A low-cost autonomous navigation system for a quadrotor in complex outdoor environments

Hanzhang Wang and Yisha Liu

Abstract
In this article, we design a low-cost navigation system for a quadrotor working in unknown outdoor environments. To reduce the computing burden of the quadrotor, we build a separated system and transfer the computing resources from onboard side to ground station side. Both sides’ communication is guaranteed by 5G wireless networks. We utilize a stereo camera to acquire point clouds and build Octomap for the quadrotor’s navigation. Then, the trajectory is generated in two stages. In the first stage, a modified RRT*-CONNECT algorithm is adopted to generate a set of collision-free waypoints. In the second stage, a curve fitting algorithm is utilized to get a smooth piecewise Bezier trajectory. The advantage of the proposed method is to optimize the path into a safe, smooth, and dynamically feasible trajectory in real time. The modules of the state estimation, dense mapping, and motion planning are integrated into a DJI Matrice 100 quadrotor. Finally, simulation and experiments are both conducted to show the validity and practicality of our method.

Keywords
Path planning, RRT*-CONNECT, cluttered outdoor environments, quadrotors, 5G

Date received: 29 September 2019; accepted: 5 January 2020

Introduction
Micro aerial vehicles (MAVs) are becoming more and more popular in many fields. It is convenient to utilize such platforms to execute flight missions in complex scenes for their wide coverage, high mobility, and agility. As the applications in the fields of bridge inspection, power line inspection, forest fire prevention, search, and rescue, the demand for MAVs’ autonomous flight in unknown scenes has become extremely larger than before.

The purpose of this article is to build an inexpensive platform for a quadrotor to navigate autonomously in complex environments, such as woods and ruins. As shown in Figure 1, our plan is to make a quadrotor able to fly in unknown woods autonomously. Nowadays, the huge breakthroughs in localization, mapping, navigation, and planning have made autonomous onboard flight possible in complex places.1–3 These works need the quadrotors to carry on an advanced onboard computer to process localization, mapping, planning, and flight control simultaneously. However, in cluttered outdoor scenes such as forests and ruined places, these application scenarios are unfriendly and accidents may happen sometimes, which is a deadly threat for the quadrotors to carry out missions in these environments. Once the quadrotors are missing or broken badly, it will cause significant economic losses to

Corresponding author:
Yisha Liu, Information Science and Technology College, Dalian Maritime University, Dalian, Liaoning 116026, China.
Email: liuyisha@dlmu.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
users. So we will design a low-cost and separated navigation system to solve this problem. The system is combined with a lightweight quadrotor and a ground station like other state-of-the-art ones. The quadrotor carries a low-cost Intel NUC (Intel I5 5300U, DDR3L), by which it is easy to:

1. control the autopilot by serial bus;
2. get the video stream from onboard camera; and
3. transmit the video stream data to ground station.

We use 5G wireless communication to build a high bandwidth data transmission between the quadrotor and the ground station. The quadrotor’s autonomous flight in unknown environments needs sufficient perception of surroundings to localize itself and create a dense globally consistent map of the environment. We transfer this work from the onboard computer to the ground station since the low-cost computer has weaker computing performance than the one in our previous work. In mobile robotics, various vision-based and laser-based algorithms have been proposed and used for simultaneous localization and mapping (SLAM). In our previous work, a 3D-LiDAR is used to locate itself and plan trajectories on Octomap, which is generated from raw point clouds. But we use an inexpensive and lightweight stereo camera (Intel Realsense D435) for perception in this article. We deploy the VINS-MONO for localization and Dense Surfel Mapping to get local dense point cloud map. We also use Octomap to perform three-dimensional (3-D) map building for its efficient memory usage. The Octomap is transformed from raw point clouds obtained by stereo camera in real-time navigation. We use the flexible collision library to detect whether the quadrotor and the obstacles are overlapping. A set of collision-free waypoints are generated by a modified RRT*-CONNECTED algorithm on Octomap. Then, a smooth trajectory consisting of multiple Bezier curves is generated, which is proposed by Gao et al. The safety and dynamic feasibility of the trajectory are guaranteed by introducing a set of constraints. All operations can be demonstrated on the Rviz, which is a 3-D visualization tool of the robot operating system (ROS).

We sum up the main contributions of this article as follows. Firstly, we design a low-cost separated navigation system, which is combined with a lightweight quadrotor and a ground station based on 5G communication. Then, a modified RRT*-CONNECT algorithm is proposed for the quadrotor to generate a set of optimized collision-free waypoints on Octomap. Moreover, our system integrates the functions of real-time state estimation, mapping, planning, and flight control for the quadrotor working in unknown outdoor complex environment.

The outline of this article is as follows. We present the related literature in the “Related work” section and introduce the software and hardware architecture of the system in the “System overview” section. A feasible path planning method is detailed in the “Path planning” section. The results of both the simulation and outdoor experiments are presented in the “Experimental results” section. Finally, the article is concluded in the “Conclusions” section.

**Related work**

Basically, recent works on practical quadrotor path planning have separated the trajectory generation problem in two steps. The first step is path finding. There are a lot of works of MAV’s path finding in recent years, and the algorithms are diverse, ranging from the searching-based A* to sampling-based RRT. They all have been applied effectively in different scenes. The searching-based algorithms can be used in a dispersed space which is modeled...
from a real-world scene. The sampling-based motion planning is an algorithm proposed in last decade and has attracted great attention. In 2000, Kuffner et al. proposed RRT-CONNECT,\(^\text{19}\) which is a bidirectional search algorithm. The search speed and efficiency have been significantly improved over the original RRT algorithm. MIT’s Karaman et al. proposed RRT*.\(^\text{20}\) Compared with RRT, this algorithm can not only find a fast and feasible solution but also ensure asymptotic optimality. CMU’s Gammell et al. proposed a modified informed RRT*,\(^\text{21}\) which can reduce the search range by constantly finding a better path by minimizing the sampling space. Based on these methods, we will combine the RRT-CONNECT with RRT* and add some evaluation conditions in the generation of random points.

The second step is path optimizing. After finding a set of collision-free waypoints, we cannot execute flight mission on them directly. We have to generate another smooth, safe, and dynamically feasible trajectory based on these waypoints we have found. Many methods\(^\text{22,23}\) have been adopted to optimize the trajectory. Mellinger et al.\(^\text{24}\) were the first people to parameterize the path into a minimum snap trajectory. The trajectory is represented by piecewise polynomial functions, and the trajectory optimization process is modeled as a quadratic programming problem. Richter et al. proposed an analytic solution for minimum snap trajectory generation.\(^\text{25}\) Gao et al. come up with a way to create a corridor which is composed of spheres on each control point and formulate the problem as quadratically constrained quadratic programming.\(^\text{26}\) Gao et al. proposed a method to model the trajectory with multiple Bezier segments by deploying Bernstein polynomial basis.\(^\text{15}\)

Usually, MAV’s state estimation relies on GPS in outdoor environments. When the MAV flies in cluttered outdoor or indoor scenes, it relies on ultrasonic or optic flow algorithm to localize itself. In the study by Mellinger et al.,\(^\text{24}\) the position and orientation of a quadrotor were captured by a Vicon system. But as the application of the MAV become more and more wide, we need the quadrotor to execute missions in some complex outdoor environments, such as woods and mountains, where the GPS signal is poor and lots of obstacles exist. The quadrotor must equip with sufficient perceptual ability to estimate its state and map its surroundings. Various laser-based and vision-based algorithms have been proposed with the rapid development in SLAM. A laser scanner\(^\text{25}\) was used to re-localize against a previously recorded map and a global plan was calculated before flying. The laser-based strategy was also used in our previous work\(^\text{6}\) for its long-range perceptual ability and immunity to illuminated conditions. But the heavyweight and the high cost of Velodyne VLP-16 are obvious in the practical applications of the quadrotor San Jose, California, United States. In these years, the vision-based strategy is very popular for its lightweight and low cost. These strategies use monocular camera,\(^\text{27}\) stereo camera,\(^\text{28}\) monocular camera with IMU,\(^\text{12}\) or stereo camera with IMU.\(^\text{13}\) In this article, we will deploy the VINS-MONO for localization and Dense Surfel Mapping to get local dense point cloud map.

There are three forms to represent the map. The first form is to utilize the raw point cloud as the map and the trajectory is generated on it directly. The distance between one waypoint of the trajectory and its nearest point cloud is calculated by utilizing KD-tree\(^\text{26}\) to store the information of all point clouds. But this way needs to store all point cloud information, and it will become slow when we send the target position directly on Rviz. The second form is voxel grid, which reduces the number of point clouds using the center point to represent neighboring point clouds in one cube unit. It is efficient for representing small scenes, but it cannot store historical point cloud information. Another way is Octomap, which is memory efficient. It just needs the sensor to publish local point clouds and can record all history points. In this research, we will deploy Octomap to represent the global map.

**System overview**

**Hardware architecture**

The hardware equipment is shown in Figure 2, and the overall hardware architecture pipeline is shown in Figure 3.

**Flight platform.** We use a commercial DJI M100 as the flight platform. It is controlled by an onboard autopilot and can be accessed through DJI’s Onboard SDK, which contains a set of functions that allow the user to send control commands and receive telemetry data from the drone. The quadrotor carries an inexpensive Intel NUC (Intel i5-5300U) and an Intel Real sense D435 as perception sensor. A Netcore NW392 gigabit wireless network card (802.11ac) is used for the quadrotor to communicate with the ground station. The quadrotor does not rely on the GPS in autonomous mode. But when there is emergency happening, we have to switch autonomous state to remote control state. The DJI M100 has to rely on the GPS to localize itself in remote control state, so we do not remove the GPS.
Ground station. We use a laptop (MSI GP62M with Intel i7 7700HQ quad-core processor) as a ground station, which communicates with flight platform by HUAWEI Honor X2 router. The laptop and the router are connected by a Gigabit cable. The maximum speed of the onboard computer side is up to 863 m/s, and the ground station side’s maximum speed is up to 1000 m/s. According to our initial thoughts, we treat the flight platform as an entire low-cost sensor to explore the unknown scene and transfer the computing resource from the quadrotor to the ground station. The huge amount of data transmission between two sides is guaranteed by the high bandwidth of 5G communication.

Software architecture

The entire interactive software architecture is shown in Figure 4. We use ROS as our development environment. Each dashed box represents a ROS node running at different rate. The visualized platform is based on Rviz, which is a 3-D visualization interface of ROS. Firstly, the monocular color stream, dual IR stream, and depth stream are obtained from stereo camera D435 by running Intel Realsense SDK. The inertial measurement unit (IMU) data are obtained by running Onboard SDK. The video stream and the IMU data are transmitted to perception module by building SSH connection between both sides. The VINS-MONO is used for localization by merging the monocular color stream and IMU data. The Surfel Mapping module is used to build a local dense point cloud map by fusing the pose data from VINS-MONO and the depth data from onboard side. Then, these point clouds are transformed to Octomap for getting a globally consistent map. A collision-free and dynamically feasible Bezier trajectory is generated by running path planning module. A set of kinodynamic commands are generated by calculating the desired states of the quadrotor and transmitted to Onboard SDK online by wireless communication.
Path planning

Generating a set of collision-free waypoints

RRT* is a fast random sampling algorithm which guarantees asymptotic optimality by selecting the local nearest node. However, RRT* does not consider the target’s position as guidance. Therefore, we proposed a Heuristic RRT* algorithm in our previous work and employed this method in the flight mission. To improve the efficiency to find a set of optimal collision-free waypoints, we combine the RRT-CONNECT with RRT* and add some evaluation conditions in the generation of random points. In Figure 5, the red point is the start position and the green point is the goal position. The start point generates a red tree toward the goal point and the goal point generates a blue tree toward the start point. A greedy heuristic strategy is used in trees’ generation. When the closest distance between two trees is less than a threshold, two trees are connected with a green line in the yellow circle.

The expansion from both sides and the greedy choice are the core of our improved algorithm, which are demonstrated in Algorithm 1. Let $G$ be the map in the configuration space which consisted of vertices $V$ and edges $E$. Denote the straight line connected by point $q_1$ and point $q_2$ as $LINE(q_1, q_2)$. The random exploring map $G_1$ is expanded from the initial node $q_{init}$, which is also the start position of the quadrotor. Another random exploring map $G_2$ is also initialized with the goal node $q_{goal}$, which is located at the goal position. The generation of the $q_{rand}$ is chosen by $Sample(q_{init}, q_{goal}, p, G_1)$. Firstly, heuristic information is given to generate the random tree in the direction of $q_{goal}$ point. Then, we check whether $\theta_n$ which consisted of $LINE(q_{parent}, q_{nearest})$ and $LINE(q_{nearest}, q_{rand})$ is suitable for flight. From Figure 6, we can get that if $\theta_n$ is small, the quadrotor needs to fly a sharp turn to follow the path, and this kind of small angle is not good for generating Bezier curve in the next section. So we only take the $q_{rand}$ with $\theta_n > 90^\circ$ to make sure the quadrotor avoids emergent brake. Since we only concern the translational movement of the quadrotor, the yaw angle’s planning is left as our future work. The map $G_1$ is locally trimmed by $Reconnect(q_{new}, q_{nearest}, G_1)$ in Algorithm 2 which uses the asymptotic optimality strategy of RRT* algorithm. Then, we get the $q_{new}$ point and the trimmed map $G_1$. The following steps are the same as RRT-CONNECT algorithm basically, which is processed by lines 5–30 in Algorithm 1.

A feasible path optimization method

Mellinger et al. have suggested a way to represent the MAVs’ full-state space by 4-D variables consisting of 3-D coordinates ($x, y, z$) and yaw angle $\psi$. A piecewise parameterized trajectory can be used for MAVs’ path following on the basis of differential flatness. Since the trajectory optimization process can be formulated as a Quadratic Programming, we utilize the minimum jerk of the trajectory as objective function to generate smooth trajectory. The trajectory is represented by Bernstein polynomial basis which is written as

![Algorithm 1: RRT*-CONNECT.](image-url)
Figure 6. Two situations of qrand’s growth direction. (a) \( \theta n \leq 90^{\circ} \). (b) \( \theta n > 90^{\circ} \).

Algorithm 2. Sample(qinit, qgoal, p, G1).

1: qrand \leftarrow \text{ChooseTarget}(\text{qinit}, \text{qgoal}, p)
2: qnearest \leftarrow \text{Nearest}(G1, \text{qrand})
3: if FlightYawLimit(qparent, qnearest, qrand) then
4: qnew \leftarrow \text{Steer}(qnearest, qrand); qparent \leftarrow qnew
5: Reconnect(qnew, qnearest, G1)
6: end if

\[
B_j(t) = c^0_j b^0_n(t) + c^1_j b^1_n(t) + \cdots + c^n_j b^n_n(t) = \sum_{i=0}^{n} c^i_j b^i_n(t) \tag{1}
\]

where \([c^0_j, c^1_j, \ldots, c^n_j]\) is a set of control points of the \(j\)th piece of Bezier curve and \(n\) is the degree of Bernstein basis.

The trajectory is divided into multiple segments by time. Each segment is represented by a piece of Bezier curve as

\[
f_i(t) = \begin{cases} 
\frac{p_1 \cdot \sum_{i=0}^{n} c^i_j b^i_n(t)}{p_1}, & t \in [T_0, T_1] \\
\frac{p_2 \cdot \sum_{i=0}^{n} c^i_j b^i_n(t)}{p_2}, & t \in [T_1, T_2] \\
\vdots \\
\frac{p_m \cdot \sum_{i=0}^{n} c^i_j b^i_n(t)}{p_m}, & t \in [T_{m-1}, T_m] 
\end{cases} \tag{2}
\]

where \(m\) is the number of segments of the trajectory and \(p_1, p_2, \ldots, p_m\) are the parameter vectors on every segment of the Bezier curve to map the time period from \([0, 1]\) to \([T_{i-1}, T_i]\) in each curve. \(i\) represents the number of the control point. \(j\) represents the number of the segments. \(\mu\) is any of \(x, y,\) and \(z\) dimensions. \(T_1, T_2, \ldots, T_m\) are end times of each segment. The cost function is written as follows

\[
J = \sum_{\mu \in \{x, y, z\}} \int_0^T \left( \frac{d^k f_i(t)}{dt^k} \right)^2 dt \tag{3}
\]

where \(k\) represents the \(k\)th-order derivative of the objective function. We set \(k = 3\) as the original article.\(^{15}\)

It is in a quadratic form \(p^T Q_p p\). \(p\) represents the vector consisting of all control points \([c^0_j, c^1_j, \ldots, c^n_j]\). \(Q_p\) is the Hessian matrix of the objective function and can be converted into minimum snap trajectory form. Then, a set of constrains are enforced during the path generation to ensure the dynamical feasibility of the trajectory.

**Waypoints constraints.** Bezier curve has the property that the head point and tail point are connected by the curve, and other points are not in the curve. We add position, velocity, acceleration, and their derivatives as equality constraints on both head point and tail point. A \(lth\) \((h < n)\)-order derivative of head point of the \(\mu\) dimension can be represented as \(c^l_{\mu,1} b^i_n(t) \cdot s^{(1-l)}_m = d^{(l)}_{\mu,1}\) and at the tail point of the \(\mu\) dimension, we can get \(c^l_{\mu,m} b^i_n(t) \cdot s^{(1-l)}_n = d^{(l)}_{\mu,m}\).

**Continuity constrains.** For each two adjacent curve segments of the trajectory, they must be continuous. So the continuous equality constrains are added to any two adjacent segments. For the \(j\)th and \((j+1)\)th piece curves, we have

\[
c^l_{\mu,j} b^i_n(t) \cdot s^{(1-l)}_j = c^l_{\mu,j+1} b^i_n(t) \cdot s^{(1-l)}_{j+1} \tag{4}
\]

**Safety constrains.** In the study by Gao et al.,\(^{15}\) a collision-free sphere area is generated by considering the nearest obstacles. And the area is enlarged to a cube by gauging near grids along the direction of the 3-D axis. This method can ensure the trajectory constrained in the middle of the corridors, but it takes much time in finding the corridor, so we just follow the same way in our previous work,\(^{6}\) to add a cube constrains with a fixed size on each waypoint: \(-x_m \leq c^l_{ij} \leq x_m, -y_m \leq c^l_{ij} \leq y_m\), and \(-z_m \leq c^l_{ij} \leq z_m\). These constrains are shown as orange cubes in Figure 7.

**Kinodynamic constrains.** The velocity and acceleration of MAV must be fixed within a range to ensure the dynamic feasibility of the trajectory. For the \(j\)th control point of the \(j\)th segment, we add inequality constraints

\[
v^+_m \leq n \cdot (c^l_{ij} - c^{l-1}_{ij}) \leq v^-_m
\]
\[ a_m^+ \leq n \cdot (n - 1) \cdot \left( c_{ij}^2 - 2c_{ij}^{-1} + c_{ij}^{-2} \right) / p_j \leq a_m^- \]  \hspace{1cm} (5)

where \( v_m \) is the maximum velocity and \( a_m \) is the maximum acceleration of the MAV.

The trajectory optimization process is modeled as a convex quadratic programming with constraints, and convex solvers are used to find the solution within polynomial time.

\[
\min c^T Qc, \quad \text{s.t.} \quad A_e c = b_e, \quad A_i c \leq b_i
\]  \hspace{1cm} (6)

Experimental results

In this section, we present the time and spatial cost comparison result between our proposed algorithm and Heuristic RRT*. We also demonstrate the simulation and the real experimental results of autonomous flight in woods. From Figure 8, it can be seen that the simulation of the path finding process of RRT*-CONNECT and Heuristic RRT* are both executed on a known map. Figure 9 shows a simulation test on a known map. Figure 10 shows the real trajectory of autonomous flight in woods. The flight is fully autonomous in the unknown woods. Each path planning of the quadrotor is global. When the collision occurs, a new trajectory would be generated from its current position to the goal position.

Comparison results

Figure 8 shows one of our comparative tests. We test our proposed algorithm and Heuristic RRT* in Figure 8(a) and (b). Firstly, we generate five different maps and each of them has six random cubic obstacles in a \( 30 \times 30 \times 30 \) m³ cubic space. Then, we performed 100 comparative tests on each map. Table 1 shows five sets of experimental data. Each set represents the mean value of 100 comparative experiments on time and space. The result shows RRT*-CONNECT is more efficient in time cost. We test the path finding algorithms on our ground station (Intel i7 7700HQ).

![Figure 7. A green polynomial Bezier trajectory is generated based on a set of black collision-free waypoints with orange cube constraints.](image1)

![Figure 8. The red point (2, 2, 2) (m) is the start position and the green point (6, 28, 5) (m) is the target position. Two algorithms are both operated in a \( 30 \times 30 \times 30 \) m³ with six cuboid obstacles. In (a), the path finding process of RRT*-CONNECT is shown in blue and red lines, and the green line are the connection of waypoints of the final collision-free path. In (b), the path finding process of Heuristic RRT* is shown in gray lines, and the blue line are the connection of waypoints of the final collision-free path. (a) The results of RRT*-CONNECT and (b) the result of Heuristic RRT*.](image2)
Simulation results

Figure 9 shows one of autonomous flight tests in simulation. An Octomap is transformed from a dense point cloud data file, which is sampled by Velodyne VLP-16. A white smooth Bezier trajectory is generated using Bernstein polynomial basis. Then, a set of high-frequent kinodynamic commands are calculated and transmitted to the quadrotor for executing the flight.

Outdoor experimental results

Our real flight experiment is operated in woods besides our laboratory building. This environment contains many woods with cluttered branches, which is shown in Figure 1. The whole process of path planning is demonstrated in Figure 10. In Figure 10(a), the ground station generates a trajectory which is collision free with current obstacles and the quadrotor follows the path. When a new perceived obstacle collides with the original path, the ground station will regenerate a new trajectory. This flight process replans twice, which is shown in Figure 10(b) and (c). For more details please refer our video, which can be viewed or downloaded at the website with URL: https://v.youku.com/v_show/id_XNDM4MDg5MDAyNA==.html?spm=a2hzp.8244740.0.0.

Conclusion

This article is focused on how to design and implement a low-cost separated system for quadrotors to carry out...
autonomous path planning in cluttered outdoor scenes. To accomplish online local map building, 3-D point cloud acquired by the onboard stereo camera is transformed to Octomap, which is a memory efficient way to represent the unstructured outdoor scenes precisely. Furthermore, an improved 3-D path finding algorithm named as RRT*-CONNECT is proposed to generate a set of collision-free waypoints and then a smooth and dynamically feasible trajectory with a set of constraints is generated using curve fitting algorithm. Experimental results show that our proposed low-cost autonomous system is a feasible solution for a quadrotor to conduct path planning in unknown cluttered woods environment and can reduce the economic losses to users when the quadrotors are missing or broken.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National Key Research and Development Project under grant 2018YFB1307503 and in part by the National Natural Science Foundation of China under grant 61973049.

ORCID iD
Yisha Liu https://orcid.org/0000-0003-2083-8612

References
1. Chen J, Liu T, and Shen S. Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments. In: *IEEE International conference on robotics and automation (ICRA)*, Stockholm, Sweden, 16–21 May 2016.
2. Liu S, Mohta K, Atanasov N, et al. Search-based motion planning for aggressive flight in SE(3). *IEEE Robot Autom Lett* 2018; 2439–2446.
3. Gao F, Lin Y, and Shen S. Gradient-based online safe trajectory generation for quadrotor flight in complex environments. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Vancouver, BC, Canada, 24–28 September 2017.
4. Sa I, Kamel M, Burri M, et al. Build your own visual-inertial drone: a cost-effective and open-source autonomous drone. *IEEE Robot Autom Mag* 2017; 25: 89–103.
5. Gao F, Wang L, Wang K, et al. Optimal trajectory generation for quadrotor teach-and-repeat. *IEEE Robot Autom Lett*, 2019; 4: 1493–1500.
6. Wang H, Liu Y, and Deng H. A novel interactive system for quadrotor path planning in cluttered outdoor environments. In: *IEEE international conference on real-time computing and robotics (RCAR)*, Irkutsk, Russia, 4–9 August 2019.
7. Engel J, Thomas S, and Cremers D. LSD-SLAM: large-scale direct monocular SLAM. In: *European conference on computer vision (ECCV)*, Zurich, Switzerland, 6–12 September 2014.
8. Engel J, Kolton V, and Cremers D. Direct sparse odometry. *IEEE Pattern Anal Mach Intell* 2017; 40: 611–625.
9. Endres F, Hess J, Sturm J, et al. 3-D mapping with an RGB-D camera. *IEEE Robot Trans* 2014; 30: 177–187.
10. Zhang J and Singh S. Visual-Lidar odometry and mapping: low-drift, robust, and fast. In: *IEEE international conference on robotics and automation (ICRA)*, Seattle, WA, USA, 26–30 May 2015.
11. Hornung A, Wurm KM, Bennewitz M, et al. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonom Robots* 2013; 34: 189–206.
12. Qin T, Li P, and Shen S. VINS-MONO: a robust and versatile monocular visual-inertial state estimator. *IEEE Robot Trans* 2018; 34: 1004–1020.
13. Wang K, Gao F, and Shen S. Real-time scalable dense surfel mapping. In: *IEEE international conference on robotics and automation (ICRA)*, Montreal, Canada, 20–24 May 2019.
14. Pan J, Chitta S, and Manocha D. FCL: A general purpose library for collision and proximity queries. In: *IEEE international conference on robotics and automation (ICRA)*, Brisbane, Australia, 21–26 May 2012.
15. Gao F, Wu W, Lin Y, et al. Online safe trajectory generation for quadrotors using fast marching method and Bernstein basis polynomial. In: *IEEE international conference on robotics and automation (ICRA)*, Brisbane, Australia, 21–26 May 2018.
16. Koub A. *Robot operating system (ROS): the complete reference* (vol. 1). Riyadh, Saudi Arabia: Springer Publishing Company, 2016.
17. Hart PE, Nilsson NJ, and Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans Syst Sci Cybern* 1968; 4: 100–107.
18. LaValle SM. *Randomized kinodynamic planning*. *Int J Robot Res* 2001; 20(5): 378–400.
19. Kuﬀner JJ and Lacalle SM. RRT-connect: an efﬁcient approach to single-query path planning. In: *IEEE international conference on robotics and automation (ICRA)*, San Francisco, CA, USA, 24–28 April 2002.
20. Karaman S and Frazzoli E. Sampling-based algorithms for optimal motion planning. *Int J Robot Res* 2011; 30: 846–894.
21. Gammell JD, Srinivasa SS, and Barfoot TD. Informed RRT*: Optimal incremental path planning focused through an admissible ellipsoidal heuristic. In: *IEEE international conference on intelligent robots and systems (IROS)*, Chicago, Illinois, USA, 14–18 September 2014.
22. Ratliff N, Zucker M, Bagnell JA, et al. CHOMP: gradient optimization techniques for efﬁcient motion planning. In: *IEEE international conference on robotics and automation (ICRA)*, Kobe, Japan, 12–17 May 2009.
23. Schulman J, Ho J, Lee A, et al. Finding locally optimal, collision-free trajectories with sequential convex
optimization. In: Robotics: Science and Systems, Berlin, Germany, 24–28 June 2013.

24. Mellinger D and Kumar V. Minimum snap trajectory generation and control for quadrotors. In: IEEE international conference on robotics and automation (ICRA), Shanghai, China, 9–13 May 2011.

25. Richter CA, Bry AP, and Roy N. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. Robot Res 2016; 114: 649–666.

26. Gao F and Shen S. Online quadrotor trajectory generation and autonomous navigation on point clouds. In: IEEE international symposium on safety, security, and rescue robotics (SSRR), Lausanne, Switzerland, 23–27 October 2016.

27. Mur-Artal R, Montiel JMM, and Tardos JD. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot 2015; 31: 1147–1116.

28. Labbé M and Michaud F. Long-term online multi-session graph-based SPLAM with memory management. Autonom Robots 2017; 42: 1133–1150.