this assignment, all other walls will have their orientations at $n\alpha$, where α is the preset constant angle displacement (in most cases, 90°) and n can be any integers. By choosing any wall currently observable and let its direction be β_b in the MAV body frame, we have its map frame direction β_m at:

$$\begin{aligned}\beta_m &= \psi_t + \beta_b \\ &= \psi_{t-1} + \Delta\psi_t + \beta_b\end{aligned}$$

where ψ_t and ψ_{t-1} are the MAV heading angles in the current frame and previous frame respectively, $\Delta\psi_t$ is the inter-frame heading movement. From the second assumption, one can easily deduce that $(\psi_{t-1} + \Delta\psi_t + \beta_b)$ is divisible by α, which leads to

$$\Delta\psi_t = -[(\psi_{m,t-1} + \beta_l)\,\%\,\alpha], \tag{1}$$

where the operator % is defined in this manuscript as:

$$a\%b = \begin{cases} (a \mod b) & , \text{ if } (a \mod b) \le b/2 \\ (a \mod b) - b & , \text{ otherwise} \end{cases} \tag{2}$$

After obtaining $\Delta\psi_t$, the UAV heading can be updated as

$$\psi_t = \psi_{t-1} + \Delta\psi_t. \tag{3}$$

Using this method, the MAV heading angle is tractable in each frame provided that the initial heading ψ_0 is known. However, it should be noted that this heading tracking algorithm only works when the MAV inter-frame rotational motion is less than $\alpha/2$, which limits the yawing speed of the MAV to 450°/s for $\alpha = 90°$, or 225°/s for $\alpha = 45°$ as the laser scanned frames are obtained at 10 Hz speed. As this method compares the orientation of each walls in any frame to the MAV initial heading angle, it solves the yaw drifting problem commonly exists in conventional Lidar SLAM solution [9].

3.3 Point feature association

As mentioned in the subsections above, the end points of the line clusters are recorded as point features, in which most of them should represent physically the existing corners or ending points of walls in the indoor environment. In this step, locally observed point features are associated from one frame to the next such that the incremental planar displacement of the MAV can be tracked. It is done by rotating the locally observed point features to the map x-y direction based on the MAV heading angle ψ_t estimated previously. As a result, only translational motion needs to be taken care for the inter-frame movement after the frames are correctly orientated. By considering the fact that this translational motion between frames of 10 Hz is very small, the nearest neighbor searching is a good choice to associate them. The 2-D transformation from the laser scanner local frame to the map frame can be calculated as

$$\mathbf{p}_\mathrm{g} = \mathbf{R}_t \times \mathbf{p}_\mathrm{l}, \tag{4}$$

where $\mathbf{p}_\mathrm{l}$ is the local feature point, $\mathbf{p}_\mathrm{g}$ is the same feature point rotated to the map x-y direction, and $\mathbf{R}_t$ is the rotation matrix, which is obtained as

$$\mathbf{R}_t = \begin{bmatrix} \cos\psi_t & \sin\psi_t \\ -\sin\psi_t & \cos\psi_t \end{bmatrix}. \tag{5}$$

3.4 Line feature association

Line features are not only used to track rotation, but also to supplement translational motion estimation when the number of point features are too low. First, all locally observed

lines are rotated to the map x-y direction and then classified into horizontal lines and vertical lines. For each group of lines, they are compared exhaustively with those in the previous scan. If their respective d_i and α_i parameters (see Fig. 3) are sufficiently close they are associated. The difference in d_i of the two frames represent the translational motion of the MAV in either x-axis or y-axis for associated horizontal lines and associated vertical lines respectively.

3.5 Position tracking

Similar to rotation tracking, the current position can be computed based on the previous-frame position $[x_{t-1}, y_{t-1}]$ and an incremental change $[\Delta x_t, \Delta y_t]$ as

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} \Delta x_t \\ \Delta y_t \end{bmatrix}, \tag{6}$$

where

$$\begin{bmatrix} \Delta x_t \\ \Delta y_t \end{bmatrix} = \frac{\sum w_{\mathrm{p}}(\mathbf{p}_t - \mathbf{p}_{t-1})}{\sum w_{\mathrm{p}}} + \begin{bmatrix} \frac{\sum w_{\mathrm{l},x}(d_{x,t} - d_{x,t-1})}{\sum w_{\mathrm{l},x}} \\ \frac{\sum w_{\mathrm{l},y}(d_{y,t} - d_{y,t-1})}{\sum w_{\mathrm{l},y}} \end{bmatrix}. \tag{7}$$

It can be treated as an average displacement of all the associated features. By considering the laser scanner noise model, i.e. points further away and lines shorter are relatively more noisy, the matched point features are given different weights w_{p}, $w_{\mathrm{l},x}$ and $w_{\mathrm{l},y}$ in calculating the average displacement. The closer the point feature or the longer the line feature, the larger the weight.

3.6 Height Computation

In the case where the ground is completely clean and flat, the height measurement of the MAV can be obtained via a sonar sensor or a one-point laser range finder. However, for the cases when the MAV needs to fly over random obstacles such as tables, chairs and windowsills for most practical implementation, these sensors might fail as the MAV cannot distinguish between the actual floor surface and the surfaces of protruding objects. An alternative is the barometer sensor. However its accuracy will be an issue to the MAV to fly in a confined indoor space. To solve this problem, a 4 meters Hokuyo URG-04LX laser scanner is mounted vertically to estimate the actual flying height of the MAV. The height calculation algorithm with robust floor identification is developed and integrated into the navigation system.

In estimating the flying height, *split-and-merge* method is again applied. After filtering line segments with dissimilar gradients to the ground plane, the left over are sorted by their perpendicular distances to the MAV. The furthest line segments are kept, and among them the longest one is assumed to be the true ground. Then, the MAV height can be calculated as the perpendicular distance of this line to the MAV center of gravity (CG), compensated by the offset of the MAV attitude angles. Using this method, an accurate height measurement can be obtained as long as the laser scanner projects a portion of its laser beams onto the true ground in every frame. It is proven to work well and is robust in actual flight tests for the case when the MAV flies over protruding objects on the ground.

4 CONTROL

Controls of a MAV is usually separated into two different loops, with an inner-loop running at a higher rate stabilizing the aircraft in the attitude dynamics, and an outer-loop running at a lower rate to control the position or linear velocity of the aircraft [10]. As mentioned in the previous sections, the platform is stabilized in the attitude dynamics by the 3DR Pixhawk Autopilot controller. The Pixhawk hardware comes with open source attitude stabilizer, which is available freely from the Internet [11] [12]. Thus in this manuscript, only the design of the outer-loop controller for position tracking will be mentioned.

Here, a RPT control concept developed in [13] is applied to the MAV trajectory tracking. Theoretically, a system controlled by this method is able to track any given reference with arbitrarily fast settling time subjected to disturbances and initial conditions. The basic idea can be generalized as follows. Given a linear time invariant system

$$\Sigma = \begin{cases} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} + E\boldsymbol{w} \\ \boldsymbol{y} &= C_1\boldsymbol{x} + D_1\boldsymbol{w} \\ \boldsymbol{h} &= C_2\boldsymbol{x} + D_2\boldsymbol{u} + D_{22}\boldsymbol{w} \end{cases}, \tag{8}$$

with $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{y}, \boldsymbol{h}$ being the state, control input, disturbance, measurement and controlled output respectively, the task of RPT controller is to formulate a dynamic measurement control law of the form of

$$\begin{aligned} \dot{\boldsymbol{v}} &= A_c(\varepsilon)\boldsymbol{v} + B_c(\varepsilon)\boldsymbol{y} + G_0(\varepsilon)r + \ldots + G_{\kappa-1}(\varepsilon)r^{\kappa-1}, \\ \boldsymbol{u} &= C_c(\varepsilon)\boldsymbol{v} + D_c(\varepsilon)\boldsymbol{y} + H_0(\varepsilon)r + \ldots + H_{\kappa-1}(\varepsilon)r^{\kappa-1}, \end{aligned}$$

so that when an proper $\varepsilon^* > 0$ is chosen,

1. The resulted closed-loop system is asymptotically stable subjected to zero reference.
2. If $e(t,\varepsilon)$ is the tracking error, then for any initial condition $\boldsymbol{x}_0$, there exists:

$$\|e\|_p = \left(\int_0^\infty |e(t)^p| dt\right)^{1/p} \to 0, \quad as \quad \varepsilon \to 0. \tag{9}$$

Similar to the case introduced in [14], the outer dynamics of a quadrotor MAV is differentially flat. That means all its state variables and inputs can be expressed in terms of algebraic functions of flat outputs and their derivatives. A proper choice of flat outputs could be

$$\sigma = [x, y, z, \psi]^{\mathrm{T}}. \tag{10}$$

It can be observed that the first three outputs, x, y, z, are totally independent. In other words, we can consider the MAV as a mass point with constrained velocity, acceleration, jerk and any higher order. Hence, a stand-alone RPT controller based on multiple-layer integrator model in each axis can be designed to track the corresponding reference in that axis. For the x-axis or the y-axis, the nominal system can be written as

$$\begin{cases} \dot{\boldsymbol{x}}_n = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}_n + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_n \\ \boldsymbol{y}_n = \boldsymbol{x}_n \end{cases}, \tag{11}$$

where $\boldsymbol{x}_n$ contains the position and velocity state variables and u_n is the desired acceleration.

To achieve better tracking performance, it is common to include an error integral to ensure zero steady-state error subjected to step inputs. This requires an augmented system to be formulated as

$$\begin{cases} \dot{\boldsymbol{x}}_{xy} = \begin{bmatrix} 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_{xy} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_{xy} \\ \boldsymbol{y}_{xy} = \boldsymbol{x}_{xy} \\ h_{xy} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_{xy} \end{cases}, \tag{12}$$

where $\boldsymbol{x}_{xy} = \begin{bmatrix} \int (p_e) & p_r & v_r & a_r & p & v \end{bmatrix}^{\mathrm{T}}$ with p_r, v_r, a_r as the position, velocity and acceleration references in the controlled axis, p, v as the actual position and velocity and $p_e = r_p - p$ as the tracking error of position. By following the procedures in [15], an linear feed back control law of the form below can be acquired,

$$u_{xy} = F_{xy} \boldsymbol{x}_{xy}, \tag{13}$$

where

$$F_{xy} = \begin{bmatrix} \dfrac{k_i \omega_n^2}{\varepsilon^3} & \dfrac{\omega_n^2 + 2\zeta\omega_n k_i}{\varepsilon^2} & \dfrac{2\zeta\omega_n + k_i}{\varepsilon} \\ 1 & -\dfrac{\omega_n^2 + 2\zeta\omega_n k_i}{\varepsilon^2} & -\dfrac{2\zeta\omega_n + k_i}{\varepsilon} \end{bmatrix}. \tag{14}$$

Here, ε is a design parameter to adjust the settling time of the closed-loop system. ω_n, ζ, k_i are the parameters that determines the desired pole locations of the infinite zero structure of (12) through

$$p_i(s) = (s + k_i)(s^2 + 2\zeta\omega_n s + \omega_n^2) \tag{15}$$

The z-axis control is similar but in a lower-order form. As the inner-loop is directly looking for velocity reference in this axis, it is straight forward to model the outer loop as a single integrator from velocity to position, and it leads to the augmented system as

$$\begin{cases} \dot{\boldsymbol{x}}_z = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_z + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_z \\ \boldsymbol{y}_z = \boldsymbol{x}_z \\ h_z = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_z \end{cases}. \tag{16}$$

where $\boldsymbol{x}_z = \begin{bmatrix} \int (p_e) & p_r & v_r & p \end{bmatrix}^{\mathrm{T}}$. This leads to a linear feedback control law of

$$u_z = F_z \boldsymbol{x}_z, \tag{17}$$

where

$$F_z = \begin{bmatrix} -\dfrac{\omega_n^2}{\varepsilon} & \dfrac{2\omega_n\zeta}{\varepsilon^2} & 1 & -\dfrac{2\omega_n\zeta}{\varepsilon^2} \end{bmatrix}.$$

Theoretically, when the design parameter ε is small enough, the RPT controller can give arbitrarily fast responses. However, due to the constraints of the MAV physical dynamics and its inner-loop bandwidth, it is safer to limit the bandwidth of the outer loop to be much smaller than that of the inner-loop dynamics. For the case of the designed MAV, the following design parameters are used:

$$x,y \text{ axis}: \begin{cases} \varepsilon &= 1 \\ \omega_n &= 0.99 \\ \zeta &= 0.707 \\ k_i &= 0.25 \end{cases} \quad z \text{ axis}: \begin{cases} \varepsilon &= 1 \\ \omega_n &= 0.559 \\ \zeta &= 2 \end{cases}$$

5 Results

The proposed localization and navigation algorithm was first tested using Matlab on a set of data collected by a horizontally scanning laser scanner mounted onboard of the MAV. Autonomous flight experiments were conducted in one level of the office. The flight results are shown in Fig. 4-6 where red dots indicate the estimated path of the MAV and blue dots are the reconstructed map. It is clear that the result of using both point and line features to compute translational motion of the MAV is better than the other two cases, which uses only one type of features.

In the IMAV 2014 indoor competition, the localization and navigation algorithm was implemented onboard of the MAV in the Mastermind computer. Localization, navigation, and map display on GCS were all running in real time. Fig. 7 shows the reconstructed map after the flight finished. The indoor environment was accurately constructed autonomously and via the USB camera, the operator at the GCS had successfully identified all items in various rooms. Our team who

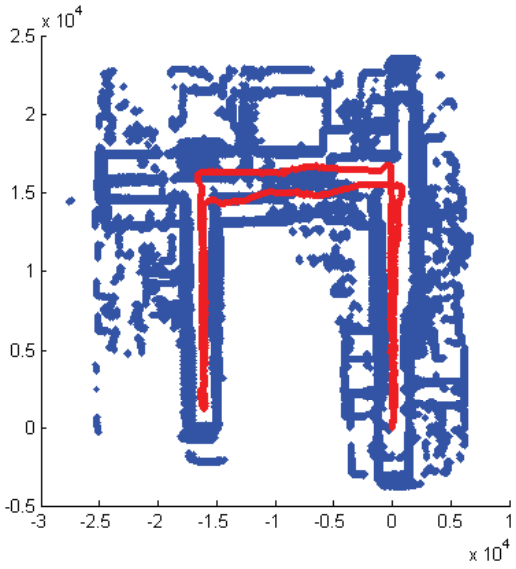

Figure 4: Result of trajectory and map using only point features for translational motion estimation

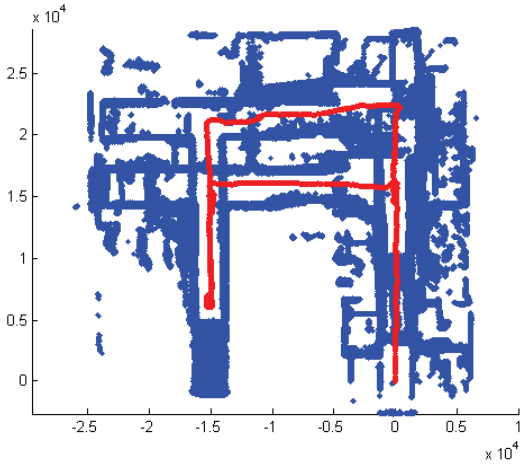

Figure 5: Result of trajectory and map using only line features for translational motion estimation

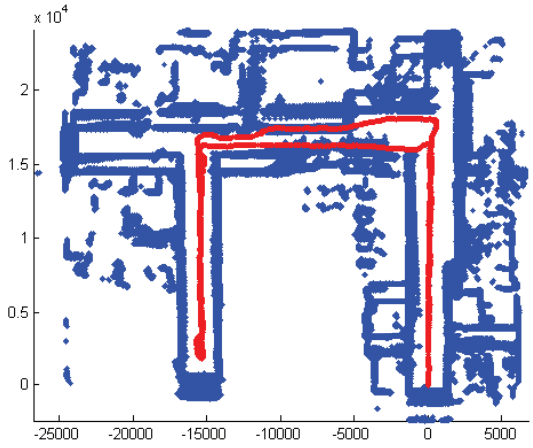

Figure 6: Result of trajectory and map using both point and line features for translational motion estimation

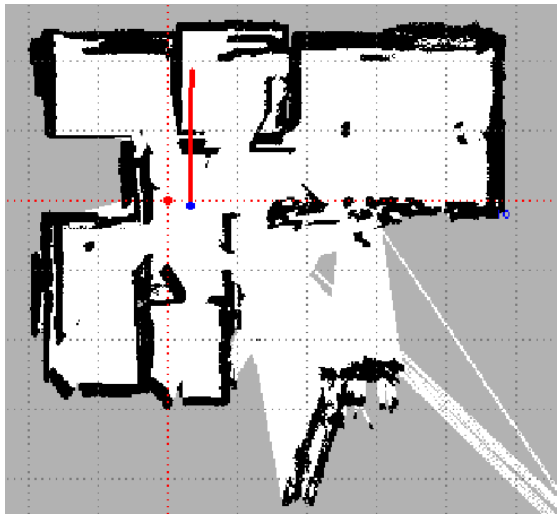

Figure 7: Result of map reconstruction in the IMAV competition fly-off

implemented this algorithm to the customized small scale MAV has won the competition after completing the tasks autonomously.

6 Conclusions

An effective method to localize and navigate an MAV in unknown indoor environment is proposed in this manuscript. The proposed algorithm requires much lesser computational intensity compared to current state-of-art technologies. It is formulated and implemented onboard in a quadrotor MAV with a dimension lesser than 90 cm and weight lower than 3 kg. The algorithm is running onboard in the MAV in real-time. Flight results have shown that the algorithm works in different indoor environment, including solving the mission element of IMAV 2014 indoor tasks.

REFERENCES

[1] D. Mellinger, N. Michael, M. Shomin, and V. Kumar. Recent advances in quadrotor capabilities. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2964–2965, Shanghai, China, 2011.

[2] A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with an application in autonomous mine mapping. In *IEEE International Conference on Robotics and Automation*, 2004.

[3] Y. Zhuang, K. Wang, W. Wang, and H. Hu. A hybrid sensing approach to mobile robot localization in complex indoor environments. *International Journal of Robotics and Automation*, 27(2):198–205, 2012.

[4] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *IEEE International Conference on Robotics and Automation*, pages 2878–2883, 2009.

[5] B. Ben Moshe, N. Shvalb, J. Baadani, I. Nagar, and H. Levy. Indoor positioning and navigaton for micro UAV drones - work in progress. In *IEEE 27th Convention of Electrical & Electronics Engineerings*, Israel, 2012.

[6] F. Wang, K. Wang, S. Lai, S. K. Phang, B. M. Chen, and T. H. Lee. An efficient uav navigation solution for confined but partially known indoor environments. In *11th IEEE International Conference on Control and Automation*, pages 1351–1356, Taichung, Taiwan, 2014.

[7] F. Wang, J. Q. Cui, S. K. Phang, B. M. Chen, and T. H. Lee. A mono-camera and scanning laser ranger finder based uav indoor navigation system. In *2013 International conference on Unmanned Aircraft Systems*, pages 693–700, Atlanta, US, 2013.

[8] G. Borges and M. Aldon. A split-and-merge segmentation algorithm for line extraction in 2D range images. In *15th International Conference on Pattern Recognition*, 2000.

[9] J. Zhang and S. Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems X*, Rome, Italy, 2014.

[10] S. K. Phang, K. Li, K. H. Yu, B. M. Chen, and T. H. Lee. Systematic design and implementation of a micro unmanned quadrotor system. *Unmanned Systems*, 2(2):121–141, 2014.

[11] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys. Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 5(1-2):21–39, 2012.

[12] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys. Pixhawk: A system for autonomous flight using onboard computer vision. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2992–2997, Shanghai, China, 2011.

[13] G. Cai, B. M. Chen, and T. H. Lee. Unmanned rotorcraft systems. Springer, New York, 2011.

[14] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2520–2525, Shanghai, China, 2011.

[15] B. M. Chen. H infinity control and its applications. Springer, New York, 1998.