

TDP: CW-Lion (China), Indoors Category

Jialiang Wang¹*, Yanqi Zhao¹, Zongzhou Wu¹, Xinyu Guo², Handa Wang², Haotian Chen² and Jerry Tang³

¹The Chinese University of Hong Kong, Hong Kong SAR, China

²Wuhan University, Hubei, China

³Hong Kong Centre for Logistics Robotics, Hong Kong SAR, China

ABSTRACT

This Technical Description Paper outlines the CW-Lion team's solution for the IMAV 2025 indoor competition. The lightweight quadrotor, featuring LiDAR, cameras, IMU, and microphone arrays, supports autonomous flights in GNSS-denied environments.

Mission 1 uses a deep learning model for point cloud-based tunnel detection and localization, achieving 10 cm accuracy through voxel down-sampling, PCA, and depth clustering.

Mission 2 utilizes a compliant Chinese brush for whiteboard line drawing, with LiDAR odometry and PD control ensuring stable contact.

Mission 3 employs a microphone array for sound localization via time difference of arrival, yielding azimuth angles for target tracking.

Mission 4 enables landing on smoke-obscured moving platforms using arc-based ellipse detection, PnP estimation, and PCA trajectory fitting for robust descent.

1 PLATFORM OVERVIEW

The autonomous MAV platform is engineered for precise positioning, navigation, and mapping in GNSS-denied environments, optimized for IMAV indoor scenarios. As illustrated in Figure 1, the platform is constructed from lightweight materials, such as carbon fiber, PVC, and TPU, ensuring durability and flexibility. The MAV has a maximum take-off weight of 1850 g, a payload capacity of 1000 g, and compact dimensions of 216 mm × 216 mm × 95 mm. Powered by a Sony 18650 3000 mAh, 22.2 V 6S battery, it achieves a maximum flight time of 16 minutes outdoors and 15 minutes indoors. The platform is equipped with multi-modal sensors, including an OMNIVISION OS02C10 camera for visual data, a Livox Mid360 LiDAR for 3D ranging and mapping, and a BMI088 IMU for motion sensing. The comprehensive design of the proposed indoor MAV is depicted in Figure 1.

The onboard computing unit, an NVIDIA Jetson Orin NX, incorporates a dual-core NVIDIA ARMv8 CPU, a 256-core NVIDIA GPU, and 16 GB of DDR4 memory. This con-

figuration enables real-time sensor data processing and execution of sophisticated algorithms, such as LiDAR-based Simultaneous Localization and Mapping (SLAM) and motion planning. To facilitate fully autonomous flight, the MAV operates in offboard mode. Leveraging LiDAR-based SLAM for precise localization, mission-specific perception algorithms are executed to determine the MAV's intended actions. These commands are transmitted to the motion planner, which generates an obstacle-free trajectory to the designated location.

To enable fully autonomous flight, the MAV operates in offboard mode. Using LiDAR SLAM for accurate localization, various mission-related perception algorithms are executed to determine the desired action of the MAV. These commands are subsequently sent to motion planner, which generates an obstacle-free trajectory towards the planned location.

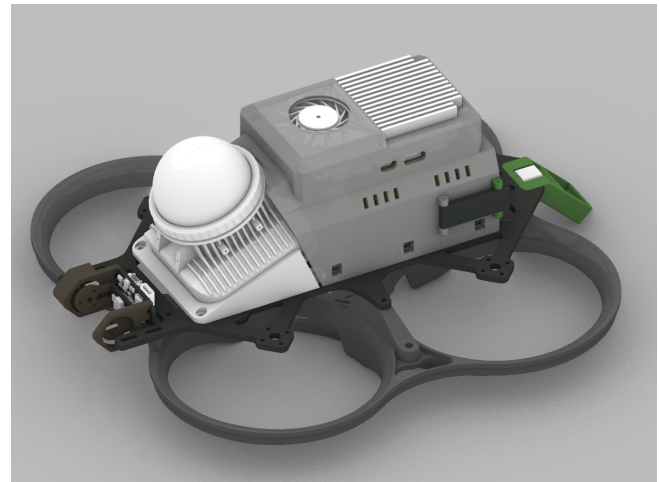


Figure 1: Platform design

2 SOLUTION TO MISSION 1

Mission 1 aims to navigate through five 0.5 m × 1 m tunnels spanning a total length of 2 m, which requires accurate trajectory setpoints. To achieve this, we design a 3D object detection framework based on point clouds. Given a point cloud $\mathbf{P} \in \mathbb{R}^{n \times 3}$, the algorithm outputs 3D poses of the center point $\mathbf{P}_i \in SE(3)$, $i = 1, 2, 3, 4, 5$.

As illustrated in Figures 2, we first utilize the PV-RCNN++ [1] model to detect the candidate target box point clouds. Subsequently, Principal Component Analysis (PCA)

*Email address: jialiang.wang@link.cuhk.edu.hk

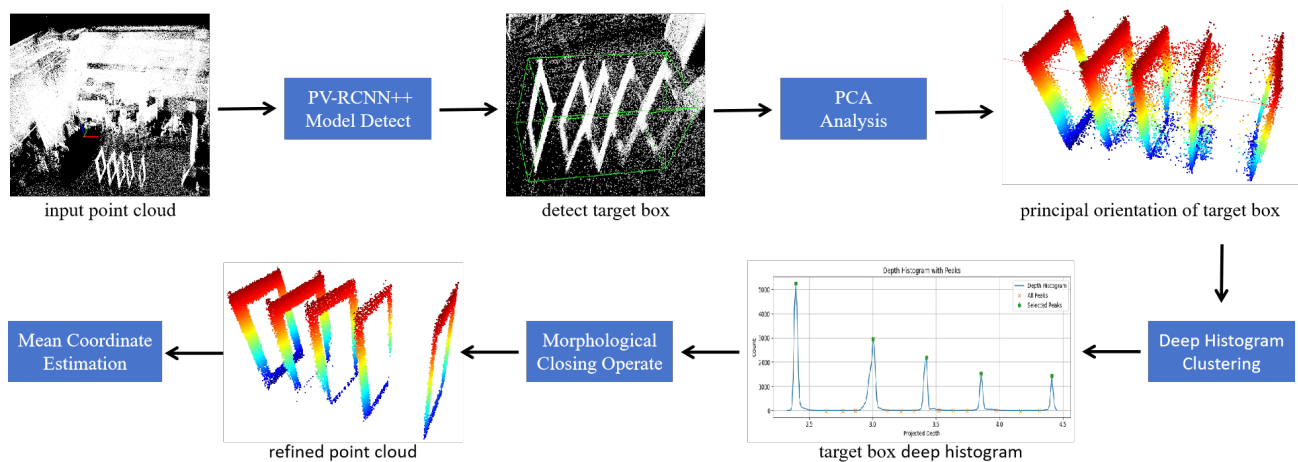


Figure 2: Overview of the tunnel detection algorithm

is performed on the extracted point clouds to estimate their orientations. Based on the depth distribution along the estimated orientation, we apply clustering to separate the point clouds of the five boxes. Finally, each clustered point cloud is denoised, and the center coordinates of each box are obtained by computing the mean of the remaining points. The following provides the detailed implementation of this procedure.

We utilize point cloud data generated by Fast-LIO [2] mapping as the input to our framework. Since the accumulated multi-frame point clouds contain a very large number of points, direct processing would be computationally expensive and memory-intensive. To address this, we first apply voxel-based downsampling with a resolution of 1 cm, which not only reduces the data volume but also preserves the essential geometric structures required for subsequent detection. Following this preprocessing, a self-trained PV-RCNN++ model is employed to perform the target box detection task.

The PV-RCNN++ model is particularly well-suited for this scenario as it integrates voxel-based feature extraction with point-wise refinement, thereby capturing both large-scale geometric context and fine-grained local details. Compared with the original PV-RCNN [3] model, PV-RCNN++ introduces enhanced feature interaction mechanisms and a more efficient refinement strategy, which lead to improved detection accuracy and computational efficiency. Its multi-scale feature aggregation mechanism allows the model to effectively encode information across different resolutions, while the Region-of-Interest (RoI) refinement module further improves localization accuracy. These advancements enable PV-RCNN++ to maintain greater robustness in cluttered or partially occluded environments, ensuring reliable detection even when the target boxes overlap or are partially hidden.

The detection output of the model is represented as a seven-dimensional vector, which explicitly encodes the center coordinates (x, y, z) , the dimensions of the target box (l, w, h) , and the horizontal rotation angle r in radians. This

compact yet expressive representation provides not only spatial localization but also orientation information, which serves as a crucial foundation for the subsequent steps of point cloud extraction, orientation estimation, and center localization.

Subsequently, we extract the partial point clouds corresponding to the detected target boxes from the input data. Since most point cloud frames are captured from the frontal viewpoint of the target boxes, the preceding boxes inevitably cause partial occlusions to the subsequent ones. This occlusion introduces a significant number of redundant and noisy points that overlap across different target boxes, thereby complicating the separation of individual box instances. To mitigate this issue, an outlier removal procedure is first applied to suppress scattered noise and retain the structural consistency of each target box point cloud. Following this denoising step, Principal Component Analysis (PCA) is conducted to analyze the geometric distribution of the remaining points. By identifying the dominant eigenvector of the covariance matrix, the principal orientation of the target box is estimated, providing a three-dimensional direction vector that serves as the basis for subsequent point cloud separation and center localization.

To further achieve a reliable separation of the point clouds corresponding to the five target boxes along the estimated orientation, we conduct a detailed analysis of the depth distribution within each detected target box. Specifically, the depth values of all points are projected onto the orientation axis, and a one-dimensional histogram is constructed to characterize the distribution. This histogram typically exhibits multiple prominent peaks, with each peak corresponding to a distinct target box along the depth direction. To isolate individual boxes, we select the nearest significant peak and apply a filtering process to retain only those points whose depth values lie within a predefined tolerance range around the peak (e.g., ± 0.05 m). The retained points are then aggregated, and their average coordinates are calculated to determine the center of the corresponding target box. By focusing on the dom-

inant depth peaks, this approach not only suppresses random outliers and spurious measurements but also mitigates the effects of occlusion and overlapping point distributions. Consequently, it ensures a more robust separation of adjacent boxes, making it particularly effective in scenarios where the depth distribution is corrupted by noise or affected by partial visibility.

Finally, for each separated target box point cloud, we perform an outlier removal step to suppress residual scattered points and preserve the structural integrity of the data. Since the point clouds corresponding to the target boxes may suffer from partial incompleteness caused by occlusion, sparse sampling, and sensor noise, additional refinement is necessary to achieve reliable center estimation. To this end, we iteratively apply morphological closing operations until convergence, which progressively eliminates small voids, smooths irregular boundaries, and improves the overall continuity of the point cloud representation. This iterative process ensures that local discontinuities and fragmented structures are effectively corrected, thereby enhancing the geometric integrity of the target box. Once the refined point cloud is obtained, we compute the mean of the remaining point coordinates to determine the final center location of each target box, providing a stable and robust estimation that is less sensitive to noise and missing data.

In terms of inference, due to the limitations of sparse 3D convolutions, a dynamic graph inference mode is adopted. On the Orin platform, the initial inference requires approximately 4 seconds, while subsequent inferences take about 2 seconds each. Regarding accuracy, since no large-scale quantization is applied to the model weights and activations, inference is performed using the original data type (FP32), and the error of the estimated center coordinates is within 10 cm along each spatial direction. In terms of memory consumption, the CPU and GPU share a unified memory, with the overall usage maintained at around 4 GB.

3 SOLUTION TO MISSION 2

The objective of Mission 2 is to draw a continuous horizontal line on a whiteboard that is mounted horizontally on the wall. The longer the line, the higher the score. Since the competition rules do not specify the type of writing tool, our system adopts a Chinese brush as the end-effector. To ensure stable contact, the brush is mounted with a spring-based compliant mechanism, which provides a gentle and consistent contact force when approaching the whiteboard. This design avoids direct collision with the surface and reduces the risk of line discontinuity caused by small flight disturbances. For navigation and localization, the system mainly relies on the mid360 LiDAR odometry integration, which provides global pose estimation. Based on the competition's disclosed whiteboard coordinates (or on-site specification), the MAV first flies to the target neighborhood. It then adjusts its attitude and lateral position, approaching the whiteboard slowly along

the board's normal direction. Contact is detected through the slight compression of the compliant mechanism, after which the MAV proceeds with a uniform horizontal motion to draw the line. Once the target length is achieved or any anomaly is detected, the MAV immediately retracts and hovers. This workflow, without additional sensor dependencies, leverages the compliance of the brush and the tolerance of the elastic mechanism to ensure both line continuity and flight safety.

3.1 Mechanical Design for Line Drawing

Height control in the drawing task is relatively straightforward, but maintaining a consistent lateral distance to the board presents challenges. If the writing tool is mounted as a rigid body, the drone must ensure proper contact force with the board. This would require implementing explicit force control in addition to trajectory tracking, which complicates the control architecture.

A more effective approach is to simply rely on trajectory tracking and instead design the drawing mechanism so that it can adapt passively to lateral deviations without imposing significant contact or friction forces back onto the MAV. This means the contact between the tool and the board must be soft. In this way, even with small lateral deviations, the MAV can continue to track its path while the tool maintains writing contact without destabilizing the vehicle.

Based on these requirements, we selected a Chinese paint brush as the writing tool. The brush is made of wool with a head length of approximately 8 cm. Its soft bristles deform under varying levels of contact, allowing consistent writing quality across a range of contact depths. Figure 3 demonstrates the brush's ability to tolerate variations in contact distance through deformation.



Figure 3: Chinese paint brush as the chosen writing tool. The figure shows the MAV emulating drawing on the wall using the paint brush. The paint brush can maintain good contact with the wall despite lateral position variations.

The brush is mounted at the middle of the drone's body, so that the contact and friction forces generate the smallest possible moment arm relative to the MAV's center of mass. This minimizes the disturbance transmitted back to the con-

trol system and allows stable trajectory tracking while writing.

3.2 Control and Drawing Logic

In this mission, the brush is mounted horizontally on the MAV body, and the MAV moves forward along its flight direction to draw on a vertical whiteboard mounted on the wall. The control strategy focuses on attitude alignment, lateral distance control, and forward trajectory tracking. The MAV first uses the mid360 LiDAR odometry to obtain its global pose and flies to the vicinity of the whiteboard. It then adjusts its heading so that the MAV's forward direction aligns with the drawing direction, while keeping the brush axis parallel to the whiteboard surface. The desired yaw angle can be expressed as

$$\psi_{des} = \arctan 2(t_y, t_x)$$

where $\mathbf{t} = [t_x, t_y]$ is the unit vector of the whiteboard drawing direction in the horizontal plane. The yaw angle ψ_{des} aligns the MAV's forward direction with the drawing direction. Pitch and roll angles are maintained at level to ensure the brush remains horizontal relative to the whiteboard, preventing skewed lines.

Lateral distance control is critical for this task. The MAV adjusts its lateral distance (normal to the whiteboard) to ensure the brush lightly contacts the whiteboard. Let y denote the MAV's lateral position, y_{board} the whiteboard surface location, and d_{pen} the lateral offset of the brush relative to the MAV. The desired lateral position is then

$$y_d = y_{board} - d_{pen}$$

The lateral contact force can be implemented or estimated using the brush's compliance mechanism. A PD control law can be applied:

$$F_y = K_p(y_d - y) + K_d(\dot{y}_d - \dot{y})$$

where y and $\dot{y}$ are the MAV's current lateral position and velocity, K_p and K_d are control gains, and F_y is the output control force to maintain gentle contact between the brush and the whiteboard, ensuring continuous lines without collisions.

Once the lateral contact is established, a forward trajectory is generated to draw horizontally along the brush axis:

$$x_{des}(t) = x_{start} + v_{draw}t$$

where $x_{des}(t)$ is the desired position along the forward direction, x_{start} is the starting contact point.

The trajectory is tracked using a hierarchical position-attitude controller to ensure the brush draws continuous and smooth lines on the whiteboard surface. During the drawing process, the system continuously monitors lateral force and forward trajectory errors. If any lateral deviation or trajectory error exceeds a safety threshold, an immediate retreat along

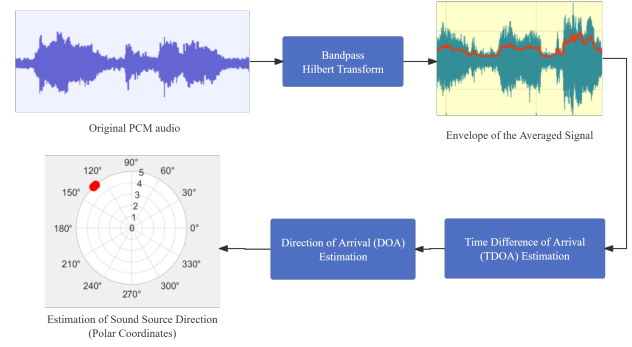


Figure 4: Overview of the sound localization algorithm

the lateral direction is triggered, and the MAV hovers at a safe distance, ensuring both MAV and whiteboard safety. This control strategy leverages the horizontal brush layout and the compliance mechanism to achieve stable, continuous wall-surface drawing without additional sensors.

4 SOLUTION TO MISSION 3

Mission 3 requires the MAV to locate a box of a certain size, which has a speaker inside producing the sound of somebody crying out for help. We devise a sound positioning algorithm to accurately orient the vehicle towards the box. As is shown in Figure 4, given an 8-channel, 16 bit PCM sound file with a 16 kHz sampling rate, the algorithm outputs the azimuth angle of the target box relative to the drone in the polar coordinate system $\theta \in (0^\circ, 360^\circ)$. The MAV then tracks the target angle and locates the box to initiate med-kit release.

As the visual localization of the box is relatively straightforward, we mainly present the sound localization algorithm in this paper. This algorithm is designed to detect and locate acoustic events (such as the word "Help") using a circular microphone array and consists of several stages: signal acquisition, event detection, time difference of arrival (TDOA) estimation, and direction of arrival (DOA) estimation via geometric fitting.

4.1 Signal Acquisition and Preprocessing

The audio data is read from a multi-channel PCM file. Let $M = 6$ be the number of microphones used. The recorded signal at the m -th microphone is denoted as $x_m(t)$. The signals are bandpass-filtered to retain frequencies between f_{low} and f_{high} :

$$H(z) = \frac{\sum_{k=0}^K b_k z^{-k}}{\sum_{k=0}^K a_k z^{-k}}$$

where $K = 4$ is the filter order, and the passband is set to [300,7600] Hz to avoid aliasing and reduce noise.

4.2 Event Detection

The envelope of the averaged signal across microphones is computed using the Hilbert transform:

$$e(t) = \left| \mathcal{H} \left(\frac{1}{M} \sum_{m=1}^M x_m(t) \right) \right|$$

A moving average is applied to smooth the envelope:

$$e_{\text{smooth}}(t) = \frac{1}{W} \sum_{k=-W/2}^{W/2} e(t+k)$$

where $W = 500$ is the window length. Events are detected when $e_{\text{smooth}}(t)$ exceeds a threshold, and are merged or discarded based on duration and proximity constraints.

4.3 Time Difference of Arrival (TDOA) Estimation

For each detected event, the generalized cross-correlation with phase transform (GCC-PHAT) is used to estimate TDOA between microphone pairs [4]. For microphones i and j , the cross-power spectrum is:

$$S_{ij}(f) = X_i(f) \cdot X_j^*(f)$$

The PHAT-weighted cross-spectrum is:

$$S_{ij}^{\text{PHAT}}(f) = \frac{S_{ij}(f)}{|S_{ij}(f)| + \epsilon}$$

The time-domain cross-correlation is obtained via inverse Fourier transform:

$$R_{ij}(\tau) = \mathcal{F}^{-1} \{ S_{ij}^{\text{PHAT}}(f) \}$$

The TDOA τ_{ij} is the time lag that maximizes $R_{ij}(\tau)$

4.4 Direction of Arrival (DOA) Estimation

The microphones are arranged uniformly on a circle of radius $r = 0.03975$ m. The position of the m -th microphone is:

$$\mathbf{p}_m = r [\cos \theta_m, \sin \theta_m, 0]^T, \quad \theta_m = \frac{2\pi(m-1)}{M}$$

For a sound source from direction ϕ , the theoretical TDOA between microphones i and j is:

$$\tau_{ij}^{\text{theory}}(\phi) = \frac{(\mathbf{p}_j - \mathbf{p}_i) \cdot \mathbf{d}(\phi)}{c}$$

where $\mathbf{d}(\phi) = [\cos \phi, \sin \phi, 0]^T$ is the direction vector, and $c=343$ m/s is the speed of sound. The best estimate $\hat{\phi}$ is found by minimizing the mean squared error between theoretical and observed TDOAs:

$$\hat{\phi} = \arg \min_{\phi} \sum_{i < j} \left(\tau_{ij} - \tau_{ij}^{\text{theory}}(\phi) \right)^2$$

A grid search over $\phi \in [0, 2\pi]$ with 1° resolution is used to find the optimal azimuth.

5 SOLUTION TO MISSION 4

Mission 4 requires autonomous landing on a $1 \text{ m} \times 1 \text{ m}$ moving platform obscured by smoke, necessitating precise estimation of the board's 3D center coordinates. To accomplish this, we developed a vision-based shape detection and localization algorithm. Given an input image $\mathbf{I}$ and camera parameters (intrinsic matrix $\mathbf{K}$, distortion coefficients $\mathbf{d} = [k_1, k_2, p_1, p_2, p_3]$), the algorithm outputs the board's 3D center position $\mathbf{P}_{\text{camera}}^{\text{board}}$ in the camera coordinate system. Our solution comprises three stages: identifying elliptical arcs and H-shaped features, estimating the board's position via keypoints, and guiding the MAV to the platform using position control. Figure 5 illustrates the detection and localization algorithm.

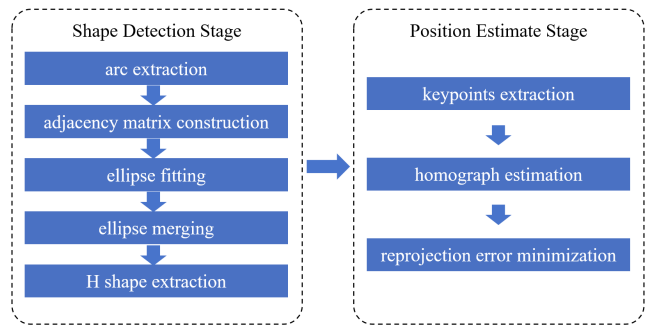


Figure 5: Overview of the detection and localization algorithm for autonomous landing

5.1 Shape Detection

Smoke-induced partial occlusions complicate detection of the board's full shape. As projected circles appear as ellipses and may be segmented into arcs by smoke, we adopt the AAMED arc-based ellipse detection method [5]. Initially, edge pixels are extracted using the Canny edge detector [6] and contour tracing. Contours are then polygonized using the Douglas-Peucker algorithm [7] and segmented into arcs based on convexity and curvature.

Post-extraction, an arc adjacency matrix $\mathbf{L}$ is constructed, where L_{ij} indicates whether arc j can follow arc i , based on endpoint distance, curvature, and convexity consistency [8]. The matrix $\mathbf{L}$ is traversed to identify valid arc connections. A set of arcs $\{r_1, \dots, r_m\}$ is a valid ellipse candidate if it meets:

$$\begin{cases} L_{r_1 r_2} = L_{r_2 r_3} = \dots = L_{r_{m-1} r_m} = 1 \\ \sum_{k=1}^m \Delta \theta_{r_k} \geq \Theta_{\min} \end{cases}$$

where $\Delta \theta_{r_k}$ is the angular change of arc r_k , and $\Theta_{\min}$ is the angular coverage threshold.

A least-squares method [9] estimates ellipse parameters from these candidates, with the optimization objective:

$$\min_{\alpha} \sum_{(x,y) \in P} (z(x,y) - \alpha)^2$$

where $\alpha = [A, B, C, D, E, F]^T$ represents ellipse parameters, $z(x, y) = [x^2, xy, y^2, x, y, 1]^T$, and P is the set of arc pixels. The smallest eigenvalue of $\sum_{(x,y) \in P} z(x, y) z(x, y)^T$ yields the best-fit ellipse. The ellipse's geometric attributes are then computed:

$$\begin{cases} \theta = \frac{1}{2} \arctan\left(\frac{2B}{A-C}\right) \\ (x_0, y_0) = \left(\frac{BE-2CD}{4AC-B^2}, \frac{BD-2AE}{4AC-B^2}\right) \\ a = \sqrt{\frac{-2(Ax_0^2+Bx_0y_0+Cy_0^2+Dx_0+Ey_0+F)}{A+C-\sqrt{(A-C)^2+B^2}}} \\ b = \sqrt{\frac{-2(Ax_0^2+Bx_0y_0+Cy_0^2+Dx_0+Ey_0+F)}{A+C+\sqrt{(A-C)^2+B^2}}} \end{cases}$$

where θ is the rotation angle, (x_0, y_0) is the center, and a, b are the semi-major and semi-minor axes.

Fitted ellipses are validated by clustering based on parameter similarity, retaining one ellipse per cluster. Assuming the largest ellipse corresponds to the board, we select it for further processing. The region within the ellipse is extracted, binarized adaptively, and subjected to connectivity analysis to identify the H shape. This shape detection provides keypoints for position estimation, as described below.

5.2 Position Estimation

The Perspective-n-Point (PnP) algorithm [10] estimates the platform's 3D position using 3D points and their 2D image projections. 2D points are derived from the detection results. If an H-shape is detected, 12 corner points are extracted using Canny edge detection [6] and Douglas-Peucker simplification [7]. If an ellipse is detected, five keypoints are used: the center and the top, bottom, left, and right points. 3D points are defined in a world coordinate system anchored to the platform, with the board's center as the origin and its surface as the xOy plane. Known board dimensions allow determination of 3D point coordinates.

As keypoints are coplanar, an efficient PnP solver is used. An initial estimate is computed using a homography-based method with four points:

$$\begin{cases} X_i h_{11} + Y_i h_{12} + h_{13} - u_i X_i h_{31} - u_i Y_i h_{32} - u_i h_{33} = 0 \\ X_i h_{21} + Y_i h_{22} + h_{23} - v_i X_i h_{31} - v_i Y_i h_{32} - v_i h_{33} = 0 \\ \sum_{i=1}^3 \sum_{j=1}^3 h_{ij} = 1 \end{cases}$$

where $\mathbf{H} = (h_{ij})_{3 \times 3}$ is the homography matrix, $\mathbf{X}_i = (X_i, Y_i, 0)$ and $\mathbf{u}_i = (u_i, v_i)$ are the 3D and 2D point coordinates, respectively. This initial solution is refined via Levenberg-Marquardt optimization, minimizing reprojection error:

$$\mathbf{R}, \mathbf{t} = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_i \|\mathbf{u}_i - \pi(\mathbf{K}, \mathbf{d}, \mathbf{R}, \mathbf{t}, \mathbf{X}_i)\|^2$$

where $\pi(\cdot)$ is the projection function, and $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector from the world to the camera coordinate system. The board's 3D center position is

$\mathbf{P}_{\text{camera}}^{\text{board}} = \mathbf{t}$. This position estimation enables robust landing, as detailed below.

5.3 Autonomous Landing

We present a line-tracking-based method for dynamic target tracking and landing, ensuring robustness despite temporary visual occlusions. This approach uses historical position data to infer the target's trajectory, enabling predictive corrections and stable descent under partial visibility. The algorithm employs a sliding window to collect historical target positions over a configurable interval. With at least five data points, PCA fits the trajectory to a straight line. The covariance matrix yields the line's center of mass and principal direction, providing a robust estimate of the target's motion path.

Proportional control in the XY plane, scaled by a gain factor, aligns the MAV with the platform's center. Track error correction enhances lateral stability by projecting the MAV's position onto the fitted line and adjusting it to mitigate deviations from abrupt maneuvers or brief occlusions. The system's response to lost visual tracking depends on the time elapsed since the last valid observation. For short-term losses, the current setpoint is projected onto the latest fitted line, enabling predictive hovering along the inferred trajectory. For extended losses, descent is paused, and the MAV hovers at the line's center of mass or last known position until visual recovery, ensuring safety.

Position updates are smoothed by limiting single-step changes to under 0.2 m, preventing destabilizing abrupt commands. Z-axis descent is regulated, halting at a predefined altitude or during visual loss to prioritize safety. Trajectory extrapolation reduces susceptibility to noise and smoke-induced signal loss, enhancing robustness. Simulations, as shown in Figure 6, confirm stable landings on linearly moving platforms.

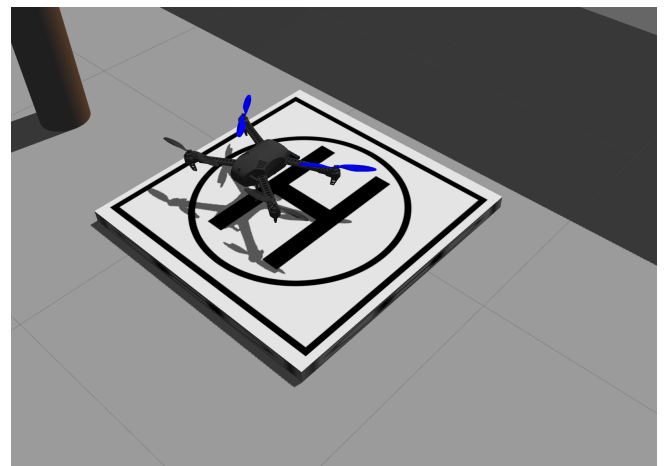


Figure 6: Autonomous landing on a moving platform in simulation

6 CONCLUSION

In this paper, we have detailed the design and algorithmic solutions for the IMAV 2025 indoor competition. By integrating advanced sensing modalities, including LiDAR for SLAM-based mapping, visual cameras for object detection, and microphone arrays for sound localization, the platform achieves high-precision autonomy in GNSS-denied and cluttered environments.

Preliminary evaluations indicate promising performance, with tunnel localization errors under 10 cm, drawing continuity maintained via compliant writing tool, sound azimuth accuracy within 10 degrees, and stable landings on moving targets. Future enhancements could include multi-sensor fusion for enhanced robustness and adaptive learning for varying environmental conditions.

REFERENCES

- [1] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN++: point-voxel feature set abstraction with local vector representation for 3d object detection. *Int. J. Comput. Vis.*, 131(2):531–551, 2023.
- [2] Wei Xu and Fu Zhang. FAST-LIO: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics Autom. Lett.*, 6(2):3317–3324, 2021.
- [3] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: point-voxel feature set abstraction for 3d object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10526–10535. Computer Vision Foundation / IEEE, 2020.
- [4] Charles H Knapp and G Clifford Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, 1976.
- [5] Cai Meng, Zhaoxi Li, Xiangzhi Bai, and Fugen Zhou. Arc adjacency matrix-based fast ellipse detection. *IEEE Transactions on Image Processing*, 29:4406–4420, 2020.
- [6] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.
- [7] Dilip K. Prasad, Chai Quek, Maylor K.H. Leung, and Siu-Yeung Cho. A parameter independent line fitting method. In *The First Asian Conference on Pattern Recognition*, pages 441–445, 2011.
- [8] Dilip K. Prasad, Maylor K.H. Leung, and Siu-Yeung Cho. Edge curvature and convexity based ellipse detection method. *Pattern Recognition*, 45(9):3204–3221, 2012. Best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA’2011).
- [9] Dilip K. Prasad, Maylor K.H. Leung, and Chai Quek. Ellifit: An unconstrained, non-iterative, least squares based geometric ellipse fitting method. *Pattern Recognition*, 46(5):1449–1465, 2013.
- [10] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision*, 81(2):155–166, February 2009.