LiDARFlow: Real-Time Panel-Based MAV Guidance in Unknown Environments

João Machado[1]*, Zeynep Bilgin[2], Matthieu Verdoucq[1], Murat Bronz[1]
[1]ENAC, École Nationale de l'Aviation Civile, Université de Toulouse, Toulouse, France
[2]CNRS, Centre national de la recherche scientifique, France

Abstract

This paper presents a guidance algorithm for micro aerial vehicles operating in unknown, cluttered environments using only onboard sensing. The method is based on a panel formulation originally derived from aerodynamic potential-flow theory and generates smooth, collision-free guidance vectors from locally perceived obstacles. The approach is extended to unknown environments by constructing and updating the obstacle representation online from onboard LiDAR measurements. The resulting obstacle-avoidance field is integrated with a nominal guiding vector field to produce the final control input. The system is experimentally validated in indoor flight tests under two scenarios: waypoint navigation and directional guidance. In both cases, the vehicle successfully completes its task while avoiding all obstacles in real time using only onboard perception. The results demonstrate that the method is computationally lightweight and suitable for onboard implementation, with point-cloud processing identified as the main practical limitation. These results support the feasibility of lightweight onboard guidance in unknown environments.

Keywords: micro aerial vehicles, autonomous navigation, obstacle avoidance, LiDAR, panel method, unknown environments.

http://www.imavs.org/

1 Introduction

Advances in technology have made autonomous micro aerial vehicles (MAVs increasingly important across a wide range of applications such as search and rescue and disaster relief. Thanks to their small size and low weight, MAVs can reach locations inaccessible to humans, be deployed in large numbers, and provide vantage points that would otherwise be unattainable. In many of these scenarios, MAVs must operate in unknown environments for which no prior map is available. For example, in forests or disaster zones, pre-mapping all obstacles is often infeasible. Consequently, onboard sense-and-avoid capability is essential for mission success. Operating reliably under these conditions requires a robust guidance system that is lightweight enough to run onboard the MAV.

*Email address: joao.machado@enac.fr, zeynep.bilgin@irit.fr , matthieu.verdoucq@enac.fr, murat.bronz@enac.fr

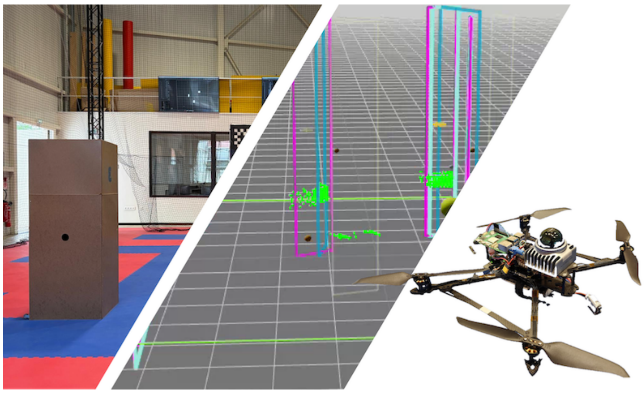


Figure 1: Stage overview of the project. The obstacles, represented on the left, are reconstructed with LiDAR data, as illustrated in the middle. The model of the drone used during the experiments can be seen on the write. It runs onboard with a Raspberry Pi running the nominal guidance and *PGFlow* algorithms, producing velocity commands that are passed to the onboard paparazzi micro-controller.

In many of these applications, MAVs are deployed by personnel who may not have extensive experience in piloting aerial robots. This motivates the development of guidance systems that reduce operator workload by abstracting low-level control tasks such as continuous obstacle avoidance, allowing users to focus on high-level mission objectives. Such capabilities are particularly valuable in time-critical missions where trained operators may be limited or unavailable.

1.1 Related work

Existing approaches to UAV guidance in unknown environments fall broadly into two categories: optimization-based methods and reactive methods. The first formulates guidance as a trajectory optimization problem and has converged on two competing paradigms. Hard-constraint corridor and mixed-integer (MIQP) planners such as FASTER [1], MADER [2], HDSM [3], DYNUS [4], and SANDO [5] guarantee collision-free trajectories. Whereas soft-constraint gradient-based planners like EGO-Planner [6], EGO-Swarm [7], and SUPER [8] embed collision risk as a penalty term to achieve faster computation at the cost of any formal safety guarantee. These methods typically pair a

http://www.imavs.org/

global planner with a local planner that compute and refine a safe path online. Their main drawbacks are that they generally rely on constructing and maintaining a map of the environment and that solving the underlying optimization is computationally demanding, which can limit the achievable replanning rate on the constrained hardware of a small MAVs.

The second category comprises reactive methods, which compute control commands directly from instantaneous sensor data. Recent reactive guidance for UAVs in cluttered, unknown environments has developed along three main lines. The first is control barrier function (CBF) safety filters operating directly on LiDAR or point-cloud data, exemplified by the composite CBF filter of Harms et al. [9], which steers an onboard LiDAR quadrotor through unknown indoor and outdoor clutter while handling thousands of constraints. The second is velocity obstacle and ORCA schemes, increasingly used together with model-predictive control [10] and reinforcement learning [11] for multi-agent and dynamic obstacle avoidance. The third is potential field, guidance vector field methods, which remain heavily used but are increasingly embedded as components within optimisation frameworks as local planners rather than deployed as standalone controllers. Because they act on live sensor measurements rather than on a global map, reactive methods offer low computational cost, short reaction latency, and map free operation, making them well suited to the limited onboard resources of an MAV. Their classical weaknesses lie in completeness, optimality, oscillation, and the absence of continuous time safety guarantees. Potential field methods in particular remain widely used but are well known to suffer from local minima, where competing attractive and repulsive forces can trap the vehicle short of its goal. In [12] this is mitigated by embedding the field within an MPC formulation .

The guidance method proposed in this work superposes the velocity vector fields generated by two algorithms. The nominal motion is generated using the convergent guiding vector field proposed in [13], while obstacle avoidance is achieved using the harmonic potential field approach of [14, 15]. Both algorithms are individually free of local minima under their respective assumptions, while being computationally light. Although no formal proof exists that their superposition preserves this property, [13] provides an intuitive argument for why undesired local minima are unlikely to arise. The practical effectiveness of the proposed guidance strategy is further demonstrated through the experimental results presented in section 4.

The harmonic potential field approach of [14] assumes complete *a priori* knowledge of obstacle locations and geometries. Although several studies have extended potential flow guidance to unknown environments, important limitations remain. In [16], obstacle avoidance was demonstrated for an aerial vehicle operating in an unknown environment, but only in hardware-in-the-loop simulation. The related panel method framework proposed in [17] was validated in unknown environments using a ground robot rather than an aerial platform. More recently, [18] considered potential flow guidance for UAVs in unknown environments, but validation was limited to simulation. This work addresses these limitations by proposing an onboard lightweight system that integrates LiDAR-based obstacle detection with a collision-free guidance strategy that operates without prior knowledge of the environment and is validated through real flight experiments.

1.2 Contributions

This paper presents a computationally lightweight onboard guidance system for MAV navigation in unknown environments using only onboard LiDAR sensing. The proposed approach combines a convergent guiding vector field with harmonic potential-flow obstacle avoidance and operates without prior knowledge of obstacle locations or geometries. The system is validated experimentally in two representative scenarios: waypoint navigation and commanded-direction flight.

The main contributions are:

- A guidance system that autonomously avoids obstacles, reducing operator workload by removing the need for manual obstacle avoidance;

- Full onboard integration of a computationally light functional LiDAR obstacle detection system and collision-free guidance algorithm.

An overview of the project can be seen in Fig. 1.

2 PROBLEM FORMULATION

The objective is to integrate LiDAR dynamic obstacle detection with a collision-free guidance algorithm. To validate the proposed system, two representative flight scenarios are considered in an unknown environment with unknown and dynamic obstacles. In both cases, the MAV has no prior map of the environment and must continuously detect and avoid obstacles in real time using onboard sensing.

The first scenario considers waypoint navigation. The MAV is initially located at an arbitrary position and is commanded by a human operator to reach a goal waypoint. Upon reaching this location, the operator may update the mission by assigning a new waypoint (e.g., returning to the starting location). The vehicle must therefore repeatedly track user-defined waypoints while autonomously handling all obstacle avoidance.

The second scenario considers directional guidance. Instead of fixed waypoints, the operator specifies a desired direction of motion that can be updated during flight. The MAV is required to continuously progress along the commanded direction while autonomously avoiding obstacles encountered along its path.

Together, these two scenarios evaluate the system under operational modes where operators do not need extensive piloting expertise, relying on autonomous obstacle avoidance to safely execute missions in cluttered and unknown environments.

3 SYSTEM ARCHITECTURE

The guidance algorithm presented in [13] is used to generate the drone's nominal motion. In addition to its low computational cost, the algorithm is particularly well suited to the proposed approach because it is based on the superposition of vector fields. The original formulation does not address obstacle avoidance and so it is augmented with an obstacle-avoidance module described in the next section, which likewise relies on the superposition of velocity vector fields.

Obstacles are detected using an onboard Livox MID-360 LiDAR sensor, which produces dense three-dimensional point clouds which are processed into usable obstacles and sent to the guidance algorithm. The full system architecture can be observed in Fig. 2.

3.1 Nominal motion

To apply the algorithm in [13], it is necessary to compute the normal and tangent vectors to the point and line manifolds. The position of a point can be expressed in a three-dimensional Cartesian space by:

$$x - p_x = 0 \quad y - p_y = 0 \quad z - p_z = 0$$

where $(p_x, p_y, p_z) \in \mathbb{R}^3$. Defining $F_p(x,y,z) := \begin{bmatrix} F_{p,1} & F_{p,2} & F_{p,3} \end{bmatrix}^\intercal = \begin{bmatrix} x-p_x & y-p_y & z-p_z \end{bmatrix}^\intercal$, the normal and tangent manifold vectors can be obtained [13]:

$$\nabla F_{p,1} := \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\intercal \quad \nabla F_{p,2} := \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\intercal$$
$$\nabla F_{p,3} := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\intercal \quad \tau_p := \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\intercal$$

Accordingly, a line parallel to xOy can be defined as:

$$\begin{aligned} &z - z_0 = 0 \\ &-\sin\alpha(x-x_0) + \cos\alpha(y-y_0) = 0 \end{aligned}$$

where $(x_0, y_0, z_0) \in \mathbb{R}^3$ is an offset given to the line and α the desired angle between the line and the x-axis. Defining $F_l(x,y,z) := \begin{bmatrix} F_{l,1} & F_{l,2} \end{bmatrix}^\intercal = \begin{bmatrix} z-z_0 & -\sin\alpha(x-x_0)+\cos\alpha(y-y_0) \end{bmatrix}^\intercal$, the normal and tangent manifold vectors can be obtained:

$$\nabla F_{l,1} := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\intercal \qquad \nabla F_{l,2} := \begin{bmatrix} -\sin\alpha & \cos\alpha & 0 \end{bmatrix}^\intercal$$
$$\tau_l := \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \end{bmatrix}^\intercal$$

These can be directly introduced into the guidance algorithm to compute a velocity command prior to obstacle detection, V:

$$\begin{aligned} \text{Point:} \quad & V = -v_d \frac{k_c}{3} \sum_{i=1}^{3} F_{p,i} \frac{\nabla F_{p,i}}{\|\nabla F_{p,i}\|} \\ \text{Line:} \quad & V = v_d \left(\frac{\tau_l}{\|\tau_l\|} - \frac{k_c}{2} \sum_{i=1}^{2} F_{l,i} \frac{\nabla F_{l,i}}{\|\nabla F_{l,i}\|} \right) \end{aligned} \tag{1}$$

where v_d is the desired velocity along the tangent vector[1] and k_c is an arbitrary convergence gain. Note that for the line case, the drone is not converging to a particular line path. It is following the direction of the line, and so (1) reduces to:

$$\text{Line:} \quad V = v_d \frac{\tau_l}{\|\tau_l\|}$$

See [13] for a detailed derivation of the algorithm.

3.2 Obstacle avoidance with PGFlow

For obstacle avoidance, the PGFlow algorithm [15, 19] is employed. PGFlow is a fluid dynamics inspired method that models obstacles as collections of one dimensional panels in a two-dimensional plane. Each panel is represented by a line source (or vortex, depending on the formulation), generating an irrotational velocity field around the obstacle. Fig. 3 illustrates how an obstacle is decomposed into panels.

Since obstacle avoidance is generally more critical in the horizontal plane, only the x and y components of the repulsive field are considered. Although the method can be extended to three dimensions, this significantly increases computational complexity. Vertical collision avoidance can instead be handled through a simpler repulsion mechanism acting along the z-axis.

Given a panel i with vertices $p_{oi_1} \to (x_{oi_1}, y_{oi_1})$ and $p_{oi_2} \to (x_{oi_2}, y_{oi_2})$, its control point is defined as the centre point of the panel.

$$p_{ci} := \frac{(p_{oi_2} - p_{oi_1})}{2} + p_{oi_1} \to (x_{ci}, y_{c_i})$$

The velocity command given to the drone is:

$$\begin{aligned} \dot{x} &= V_x - \sum_{i=1}^{m} \frac{\lambda_i}{2\pi} \int_i \frac{\partial}{\partial x} \ln R_i \mathrm{d}l_i \\ \dot{y} &= V_y - \sum_{i=1}^{m} \frac{\lambda_i}{2\pi} \int_i \frac{\partial}{\partial y} \ln R_i \mathrm{d}l_i \end{aligned}$$

where $V = \begin{bmatrix} V_x & V_y \end{bmatrix}^\intercal$ is the velocity command prior to *PGFlow*, m is the total number of panels, λ_i is the strength of panel i, R_i is the distance from the drone to a point on panel i, and $\mathrm{d}l_i$ is the differential length element along panel i.

The panel strengths are computed such that the normal velocity at each panel control point satisfies $V_i \geq 0$, preventing trajectories from penetrating the obstacle boundary and thereby producing a repulsive effect:

[1] for the point case, since the tangent vector is **0**, this can be simply treated as a gain and the total convergence gain becomes $k_c' := v_d k_c$.

http://www.imavs.org/

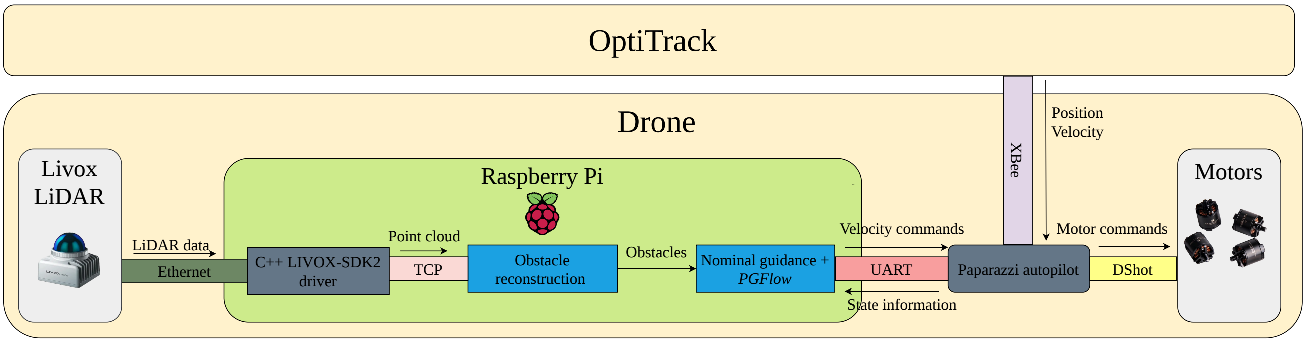


Figure 2: System architecture and respective data flow. LiDAR sends captured data to the Raspberry Pi where it is then processed by the different modules. The guidance algorithm generates the velocity commands sent to the autopilot. The autopilot fuses the vehicle's position and velocity captured by OptiTrack into the full vehicle's state, sending it to the Raspberry Pi. It also convert the velocity commands into the appropriate motor commands.

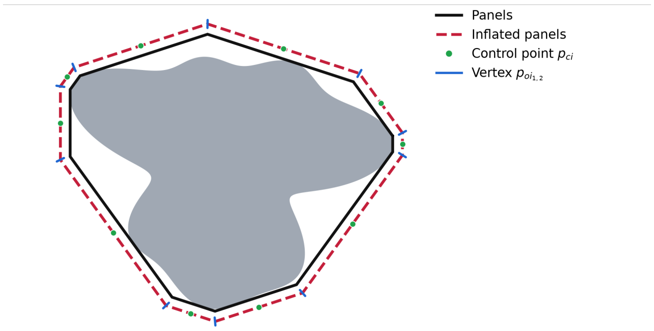


Figure 3: The obstacle (gray) is decomposed into panels (black) that act as a line source for the guidance of the drone. The panels are inflated (red) by at least half the drone's size so that collision avoidance is guaranteed. The control points (green) are located at each panel's half length.

For panel i :

$$\frac{\lambda_i}{2} + \sum_{i \neq j}^{m} \frac{\lambda_j}{2\pi} I_{ij} = -V_i + V(x_{ci}, y_{ci}) \cdot \vec{n_i}$$

where I_{ij} is the panel interaction matrix [15], and $\mathbf{n}_i$ is the outward-pointing normal vector of panel i. See [15] and [14] for a detailed derivation and discussion of the algorithm.

To account for the physical dimensions of the drone, obstacles must be inflated by at least half the characteristic size of the vehicle. Under these conditions, the method guarantees that the drone does not collide with the obstacles. This is also validated empirically on section 4.

In the proposed implementation, the obstacle panels used by PGFlow are not known a priori. They are extracted from onboard LiDAR data.

3.3 *LiDAR-based obstacle extraction*

Obstacle detection is performed using a Livox MID-360 solid-state LiDAR operating at approximately 10 Hz. At each scan, the sensor produces a high-density point cloud in the LiDAR local coordinate frame. The objective of the perception layer is to extract a compact set of obstacles that can be directly used by PGFlow.

The LiDAR scans are acquired onboard by a Raspberry Pi using a C++ driver based on Livox-SDK2 and published through a ZeroMQ stream. The drone's position is provided by the OptiTrack via XBee radio modem fused onboard with the drone's IMU. The LiDAR scans are then associated with the corresponding state estimate and transformed into a common ENU frame.

To reduce computational cost, the point cloud is first downsampled using a voxel grid with voxel side length $s = 0.2$ meters. Each point $p_i = (x_i, y_i, z_i)$ is assigned to a voxel index

$$K_i = 1000003\, k_{x_i} + 1001\, k_{y_i} + k_{z_i},$$

where

$$k_{x_i} = \left\lfloor \frac{x_i}{s} \right\rfloor, \qquad k_{y_i} = \left\lfloor \frac{y_i}{s} \right\rfloor, \qquad k_{z_i} = \left\lfloor \frac{z_i}{s} \right\rfloor.$$

All points assigned to the same voxel are replaced by their centroid. The resulting cloud is then cropped to the region of interest, which corresponds to the size of the used portion of the arena,

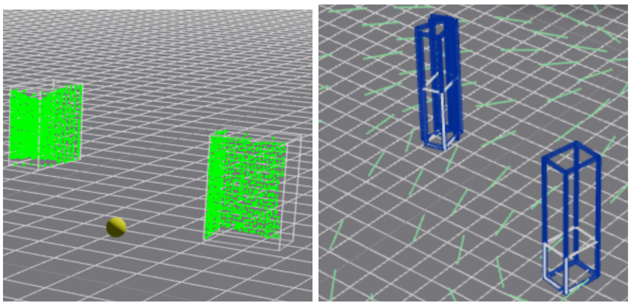
$$x, y \in [-3, 3]\ \mathrm{m}, \qquad z \in [0.5, 3]\ \mathrm{m},$$

and points closer than 0.2 m to the sensor are discarded to remove returns from the vehicle body or from very near-field artifacts.

The filtered points are projected onto the horizontal plane and grouped into clusters. Candidate obstacle faces are extracted from each cluster using iterative RANSAC line fitting, allowing multiple planar faces to be identified within a single cluster. The extracted line segments are filtered according to geometric consistency, including minimum length, point support, continuity, and thickness, to reject spurious detections and over-extended fits.

http://www.imavs.org/

http://www.imavs.org/



(a) (b)

Figure 4: Raw pointcloud generated by the LiDAR (a), in green, and resulting obstacle segments, in blue, after processing by RANSAC (b).

The remaining segments are associated with a temporal segment map, where compatible observations update existing segments and unmatched observations initialize new ones. Repeated detections increase the confidence of a segment, while temporary occlusions are handled by maintaining segments for a limited duration before removal.

The perception layer outputs an obstacle-indexed collection of segments, each defined by its endpoints, tangent direction, outward normal, and confidence. This compact representation is passed directly to the PGFlow guidance algorithm for real-time obstacle avoidance. The raw LiDAR generated pointcloud can be observed in Fig. 4a. Fig. 4b shows the obstacle segments generated from that same pointcloud. The full LiDAR obstacle detection and processing pipeline can be observed in Fig. 5.

4 RESULTS

4.1 *Experimental setup*

Indoor experiments are conducted in the Toulouse Occitanie Drone Flight Arena at ENAC, Toulouse, France. The flight arena measures 10×10 meters, and all experiments are performed using a quadrotor equipped with the Paparazzi autopilot system [20]. The arena is equipped with a 16-camera OptiTrack motion capture system, which provides a highly accurate real-time state estimation of the vehicle.

During flight, the vehicle's position and velocity are estimated by the OptiTrack system and transmitted via XBee radio modem and fused onboard with Paparazzi's IMU for full state estimation. Obstacle perception is performed exclusively onboard using the LiDAR sensor described in the previous section.

The obstacles are placed at fixed locations in the arena, as shown in Fig. 6. However, their positions are not known *a priori* to the drone and are assumed to be non-static from the vehicle's perspective. Consequently, the drone continuously scans the environment and executes the obstacle detection algorithm at every time step.

The physical specifications of the vehicle can be found in Table 1.

Table 1: Physical specifications of the vehicle used for the experiments.

Specification	Value
Mass	1200 g
Maximum thrust	40 N
Battery	7000 mAh, 11.1 V (3-cell)
Motors	T-motor air gear 450 V2
Flight time	∼20 min
LiDAR	Livox Mid-360, 10 Hz
Companion computer	Raspberry Pi 5
Autopilot	Tawaki V2.0 (Paparazzi)

4.2 *Scenario 1: Waypoint guidance*

In the first scenario, the vehicle is required to navigate from waypoint $A = (-2, -4)$ meters to waypoint $B = (2, 4)$ meters and return to the starting position. The chosen parameters for (1) are $v_d = 0.5$ meters per second and $k_c = 1$. The maximum velocity is limited to 0.5 meters per second for safety. Two obstacles are placed at $(-1.5, -1.75)$ meters and $(-0.5, 2)$ meters.

The resulting trajectory is shown in Fig. 7a. The vehicle successfully reaches both waypoints while maintaining collision-free operation throughout the experiment.

4.3 *Scenario 2: Directional guidance*

In the second scenario, the vehicle is tasked with following a reference direction. The commanded direction is initially $(0, 1)$ and is later changed to $(1, 0)$. The chosen parameters for (1) are $v_d = 0.5$ meters per second and $k_c = 1$. The maximum velocity is again limited to 0.5 meters per second. Three obstacles are placed at $(-2, 1)$ meters, $(1.5, 0.5)$ meters, and $(-1, -2)$ meters.

The resulting trajectory is shown in Fig. 7b. The vehicle is able to track the commanded direction while avoiding the obstacles.

5 DISCUSSION

The experimental results demonstrate the feasibility of the proposed guidance system in unknown environments using only onboard LiDAR sensing. In both validation scenarios, the MAV successfully achieved its objectives, reaching the requested waypoints in the waypoint-navigation scenario and following the commanded direction in the directional-guidance scenario while avoiding all detected obstacles. However, the resulting trajectories exhibit noticeable oscillations and sharp turns rather than smooth paths. This behaviour is primarily attributed to instabilities in the obstacle detection algorithm. The estimated obstacles changed dimensions erratically between frames. These fluctuations caused corresponding variations in the repulsive forces generated by *PGFlow*, leading to oscillatory trajectory corrections.

The MAV operated at a relatively low speed of approximately $0.5\,\mathrm{m/s}$. Further validation is required before evaluating the proposed guidance system at higher flight speeds.

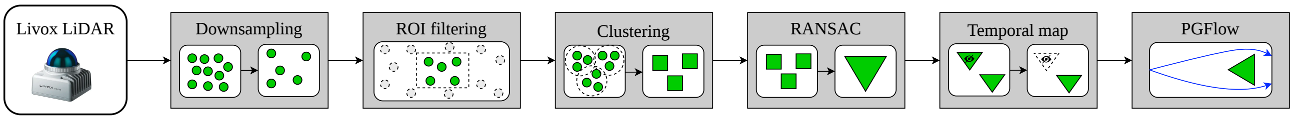
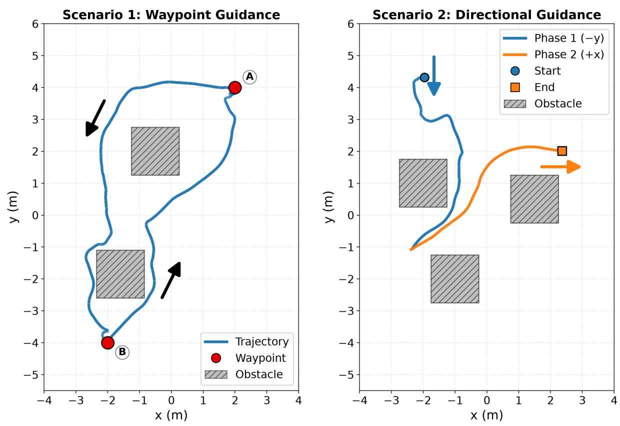


Figure 5: LiDAR's obstacle detection and processing pipeline. The raw data is captured by the Livox LiDAR sensor and then processed along the pipeline until it can be used by *PGFlow* to generate the appropriate velocity commands for the drone.

(a) (b)

Figure 6: Experimental setups for the go to waypoint, (a), and follow direction, (b) scenarios. On (a) two obstacles are placed between the waypoints the drone has to reach. On (b) the drone has to navigate around three obstacles while following the nominal direction command.



(a) (b)

Figure 7: Experimental drone trajectories for the first, (a), and second, (b), scenarios. (a) The drone starts at waypoint A, flies to waypoint B, and returns to waypoint A while moving counter-clockwise (black arrows). The trajectory (blue) successfully avoids the obstacles (zebra-patterned squares). (b) For each coloured trajectory segment, the commanded direction is indicated by the arrow of the corresponding colour. The resulting trajectory successfully avoids the obstacles.

Additionally, all obstacles considered in the experiments were static. The current approach is not designed to handle fast-moving obstacles, as it relies solely on their instantaneous positions. To address this limitation, a future extension could incorporate an obstacle motion prediction module to estimate future obstacle positions.

Rather than continuously performing low-level obstacle avoidance, the operator is only required to specify a desired direction of travel while the vehicle autonomously reacts to the surrounding environment. Although no formal user study was conducted, this mode of operation has the potential to reduce operator workload and the level of piloting expertise required, making the approach attractive for applications such as search and rescue, where personnel may not have extensive experience operating MAVs.

The proposed guidance algorithm is computationally efficient, provided that the number of obstacle panels remains limited. At each obstacle update, an $n \times n$ matrix inversion is required, resulting in a computational complexity of $O(n^3)$, where n denotes the total number of panels. Consequently, the computational cost increases rapidly as additional panels are introduced. In practice, however, the number of panels can be kept small by removing redundant or distant panels while preserving an adequate approximation of the obstacle geometry. The *PGFlow* algorithm runs comfortably at 100 Hertz. As the number of panels increases, the maximum frequency at which the obstacle matrix can be computed decreases. The relationship between this frequency and the maximum allowed number of detected panels is summarized in Table 2. Even at higher frequencies the maximum allowed number of detected panels remains sufficiently high for reliable real-time obstacle avoidance.

Table 2: Relation between the frequency at which the obstacle detection algorithm is run and the maximum number of detected panels allowed for real-time safe obstacle avoidance.

Frequency (Hz)	1	5	10	20
Maximum number of panels	2025	1100	725	506

Obstacle perception proved to be one of the most challenging components of the system. The dense LiDAR point cloud makes it difficult to segment individual obstacles and group the corresponding points, resulting in an unstable obstacle estimation algorithm, where object dimensions would

http://www.imavs.org/

http://www.imavs.org/

change erratically between frames. Furthermore, the complete point cloud is currently transmitted from the C++ LiDAR driver to the Python guidance module via TCP before any processing is performed. A more efficient implementation would perform the filtering and obstacle extraction directly within the C++ driver, reducing both communication overhead and computational cost. Machine-learning-based methods would probably be a better fit to convert the point cloud into concrete obstacles.

Several simplifying assumptions were made in this work. The vehicle state was obtained from an external OptiTrack motion capture system, eliminating the need for onboard state estimation. Furthermore, guidance was only considered in the horizontal plane, with motion along the z-axis being neglected. Consequently, the experimental validation focuses solely on the proposed guidance strategy.

Only convex obstacles were considered. Concave obstacles are more challenging to navigate around and may introduce stagnation points (local minima) in the guidance field. In practice, concave obstacles can be approximated using multiple convex panels, albeit at the cost of approximation accuracy.

Finally, the proposed system was validated using a single MAV. Extending the approach to multi-vehicle operation is a natural continuation of this work.

6 CONCLUSION

This paper presented a fully onboard integrated guidance system for MAV navigation in unknown environments, addressing two scenarios: convergence to user-defined waypoints and tracking of a commanded direction. Obstacle detection is performed using an onboard LiDAR sensor, while obstacle avoidance is ensured by a guidance algorithm [14, 15]. The nominal motion is generated using a guiding velocity vector field approach [13].

The system was experimentally validated in an indoor environment with unknown convex obstacles. In both scenarios, the MAV successfully achieved its objectives without encountering local minima, while maintaining low computational cost. It has not been tested with concave obstacles.

These results demonstrate the feasibility of fully onboard guidance and obstacle avoidance in unknown environments using only onboard sensing. This enables intuitive high-level safe control with minimal operator expertise.

REFERENCES

[1] Jesus Tordesillas, Brett T. Lopez, Michael Everett, and Jonathan P. How. FASTER: Fast and safe trajectory planner for navigation in unknown environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019. Journal version: IEEE Trans. Robotics, vol. 38, 2022, DOI 10.1109/TRO.2021.3100142.

[2] Jesus Tordesillas and Jonathan P. How. MADER: Trajectory planner in multiagent and dynamic environments. *IEEE Transactions on Robotics*, 38(1), 2022.

[3] Charbel Toumieh and Dario Floreano. High-speed motion planning for aerial swarms in unknown and cluttered environments. *IEEE Transactions on Robotics*, 2024.

[4] Kota Kondo, Mason Peterson, Nicholas Rober, Juan Rached Viso, Lucas Jia, Jialin Chen, Harvey Merton, and Jonathan P. How. DYNUS: Uncertainty-aware trajectory planner in dynamic unknown environments, 2025. arXiv preprint.

[5] Kota Kondo, Jesús Tordesillas, and Jonathan P. How. SANDO: Safe autonomous trajectory planning for dynamic unknown environments, 2026. arXiv preprint.

[6] Xin Zhou, Zhepei Wang, Chao Xu, and Fei Gao. EGO-planner: An ESDF-free gradient-based local planner for quadrotors. *IEEE Robotics and Automation Letters*, 6(2), 2021.

[7] Xin Zhou, Jiangchao Zhu, Hongyu Zhou, Chao Xu, and Fei Gao. EGO-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[8] Yunfan Ren, Fangcheng Zhu, Guozheng Lu, Yixi Cai, Longji Yin, Fanze Kong, Jiarong Lin, Nan Chen, and Fu Zhang. SUPER: Safety-assured high-speed navigation for MAVs. *Science Robotics*, 10(98):eado6187, 2025.

[9] Marvin Harms, Martin Jacquet, and Kostas Alexis. Safe quadrotor navigation using composite control barrier functions. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6343–6349, 2025.

[10] Jihao Huang, Xuemin Chi, Jun Zeng, Zhitao Liu, and Hongye Su. GeoPro-VO: Dynamic obstacle avoidance with geometric projector based on velocity obstacle, 2024. arXiv preprint.

[11] Gang Zhong, Yupu Liu, Sen Du, Fei Wang, Jinlun Zhou, and Honghai Zhang. 3D RVO-enhanced multi-agent deep reinforcement learning for collision avoidance in urban structured airspace. *Aerospace Science and Technology*, 2025.

[12] Nikita Vaibhav Pavle, Shrreya Rajneesh, Rakesh Kumar Sahoo, and Manoranjan Sinha. Obstacle avoidance of UAV in dynamic environments using direction and velocity-adaptive artificial potential field, 2025. arXiv preprint.

[13] João Machado, Matthieu Verdoucq, Christophe Jouffrais, and Murat Bronz. Swarm choreography made simple: Superposed guiding vector fields for rigid formation path following. In *14th International Conference on Unmanned Aircraft Systems (ICUAS)*, Corfu, Greece, 2026.

[14] Zeynep Bilgin, Ilkay Yavrucuk, and Murat Bronz. Urban air mobility guidance with panel method: Experimental evaluation under wind disturbances. *Journal of Guidance, Control, and Dynamics*, 2024.

[15] Farbod Fahimi. Obstacle avoidance using harmonic potential functions. In Farbod Fahimi, editor, *Autonomous Robots: Modeling, Path Planning, and Control*, pages 1–49. Springer US, 2009.

[16] G. C. S. Cruz and P. M. M. Encarnação. Obstacle avoidance for unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 65(1–4):203–217, 2012.

[17] Jasmin Velagić, Lamija Vuković, and Belma Ibrahimović. Mobile robot motion framework based on enhanced robust panel method. *International Journal of Control, Automation and Systems*, 18(5):1264–1276, 2020.

[18] J. Rudzki, Zeynep Bilgin, and Ilkay Yavrucuk. Enhancing unmanned rotorcraft guidance with LIDAR and ADS-B integrated PGFlow algorithm. In *50th European Rotorcraft Forum (ERF)*, Germany, 2024.

[19] Zeynep Bilgin, Ilkay Yavrucuk, and Murat Bronz. Urban air mobility guidance with panel method: Experimental evaluation under wind disturbances. *Journal of Guidance, Control, and Dynamics*, 47(6):1080–1096, 2024.

[20] Gautier Hattenberger, Murat Bronz, and Michel Gorraz. Using the Paparazzi UAV System for Scientific Research. In *International Micro Air Vehicles Conference and Flight Competition*, pages 247–252, Delft, Netherlands, 2014.

http://www.imavs.org/