Fast Obstacle Avoidance Motion in Small Quadcopter operation in a Cluttered Environment

Chaitanyavishnu S. Gadde, Mohitvishnu S. Gadde, Nishant Mohanty, and Suresh Sundaram

Abstract—The autonomous operation of small quadcopters moving at high speed in an unknown cluttered environment is a challenging task. Current works in the literature formulate it as a Sense-And-Avoid (SAA) problem and address it by either developing new sensing capabilities or small form-factor processors. However, the SAA, with the high-speed operation, remains an open problem. The significant complexity arises due to the computational latency, which is critical for fast-moving quadcopters. In this paper, a novel Fast Obstacle Avoidance Motion (FOAM) algorithm is proposed to perform SAA operations. FOAM is a low-latency perception-based algorithm that uses multi-sensor fusion of a monocular camera and a 2-D LIDAR. A 2-D probabilistic occupancy map of the sensing region is generated to estimate a free space for avoiding obstacles. Also, a local planner is used to navigate the high-speed quadcopter towards a given target location while avoiding obstacles. The performance evaluation of FOAM is evaluated in simulated environments in Gazebo and AIRSIM. Real-time implementation of the same has been presented in outdoor environments using a custom-designed quadcopter operating at a speed of 4.5 m/s. The FOAM algorithm is implemented on a low-cost computing device to demonstrate its efficacy. The results indicate that FOAM enables a small quadcopter to operate at high speed in a cluttered environment efficiently.

Index Terms—Multi-sensor fusion, 3-D probabilistic occupancy map, LIDAR, monocular camera, local planner

I. INTRODUCTION

Recent advancements in Unmanned Aerial Vehicle (UAV) technologies have accelerated the deployment of UAVs for various applications like visual inspection [1], reconnaissance missions [2], agriculture [3] etc. The current increase in humans’ reliance on UAVs is due to their mechanical simplicity and ease of control. Therefore, there is a need to develop robust automation algorithms to enhance the capabilities of a UAV to operate in cluttered environments.

In the literature, the problem of obstacle avoidance has been extensively studied. Studies like [4] have been focused on coordinated flights to prevent UAVs from crashing into each other. Similarly, [5] uses proportional navigation for the same. Many works in UAV navigation like [6] use sensor data streamed to the ground station for processing, thus leading to huge latency issues. Also, in these works, a known map of the environment is used. In an uncertain environment, it will not yield successful results due to changes in the map. Therefore a sensor-based algorithm enables UAVs to act dynamically based on their observed environment.

In the literature for sensor-based obstacle avoidance, multi-sensor fusion techniques have been used to enhance the UAV’s capabilities. For example, in [7] one radar, four cameras, and two onboard computers have been used. However, it is not a viable option for low-cost fast moving drone operating in a cluttered environment. To overcome this problem, many algorithms using complex local [8]–[11] and global [12] planners have been designed for obstacle avoidance. However, these algorithms have not shown promising results because their planning modules were constrained due to the sensor’s field of vision. Recently a state-of-the-art approach for UAV control with aggressive maneuvers has been presented by [13], which could be used for real-time applications. However, these methods require accurate instantaneous state feedback, which can only be delivered by motion capture systems and perform well in indoor environments only. Thus, there is a need to develop a cheap multi-sensor approach to efficiently use the available sensor data for instantaneous decision-making and control in outdoor environments.

Recently in literature, the fusion of laser scanners and RGB-D cameras has been used for obtaining accurate estimates of the environment. In [14], RGB-D sensors is used to obtain point-cloud data of the environment but these sensors are restricted to very low range in outdoor conditions. Accurate estimation of the obstacles can be obtained either by using stereo cameras [15] or monocular cameras by creating depth maps using various Simultaneous Localization and Mapping (SLAM) algorithms [16]. However, these works are computationally intensive for on-board computers and yield in lower control frequency in position and velocity feedback.

In the absence of data of the entire environment, learning-based motion planners have been commonly used. Approaches related to learning motion policies and depth from input data were demonstrated by [17]. It used orthogonal structures of indoor scenes to estimate the vanishing point to navigate the MAV in the corridors by maneuvering towards dominant vanishing points. Other works [18]–[20] have applied various learning techniques like imitation learning to navigate a UAV. In [21], autonomous flight is achieved at a stretch of 3 km, at a speed of 1.5 m/s in a cluttered forest environment using the imitation learning method. However, in most cases, due to high computational complexity, the UAV streams data to an external computer and then receives higher-level control commands. Hence, there is still a need to develop a cost-effective, less computationally intensive obstacle avoidance
algorithm for high-speed UAVs in a cluttered environment.

In this paper, we propose a novel Fast Obstacle Avoidance Motion (FOAM) algorithm for sense-and-avoid operations in a cluttered environment. A low-cost monocular camera is used to capture images, which are then processed to obtain the probabilistic occupancy map (POM). A 2D Lidar is also used to generate POM in horizontal plane and is fused with that of camera images. The resultant POM is used to determine the free space by formulating it as an optimization problem. Here, local planner uses this directional information to generate high-level commands like the desired yaw and linear velocity. It enables the quadcopter to traverse the least occupied path to the desired goal. The performance of the proposed algorithm is evaluated in both simulation and actual environments. First, Gazebo simulations were carried out to verify the performance of the algorithm in a cluttered bugtrap environment. Later, simulations in the AIRSIM simulator were conducted in a forest environment to realize the near outdoor-like conditions. Finally, the proposed algorithm was tested on a custom-built quadcopter that uses Pixhawk autopilot in an outdoor environment. The results clearly indicate that FOAM is computationally less intensive and can handle obstacle at a speed of 4.5m/s efficiently.

The remaining paper’s organisation is as follows: Section II presents the the formulation of the problem, followed by the description of the proposed algorithm. Section III describes the simulations carried out along with the experimental setup followed by the conclusion.

II. Fast Obstacle Avoidance Motion (FOAM) Algorithm

In this section, the problem definition of sense-and-avoid using a high speed quadcopter is provided. The following subsection describes the generation of probability occupancy map (POM) based on monocular camera, followed by POM based on 2D lidar. Later, the fusion of the occupancy map is described along with the information about flight control. Finally, the FOAM algorithm is described followed by a formal description of the same is provided.

A. Problem Definition

In this section the sense and avoid (SAA) problem is formulated for a quadcopter $Q$ with $m$ obstacles ($O_1, O_2, \ldots, O_m$) within a region $A \subset \mathbb{R}^2$. A typical SAA problem with a quadcopter and obstacles is shown in Fig. 1. The initial launch position $(X_o, \in \mathbb{R}^3)$ and the goal position $(X_g, \in \mathbb{R}^3)$ are user defined and can be anywhere within $A$. The objective of $Q$ is to move from the initial launch position $x_o$ to the goal position $x_g$ at a high speed $v$ (i.e., $\geq 4$ m/s). Also due to the presence of obstacles in the path it has to avoid them to achieve a collision free maneuver. The mission is assumed to be a success if $Q$ manages to reach the goal position without any collision. However the mission is terminated if the quadcopter fails to reach the goal position within $t^{max}$, or in case there is a collision. Before the start of the mission, the quadcopter is initialized at $x_o$ with velocity $v = 0$ along with a random heading angle $\psi$. Also the quadcopter is made to maintain a fixed height of $h$ along the $z$ axis all the time. During the mission the kinematics equations for the control of the quadcopter can be given as: $\dot{x} = v \cdot \cos(\psi); \dot{y} = v \cdot \sin(\psi)$; where, $\dot{x}$ and $\dot{y}$ represents the velocity of the $Q$ along Cartesian $x$ and $y$ axes.

The quadcopter is equipped with a monocular camera and a 2-D lidar. The 2D lidar provides point-cloud data at a radius of $r$. Every point in the point cloud is associated with two values, i.e., the range measurements ($z_i \leq r$) and the angle from the evader frame of reference. The monocular camera provides visual sensor information within a $90^\circ$ field of view. The orientation of the camera is kept similar to that of the quadcopter. In addition to LIDAR and monocular vision, the quadcopter also has a flight controller that provides inertial measurements and position information of the vehicle. Based on the sensing information, the quadcopter is expected to identify the boundary of obstacles in the sensing region and follow an obstacle free paths. For example in Fig. 1, the vehicle senses the obstacle $(o_1)$ partially and generate a feasible path to reach the goal point as shown in yellow coloured path.

In order to ensure that the quadcopter reaches the goal position optimally, some additional constraint on its motion has been added. To ensure optimally the quadcopter requires to minimize the deviation from the ideal straight line path joining the initial position and the goal while performing the mission. This has to be implemented using a local planner by minimizing the distance between the quadcopter’s position $(x, y)$ from the line joining $(x_o, y_o)$ and $(x_g, y_g)$. Hence, the final mission of the quadcopter is to reach the target location by avoiding collision while having minimum deviation from the optimal path.

B. Probabilistic Occupancy Map for FOAM

The problem of detecting obstacles from the current frame $f$ at time $t$ is converted to a free space determination problem.
features. Let \( n_i \) be the number of features present in \( i \)th sector, \( v_j \) is the flow for given feature \( x_i \) and \( N = \sum_{i=1}^{m} n_i \) be the total number of features with reliable optical flow values in the image frame. Using the flow values of these features, we compute the Probability of Occupancy Map (POM) for \( i \)th sector as given below,

\[
P^C_i (I_t) = \sum_{j=1}^{n_i} \frac{k \cdot x_i \in s_i}{v_j} \frac{v_j}{\epsilon + \sum_{j=1}^{N} v_j} , \quad i = 1, 2, \ldots, m \tag{1}
\]

where, \( s_i \) is the \( i \)th sector and the value of \( \epsilon \) is close to zero.

Note that the flow values for the features detected very far away from the camera will be low, and flow values for features near the camera will be high. Hence, the occupancy value of the sector will be small if the obstacle is far away from the quadcopter and vice-versa. The computational requirement of corner features and optical flow is small, and one achieves 30 frames per second processing speed to detect the free space.

Similarly, a long range 2D lidar is used for depth estimation of the obstacles planar to the quadcopter. In order to estimate the occupancy in the camera plane, the field of view (FOV) of lidar is restricted to the camera’s FOV, as shown in Fig. 2. The effective depth of the 2D lidar is restricted to \( d_{max} \). Based on the resolution of LIDAR, one will always have a finite number of points in each sector. Note that the depth information of points beyond the desired sensing radius is considered effective depth. It is set at 10m in the simulation. Thus, the probability occupancy map using 2D point cloud data for the \( i \)th sector is given as

\[
P^L_i (I_t) = \sum_{j=1}^{n_i} \frac{k \cdot p_i \in s_i}{d_{max} - d_j} \frac{d_{max} - d_j}{\epsilon + \sum_{j=1}^{N} d_{max} - d_j} , \quad i = 1, 2, \ldots, m \tag{2}
\]

From the equation, we can conclude that the probability will be zero if the sector does not have any obstacles. The probability occupancy map obtained from camera and LIDAR point cloud data are fused as:

\[
P_i (I_t) = w_CP^C_i (I_t) + w_LP^L_i (I_t) \quad i = 1, 2, \ldots, m \tag{3}
\]

where \( w_C \) and \( w_L \) are the weights for camera and LIDAR POM respectively and \( w_C + w_L = 1 \).

**C. Flight Control**

The algorithm is designed to send high-level commands to the quadrotor – (1) forward translational speed, (2) constant height, and (3) rotational velocity (heading/desired yaw). The forward velocity of the complete system is constant throughout the whole mission and can be modified by the user based on the application of the vehicle. Fig. 3 shows the schematic diagram of the flight control used for this algorithm. As
Algorithm 1 Pseudocode FOAM

Data: Current image frame \( f' \), Lidar Data \( L \), GPS Data \( G \); 
\( \epsilon \) \( \leftarrow \) positive value close to zero; 
\( H \) \( \leftarrow \) empty list; 
\( P \) \( \leftarrow \) empty list;

Algorithm Parameters: \( W_C, W_L \mid W_G = 2, D_{max}, M \)

function FOAM

for \( i \) in \( 1, 2, \ldots, M \) do
  \( P_i^C \leftarrow \text{POM image}(\cdot) \)
  \( P_i^L \leftarrow \text{POM lidar}(\cdot) \)
  \( P_i \leftarrow W_C P_i^C + W_L P_i^L \)
  \( P \leftarrow \text{append}(P_i) \)
end for

\( s_d \leftarrow \text{minYawCost}(P) \)
\( \text{currYaw} \leftarrow \text{from GPS Data G} \)
if \( s_d > (\frac{M+1}{2}) \) then
  \( \text{yaw} \leftarrow \text{currYaw} + 10 \times (i - (\frac{M}{2})) \)
else if \( s_d < (\frac{M+1}{2}) \) then
  \( \text{yaw} \leftarrow \text{currYaw} - 10 \times ((\frac{M}{2}) - i) \)
else
  \( \text{yaw} \leftarrow \text{currYaw} \)
end if

function POM IMAGE

Get \( f_k \) from camera
\( f_k \leftarrow \text{convert} \ f'_k \ \text{to gray-scale image} \)
\( X_i \leftarrow \text{corner features for sector } s_i \ \text{in} \ f_k \) \([22]\)
\( V_i \leftarrow \text{optical flow values for given} \ X_i \ \text{in frames} \)
\( f_{k-1}, f_{k-2}, f_k, f_k > 3 \) \([23]\)
calculate \( P_i^C \) using Equation (1)
return \( P_i^C \)

function POM LIDAR

Get \( L \) from lidar
\( D_j \leftarrow \text{lidar data in sector } S_i \ \text{in frame} \ f_k, D_j \ \in \ L \)
calculate \( P_j^L \) using Equation (2)
return \( P_j^L \)

function minYawCost (P)

\( P_{min} = \text{min}(P) \)
for \( l \) in \( 0, 1, \ldots, (M - 1) \) do
  if \( P = P_{min} \) then
    \( H \leftarrow \text{append}(l + 1) \)
  else
    \( H \leftarrow \text{min}(\text{abs}(h - (\frac{M+1}{2})) \ \forall h \ \in \ H) \)
end for

mentioned, the required high-level commands like the desired yaw, height of the vehicle, etc., are sent to the velocity controller present in the Pixhawk autopilot. These velocity commands are sent via mavlink protocol while using the MAVROS package to the autopilot. The Pixhawk is connected to the on-board computer via a serial connection. The outer loop uses a PID controller for velocity control. It computes the desired thrust, roll, pitch and yaw and sends it to the attitude controller. The PID control of the attitude controller then realises the required motor speed on the quadrotor for the optimal motion. The above-mentioned process is completely realised inside the Pixhawk autopilot of the quadcopter.

D. FOAM Algorithm

The FOAM algorithm is designed such that it is computationally less intensive. The system architecture of the same is shown in Fig. 4. The figure illustrates various components of FOAM and its integration into the small quadcopter. First, FOAM generates POM based on the data from the monocular camera for the \( xz \)-plane. Then it generates POM using the point cloud data from 2D-LIDAR for the \( xy \)-plane. Next, free space is determined by formulating an optimization problem by fusing the POM’s. The solution is to determine the free space at time \( t \). Finally, the local planner generates the necessary commands to the Pixhawk flight controller such that the quadcopter avoids the obstacle and move towards the goal position. The above-mentioned algorithm is summarized and given in algorithm 1.

III. PERFORMANCE EVALUATION

In order to evaluate the performance of FOAM, different test scenarios are considered. Custom environments were created and tested using the Gazebo simulator using PX4 Flight Stack in the back-end for the 3DR IRIS quadcopter. Simulations were also conducted on AirSim to verify the performance of the algorithm. Finally, to evaluate the entire proposed system, experiments were conducted in an outdoor environment using a custom-built quadrotor vehicle that would autonomously navigate based on the high-level commands sent from the onboard computer. The videos related to the evaluation can be found in the following link: https://bit.ly/3sFtVyP.

A. PX4 ROS Gazebo Simulations

The proposed algorithm was tested in various environments. A custom python code was designed to compute and provide the required control commands to the 3DR IRIS quadrotor. The 3DR IRIS model has a monocular camera and 2-D lidar plugins integrated into it. The monocular camera uses a horizontal field of view of 90° and a sensing range of 10 m.
for the 2-D lidar. The sample Gazebo environment is shown in Fig. 5.

B. AIRSIM Simulation

Next, the algorithm was tested on AirSim, an open-source simulator to realize the near outdoor-like conditions. The quadcopter model uses a built-in monocular camera along with the 2D LIDAR for perception. Different instances in Fig. 6 show a typical forest environment in an AirSim simulator. The algorithm was also tested on a custom-designed cluttered environment in AirSim to test its robustness. The simulation was carried out using Intel i7 8750H 3.0 GHz processor, powered with 16GB of DDR4 memory, which also housed an NVIDIA RTX 2070 Max-Q graphics memory.

C. Outdoor Experiments

Finally, to evaluate the proposed system for high speeds, a custom-built quadcopter shown in Fig. 7 was tested in an outdoor environment. The quadrotor is composed of TBS 500 Carbon fiber frame. The quadrotor houses an of the shelf ARM-based computer, NVIDIA Jetson Nano, for onboard computation of perception, control, and planning. Pixhawk 4 is the flight controller, which uses the PX4 flight stack. The vehicle has a 35 percent hover throttle on a 5000 mAh 3S Lithium Polymer (LiPo) battery. The quadcopter is equipped with a Logitech C930e web camera. The camera module packs up with a 1080p resolution, H.264 video compression with Scalable Video Coding and UVC 1.5 encoding to minimize its dependence on computer and network resources and a wide 90-degree field of view. Slamtec RP LIDAR S1 is used for the depth estimation. It provides 360-degree scan field, 5.5hz/10hz rotating frequency with 40-meter ranger distance measurement with more than 9.2K samples per second. A pole was erected in the middle of a field. The quadcopter was placed at the initial launch position. The destination position was given as an input and the program was run. During the mission the quadcopter was able to avoid the pole in between while
operating at a desired velocity of 4.5 m/s. Also the height along z-direction was maintained, i.e, 2 m. An image instance of the same is given in fig. 9. Fig. 8 shows the commanded velocity and estimated velocity. The experiment shows that FOAM can perform sense and avoid in outdoor environments at a very high speed. Also, it demonstrates the low latency of the proposed FOAM architecture by using it on low cost computing devices.

IV. CONCLUSION
In this paper, the novel Fast Obstacle Avoidance Motion (FOAM) algorithm is proposed to perform sense and avoid (SAA) operations. FOAM is a low-latency perception-based algorithm that uses multi-sensor fusion of a monocular camera and a 2-D LIDAR. It has been shown to perform obstacle avoidance at a speed of 4.5 m/s in outdoor environments. Also, it has been shown to perform high-speed maneuvers in a cluttered environment in both Gazebo and AirSim.

REFERENCES
[1] T. Özaslan, G. Loianno, J. Keller, C. Taylor, V. Kumar, J. Wozencraft, and T. Hood, “Autonomous navigation and mapping for inspection of penstocks and tunnels with mavs,” IEEE Robotics and Automation Letters, vol. 2, no. 3, pp. 1740–1747, Jul. 2017.
[2] S. Waharte and N. Trigoni, “Supporting search and rescue operations with uavs,” in 2010 International Conference on Emerging Security Technologies, 2010, pp. 142–147.
[3] B. Vroegindeweij, S. van Wijk, and E. van Henten, “Autonomous unmanned aerial vehicles for agricultural applications,” 2014.
[4] R. W. Beard and T. W. McLain, “Multiple uav cooperative search under collision avoidance and limited range communication constraints,” in 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475), vol. 1, 2003, pp. 25–30 Vol.1.
[5] S.-C. Han, H. Bang, and C. Yoo, “Proportional navigation-based collision avoidance for uavs,” International Journal of Control, Automation and Systems, vol. 7, pp. 553–565, 2009.
[6] S. Mannar, M. Thummalapeta, S. K. Saksena, and S. Omkar, “Vision-based control for aerial obstacle avoidance in forest environments,” IFAC-PapersOnLine, vol. 51, no. 1, pp. 480–485, 2018. 5th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2018.
[7] G. Fasano, D. Accardo, A. Moccia, C. Carbone, U. Cingilio, F. Corrado, and S. Luongo, “Multi-sensor-based fully autonomous non-cooperative collision avoidance system for unmanned air vehicles,” J. Aerosp. Comput. Inf. Commun., vol. 5, pp. 338–360, 2008.
[8] S. Daftry, S. Zeng, A. Khan, D. Dey, N. Melik-Barkhudarov, J. Bagnell, and M. Hebert, “Robust monocular flight in cluttered outdoor environments,” ArXiv, vol. abs/1604.04779, 2016.
[9] Sikang Liu, M. Watterson, S. Tang, and V. Kumar, “High speed navigation for quadrotors with limited onboard sensing,” in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 1484–1491.
[10] B. T. Lopez and J. P. How, “Aggressive 3-d collision avoidance for high-speed navigation,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 5759–5765.
[11] P. R. Florence, J. Carter, and R. Tedrake, “Integrated perception and control at high speed: Evaluating collision avoidance maneuvers without maps,” in WAFR, 2016.
[12] B. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, “Continuous-time trajectory optimization for online uav replanning,” 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5332–5339, 2016.
[13] J. Tordesillas, B. Lopez, J. Carter, J. Ware, and J. How, “Real-time planning with multi-fidelity models for agile flights in unknown environments,” 2019 International Conference on Robotics and Automation (ICRA), pp. 725–731, 2019.
[14] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Estimation, planning, and mapping for autonomous flight using an rgb-d camera in gps-denied environments,” The International Journal of Robotics Research, vol. 31, pp. 1320 – 1343, 2012.
[15] S. W. Chen, G. V. Nardari, E. S. Lee, C. Qu, X. Liu, R. A. F. Romero, and V. Kumar, “Sloam: Semantic lidar odometry and mapping for forest inventory,” IEEE Robots and Automation Letters, vol. 5, no. 2, pp. 612–619, 2020.
[16] A. Bachrach, R. He, and N. Roy, “Autonomous flight in unknown indoor environments,” International Journal of Micro Air Vehicles, vol. 1, pp. 217 – 228, 2009.
[17] C. Bills, J. Chen, and A. Saxena, “Autonomous mav flight in indoor environments using single image perspective cues,” 2011 IEEE International Conference on Robotics and Automation, pp. 5776–5783, 2011.
[18] S. Schaal, “Is imitation learning the route to humanoid robots?” Trends in Cognitive Sciences, vol. 3, pp. 233–242, 1999.
[19] B. Schölkopf, J. Platt, and T. Hofmann, Boosting Structured Prediction for Imitation Learning, 2007, pp. 1153–1160.
[20] B. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” Robotics Auton. Syst., vol. 57, pp. 469–483, 2009.
[21] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, “Learning monocular reactive uav control in cluttered natural environments,” in 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 1765–1772.
[22] Jiao Zhou and Tomasi, “Good features to track,” in 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.
[23] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in International Joint Conference on Artificial Intelligence (IJCAI), 04 1981.