International Journal of Control, Automation and Systems 18(X) (2020) 1-16
http://dx.doi.org/10.1007/s12555-019-0797-7
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Collision-free Autonomous Navigation of A Small UAV Using Low-cost Sensors in GPS-denied Environments

Wonkeun Youn, Hayoon Ko, Hyungsik Choi, Inho Choi, Joong-Hwan Baek, and Hyun Myung*

Abstract: This paper proposes a novel complete navigation system for autonomous flight of small unmanned aerial vehicles (UAVs) in GPS-denied environments. The hardware platform used to test the proposed algorithm is a small, custom-built UAV platform equipped with an onboard computer, RGB-D camera, 2D light detection and ranging (LiDAR), and altimeter. The error-state Kalman filter (ESKF) based on the dynamic model for low-cost IMU-driven systems is proposed, and visual odometry from the RGB-D camera and height measurement from the altimeter are fed into the measurement update process of the ESKF. The pose output of the ESKF is then integrated into the open-source simultaneous location and mapping (SLAM) algorithm for pose-graph optimization and loop closing. In addition, the computationally efficient collision-free path planning algorithm is proposed and verified through simulations. The software modules run onboard in real time with limited onboard computational capability. The indoor flight experiment demonstrates that the proposed system for small UAVs with low-cost devices can navigate without collision in fully autonomous missions while establishing accurate surrounding maps.

Keywords: Error state Kalman filter, navigation, path planning, simultaneous localization and mapping (SLAM), unmanned aerial vehicle (UAV).

1. INTRODUCTION

In recent years, small unmanned aerial vehicles (UAVs) such as multicopters have received much attention for various applications (e.g., search and rescue operations [1], agricultural drones [2], and industrial inspections [3]). A MEMS-based inertial measurement unit (IMU) is typically composed of a three-axis accelerometer, a three-axis gyroscope, and a three-axis magnetometer, and such devices become an attractive solution for small UAVs due to their light weight, compact size, low power consumption, and low cost [4, 5]. An inertial navigation system (INS) alone can provide a reliable state estimation solution for a short period of time, but the INS error will accumulate due to the mathematical integration of the INS signal [6]. Specifically, MEMS-based IMUs inevitably contain time-varying bias, scaling and cross-coupling errors, and random noise to a certain extent, resulting in error accumulation [7]. As the typical solution, GPS that provides an absolute position/velocity update is integrated to prevent the solution of the INS from drifting [8].

However, challenging applications such as search/rescue operations require UAVs to navigate GPS-degraded or GPS-denied environments such as downtown areas, forests, caves, and indoors [9]. Such a situation causes the severe degradation and the complete blockage of the GPS signal. To address these issues, alternative INS-aiding solutions for the position/velocity updates have been presented based on light detection and ranging (LiDAR) [10], radio detection and ranging (RADAR) [11], and cameras [9]. Among them, LiDAR-based SLAM of UAVs is the popular approach, but it is not suitable for small UAVs since the sensor is generally large and heavy and has high power consumption and high computational load. For instance, 3D point-cloud data of a LiDAR could include half million points within a few seconds, which is a heavy computational burden with the limited onboard computational capability of small UAVs [12].

Similarly, RADAR also can be utilized as the alternative aiding source during GPS-degraded situations. A RADAR

Manuscript received September 22, 2019; revised February 10, 2020 and March 15, 2020; accepted May 6, 2020. Recommended by Associate Editor Shihua Li under the direction of Editor Myo Taeg Lim. This research was supported in part by Study on the Core Technologies of Electric Vertical Take-Off & Landing Aircraft (FR20A00) through National Research Council of Science & Technology and in part by the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Science and ICT for First-Mover Program for Accelerating Disruptive Technology Development (NRF-2018M3C1B9088328).

Wonkeun Youn, Hyungsik Choi, and Inho Choi are with the UAV System Division, Aeronautics Research and Development Head Office, Korea Aerospace Research Institute, Yuseong-gu, Daejeon 34133, Korea (e-mails: {wkyoun, chs, inho}@kari.re.kr). Wonkeun Youn and Hyun Myung are with the School of Electrical Engineering, KI-AI, KI-R, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea (e-mails: {won13y, hmyung}@kaist.ac.kr). Hayoon Ko and Joong-Hwan Baek are with the School of Electronics and Information Engineering, Korea Aerospace University, 76 Hanggongdaehang-ro, Hwajeon-dong, Deokyang-gu, Goyang-si, Gyeonggi-do, Korea (e-mails: hayoon.ko@lignex1.com, jhbaek@kau.ac.kr).
* Corresponding author.

odometry algorithm based on global nearest neighbor (GNN) has been proposed [13]. Additionally, robust ego-motion estimation based on recursive random sample consensus (RANSAC) to eliminate the negative effect of outliers has been successfully implemented [11]. Similar to LiDAR, RADAR has drawbacks for small UAVs due to its weight, size, power consumption, and cost [12]; thus, it is not suitable for small UAVs. Lastly, the motion capture system (Mocaps) can provide highly accurate pose data with low latency, and it has been widely used for visual servoing [14] tracking [15], and UAV swarms [16]. However, the motion capture system has specific requirements for the space it is operated in, depending on the camera view field, which limits the applications of UAVs.

In summary, a variety of methods are used to handle the state estimation of UAVs in indoor environments, regardless of sensors and approaches. For small UAVs to safely navigate GPS-denied environments, reliable state estimation, real-time obstacle detection and efficient collision-free path planning are required. We focus mainly on the computational efficiency of the algorithm due to the limited payload and computational power of small UAVs, as the entire system needs to run onboard and in real time. In addition, we concentrate on the fully autonomous system without the aid of external installation of sensors such as Mocaps or UWB, which require specific hardware and software programs to obtain and process data. Thus, vision-based navigation is chosen due to its light weight, compact size, and low power consumption, although it may be affected by the light condition and featureless areas [17, 18].

We present a complete system that enables autonomous navigation in indoor environments and describe a navigation system that includes localization, mapping, and planning. In this paper, our main concern is the development of a localization filter based on the INS dynamic model and a computationally efficient collision-free path planning algorithm; hence, the SLAM algorithm, including the visual odometry algorithm, is adopted from the well-known RTABMap open-source algorithm [19].

The main contributions of this paper are as follows:

1) The complete software algorithm of localization, mapping, and perception in a GPS-denied indoor environment has been developed based on the low-cost multisensor fusion of sensors such as IMUs, RGB-D sensors, and altimeters.
2) The novel efficient collision-free path planning algorithm is proposed and verified through iterative simulations.
3) The proposed algorithm is experimentally verified via simulation and in a GPS-denied indoor environment.

The paper is organized as follows: The error-state Kalman filter (ESKF), including sensor modeling and system dynamics, is presented in Section 2. In Section 3, the collision-free path planning algorithm and the simulation results are presented. In Section 4, the experimental results and discussion are presented. Conclusions and future work are presented in Section 5.

2. ERROR STATE KALMAN FILTER FOR QUATERNION DYNAMICS

Arbitrary orientation can be represented by a unit quaternion as follows [20]:

$$\mathbf{q} = q_w + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k} = \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \mathbf{e}_x \sin(\theta/2) \\ \mathbf{e}_y \sin(\theta/2) \\ \mathbf{e}_z \sin(\theta/2) \end{bmatrix},$$

where q_w is referred to as the real part and $\mathbf{q}_v = q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k}$ is the vector part of the quaternion. θ is the rotation angle, and $\bar{\mathbf{e}} = [\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z]^T$ is the axis of rotation. Note that Hamilton's quaternion convention rather than the JPL quaternion convention is used in this paper [21].

The successive quaternion product operation is defined as

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - \mathbf{p}_v{}^T \mathbf{q}_v \\ p_w \mathbf{q}_v + q_w \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \end{bmatrix},$$
$$\mathbf{q} \otimes \mathbf{q}^{-1} = [1, 0, 0, 0]^T,$$

where $\otimes$ refers to quaternion multiplication. $\mathbf{q}^{-1} = [q_w, -\mathbf{q}_v]^T$ is the inverse quaternion of $\mathbf{q}$.

The quaternion must satisfy the norm constraint, which is given as follows:

$$|\mathbf{q}| = \sqrt{\mathbf{q} \otimes \mathbf{q}^{-1}} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} = 1. \tag{1}$$

The reference frame of the strap-down IMUs is termed the body-fixed frame $\{b\}$. The navigation frame is denoted as $\{n\}$. In this paper, the local NED (north-east-down) frame is used as a navigation frame [8]. Hence, an arbitrary 3-dimensional vector on the body-fixed frame $\mathbf{v}_b$ can be transformed into the vector on the navigation frame $\mathbf{v}_n$ as follows [8]:

$$\mathbf{v}_n = \mathbf{R}_b^n(\mathbf{q})\mathbf{v}_b,$$

where the rotation matrix $\mathbf{R}_b^n(\mathbf{q}) \in$ special orthogonal group (SO(3)) denotes the relative orientation with respect to the navigation frame as follows:

$$\mathbf{R}_b^n(\mathbf{q}) = \begin{bmatrix} {q_w}^2 + {q_x}^2 - {q_y}^2 - {q_z}^2 & 2(q_x \cdot q_y - q_w \cdot q_z) & 2(q_x \cdot q_z - q_w \cdot q_y) \\ 2(q_x \cdot q_y + q_w \cdot q_z) & {q_w}^2 - {q_x}^2 + {q_y}^2 - {q_z}^2 & 2(q_y \cdot q_z - q_w \cdot q_x) \\ 2(q_x \cdot q_z - q_w \cdot q_y) & 2(q_y \cdot q_z + q_w \cdot q_x) & {q_w}^2 - {q_x}^2 - {q_y}^2 - {q_z}^2 \end{bmatrix}.$$

The conversion between Euler angle $[\phi,\ \theta,\ \psi]^T$ and the corresponding quaternion can be summarized as follows:

$$\mathbf{q} = \begin{bmatrix} \begin{pmatrix}\cos(\phi/2)\cos(\theta/2)\cos(\psi/2)\\ +\sin(\phi/2)\sin(\theta/2)\sin(\psi/2)\end{pmatrix} \\ \begin{pmatrix}\sin(\phi/2)\cos(\theta/2)\cos(\psi/2)\\ -\cos(\phi/2)\sin(\theta/2)\sin(\psi/2)\end{pmatrix} \\ \begin{pmatrix}\cos(\phi/2)\sin(\theta/2)\cos(\psi/2)\\ +\sin(\phi/2)\cos(\theta/2)\sin(\psi/2)\end{pmatrix} \\ \begin{pmatrix}\cos(\phi/2)\cos(\theta/2)\sin(\psi/2)\\ -\sin(\phi/2)\sin(\theta/2)\cos(\psi/2)\end{pmatrix} \end{bmatrix},$$

$$\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix} = \begin{bmatrix} \text{atan2}\left(\frac{2(q_w q_x+q_y q_z)}{1-2(q_x^2+q_y^2)}\right) \\ \text{asin}\left(2(q_w q_y - q_z q_x)\right) \\ \text{atan2}\left(\frac{2(q_w q_z+q_x q_y)}{1-2(q_y^2+q_z^2)}\right) \end{bmatrix}.$$

2.1. Problem formulation

Consider the following discrete nonlinear system and measurement model:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k),$$
$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k),$$

where k is the time index, $\mathbf{x}_k$ the state, $\mathbf{u}_k$ the input, and $\mathbf{z}_k$ the measurement, $f(\cdot)$ and $h(\cdot)$, the system and measurement model, respectively. The system and measurement noise are generally considered to be white Gaussian noise, such that $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$ and $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$. Here, $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ refers to a normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

Instead of the direct quaternion-based Kalman filter, the ESKF approach has several distinctive merits: the error state system is always relatively small and operates near the origin, thus preventing possible parameter singularities and gimbal lock issues while guaranteeing that the linearization validity holds at all times [22]. Moreover, the ESKF approach guarantees numerical stability and handles the quaternion in its minimal representation [23]. Thus, ESKF based on the IMU dynamic model is proposed as the basic navigation filter in this paper. Please see [24] for details of the conventional indirect Kalman filter.

The main principle of ESKF is to regard the true state ($\mathbf{x}_{t,k}$) as a composition of the nominal state ($\mathbf{x}_k$) and error state ($\delta\mathbf{x}_k$), where the error state denotes the difference between the nominal state and the true state, thus yielding the following relationship [25]:

$$\mathbf{x}_{t,k} = \mathbf{x}_k \oplus \delta\mathbf{x}_k, \tag{2}$$

where $\oplus$ denotes a typical addition operation for all state variables $\in \mathbb{R}^3$ except for the quaternion state. For the quaternion state, the error state is defined in the local frame as follows:

$$\mathbf{q}_{t,k} = \mathbf{q}_k \otimes \delta\mathbf{q}_k, \tag{3}$$

where $\delta\mathbf{q}_k = [1, \delta\boldsymbol{\theta}_k/2]^T$ is the orientation error in SO(3) expressed as a unit quaternion [26]. The orientation error state $\delta\mathbf{q}_k$ is minimal, having the same number of parameters as degrees of freedom. This feature prevents the problems associated with over-parametrization and reduces the risk of singularity of the associated covariance matrix resulting from the quaternion norm constraint [27].

In the following subsections, the state of the ESKF for attitude estimation is denoted as a 16-element true state vector ($\mathbf{x}_{t,k}$), a 16-element nominal state vector ($\mathbf{x}_k$), and a 15-element error state vector ($\delta\mathbf{x}$) as follows:

$$\mathbf{x}_{t,k} = [\mathbf{q}_{t,k}\ \mathbf{v}_{t,k}\ \mathbf{p}_{t,k}\ \boldsymbol{\omega}_{bt,k}\ \mathbf{a}_{bt,k}]^T \in \mathbb{R}^{16},$$
$$\mathbf{x}_k = [\mathbf{q}_k\ \mathbf{v}_k\ \mathbf{p}_k\ \boldsymbol{\omega}_{b,k}\ \mathbf{a}_{b,k}]^T \in \mathbb{R}^{16},$$
$$\delta\mathbf{x}_k = [\delta\boldsymbol{\theta}_k\ \delta\mathbf{v}_k\ \delta\mathbf{p}_k\ \delta\boldsymbol{\omega}_{b,k}\ \delta\mathbf{a}_{b,k}]^T \in \mathbb{R}^{15},$$

where $\mathbf{q}_{t,k}$, $\mathbf{v}_{t,k}$, and $\mathbf{p}_{t,k}$ are the true quaternion, velocity, and position expressed in the navigation frame, and $\boldsymbol{\omega}_{bt,k}$ and $\mathbf{a}_{bt,k}$ are true gyroscope and accelerometer biases in the body frame, respectively.

2.2. Sensor modeling

The typical IMUs measure the three-axis accelerations, three-axis angular rates, and three-axis magnetic fields with respect to the body-fixed frame. The measurements given by the IMUs are corrupted by white Gaussian noise and slowly varying bias terms. The following model represents the relationship between the measured IMU signals and the true signals. Notably, the effects of misalignment and scale factor error of IMUs are neglected since the effect of these parameters is minimized through calibration.

2.2.1 Gyroscope

A three-axis gyroscope measures the angular rate about each of the three axes as follows:

$$\boldsymbol{\omega}_{m,k} = \boldsymbol{\omega}_{t,k} + \boldsymbol{\omega}_{bt,k} + \boldsymbol{\omega}_{n,k}, \tag{4}$$

where $\boldsymbol{\omega}_{m,k} \in \mathbb{R}^3$ is the measured angular rate signal, $\boldsymbol{\omega}_{t,k} \in \mathbb{R}^3$ the true angular rate signal, $\boldsymbol{\omega}_{bt,k} \in \mathbb{R}^3$ the slowly varying bias term of the gyroscope, and $\boldsymbol{\omega}_{n,k} \in \mathbb{R}^3$ zero-mean Gaussian noise.

2.2.2 Accelerometer

The accelerometer measures the specific force as follows [28]:

$$\mathbf{a}_{m,k} = \mathbf{R}_n^b(\mathbf{q}_k)(\mathbf{a}_{t,k} - \mathbf{g}) + \mathbf{a}_{bt,k} + \mathbf{a}_{n,k}, \tag{5}$$

where $\mathbf{a}_{m,k} \in \mathbb{R}^3$ is the measured acceleration, $\mathbf{a}_{t,k} \in \mathbb{R}^3$ the true acceleration signal, $\mathbf{a}_{bt,k} \in \mathbb{R}^3$ the slowly varying accelerometer bias term, $\mathbf{a}_{n,k}$ zero-mean Gaussian noise, and $\mathbf{g}$ the gravitational acceleration in the navigation frame.

2.2.3 Magnetometer

The magnetometer measures magnetic fields as follows [29]:

$$\begin{aligned}\mathbf{m}_{m,k} &= \mathbf{R}_n^b(\mathbf{q}_k)\cdot\mathbf{B} + \mathbf{m}_{n,k} \\ &= \mathbf{R}_n^b(\mathbf{q}_k)\cdot\begin{bmatrix} |\mathbf{B}|\cos(\alpha_k)\cos(\beta_k) \\ |\mathbf{B}|\cos(\alpha_k)\sin(\beta_k) \\ |\mathbf{B}|\sin(\alpha_k)\end{bmatrix} + \mathbf{m}_{n,k}, \quad (6)\end{aligned}$$

where $\mathbf{m}_m \in \mathbb{R}^3$ is the measured magnetic field, $\mathbf{m}_n \sim N(0,\sigma_{\mathbf{B}}^2)$ is zero-mean Gaussian noise, $\mathbf{B}$ is the magnetic field vector of the Earth expressed in the navigation frame, $|\mathbf{B}|$ is the magnitude of the magnetic flux density, α_k is the inclination angle of the Earth's magnetic field, and β_k is the declination angle. $|\mathbf{B}|$, α_k, and β_k vary depending on the geodetic position, and time and can be computed from the database of the World Magnetic Model (WMM) [8].

Magnetic field measurements have primarily been utilized for extracting yaw angle information, but the Earth's magnetic field has been widely utilized for various applications, including human body motion tracking [29], spacecraft attitude estimation [30], UAV attitude estimation [31,32], and indoor SLAM [33,34].

2.3. System dynamics

In the ESKF formulations, true state, nominal state, and error state dynamics are addressed. The true system dynamics, including noise, can be represented as follows [35]:

$$\begin{aligned}\mathbf{x}_{t,k} &= \begin{bmatrix}\mathbf{q}_{t,k}\\ \mathbf{v}_{t,k}\\ \mathbf{p}_{t,k}\\ \boldsymbol{\omega}_{bt,k}\\ \mathbf{a}_{bt,k}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{q}_{t,k-1}\otimes\mathbf{q}\{(\boldsymbol{\omega}_{m,k}-\boldsymbol{\omega}_{bt,k-1}-\boldsymbol{\omega}_{n,k-1})\Delta t\} \\ \mathbf{v}_{t,k-1}+\left(\mathbf{R}^n_{b,t,k-1}(\mathbf{a}_{m,k}-\mathbf{a}_{bt,k-1}-\mathbf{a}_{n,k-1})+\mathbf{g}\right)\Delta t \\ \mathbf{p}_{t,k-1}+\mathbf{v}_{t,k-1}\Delta t \\ +\frac{1}{2}\left(\mathbf{R}^n_{b,t,k-1}(\mathbf{a}_{m,k}-\mathbf{a}_{bt,k-1}-\mathbf{a}_{n,k-1})+\mathbf{g}\right)\Delta t^2 \\ \boldsymbol{\omega}_{bt,k-1} \\ \mathbf{a}_{bt,k-1}\end{bmatrix}, \quad (7)\end{aligned}$$

where $\mathrm{q}\{\omega\Delta t\} = \begin{bmatrix}\cos(\|\omega\|\Delta t/2)\\ \frac{\omega}{\|\omega\|}\sin(\|\omega\|\Delta t/2)\end{bmatrix}$.

The nominal dynamics refer to the system dynamics without considering noise or perturbations, and they can be represented as follows:

$$\begin{aligned}\mathbf{x}_k &= \begin{bmatrix}\mathbf{q}_k\\ \mathbf{v}_k\\ \mathbf{p}_k\\ \boldsymbol{\omega}_{b,k}\\ \mathbf{a}_{b,k}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{q}_{k-1}\otimes\mathbf{q}\{(\boldsymbol{\omega}_{m,k}-\boldsymbol{\omega}_{b,k-1})\Delta t\} \\ \mathbf{v}_{t,k-1}+\left(\mathbf{R}^n_{b,k-1}(\mathbf{a}_{m,k}-\mathbf{a}_{b,k-1})+\mathbf{g}\right)\Delta t \\ \mathbf{p}_{k-1}+\mathbf{v}_{k-1}\Delta t \\ +\frac{1}{2}\left(\mathbf{R}^n_{b,k-1}(\mathbf{a}_{m,k}-\mathbf{a}_{b,k-1})+\mathbf{g}\right)\Delta t^2 \\ \boldsymbol{\omega}_{b,k-1} \\ \mathbf{a}_{b,k-1}\end{bmatrix}. \quad (8)\end{aligned}$$

To obtain the linearized equation of error dynamics, each true state equation in (7) is reformulated as its composition of nominal and error states in (8) and simplified up to all second-order infinitesimals, yielding the following error state dynamics:

$$\begin{aligned}\delta\mathbf{x}_k &= \begin{bmatrix}\delta\boldsymbol{\theta}_k\\ \delta\mathbf{v}_k\\ \delta\mathbf{p}_k\\ \delta\boldsymbol{\omega}_{b,k}\\ \delta\mathbf{a}_{b,k}\end{bmatrix} \quad (9)\\ &= \begin{bmatrix}\delta\boldsymbol{\theta}_{k-1}+\mathbf{R}^n_{b,k-1}\{(\boldsymbol{\omega}_m-\boldsymbol{\omega}_{b,k-1})\Delta t\}\,\delta\boldsymbol{\theta}_{k-1} \\ -\delta\boldsymbol{\omega}_{b,k-1}\Delta t+\theta_i \\ \delta\mathbf{v}_{k-1}+\left(-\mathbf{R}^n_{b,k-1}[\mathbf{a}_{m,k}-\mathbf{a}_{b,k-1}]_\times\delta\boldsymbol{\theta}_{k-1}\right. \\ \left.-\mathbf{R}^n_{b,k-1}\delta\mathbf{a}_{b,k-1}\right)\Delta t+\mathbf{v}_i \\ \delta\mathbf{p}_{k-1}+\delta\mathbf{v}_{k-1}\Delta t \\ \delta\boldsymbol{\omega}_{b,k-1}+\boldsymbol{\omega}_i \\ \delta\mathbf{a}_{b,k-1}+\mathbf{a}_i\end{bmatrix}.\end{aligned}$$

where θ_i, $\mathbf{v}_i$, $\boldsymbol{\omega}_i$, and $\mathbf{a}_i$ are the random impulses to orientation, velocity, and IMU biases, respectively. Their covariance can be computed by integrating the covariances of $\boldsymbol{\omega}_n$, $\mathbf{a}_n$, $\boldsymbol{\omega}_\omega$, and $\mathbf{a}_\omega$ during Δt as follows:

$$\begin{aligned}\Upsilon_i &= \sigma^2_{\boldsymbol{\omega}_n}\Delta t^2\,\mathrm{I}\ \left[\mathrm{rad}^2\right], \\ \mathbf{V}_i &= \sigma^2_{\mathbf{a}_n}\Delta t^2\,\mathrm{I}\ \left[\mathrm{m}^2/\mathrm{s}^2\right], \\ \Omega_i &= \sigma^2_{\boldsymbol{\omega}_\omega}\Delta t\,\mathrm{I}\ \left[\mathrm{rad}^2/\mathrm{s}^2\right], \\ \Psi_i &= \sigma^2_{\mathbf{a}_\omega}\Delta t\,\mathrm{I}\ \left[\mathrm{m}^2/\mathrm{s}^4\right],\end{aligned}$$

where $\sigma_{\boldsymbol{\omega}_n}$ [rad/s], $\sigma_{\mathbf{a}_n}$ [m/s^2], $\sigma_{\boldsymbol{\omega}_\omega}$ [rad/s/$\sqrt{\mathrm{s}}$], and $\sigma_{\mathbf{a}_\omega}$ [m/s^2/$\sqrt{\mathrm{s}}$] are to be determined from the information in the IMU specification sheets or the Allen variance analysis [36]. $[\mathbf{a}]_\times$ denotes the skew-symmetric matrix as follows:

$$[\mathbf{a}]_\times = \begin{bmatrix}0 & -a_z & a_y\\ a_z & 0 & -a_x\\ -a_y & a_x & 0\end{bmatrix}.$$

2.4. Error state Kalman filter formulation

The nominal state does not consider the noise term in (7) and is propagated by integrating the nominal dynamics in (8) as follows:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, 0), \quad (10)$$

where $\mathbf{u}_k = [\mathbf{a}_{m,k}\ \boldsymbol{\omega}_{m,k}]^T$ and superscripts $+$ and $-$ refer to the a posterior and a priori estimates, respectively. The

corresponding nominal state vector dynamics are represented as follows:

$$\hat{\mathbf{x}}_k^- = \begin{bmatrix} \hat{\mathbf{q}}_k^- \\ \hat{\mathbf{v}}_k^- \\ \hat{\mathbf{p}}_k^- \\ \hat{\boldsymbol{\omega}}_{b,k}^- \\ \hat{\mathbf{a}}_{b,k}^- \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{q}}_{k-1}^+ \otimes \mathbf{q}\{(\boldsymbol{\omega}_{m,k} - \boldsymbol{\omega}_{b,k})\Delta t\} \\ \hat{\mathbf{v}}_{k-1}^+ + (\mathbf{R}_b^n(\mathbf{a}_{m,k} - \mathbf{a}_{b,k}) + \mathbf{g})\Delta t \\ \hat{\mathbf{p}}_{k-1}^+ + \hat{\mathbf{v}}_k^- \Delta t \\ \hat{\boldsymbol{\omega}}_{b,k-1}^+ \\ \hat{\mathbf{a}}_{b,k-1}^+ \end{bmatrix}, \tag{11}$$

where Δt is the sampling time interval of the IMU which is almost constant. Linear evolution of $\boldsymbol{\omega}$ is assumed during Δt; then, the first-order quaternion integrator is implemented as follows [23]:

$$\hat{\mathbf{q}}_k^- = \hat{\mathbf{q}}_{k-1}^+ \otimes \left(\mathbf{q}(\bar{\boldsymbol{\omega}}_k \Delta t) + \frac{\Delta t^2}{24} \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{k-1} \times \boldsymbol{\omega}_k \end{bmatrix} \right),$$

where $\bar{\boldsymbol{\omega}}_k = 0.5(\boldsymbol{\omega}_k + \boldsymbol{\omega}_{k-1})$ is the average angular rate during time steps $k-1$ and k.

Similarly, rearranging the error state dynamics in (9) yields the following:

$$\delta\hat{\mathbf{x}}_k^- = \mathbf{F}\,\delta\hat{\mathbf{x}}_{k-1}^+, \tag{12}$$

where $\mathbf{F}$ is the transition matrix, and it can be obtained by Euler integration of (9) with respect to the error state $\delta\mathbf{x}$ as follows:

$$\mathbf{F} \cong \mathbf{I} + \mathbf{A}\Delta t,$$

where $\mathbf{F}$ is approximated only by the first-order term since the sampling time Δt is extremely small, and

$$\mathbf{A} = \partial f(\mathbf{x}, \cdot)/\partial\delta\mathbf{x} = \begin{bmatrix} \Theta & 0 & 0 & -\mathbf{I} & 0 \\ \mathbf{V} & 0 & 0 & 0 & -\mathbf{R}_b^n \\ 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where

$$\Theta = -[\boldsymbol{\omega}_{m,k} - \boldsymbol{\omega}_{b,k}]_\times, \ \mathbf{V} = -\mathbf{R}_b^n[\mathbf{a}_{m,k} - \mathbf{a}_{b,k}]_\times.$$

Generally, the prediction step (12) of the error state $\delta\hat{\mathbf{x}}_k$ can be neglected because the initial value of the error state $\delta\hat{\mathbf{x}}_k$ is set to zero and returns zero. The prior covariance $\mathbf{P}_k^-$ is updated as follows:

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}^+\mathbf{F}^T + \mathbf{F}_i\mathbf{Q}_i\mathbf{F}_i^T, \tag{13}$$

where $\mathbf{F}_i$ is the perturbation Jacobian of (9) with respect to the perturbation vector i as follows:

$$\mathbf{F}_i = \partial f(x, \cdot)/\partial\mathbf{w} = \begin{bmatrix} -\mathbf{I} & 0 & 0 & 0 \\ 0 & -\mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix},$$

where $\mathbf{w} = [\boldsymbol{\omega}_n \ \mathbf{a}_n \ \boldsymbol{\omega}_\omega \ \mathbf{a}_\omega]^T$ and $\mathbf{Q}_i$ is the covariance matrix of the perturbation impulses given by

$$\mathbf{Q}_i = \begin{bmatrix} \Upsilon_i & 0 & 0 & 0 \\ 0 & \mathbf{V}_i & 0 & 0 \\ 0 & 0 & \Omega_i & 0 \\ 0 & 0 & 0 & \Psi_i \end{bmatrix}.$$

The measurement update of ESKF will be executed whenever valid measurements are available. The Kalman gain $\mathbf{K}_k$ is calculated as follows:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k{}^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k{}^T + \mathbf{R}_k)^{-1}. \tag{14}$$

The Jacobian matrix $\mathbf{H}_k$ must be defined with respect to the error state $\delta\mathbf{x}$ and evaluated at the best true-state estimate $\hat{\mathbf{x}}_{t,k} = \mathbf{x}_k + \delta\hat{\mathbf{x}}_k$. Since the mean of the error state is zero before the measurement update, this results in $\hat{\mathbf{x}}_t = \mathbf{x}$. Then, the nominal state can be utilized as the operating points of the linearization, thus yielding [27]

$$\mathbf{H}_k \triangleq \left.\frac{\partial h}{\partial\delta\mathbf{x}}\right|_{\mathbf{x}} = \left.\frac{\partial h}{\partial\mathbf{x}_t}\right|_{\mathbf{x}} \left.\frac{\partial\mathbf{x}_t}{\partial\delta\mathbf{x}}\right|_{\mathbf{x}} = \mathbf{H}_\mathbf{x}\mathbf{X}_{\delta\mathbf{x}},$$

where

$$\mathbf{X}_{\delta x} = \begin{bmatrix} \mathbf{Q}_{\delta\boldsymbol{\theta}} & 0 \\ 0 & \mathrm{I}_{14\times 15} \end{bmatrix},$$

where

$$\begin{aligned} \mathbf{Q}_{\delta\boldsymbol{\theta}} &= \frac{\partial(\mathbf{q}\otimes\delta\mathbf{q})}{\partial\delta\boldsymbol{\theta}} = \left.\frac{\partial(\mathbf{q}\otimes\delta\mathbf{q})}{\partial\delta\mathbf{q}}\right|_{\mathbf{q}} \left.\frac{\partial\delta\mathbf{q}}{\partial\delta\boldsymbol{\theta}}\right|_{\delta\hat{\boldsymbol{\theta}}=0} \\ &= \left.\frac{\partial([\mathbf{q}]_L\delta\mathbf{q})}{\partial\delta\mathbf{q}}\right|_{\mathbf{q}} \left.\frac{\partial\begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix}}{\partial\delta\boldsymbol{\theta}}\right|_{\delta\hat{\boldsymbol{\theta}}=0} \\ &= [\mathbf{q}]_L\frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned}$$

where $[\mathbf{q}]_L$ denotes the left-quaternion-product matrix as follows:

$$[\mathbf{q}]_L = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix}.$$

2.4.1 Visual odometry

Visual odometry provides the pose data, including the position and attitude, but position information is utilized only as follows:

$$\mathbf{z}_{v,k} = \mathbf{p}(1:2) + \mathbf{n}_v, \tag{15}$$

where $\mathbf{p}(1:2) = [p_n, \ p_e]^T$ and $\mathbf{n}_v \sim N(0, \boldsymbol{\sigma}_v)$ is the measurement noise assumed to be Gaussian.

2.4.2 Altimeter

The altimeter mounted under a UAV, facing down, provides the distance to the ground that bounces its signal. The altitude measurement can be expressed as follows:

$$\mathbf{z}_{a,k} = \frac{\mathbf{p}(3)}{\mathbf{R}_b^n(3,3)} + \mathbf{n}_a, \tag{16}$$

where $\mathbf{p}(3) = p_d$ and $\mathbf{n}_a \sim N(0, \boldsymbol{\sigma}_a)$ is the measurement noise assumed to be Gaussian.

A posteriori estimate of error state $\delta\hat{\mathbf{x}}_k^+$ and a posteriori covariance $\mathbf{P}_k^+$ are updated by

$$\delta\hat{\mathbf{x}}_k^+ = \mathbf{K}_k(\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-)), \tag{17}$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-, \tag{18}$$

where $\{\mathbf{z}_{v,k}, \mathbf{z}_{a,k}\} \in \mathbf{z}_k$.

After the completion of the measurement update of ESKF, the nominal state is updated with the a posteriori error state using the following operation:

$$\hat{\mathbf{x}}_k^+ \leftarrow \hat{\mathbf{x}}_k^- \oplus \delta\hat{\mathbf{x}}_k^+, \tag{19}$$

where $\oplus$ denotes a genetic composition operator. For all state variables except quaternion, $\mathbf{p}_t = \mathbf{p} \oplus \delta\mathbf{p} = \mathbf{p} + \delta\mathbf{p}$. For quaternion, $\mathbf{q}_t = \mathbf{q} \oplus \delta\mathbf{q} = \mathbf{q} \otimes \delta\mathbf{q}$.

The update of the corresponding nominal states with a posteriori error state is represented as follows:

$$\hat{\mathbf{x}}_k^+ = \begin{bmatrix} \hat{\mathbf{q}}_k^+ \\ \hat{\mathbf{v}}_k^+ \\ \hat{\mathbf{p}}_k^+ \\ \hat{\boldsymbol{\omega}}_{b,k}^+ \\ \hat{\mathbf{a}}_{b,k}^+ \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{q}}_k^- \otimes q\{\delta\hat{\boldsymbol{\theta}}_k^+\} \\ \hat{\mathbf{v}}_k^- + \delta\hat{\mathbf{v}}_k^+ \\ \hat{\mathbf{p}}_k^- + \delta\hat{\mathbf{p}}_k^+ \\ \hat{\boldsymbol{\omega}}_{b,k}^- + \delta\hat{\boldsymbol{\omega}}_k^+ \\ \hat{\mathbf{a}}_{b,k}^- + \delta\hat{\mathbf{a}}_k^+ \end{bmatrix}. \tag{20}$$

After updating the nominal state with the error state, the mean of the error state $\delta\hat{\mathbf{x}}_k^+$ is reset to zero, and a posteriori covariance needs to be updated according to this modification as follows [25]:

$$\mathbf{P}_k^+ = \mathbf{G}_k\mathbf{P}_k^+\mathbf{G}_k^T, \tag{21}$$

where

$$\mathbf{G}_k = \left.\frac{\partial g}{\partial \delta\mathbf{x}}\right|_{\delta\hat{\mathbf{x}}^+} = \begin{bmatrix} \mathbf{I} - \left[\frac{1}{2}\delta\hat{\boldsymbol{\theta}}_k^+\right]_\times & 0 \\ 0 & \mathbf{I}_{15\times15} \end{bmatrix}.$$

Notably, after the operation of (20), the quaternion normalization constraint, as shown in (1), can be violated due to linearization and arithmetic errors. To prevent such a case, brute-force quaternion normalization by (22) should be executed if the discrepancy from the unit norm is larger than a small fraction of the noise level (i.e., 10^{-7}) [37,38]:

$$\hat{\mathbf{q}}_k^+ = \frac{\hat{\mathbf{q}}_k^+}{|\hat{\mathbf{q}}_k^+|}. \tag{22}$$

3. PATH PLANNING

3.1. The proposed RRT*-GD-smart

The rapidly exploring random trees (RRT) algorithm has been widely implemented for a collision-free path generation since it does not require precise prior information of the surrounding obstacles and instead requires only simple information regarding whether the generated path is collision free or not [39]. However, the convergence rate of the conventional RRT algorithm is relatively slow, and optimality cannot be guaranteed [40].

To overcome these issues, several variants or improvements in the RRT algorithm have been proposed. Among them, the RRT*-Smart algorithm is an improved algorithm that extends the conventional RRT algorithm in terms of computational cost [40]. In the RRT*-Smart algorithm, the path is generated to the target point by modifying the connection to the low-cost node by comparing the distance cost value with the neighboring node when a new node is generated. Therefore, the path to the target point can be generated at a lower cost than the conventional RRT algorithm. In addition, to speed up the conventional RRT algorithm, RRT-GD (goal directionality) was proposed to be goal directional [41].

By combining the advantages of both RRT*-Smart and RRT-GD, we propose RRT*-GD-Smart algorithm. The proposed RRT*-GD-Smart algorithm has the following distinctive merits. An additional sequence is added to determine whether a direct path toward the target point exists whenever the node is generated, providing goal directionality toward the target point. In addition, the proposed RRT*-GD-Smart algorithm generates nodes that have goal directionality with random length, not constant length, preventing unnecessary nodes. As a result, the proposed RRT*-GD-Smart algorithm is computationally efficient with a high convergence rate while possessing the merits of both RRT*-Smart and RRT-GD.

Let $\mathbf{X}$ denote the configuration space, where $\mathbf{x}_{\text{init}}$, $\mathbf{x}_{\text{goal}}$, and $\mathbf{x}_{\text{free}}$ are the initial position, goal position, and obstacle-free region, respectively. The solution of RRT consists of $T = (V, E)$, where vertices $V \in \mathbf{x}_{\text{free}}$ and edge E connects vertices. The aim of RRT is to find out an input u that yields a collision-free path from $\mathbf{x}_{\text{init}}$ to $\mathbf{x}_{\text{goal}}$. In the proposed RRT*-GD-Smart, the random point $\mathbf{x}_{\text{new}}$ is produced as follows:

$$\mathbf{x}_{\text{new}} \leftarrow \mathbf{x}_{\text{near}} + (\mathbf{x}_{\text{rand}} - \mathbf{x}_{\text{near}}) \cdot x_{\text{scale}}, \tag{23}$$

where the probability distribution function (PDF) of x_{scale} is selected as the uniform distribution as follows:

$$p(x_{\text{scale}}) = \begin{cases} \frac{1}{x_{\text{upper}} - x_{\text{lower}}} & \text{for } x_{\text{lower}} \le x_{\text{scale}} \le x_{\text{upper}}, \\ 0 & \text{otherwise}, \end{cases}$$

and $x_{\text{upper}} = 5\,\text{m}$ and $x_{\text{lower}} = 1\,\text{m}$ are the upper and lower bounds, respectively. The principle of generating random point x_{new} is depicted in Fig. 1.

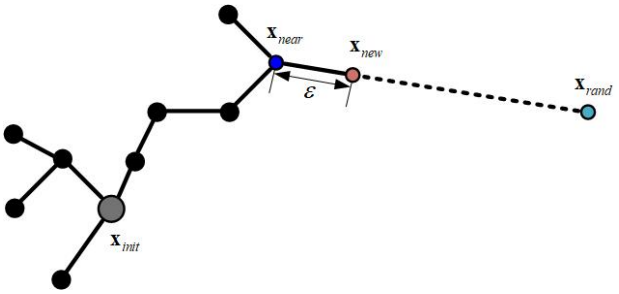


Fig. 1. Generation of random point $\mathbf{x}_{\text{new}}$ with $\varepsilon = (\mathbf{x}_{\text{rand}} - \mathbf{x}_{\text{near}}) \cdot \mathbf{x}_{\text{scale}}$.

Algorithm 1: Pseudo code for the proposed RRT*-GD-Smart

1. $T \leftarrow$ Initialize Tree();
2. $T \leftarrow$ InsertNode($\mathbf{x}_{\text{init}}$, T);
3. **for** $k = 1 : N$
4. $\quad \mathbf{x}_{\text{rand}} \leftarrow$ Sampling();
5. $\quad \mathbf{x}_{\text{nearest}} \leftarrow$ Nearest($\mathbf{x}_{\text{rand}}, T$);
6. $\quad u \leftarrow$ Steer($\mathbf{x}_{\text{rand}}, \mathbf{x}_{\text{nearest}}$);
7. $\quad$ **if GoalDirection**($\mathbf{x}_{\text{init}}, \mathbf{x}_{\text{goal}}, \mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{rand}}$)
8. $\quad\quad \mathbf{x}_{\text{new}} \leftarrow$ NewConf($\mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{rand}}, \Delta l$);
9 $\quad\quad$ **if Obstaclefree**($\mathbf{x}_{\text{new}}$, $\mathbf{x}_{\text{nearest}}$)
10. $\quad\quad\quad \mathbf{x}_{\text{near}} \leftarrow$ Near(T, $\mathbf{x}_{\text{new}}$, $|V|$);
11. $\quad\quad\quad \mathbf{x}_{\text{min}} \leftarrow$ ChooseParent($\mathbf{x}_{\text{near}}, \mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{new}}$);
12. $\quad\quad\quad T \leftarrow$ InsertNode($\mathbf{x}_{\text{min}}, \mathbf{x}_{\text{new}}, T$);
13. $\quad\quad\quad T \leftarrow$ Rewire($T, \mathbf{x}_{\text{near}}, \mathbf{x}_{\text{min}}, \mathbf{x}_{\text{new}}$);
14. $\quad\quad\quad$ **if DirectPath**($\mathbf{x}_{\text{new}}$, $\mathbf{x}_{\text{goal}}$)
15. $\quad\quad\quad\quad$ PathOptimization(T, $\mathbf{x}_{\text{init}}$, $\mathbf{x}_{\text{goal}}$)
16. **Return**: T

The goal directionality criteria of $\mathbf{x}_{\text{nearest}}$ can be expressed as follows:

$$(\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{init}}) \cdot (\mathbf{x}_{\text{rand}} - \mathbf{x}_{\text{nearest}}) > 0. \tag{24}$$

A pseudocode describing the proposed RRT*-GD-Smart is summarized in Algorithm 1. The detailed explanations of the functions defined in Algorithm 1 are listed as follows:

1) InsertNode(x, y, T): This function adds x and y to V and the connection information of x and y to E.
2) Sampling(): This function randomly samples a state $\mathbf{x}_{\text{rand}} \in \mathbf{x}_{\text{free}}$ from the obstacle-free configuration space.
3) Nearest($\mathbf{x}_{\text{rand}}$, T): This function returns the nearest node from $T = (V, E)$ in terms of the cost determined by the distance function from $\mathbf{x}_{\text{rand}}$.
4) Steer(x, y): This function derives the input u from x to y.
5) GoalDirection($\mathbf{x}_{\text{init}}, \mathbf{x}_{\text{goal}}, \mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{rand}}$): This function returns $\mathbf{x}_{\text{nearest}}$ that satisfies the goal directionality condition according to (24).
6) Obstaclefree($\mathbf{x}_{\text{new}}$, $\mathbf{x}_{\text{nearest}}$): This function returns true if the obstacle does not exist along the path between $\mathbf{x}_{\text{new}}$ and $\mathbf{x}_{\text{nearest}}$.
7) NewConf($\mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{rand}}, \Delta l$): This function generates $\mathbf{x}_{\text{new}}$ at a random distance from $\mathbf{x}_{\text{nearest}}$ toward $\mathbf{x}_{\text{rand}}$ according to (23) where $\Delta l = (\mathbf{x}_{\text{rand}} - \mathbf{x}_{\text{near}}) \cdot x_{\text{scale}}$.
8) Near($T, \mathbf{x}_{\text{new}}, |V|$): This function returns vertices at a certain radius (r) from $\mathbf{x}_{\text{new}}$.
9) ChooseParent($\mathbf{x}_{\text{near}}, \mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{new}}$): This function selects the best parent $\mathbf{x}_{\text{new}}$ from the nearby nodes.
10) Rewire($T, \mathbf{x}_{\text{near}}, \mathbf{x}_{\text{min}}, \mathbf{x}_{\text{new}}$): This function checks if the cost to the nodes in $\mathbf{x}_{\text{near}}$ is less through $\mathbf{x}_{\text{new}}$ compared to their previous costs; then, its parent is changed to $\mathbf{x}_{\text{new}}$.
11) DirectPath($\mathbf{x}_{\text{new}}$, $\mathbf{x}_{\text{goal}}$): This function returns true if the direct path between $\mathbf{x}_{\text{new}}$ and $\mathbf{x}_{\text{goal}}$ exists.
12) PathOptimization($T, \mathbf{x}_{\text{init}}, \mathbf{x}_{\text{goal}}$): This function determines an optimized path by directly connecting the nodes in the path.

3.2. Simulation

The software in the loop (SITL) was conducted in the Gazebo simulator to verify the feasibility of the proposed path planning algorithm. In the simulation, a realistic simulation environment including four obstacles with a quadrotor platform equipped with a 2D LiDAR, has been utilized, as depicted in Fig. 2. The arm length, mass, and inertia of the UAV in the simulation are set to the same to match the physical properties of the real custom-made UAV vehicle. The initial state $\mathbf{x}_{\text{init}}$ is set to $[0,\ 0,\ 0]^T$, and the UAV is given a target position. Three different path planning algorithms for the performance comparison are implemented: RRT*-Smart, RRT*-GD, and the proposed RRT*-GD-Smart.

The quantitative performance comparison results of 50 runs are summarized in Table 1. It can be seen that the

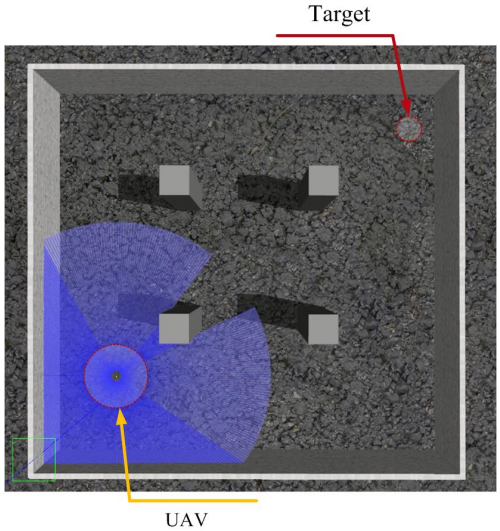


Fig. 2. Simulation environment.

Table 1. Performance comparison between RRT*-Smart, RRT-GD and the proposed RRT*-GD-Smart.

Algorithm	Path Generation Time (sec)	Number of generated nodes	Distance Cost (meter)
RRT*-Smart	6.45	31.2	19.9
RRT-GD	5.37	24.84	18.8
Proposed RRT*-GD-Smart	0.96	11.1	16.3

Table 2. Simulation and experiment parameters.

Parameter	Simulation specifications	Experiment specifications
$\mathbf{x}_{\text{init}}$	$[0,\ 0,\ 0]^T$	$[0,\ 0,\ 0]^T$
$\mathbf{x}_{\text{goal}}$	$[10,\ 10,\ 2]^T$	$[5,\ 1,\ 1]^T$
N	200	100
x_{upper}	5 m	3.5 m
x_{lower}	1 m	1 m

proposed RRT*-GD-Smart algorithm has a faster convergence rate than the RRT*-Smart and RRT-GD algorithms since it divides the length between nodes randomly, with each length ranging from 1 m to 5 m, rather than using a constant length, as shown in (23). Fig. 3 represents the collision-free path generation result of the aforementioned approaches. Specifically, the existing RRT-GD algorithm searches for the collision-free path with constant length, which results in unnecessary nodes. In contrast, the proposed RRT*-GD-Smart algorithm samples the nodes with random length, resulting in fewer nodes than the RRT-GD algorithm. The parameters used for the simulation are summarized in Table 2.

In [40], it has already been verified through the iterative simulations that the optimality of the RRT*-Smart algorithm is guaranteed due to path optimization and intelligent sampling at a much faster rate and a reduced execution time. The proposed RRT*-GD-Smart algorithm has the same optimality as its predecessor RRT*-Smart, and simulation results have demonstrated that the proposed RRT*-GD-Smart algorithm converges to an optimal solution with very few iterations and at an accelerated rate in terms of path generation time, number of generated nodes, and distance cost. A performance comparison successfully demonstrated the efficiency of the proposed RRT*-GD-Smart algorithm in terms of both time and cost.

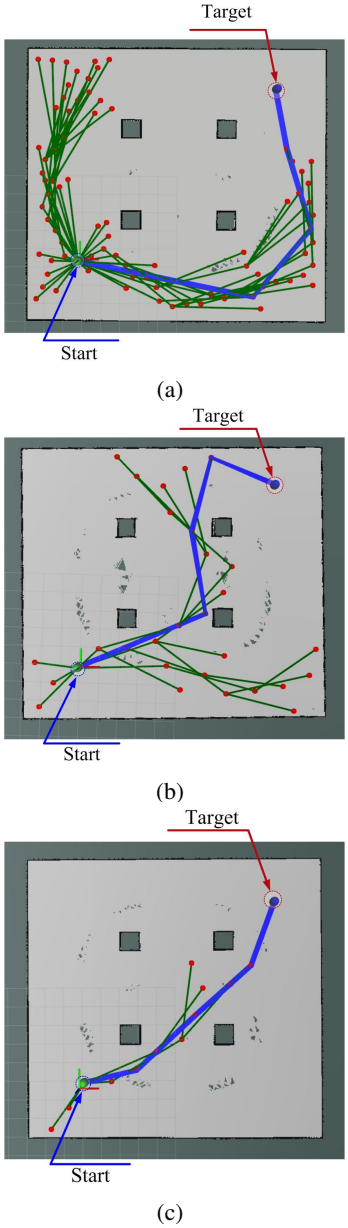


Fig. 3. The path-planning results of (a) the RRT*-Smart algorithm, (b) the RRT-GD algorithm, and (c) the proposed RRT*-GD-Smart algorithm.

4. EXPERIMENT

Fig. 4 shows an indoor experimental setup. We implemented the proposed algorithm on the customized quad-rotor UAV (diameter: 750 mm, height: 258 mm, weight: 3.5 kg), equipped with the 3D RGB-D camera (Astra Pro, Orbbec) and a 2D LiDAR (RPLiDAR A3), as shown in Fig. 5(a). The IMUs of the UAV platform consist of LSM303D-integrated accelerometers/magnetometers, and L3GD20 gyroscopes and an altimeter (LIDAR-Lite v3, Garmin) mounted under a quad-rotor UAV were utilized for the height measurement. The update rates of the visual odometry and altimeter are 10 Hz and 20 Hz, respectively, as shown in Table 3. The parameters used for the experi-

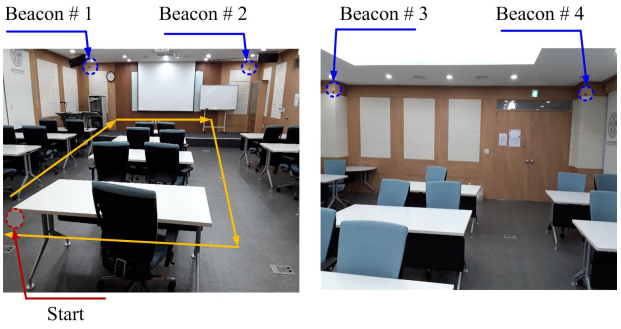


Fig. 4. Indoor experiment setup; four beacons are installed, one at each of the ceiling corners.

Table 3. Sensor specifications.

Sensor	Specifications
Visual odometry	Output rate: 10 Hz, $\sigma_{\mathbf{v}} = 0.2$ m
Altimeter	Output rate: 20 Hz, $\sigma_{\mathbf{a}} = 0.1$ m

ment are summarized in Table 2. One-hour static experiments were conducted to perform the analysis of the Allan variance and the IMUs sensor specification is summarized in Table 4.

Several devices such as a monocular camera, a stereo camera, and a 2D LiDAR have been widely utilized for obstacle detection. Among them, 2D LiDAR is generally insensitive to the light condition of the environment and has high accuracy for the detection of distant obstacles. Thus, the lightweight 2D LiDAR was mounted to generate the occupancy grid map, and the RGB-D camera was utilized only for the visual odometry algorithm. The UAV had an onboard computer (Odroid H2, Hardkernel) with an embedded Intel Celeron J4105 Processor for SLAM and the path-planning algorithm. Note that the total weight of the small UAV platform was approximately 3.5 kg, and the frame diameter and height were 750 mm and 258 mm, respectively. The total cost of the custom-built UAV platform, including sensors, was around 600 US dollars.

The overall software architecture is shown in Fig. 6. The proposed algorithm was implemented in the ROS middleware framework (ROS Kinetic) under Ubuntu 16.04 to facilitate the straight integration between the simulation and the UAV hardware, making it easier to extend in future work. The pose output from ESKF is fed into the RTABMap to build a consistent surrounding map.

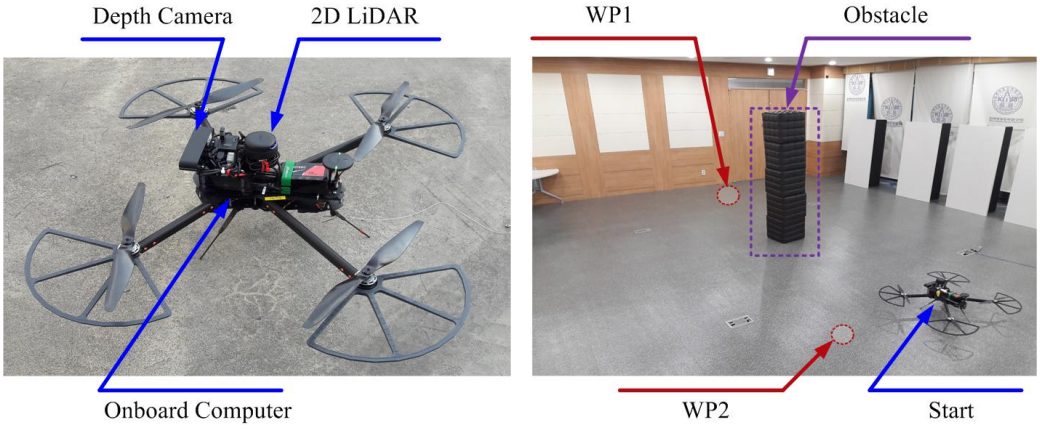


Fig. 5. (a) Custom-built quad-rotor platform that includes a RGB-D camera, a 2D LiDAR, an altimeter, and an onboard computer. (b) Experimental setup for collision avoidance tests. WP1 and WP2 refer to the target waypoints 1 and 2 in a fully autonomous mission.

Table 4. Accelerometer and gyroscope specifications.

Accelerometer	Specification		
	X	Y	Z
Bias instability ($\sigma^2_{\mathbf{a}_n}$ [m/s^2])	$5.8732 \cdot 10^{-4}$	$4.6452 \cdot 10^{-4}$	$9.3451 \cdot 10^{-4}$
Velocity random walk ($\sigma^2_{\mathbf{a}_\omega}$ [m/s$^2/\sqrt{\text{s}}$])	0.0016	0.0016	0.0028
Gyroscope	Specification		
	X	Y	Z
Bias instability ($\sigma^2_{\boldsymbol{\omega}_n}$ [rad/s])	$4.7324 \cdot 10^{-5}$	$4.7234 \cdot 10^{-5}$	$3.6552 \cdot 10^{-5}$
Angular random walk ($\sigma^2_{\boldsymbol{\omega}_\omega}$ [rad/s$/\sqrt{\text{s}}$])	$1.1828 \cdot 10^{-4}$	$1.1694 \cdot 10^{-4}$	$1.1674 \cdot 10^{-4}$

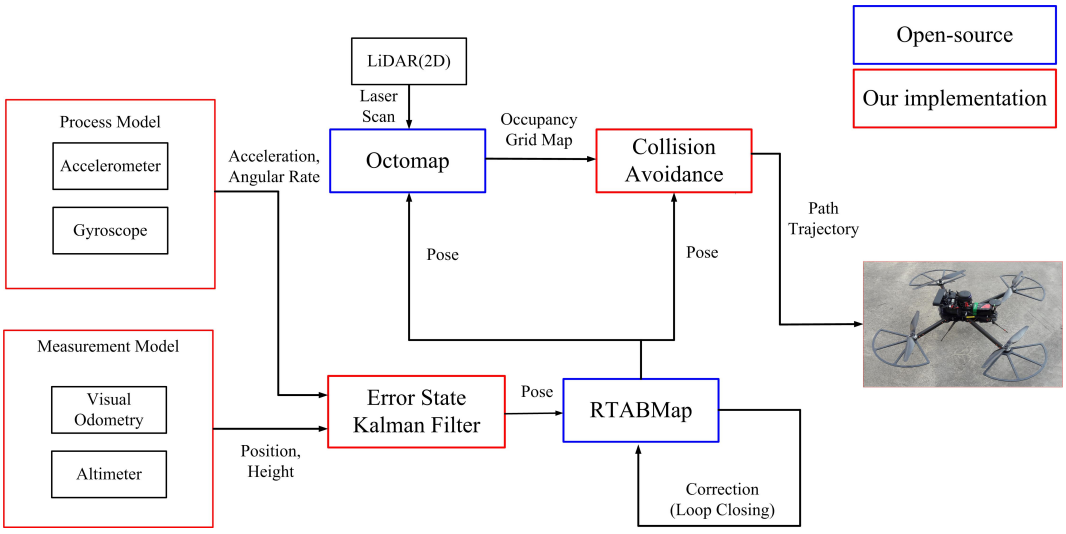


Fig. 6. Overall software architecture.

The RTABMap is a graph-based SLAM algorithm by the appearance-based loop closure detection approach [19]. The visual odometry algorithm from the RGB-D camera is also adopted from the RTABMap. The RTABMap is used because it can perform real-time localization and mapping with limited onboard computational capability [19].

Then, laser scan data from the 2D LiDAR with the corrected pose from the RTABMap are fed into the Octomap algorithm which incrementally establishes the occupancy grid map of the UAV vicinity. Octomap based on octrees that uses probabilistic occupancy estimation is adopted here since it can model arbitrary environments without prior knowledge while keeping the memory usage low [42]. The collision avoidance node then takes the occupancy grid map and the pose as inputs to generate the collision-free path position command to the UAV, as explained in Algorithm 1. Note that the development of the SLAM and visual odometry algorithm is beyond the scope of this paper.

It should be noted that the indoor environment is an area where magnetic field distortion can often occur due to the large number of objects composed of ferromagnetic materials such as tables and desks. Since the magnitude of magnetic disturbances in an indoor environment is typically larger than that of a normal magnetic field, it is desirable to discard the measured magnetic fields in the presence of magnetic field disturbances. The dip angle and magnitude of the magnetic field can be utilized as an indicator to determine whether magnetic disturbances exist [43]. Specifically, the dip angle and magnitude of a magnetic field remain nearly constant when no external magnetic distortion exists [44]. Thus, in this paper, the following two indicators are utilized for detecting a magnetic distortion:

$$\Gamma_1 = \|\mathbf{m}_m - |\mathbf{B}|\| \,/\, |\mathbf{B}|\,,$$

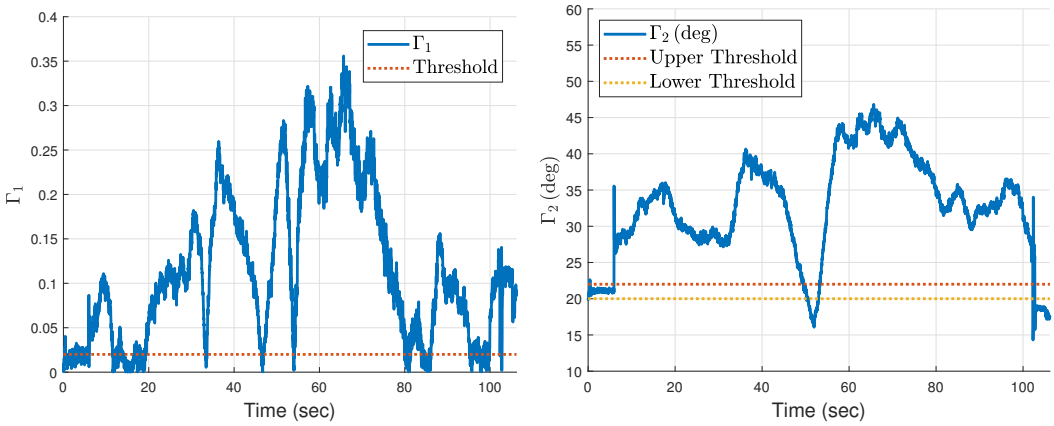
$$\Gamma_2 = \pi/2 - \operatorname{acos}(n_{\mathbf{a}} \cdot n_{\mathbf{m}})\,, \tag{25}$$

where Γ_1 and Γ_2 refer to the dip angle and normalized magnitude discrepancy, respectively, and $n_{\mathbf{a}}$ and $n_{\mathbf{m}}$ refer to the unit vectors of the measured acceleration and magnetic field, respectively.

Figs. 7(a) and (b) represent the dip angle Γ_1 and normalized magnitude discrepancy Γ_2 during the experiment, respectively. It can be seen that Γ_1 and Γ_2 significantly deviate from the threshold obtained from the engineering experience, thus implying that magnetic disturbances are present. These magnetic disturbances may be due to desks and tables made of ferromagnetic materials, as shown Fig. 4. Therefore, the magnetic field measurement is discarded in the presence of the magnetic field disturbance to prevent filter divergence.

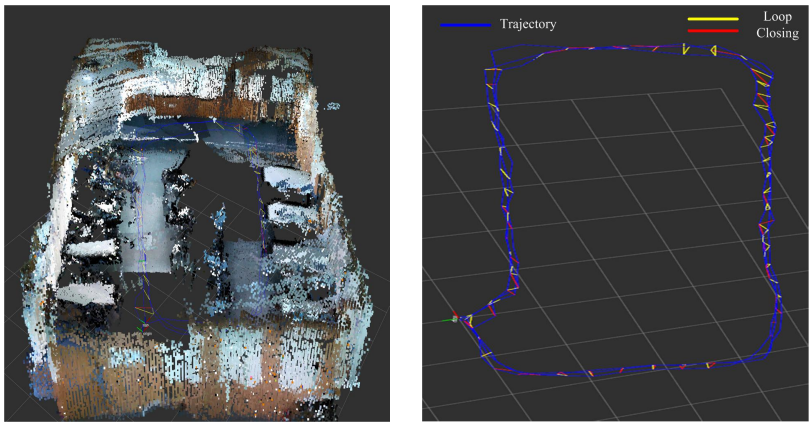
Prior to the collision avoidance experiment, the performance of mapping and localization was analyzed. To record the ground-truth data, a set of ultrasonic beacons (Starter Set HW v4.9, Marvelmind Robotics) was utilized, and the positioning accuracy of this system was reported to be less than 10 cm. Specifically, the UAV position where the mobile beacon was installed was computed using the propagation delay of the ultrasonic signal (Time-of-flight, TOF) for the set of fixed ultrasonic beacons based on triangulation. In this experiment, four stationary ultrasonic beacons are installed at each ceiling corner to have proper line-of-sight ultrasonic coverage, as shown Fig. 4. During the experiment, the UAV follows a rectangular path three times while mapping the surrounding environments using the RGB-D camera.

Fig. 10 depicts the significant drift of the ESKF standalone mode over time, while the position estimate of the proposed algorithm closely follows the ground-truth trajectory. The position estimate of the ESKF is highly reliant on the measurement from the visual odometry, and this



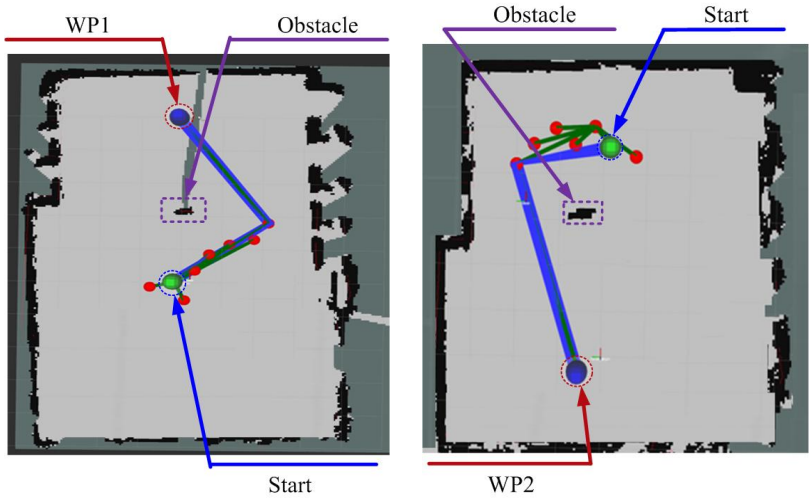
(a) (b)

Fig. 7. (a) dip angle Γ_1 (b) normalized magnitude discrepancy Γ_2 during the indoor flight.



(a) (b)

Fig. 8. (a) 3D mapping result. (b) Navigation trajectory, where the blue line refers to the navigation trajectory of the UAV and the yellow and red lines refer to loop closure detection [19].



(a) (b)

Fig. 9. Collision-free path planning using the proposed RRT*-GD-Smart during a bidirectional flight, where red dots refer to the generated node and the blue line refers to the generated collision-free path: (a) one-way, (b) round-way.

drift is caused by error accumulation of the visual odometry, as visual odometry is based on relative measurements between consecutive times. Specifically, due to the poor matching or errors in 3-D point triangulation of the visual odometry, UAV trajectories tend to drift from the ground truth over time. In contrast, the proper loop closure detection and pose graph optimization by the proposed algorithm significantly reduce this drift and correct for accumulated errors. Figs. 8(a) and 8(b) represent the 3D point-cloud-based mapping result and the estimated navigation trajectory with loop-closing constraint, respectively. It can be seen that accurate and consistent 3D mapping was constructed compared to the real environment, as shown in Fig. 4, and the drift of estimated position was successfully reduced by appearance-based loop closure detection of the RTABMap.

During the indoor collision avoidance experiment, a rectangular obstacle (0.5 m $\times$ 0.5 m $\times$ 2 m) was placed along the trajectory, as depicted in Fig. 5(b). The UAV is given a target position approximately 5 m in front of it. The UAV started in front of the obstacle and is instructed to follow the generated path toward the target position while avoiding the obstacle in fully autonomous mission. It is worth mentioning that the UAV has no prior knowledge of the environment. The flight lasts 110 seconds, and the flying altitude of the UAV was approximately set to 1 m. Fig. 9 shows collision-free path generation during the bidirectional flight test in a fully autonomous mission. After the generation of a few random nodes denoted by red dots, a straight collision-free path is generated due to the goal directionality of the proposed RRT*-GD-Smart algorithm, resulting in fast convergence. In this experiment, the resolution of the Octomap algorithm was set to be relatively high (5 cm of voxel size) to generate a detailed obstacle map. It can be also seen in Figs. 11 and 12 that the UAV was able to successfully follow the collision-free trajectory. For the video of the experiment, please visit https://www.youtube.com/watch?v=mhDPnGNgrUo.

In contrast to [46, 47], which utilize the velocity measurement by an optical flow from a down-looking camera, our system relies on position and altitude measurements determined by a visual odometry algorithm. However, this optical flow is limited to operation in textured environments. In addition, the proposed path planning algorithm is limited to simple yawing motions such as turning left or right [46, 48]. The main contribution of this paper is that our comprehensive system is able to build a consistent surrounding map due to the loop closing and then generate a collision-free path based on the high-resolution grid map obtained from a 2D LiDAR, which is significantly different from [49, 50]. Moreover, different from [51], the optimality of the proposed path planning in terms of both time and cost is first demonstrated though iterative simulations and then further verified through a real-flight experiment.

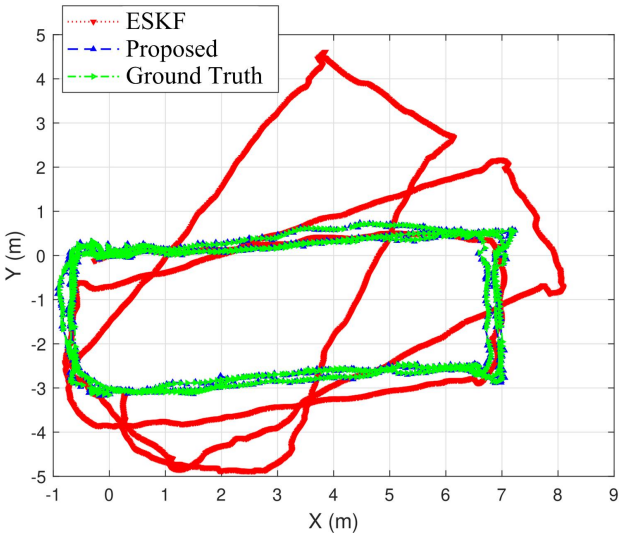


Fig. 10. Position estimate of the ESKF and the proposed algorithm.

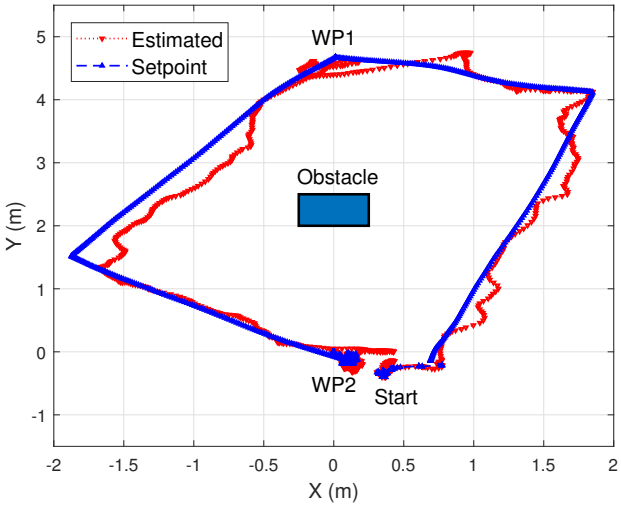


Fig. 11. The set-point trajectory and the corresponding navigation trajectory.

5. CONCLUSION

In this paper, a complete navigation system for an autonomous indoor flying UAV using low-cost sensors was presented. The proposed ESKF filter based on the INS dynamic model is integrated into the RTABMap to establish the consistent position estimate and the surrounding map. The proposed algorithm can be easily implemented using different kinds of UAVs since it does not rely on specific characteristics of the UAV platform such as the dynamic model. For instance, the proposed ESKF filter can easily be extended with other state-of-the-art SLAM algorithms, such as ORBSLAM [45]. The experimental results also show that the proposed algorithm achieves remarkable position estimation accuracy and small bias compared with

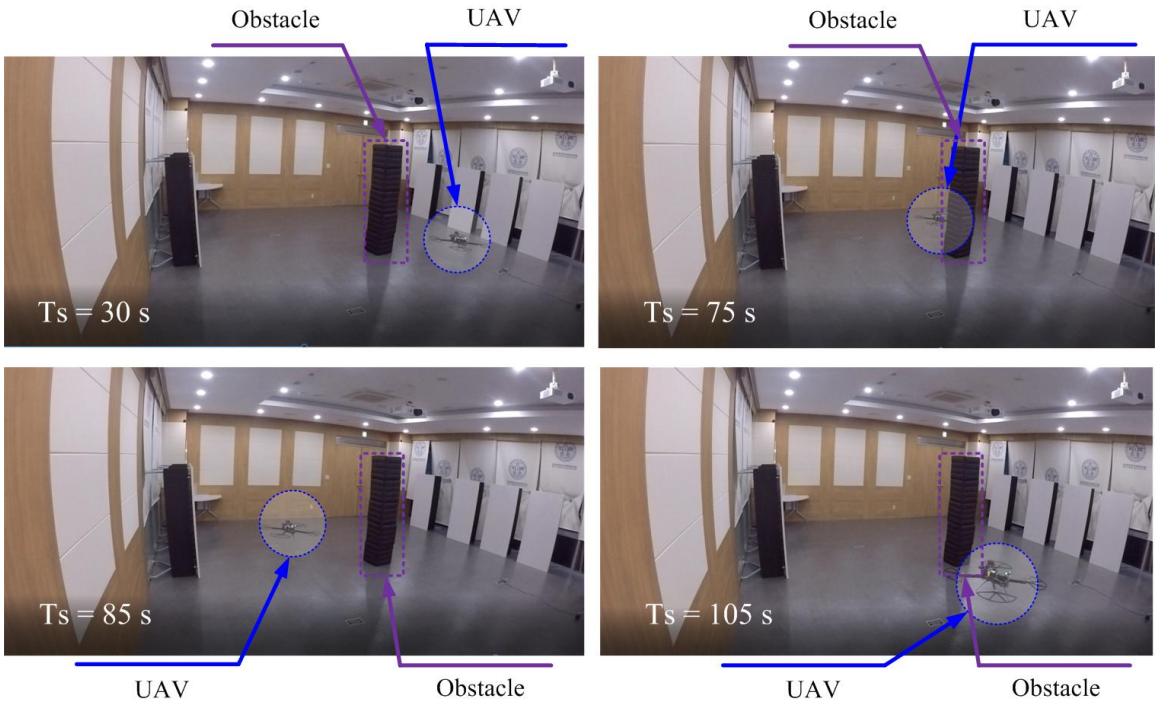


Fig. 12. Experiment for collision-free path planning over time.

the ESKF approaches alone. The proposed system does not require expensive external sensors such as Mocaps or UWBs, and all the developed software runs onboard and in real time with limited onboard computational capability. We also developed a computationally efficient real-time path planning algorithm utilizing a 2D LiDAR. The experimental result demonstrated that the proposed RRT*-GD-Smart algorithm is computationally efficient with a higher convergence rate than other approaches. In addition, the indoor collision avoidance flight test demonstrated that the proposed system can successfully navigate without collision in a fully autonomous mission.

REFERENCES

[1] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics and Automation Society*, vol. 19, no. 3, pp. 46-56, September 2012.

[2] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498-1511, December 2016.

[3] K. Máthé and L. Buşoniu, "Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection," *Sensors*, vol. 15, no. 7, pp. 14887-14916, June 2015.

[4] J. Wu, Z. Zhou, B. Gao, R. Li, Y. Cheng, and H. Fourati, "Fast linear quaternion attitude estimator using vector bservations," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 307-319, January 2018.

[5] Q. Doukhi and D. Lee, "Neural network-based robust adaptive certainty equivalent controller for quadrotor UAV with unknown disturbances," *International Journal of Control, Automation and Systems*, vol. 17, no. 9, pp. 2365-2374, September 2019.

[6] X. Zhao, G. Wang, M. Cai, and H. Zhou , "Flight and hover control system design for a mini-quadrotor based on multi-sensors," *International Journal of Control, Automation and Systems*, vol. 17, no. 2, pp. 486-499, February 2019.

[7] J. S. Jang and D. Liccardo, "Small UAV automation using MEMS," *IEEE Aerospace and Electronic Systems Magazine*, vol. 22, no. 5, pp. 30-34, June 2007.

[8] M. Barczyk and A. F. Lynch, "Integration of a triaxial magnetometer into a helicopter UAV GPS-aided INS," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 2947-2960, October 2012.

[9] G. Chowdhary, E. N. Johnson, D. Magree, A. Wu, and A. Shein, "GPS-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft," *Journal of Field Robotics*, vol. 30, no. 3, pp. 415-438, March 2013.

[10] F. Andert, N. Ammann, S. Krause, S. Lorenz, D. Bratanov, and L. Mejias, "Optical-aided aircraft navigation using decoupled visual slam with range sensor augmentation," *The Journal of Intelligent and Robotic Systems*, vol. 88, no. 2-4, pp. 547-565, December 2017.

[11] E. B. Quist, P. C. Niedfeldt, and R. W. Beard, "Radar odometry with recursive-RANSAC," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1618-1630, August 2016.

[12] S. Zahran, A. Moussa, A. B. Sesay, and N. El-Sheimy, "A new velocity meter based on hall effect sensors for UAV indoor navigation," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3067-3076, April 2019.

[13] A. F. Scannapieco, A. Renga, G. Fasano, and A. Moccia, "Experimental analysis of radar odometry by commercial ultralight radar sensor for miniaturized UAS," *Journal of Intelligent & Robotic Systems*, vol. 90, no. 3-4, pp. 485-503, June 2018.

[14] Y. H. Shin, S. Lee, and J. Seo, "Autonomous safe landing-area determination for rotorcraft UAVs using multiple IR-UWB radars," *Aerospace Science and Technology*, vol. 69, pp. 617-624, October 2017.

[15] S. Zihajehzadeh, P. K. Yoon, B.-S. Kang, and E. J. Park, "UWB-aided inertial motion capture for lower body 3-D dynamic activity and trajectory tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3577-3587, December 2015.

[16] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. R. Giordano, "Shared control: balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 57-68, September 2012.

[17] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi, "A vision-based guidance system for UAV navigation and safe landing using natural landmarks," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 57, p. 233-257, January 2010.

[18] M. Rabah, A. Rohan, M. Talha, K. Nam, and S. Kim "Autonomous vision-based target detection and safe landing for UAV," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 3013-3025, December 2018.

[19] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416-446, March 2019.

[20] D. Hoffman, M. Rehan, W. MacKunis, and M. Reyhanoglu, "Quaternion-based robust trajectory tracking control of a quadrotor hover system," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2575-2584, December 2018.

[21] G. Araguás, C. Paz, D. Gaydou, and G. P. Paina, "Quaternion-based orientation estimation fusing a camera and inertial sensors for a hovering UAV," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 1, pp. 37-53, January 2015.

[22] V. Madyastha, V. Ravindra, S. Mallikarjunan, and A. Goyal, "Extended Kalman filter vs. error state Kalman filter for aircraft attitude estimation," *Proc. AIAA Guidance, Navigation, and Control Conference*, Portland, Oregon, p. 6615, 2011.

[23] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., 2005-002, March 2005.

[24] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, Academic Press. vol. 1, 1979.

[25] A. Santamaria-Navarro, G. Loianno, J. Solà, V. Kumar, and J. Andrade-Cetto, "Autonomous navigation of micro aerial vehicles using high-rate and low-cost sensors," *Autonomous Robots*, vol. 42, no. 6, pp. 1263-1280, August 2018.

[26] F. L. Markley, "Attitude error representations for Kalman filtering," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 311-317, March 2003.

[27] J. Sola, "Quaternion kinematics for the error-state Kalman filter," arXiv:1711.02508, 2017.

[28] W. Li and J. Wang, "Effective adaptive Kalman filter for MEMS-IMU/magnetometers integrated attitude and heading reference systems," *The Journal of Navigation*, vol. 66, no. 1, pp. 99-113, January 2013.

[29] A. M. Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346-1356, Jul. 2006.

[30] P. Appel, "Attitude estimation from magnetometer and earth-albedo-corrected coarse sun sensor measurements," *Acta Astronautica*, vol. 56, no. 1-2, pp. 115-126, Jan. 2005.

[31] D. Gebre-Egziabher and G. H. Elkaim, "MAV attitude determination by vector matching," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1012-1028, July 2008.

[32] H. No, A. Cho, and C. Kee, "Attitude estimation method for small UAV under accelerative environment," *GPS Solution*, vol. 19, no. 3, pp. 343-355, July 2015.

[33] J. Jung, T. Oh, and H. Myung, "Magnetic field constraints and sequence-based matching for indoor pose graph SLAM," *Robotics and Autonomous Systems*, vol. 70, no. 3, pp.92-105, August 2015.

[34] J. Jung, S. Lee, and H. Myung, "Indoor mobile robot localization and mapping based on ambient magnetic fields and aiding radio sources," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 7, pp.1922-1934, July 2015.

[35] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *Journal of Field Robotics*, vol. 30, no. 5, pp. 803-831, August 2013.

[36] G. Zhang and L.-T. Hsu, "Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system," *Aerospace Science and Technology*, vol. 80, pp. 368-380, September 2018.

[37] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 3, Academic Press, NY, 1982.

[38] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, Chapman and Hall/CRC, Boca Raton, FL, 2004.

[39] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635-646, August 2010.

[40] F. Islam, J. Nasir, U. Malik, Y. Ayaz, and O. Hasan, "RRT*-Smart: rapid convergence implementation of RRT* towards optimal solution," *Proc. of IEEE International Conference on Mechatronics and Automation*, Chengdu, China, pp. 1651-1656, 2012.

[41] J. Ge, F. Sun, and C. Liu, "RRT-GD: an efficient rapidly-exploring random tree approach with goal directionality for redundant manipulator path planning," *Proc. of IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Qingdao, China, pp. 1983-1988, 2016.

[42] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189-206, April 2013.

[43] W. Youn, M. B. Rhudy, A. Cho, H. Myung, "Fuzzy adaptive attitude estimation for a fixed-wing UAV with a virtual SSA sensor during a GPS outage," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1456- 1472, January 2020.

[44] N. Yadav and C. Bleakley, "Accurate orientation estimation using AHRS under conditions of magnetic distortion," *Sensors*, vol. 14, no. 11, pp. 20008-20024, January 2014.

[45] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, October 2015.

[46] S. Jung, S. Cho, D. Lee, H. Lee, and D. H. Shim, "A direct visual servoing-based framework for the 2016 IROS autonomous drone racing challenge," *Journal of Field Robotics*, vol. 35, no. 1, pp. 146-166, August 2017.

[47] V. Grabe, H. H. Bülthoff, D. Scaramuzza, and P. R. Giordano, "Nonlinear ego-motion estimation from optical flow for online control of a quadrotor UAV," *The International Journal of Robotics Research*, vol. 34, no. 8, pp. 1114-1135, May 2015.

[48] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors," *IEEE Access*, vol. 3, no. 3, pp. 599-609, May 2015.

[49] M. Iacono and A. Sgorbissa, "Path following and obstacle avoidance for an autonomous UAV using a depth camera," *Robotics and Autonomous Systems*, vol. 106, no. 3, pp. 38-46, May 2018.

[50] A. Behjat, S. Paul, and S. Chowdhury, "Learning reciprocal actions for cooperative collision avoidance in quadrotor unmanned aerial vehicles," *Robotics and Autonomous Systems*, vol. 121, no. 3, pp. 103270-103286, September 2019.

[51] D. Maravall, J. de Lope, and J. P. Fuentes, "Vision-based anticipatory controller for the autonomous navigation of an UAV using artificial neural networks," *Neurocomputing*, vol. 151, no. 3, pp. 101-107, March 2015.

Wonkeun Youn received his B.S. degree from the Handong Global University, Pohang, Korea, in 2008 and his M.S. and Ph.D degrees in Robotics Program from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon in 2010 and 2020, respectively. He has been a senior researcher with the UAV System Division, Korea Aerospace Research Institute (KARI), since 2011. His current research interests include multisensor fusion, Bayesian estimation theory, multimodal target tracking, and GPS/INS-based UAV navigation.

Hayoon Ko received his B.S. and M.S. degrees from the Korea Aerospace University (KAU), Goyang, Korea, in 2017 and 2019, respectively, in information and communication engineering. His current research interests include path planning, image processing, and UAV.

Hyungsik Choi received his Ph.D. degree in aerospace engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2011. He has been a principal researcher with the UAV System Division, Korea Aerospace Research Institute (KARI), since 2002. His research interests include flight simulation and control. He has been involved in projects developing UAVs, especially in design and implementation of software for flight dynamics and control.

Inho choi received his Ph.D. degree in aerospace engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2012. He has been a principal researcher with the UAV System Division, Korea Aerospace Research Institute (KARI), since 1996. His research interests include flight control algorithm, multi-sensor fusion, collision avoidance of UAV. He has been involved in projects developing UAVs, especially in design and implementation of software for navigation and control.

Joong-Hwan Baek received his B.S. degree from the Korea Aerospace University (KAU), Goyang, Korea, in 1981 and his M.S. and Ph.D. degrees in electrical and computer engineering from Oklahoma State University in 1987 and 1991, respectively. Since 1992, he has been a professor with the School of Electronics, Telecommunications and Computer Engineering, KAU. His current research interests include image processing, computer vision, pattern recognition, and multimedia.

Hyun Myung received his B.S., M.S., and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1992, 1994, and 1998, respectively, in electrical engineering. He was a senior researcher with the Electronics and Telecommunications Research Institute, Daejeon, from 1998 to 2002; a CTO and the director of the Digital Contents Research Laboratory, Emersys Corporation, Daejeon, from 2002 to 2003; and a principal researcher with the Samsung Advanced Institute of Technology, Yongin, Korea, from 2003 to 2008. Since 2008, he has been a professor with the Department of Civil and Environmental Engineering, KAIST, where he is currently the head of the Robotics Program. From 2019, he has been a Professor with the School of Electrical Engineering. His current research interests include simultaneous localization and mapping, autonomous robot navigation, artificial intelligence, machine learning, deep learning, structural health monitoring using robotics, and swarm robots.