Real-Time Multi-Modal Active Vision for Object Detection on UAVs Equipped With Limited Field of View LiDAR and Camera

Chuanbeibei Shi, *Student Member, IEEE*, Ganghua Lai, Yushu Yu, Mauro Bellone, and Vincezo Lippiello, *Senior Member, IEEE*

***Abstract*—This letter aims to solve the challenging problems in multi-modal active vision for object detection on unmanned aerial vehicles (UAVs) with a monocular camera and a limited Field of View (FoV) LiDAR. The point cloud acquired from the low-cost LiDAR is firstly converted into a 3-channel tensor via motion compensation, accumulation, projection, and up-sampling processes. The generated 3-channel point cloud tensor and RGB image are fused into a 6-channel tensor using an early fusion strategy for object detection based on a Gaussian YOLO network structure. To solve the low computational resource problem and improve the real-time performance, the velocity information of the UAV is further fused with the detection results based on an extended Kalman Filter (EKF). A perception-aware model predictive control (MPC) is designed to achieve active vision on our UAV. According to our performance evaluation, our pre-processing step improves other literature methods running time by a factor of 10 while maintaining acceptable detection performance. Furthermore, our fusion architecture reaches 94.6 mAP on the test set, outperforming the individual sensor networks by roughly 5%. We also described an implementation of the overall algorithm on a UAV platform and validated it in real-world experiments.**

***Index Terms*—Aerial systems: applications, perception-action coupling, sensor fusion.**

I. Introduction

ACTIVE vision refers to the idea of taking perception requirements into consideration in control strategies seeking the most information content [1]. This idea has been applied in various fields of robotics for the purposes of object detection, localization or flight through complex environments [2], [3], [4] to keep the target visible while the robots are moving. In this letter, we focus on realizing a real-time multi-modal active vision system for object detection on a small-scale UAV platform. The system is mainly composed of two parts: (i) multi-modal data fusion for object detection, and (ii) a perception-aware MPC framework to achieve active vision.

The sensor setup for object detection/tracking has to take into account that RGB images captured by monocular cameras contain color and texture information, but they are strongly affected by illumination and lack of depth information. Thus, vision-based methods [5], [6] may suffer from degraded performance in adverse environments, such as those with complete darkness. Conversely, point clouds acquired by a LiDAR (Light Detection And Ranging) are produced almost independently from the ambient illumination conditions and can provide accurate 3D geometric information. The idea of combining the strength of different sensor types makes LiDAR and camera data fusion-based methods a promising approach [7]. In the fields of object detection, convolutional neural networks (CNNs) have made remarkable achievements based on RGB data [8], [9]. As one might expect, extending CNNs to multi-modal data has become a popular approach. Several studies [10], [11] have demonstrated that CNNs have brilliant performance in processing the LiDAR-camera fusion data.

Besides, in recent years, there has been a growing interest in perception-aware motion planning and control of UAVs [12]. The methodology can mainly be divided into two groups: the planning-based method, and the control-based method. The former is useful for global navigation of UAVs in unknown environments by combining path planning algorithms [4], [13], [14]. While the latter is typically based on constrained control methodology by defining the perception constraints and incorporating them into the control problem formulation [12], [15], [16], [17]. MPC is a useful tool for dealing with the perception constraints in perception-aware control [18], which can obtain improved results by integrating perception, planning and control into a single problem [2].

In this letter, we aim to utilize a sensor suite that combines a monocular camera and a limited FoV LiDAR sensor. With regards to the LiDAR sensor, solid-state LiDAR sensors have recently garnered increasing interest due to their relatively low cost, lightweight design, long-range scanning capabilities and non-repetitive scan patterns that feature an increasing density of point clouds over time [19]. These characteristics make them

Manuscript received 25 March 2023; accepted 20 August 2023. Date of publication 28 August 2023; date of current version 5 September 2023. This letter was recommended for publication by Associate Editor N. Chiba and Editor T. Ogata upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62173037, in part by the National Key R. D. Program of China, in part by the State Key Laboratory of Robotics and Systems (HIT), and in part by the Beijing Institute of Technology Research Fund Program for Young Scholars. *(Chuanbeibei Shi and Ganghua Lai contributed equally to this work.) (Corresponding author: Yushu Yu.)*

Chuanbeibei Shi, Ganghua Lai, and Yushu Yu are with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: becka0shi@gmail.com; 1216621137@qq.com; yushu.yu@bit.edu.cn).

Mauro Bellone is with the Smart City Center of Excellence, Tallinn University of Technology 19086 Tallinn, Estonia (e-mail: mauro.bellone@taltech.ee).

Vincezo Lippiello is with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80125 Naples, Italy (e-mail: vincenzo.lippiello@unina.it).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3309575, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3309575

well-suited for use in UAVs, which are load and power sensitive. Therefore, we chose to use a solid-state LiDAR in our work. In fact, solid-state LiDAR sensors have a smaller FoV compared to traditional versions. Practical solutions for obtaining a larger FoV include increasing the number of LiDARs, adjusting their setup position and orientation [20], or adding additional servo motors to adjust the FoV of the sensors [21]. However, more sensors or motors are also heavier and the amount of load that UAVs can carry is limited. Here, we present a perception-aware MPC that considers the constraints from multi-sensors to realize active vision and keep the observed object in the FoV. Besides, there are still some challenges to be addressed due to the unique requirements of small-scale UAVs and the real-time performance:

- Data from different sensors can vary in frequency and processing speed. For the point cloud acquired from a non-repetitive solid-state LiDAR, it takes time to accumulate to form dense point clouds for further up-sampling and detection.
- Small-scale UAVs typically have limited computing capabilities, while the processing of point clouds has high computational complexity.
- For detection purposes, it is crucial to keep the regions of interest (ROI) always within the limited FoV of the multi-modal sensors. To apply active vision, e.g., the perception-based MPC, the detection algorithm needs to output the position of the ROI quickly enough to satisfy the real-time requirements of control.

The purpose of this letter is to overcome the aforementioned challenges. Here, we provide an integrated multi-modal sensor fusion active vision algorithm for small-scale UAVs, which incorporates state-of-the-art methods. This algorithm will address the real-time performance requirements and take into account the specific constraints of small-scale UAVs. In summary, the main contribution of this letter is listed as follows.

- We propose a novel approach to fuse the information from solid-state LiDAR, camera, and GPS (Global Positioning System) /IMU (Inertial Measurement Unit) on UAVs. Under this framework, we can obtain the motion-compensated dense point cloud and improve the real-time performance of the LiDAR-camera fusion-based detection algorithm by using a velocity-based extended Kalman Filter (vEKF).
- A perception-aware MPC is designed to tackle perception constraints from multi-sensors and to guarantee that the ROI simultaneously falls into the FoV of multiple sensors. The perception constraints for the 2D camera and 3D LiDAR are derived carefully.
- To the best of the authors' knowledge, this is the first time that the multi-sensors (LiDAR, camera, GPS and IMU) fusion active object detection on UAVs is investigated. The proposed system has been validated in a series of real-world experiments.

II. Preliminaries

A. Frames and Notation Definition

The coordinate frames used in this work are depicted in Fig. 1(a), including the IMU body frame $\{B\}$ which is fixed at the center of UAV's mass, the world frame $\{W\}$, the LiDAR frame $\{L\}$ and the camera frame $\{C\}$. We use the sensor frame $\{S\}$ to denote both $\{L\}$ and $\{C\}$.

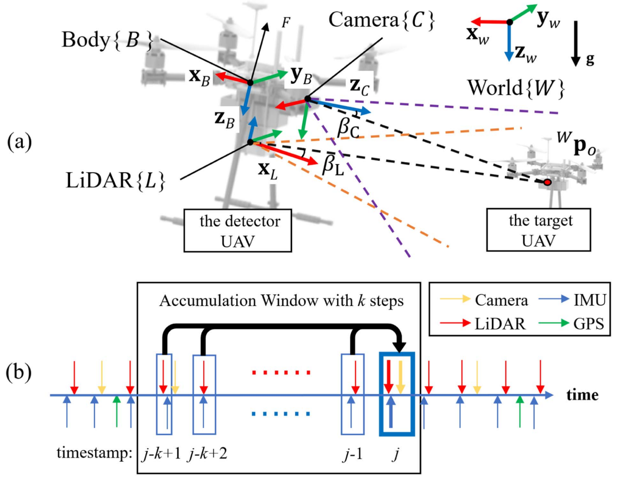

Fig. 1. (a) Illustration of the reference frames and detection scenario. Dashed lines represent the FoV of sensors. (b) Timing of the sensor measurements, data synchronization strategy (blue boxes), and point clouds accumulation process.

Each coordinate axis of frame $\{A\}$ is expressed as the orthonormal basis $\{\mathbf{x}_A, \mathbf{y}_A, \mathbf{z}_A\}$.

Given two frames $\{A\}$ and $\{D\}$, the homogeneous transformation matrix ${}^A_D\mathbf{T} \in SE(3)$ representing the transformation from frame $\{D\}$ into frame $\{A\}$ is defined as,

$$ {}^A_D\mathbf{T} := \begin{bmatrix} {}^A_D\mathbf{R} & {}^A_D\mathbf{p} \\ 0 & 1 \end{bmatrix} \tag{1} $$

where ${}^A_D\mathbf{R} \in SO(3)$ is the rotation matrix and ${}^A_D\mathbf{p} \in \mathbb{R}^3$ is the translation vector.

Let the ${}^A\mathbf{p}_o = ({}^A x_o, {}^A y_o, {}^A z_o)^T$ be the target object position in frame $\{A\}$ and ${}^A\mathcal{I}^j = \{{}^A\mathcal{P}_i^j \in \mathbb{R}^3, i = 0, 1, \ldots, M\}$ be the point cloud in frame $\{A\}$ at time instant j, where ${}^A\mathcal{P}_i^j = ({}^A x_i, {}^A y_i, {}^A z_i)^T$ denotes a single point within the point cloud. In the case of the same time instant, the time instant superscript j will be omitted for notation simplification. Denote $\mathbf{v}_A$ and $\boldsymbol{\omega}_A$ as the linear velocity and the angular velocity of frame $\{A\}$ with respect to the world frame, expressed in frame $\{A\}$. Denote $\hat{(\cdot)}$ as the estimated value of $(\cdot)$. For arbitrary two vectors $\mathbf{a} \in \mathbb{R}^3$ and $\mathbf{b} \in \mathbb{R}^3$, $\Lambda(\mathbf{a})$ is the skew-symmetric matrix of vector $\mathbf{a}$ such that $\Lambda(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$.

B. System Overview

The main goal of this letter is to realize a real-time multi-modal active vision system for object detection based on a UAV platform equipped with a limited FoV LiDAR and camera. The proposed system is applied to actively detect a target UAV in real-world experiments, as shown in Fig. 1(a).

We adopt Gaussian YOLOv3 network [9], [22], a state-of-the-art CNN-based visual object detection approach, as a baseline detector module. The network is trained with our own 6-channel RGBXYZ dataset to efficiently detect our custom UAVs. The preprocessing algorithm of point clouds is studied, which includes motion compensation, accumulation, projection, and up-sampling. A GPS/IMU fused EKF is adopted to estimate the pose and velocity information of the detector UAV. Then,

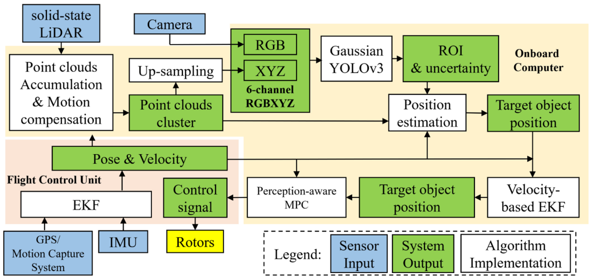

Fig. 2. Overall architecture of the multi-modal active vision on UAVs for object detection.

the detector UAV's velocity information will be further fused with detection results by a velocity-based EKF in order to increase the update frequency of the target UAV's position. A perception-aware MPC is adopted as the main tool to deal with the perception constraints from both camera and limited FoV LiDAR. The overall architecture of the proposed active vision framework is shown in Fig. 2.

III. LiDAR-Camera-GPS/IMU Fusion on UAVs

We consider fusing point clouds and RGB images into tensors of 6 channels. Firstly, point clouds in several LiDAR scans will go through motion compensation, and then it will be accumulated into a dense point cloud cluster. Then the dense point cloud cluster is converted into a 2D image (in the following denoted as XYZ) by projection and up-sampling processes. The resulting XYZ image is simply concatenated with an RGB image to produce a 6-channel tensor, which is known as early fusion. Compared to other fusion strategies like late fusion or the cross fusion [10], early fusion has the least number of model parameters and lower computational cost with little loss in performance [23]. This is crucial for a real-time system. In addition, early fusion is able to use existing methods that were based on camera images by concatenating color intensities with 3D information to process fused data [10]. Instead of estimating the target position by further modifying the network, which requires much more computation resources, we decouple the target detection and 3D position estimation, and allow the network to be substituted as needed.

A. Multi-Modal Data Preprocessing

1) Point Clouds Motion Compensation and Accumulation: Point clouds are subject to motion-based distortion as the LiDAR mounting platform moves, making the target hard to be identified. To address this issue, the poses of LiDAR are used to compensate for the motion distortion. Regarding data synchronization, we simply associate a LiDAR scan frame with its closest pose state which has the same instant as IMU. This is illustrated in Fig. 1(b). To ensure real-time performance and reduce system complexity, we only consider distortion between scan frames and neglect distortion caused by a moving target or within a scan frame, which can be mitigated by increasing the LiDAR scan rate [20]. The effect of the motion compensation is depicted in Fig. 3. As a result, we obtain an accumulated point cloud cluster ${}^{L}\mathcal{I}^{j}(k)$, which accumulates LiDAR scans within k steps up to the scan-end LiDAR frame at time instant j, using the following equation:

$$ {}^{L}\mathcal{I}^{j}(k) = \left({}^{W}_{L}\mathbf{T}^{j}\right)^{-1}\, {}^{W}_{L}\mathbf{T}^{j-k+1}\, {}^{L}\mathcal{I}^{j-k+1} \oplus \cdots \oplus {}^{L}\mathcal{I}^{j}. \quad (2) $$

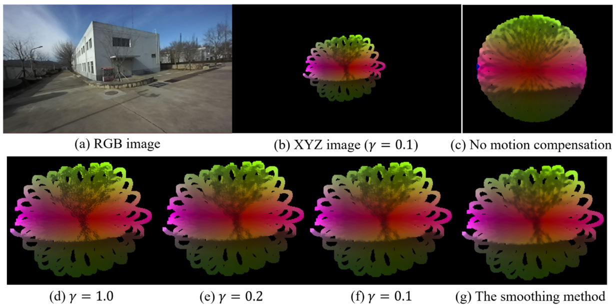
(a) RGB image (b) XYZ image (γ = 0.1) (c) No motion compensation
(d) γ = 1.0 (e) γ = 0.2 (f) γ = 0.1 (g) The smoothing method

Fig. 3. RGB image and the up-sampling XYZ image with a kernel size $l = 3$ and different discretization resolution γ [pixel], (g) is generated using the smoothing method proposed in [24]. Note that, the FoV of the solid-state LiDAR is much smaller than the camera, resulting in a significant portion of the XYZ image being black.

2) Point Clouds Projection and Up-Sampling: In the projection phase, the 3D point clouds in the LiDAR frame are projected onto the 2D image plane while preserving the shape information of the object with acceptable resolution [24]. For each point ${}^{L}\mathcal{P}_i$, the corresponding pixel coordinate $(u_i, v_i)^T$ on the image plane can be derived as:

$$ {}^{C}z_i\,[u_i, v_i, 1]^T = \mathbf{K}_{in}\left({}^{C}_{L}\mathbf{R}\,{}^{L}\mathcal{P}_i + {}^{C}_{L}\mathbf{p}\right) \quad (3) $$

where $\mathbf{K}_{in}$ is the intrinsic matrix of the camera. However, the accumulation and projection result is still sparse. We utilize an up-sampling process [24] to solve it. To improve real-time performance, we alter the process order from the open-source upsampling program, thereby reducing the time required to create intermediate values. The reordered version derives the up-sampling XYZ image $\mathcal{D}_{xyz}$ by dividing the sum of the intensity maps $\mathcal{M}_i$ element-wise by the sum of the normalization maps $\mathcal{N}_i$:

$$ \mathcal{D}_{xyz} = \frac{\sum_{i=1}^{M(k)} \mathcal{M}_i}{\sum_{i=1}^{M(k)} \mathcal{N}_i} \quad (4) $$

where $M(k)$ is the number of points in ${}^{L}\mathcal{I}(k)$. Furthermore, we adjust the non-integers pixel coordinate $(u_i, v_i)^T$ to the nearest resolution of γ (in pixels), denoted as $(\tilde{u}_i, \tilde{v}_i)$. We refer to this process as discretization. It allows us to compute and store the distance from the pixel position q inside a $l \times l$ kernel $\mathcal{K}_i$ with center at $(\tilde{u}_i, \tilde{v}_i)^T$ to the kernel center beforehand and reuse it later. Denoting the lower-index $(\cdot)_q$ as the intensity value of the map in the pixel position $q \in \mathcal{K}_i$, the expression of $\mathcal{M}_i$ and $\mathcal{N}_i$ can be written as:

$$ \begin{aligned} \mathcal{M}_{i,q} &= K_d(\|q - (\tilde{u}_i, \tilde{v}_i)^T\|)\, K_r(|{}^{L}\mathcal{P}_i|)\, |{}^{L}\mathcal{P}_i| \\ \mathcal{N}_{i,q} &= K_d(\|q - (\tilde{u}_i, \tilde{v}_i)^T\|)\, K_r(|{}^{L}\mathcal{P}_i|) \end{aligned} \quad (5) $$

where K_d is a weight that is proportional to the inverse of the distance $\|q - (\tilde{u}_i, \tilde{v}_i)^T\|$, and K_r is a penalization value [24]. A higher resolution γ only results in a slight increase in space complexity, but it can generate smoother up-sampling results, as shown in Fig. 3. The time cost comparison will be presented in Section V, where the above-mentioned two improvements are designated as **Reordered** and **Discretized**.

B. EKF for Detection Results and Vehicle Velocity Fusing

To provide high-frequency estimations of the target object position for the MPC, we designed an EKF that fuses the detection results from YOLO and the detector UAV's velocity information output from GPS/IMU.

In implementing the EKF, we assume the target object to be static. The target object position ${}^{\mathbf{W}}\mathbf{p}_o \in \mathbb{R}^3$ in the world frame can be expressed from ${}^{\mathbf{S}}\mathbf{p}_o \in \mathbb{R}^3$ as,

$$ {}^{W}\mathbf{p}_o = {}^{W}_{S}\mathbf{R}\ {}^{S}\mathbf{p}_o + {}^{W}_{S}\mathbf{p}. \tag{6} $$

Taking the time derivative of (6) yields,

$$ {}^{W}\dot{\mathbf{p}}_o = {}^{W}_{S}\dot{\mathbf{R}}\ {}^{S}\mathbf{p}_o + {}^{W}_{S}\mathbf{R}\ {}^{S}\dot{\mathbf{p}}_o + {}^{W}_{S}\dot{\mathbf{p}}. \tag{7} $$

As we assume ${}^{W}\dot{\mathbf{p}}_o = 0$, (7) can be further written as,

$$ {}^{S}\dot{\mathbf{p}}_o = -\Lambda[\boldsymbol{\omega}_S]\ {}^{S}\mathbf{p}_o - \mathbf{v}_S. \tag{8} $$

Linearizing (8) we obtain the prediction equation of the EKF.

The measurement equation of the EKF is expressed as,

$$ {}^{S}\mathbf{p}_o = {}^{S}\hat{\mathbf{p}}_o + \mathbf{n} \tag{9} $$

where $\mathbf{n} \sim \mathcal{N}\{\mathbf{0}, \boldsymbol{\Sigma}_0\}$ is the white Gaussian noise with covariance $\boldsymbol{\Sigma}_0$, and ${}^{S}\hat{\mathbf{p}}_o$ can be estimated by:

$$ {}^{S}\hat{\mathbf{p}}_o = h(u_{yolo}, v_{yolo}, {}^{L}\mathcal{I}(k)^j) \tag{10} $$

where h is a general nonlinear function and $(u_{yolo}, v_{yolo})^T$ denotes the 2D detection result obtained from YOLO.

The detailed implementation of h is outlined in Algorithm 1. We aim to leverage the accurate 3D information provided by the LiDAR sensor to estimate ${}^{S}\hat{\mathbf{p}}_o$. First, a *Projection_Filter* function is applied to the point cloud cluster ${}^{L}\mathcal{I}(k)^j$ to pick out ${}^{L}\mathcal{I}^j_{box}$ which falls within the 2D detection result box, using the projection transformation (3). Next, the point cloud ${}^{L}\mathcal{I}^j_{box}$ is downsampled and divided into $Row \times Col \times Dep$ voxel grids, which form a large 3D box denoted by $\mathbf{VG}_{Row\times Col\times Dep}$. This is accomplished using a *VoxelGrid_Filter* function. To reduce computational complexity, the 3D localization problem is converted into a 2D cluster problem by traversing the $Row \times Col$ voxel plane along with the depth dimension. Treating one voxel as the basic unit, the DBSCAN algorithm [25] is applied to solve the 2D cluster problem. The first isolated cluster of voxels which contains more than ϵ points is treated as the target point cloud ${}^{L}\mathcal{I}^j_{obj}$, and its average position can be estimated. However, there is a time delay τ of several hundred milliseconds from when ${}^{L}\mathcal{I}^j_{box}$ is produced until the target UAV is detected (time instant $j+\tau$, which will be omitted for notation simplification). To mitigate the effects of the time delay, the relative pose of the sensor, ${}^{S}_{S^j}\mathbf{T}$, is used to correct the average position, resulting in ${}^{S}\hat{\mathbf{p}}_o$.

For the uncertainty estimation, the covariance of the 2D detection result is expressed as [22],

$$ \boldsymbol{\Sigma}_1 = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}. \tag{11} $$

The Jacobian of h with respect to $(u_{yolo}, v_{yolo})^T$ can be written as,

$$ \mathbf{J} = \left[\frac{\partial h}{\partial u_{yolo}}, \frac{\partial h}{\partial v_{yolo}} \right] \in \mathbb{R}^{3\times 2}. \tag{12} $$

Algorithm 1: Observation Function h.

```
Initialization:Given the point cloud cluster ${}^{L}\mathcal{I}(k)^j$ and
  the YOLO detection result $(u_{yolo}, v_{yolo})^T$.
${}^{L}\mathcal{I}^j_{box}$ = Projection_Filter(${}^{L}\mathcal{I}(k)^j, (u_{yolo}, v_{yolo})^T$)
$\mathbf{VG}_{Row\times Col\times Dep}$ = VoxelGrid_Filter(${}^{L}\mathcal{I}^j_{box}$)
for all $\mathbf{VG}_{Rol\times Col,d_i}$, $d_i = 1, 2, \ldots, Dep$ do
  ${}^{L}\mathcal{I}^j_{obj'}$ =DBSCAN($\mathbf{VG}_{Row\times Col,d_i}$)
  if ${}^{L}\mathcal{I}^j_{obj'}$ is isolated then
    if Number_of_Points(${}^{L}\mathcal{I}^j_{obj'}$) $> \epsilon$ then
      ${}^{L}\mathcal{I}^j_{obj} ={}^{L} \mathcal{I}^j_{obj'}$
    end if
  end if
end for
${}^{S}\hat{\mathbf{p}}^j_o$ = Average_Position(${}^{L}\mathcal{I}^j_{obj}$)
$({}^{S}\hat{\mathbf{p}}_o, 1)^T = {}^{S}_{W}\mathbf{T}\ {}^{W}_{S}\mathbf{T}^j\ ({}^{S}\hat{\mathbf{p}}^j_o, 1)^T$
```

And then, the transfer of the uncertainty of (u_{yolo}, v_{yolo}) to the estimated variable ${}^{S}\hat{\mathbf{p}}_o$ can be derived as,

$$ \boldsymbol{\Sigma}_0 = \mathbf{J}\Sigma_1\mathbf{J}^T + \boldsymbol{\Sigma}_2 \tag{13} $$

where the first part $\mathbf{J}\Sigma_1\mathbf{J}^T$ corresponds to the uncertainty of the *Projection_Filter* process, and $\boldsymbol{\Sigma}_2 \in \mathbb{R}^{3\times 3}$ describes the uncertainty of estimating ${}^{S}\hat{\mathbf{p}}_o$ from the point cloud ${}^{L}\mathcal{I}^j_{box}$.

IV. Perception-Aware MPC Design

We adopt a perception-aware MPC to achieve the active vision task, which requires the ROI to fall in both the camera and LiDAR FoV while the UAV moves. Such constraints are derived as perception constraints in the MPC design.

For an underactuated UAV, the model typically has four inputs and six degrees of freedom. To reduce the computational burden of MPC, we set the thrust and angular velocity of the UAV as the input. Such an assumption is reasonable for small-scale UAVs as they can track the control angular rate signal in time. As we assume the angular velocity response of the UAV is fast enough, the equation of motion (EOM) of the UAV can be simplified as [26], [27],

$$ \begin{aligned} {}^{W}_{B}\dot{\mathbf{p}} &= {}^{W}_{B}\mathbf{v} \\ {}^{W}_{B}\dot{\mathbf{v}} &= -\frac{1}{m}\ {}^{W}_{B}\mathbf{R}F\mathbf{e}_3 + {}^{W}\mathbf{g} \\ {}^{W}_{B}\dot{\mathbf{R}} &= {}^{W}_{B}\mathbf{R}\Lambda[\boldsymbol{\omega}_B] \end{aligned} \tag{14} $$

where $\mathbf{e}_3 = (0, 0, 1)^T$, $m \in \mathbb{R}$ is the mass of UAV, $F \in \mathbb{R}$ is the net thrust, and ${}^{W}\mathbf{g} = (0, 0, g)^T$ is the gravitational acceleration. From EOM (14) we can write the state of the system as $\boldsymbol{\xi} = ({}^{W}_{B}\mathbf{p}, {}^{W}_{B}\mathbf{v}, {}^{W}_{B}\mathbf{R}) \in \mathbb{R}^6 \times SO(3)$, and the input as $\mathbf{u} = (\boldsymbol{\omega}_B, F) \in \mathbb{R}^4$. Finally, we can write the EOM (14) as $\dot{\boldsymbol{\xi}} = f_B(\boldsymbol{\xi}, \mathbf{u})$.

A. Perception Constraints Induced by LiDAR and Camera

The perception constraints are expressed as the angle between the main axis ($\mathbf{x}_L$ for the LiDAR frame and $\mathbf{z}_C$ for the camera frame) and the line connecting the center of ROI (i.e. ${}^{W}\mathbf{P}_o$) to the origin of the sensor frame, as shown in Fig. 1(a). To simplify the notation, we introduce a virtual state $c\beta := \cos\beta$ in the system. From the definition of the main axis of the sensor frame, the expression of $c\beta$ for the camera and LiDAR sensor is,

$$c\beta_C = \frac{\mathbf{e}_3^T\ {}^C\mathbf{p}_o}{\|{}^C\mathbf{p}_o\|},\quad c\beta_L = \frac{\mathbf{e}_1^T\ {}^L\mathbf{p}_o}{\|{}^L\mathbf{p}_o\|} \tag{15}$$

where $\mathbf{e}_1 = (1,0,0)^T$, $\|\cdot\|$ stands for Euclidean norm of vector $(\cdot)$. From (15) we can further obtain the time derivative of $c\beta_C$ as follows:

$$\dot{c\beta}_C = \frac{{}^C\dot{\mathbf{p}}_o^T\ \mathbf{e}_3\|{}^C\mathbf{p}_o^T\| - {}^C\mathbf{p}_o^T\ \mathbf{e}_3\frac{d\|{}^C\mathbf{p}_o\|}{dt}}{\|{}^C\mathbf{p}_o\|^2} \tag{16}$$

in which ${}^C\mathbf{p}_o$ can be obtained by coordinate transformation as,

$${}^C\mathbf{p}_o = {}^C_B\mathbf{R}\ {}^W_B\mathbf{R}^T({}^W\mathbf{p}_o - {}^W_B\mathbf{p}) + {}^C_B\mathbf{p} \tag{17}$$

where ${}^C_B\mathbf{R}$ and ${}^C_B\mathbf{p}$ are the rotation matrix and translation vector of ${}^C_B\mathbf{T}$, which is the extrinsic pose between the camera frame and the body frame. So that $\dot{c\beta}_C$ can be expressed as the function of $\boldsymbol{\xi}$ and $\boldsymbol{\omega}_B$,

$$\dot{c\beta}_C = f_\beta(\boldsymbol{\xi}, \boldsymbol{\omega}_B). \tag{18}$$

Since the relative pose from the camera frame and the LiDAR frame is fixed, $c\beta_C$ can be expressed from $c\beta_L$. Then it is not needed to include both of the two sensors in the perception constraints. Here, we can express the perception constraints as,

$$\mathcal{L}_1 \leq c\beta_C \leq \mathcal{L}_2 \tag{19}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are two constants. They are determined from the overlap between the LiDAR FoV and the camera FoV. From the relative pose between the two sensors and the overlapping FoV, an enveloping cone whose main axis is the same with the main axis of the camera can be obtained. Then $\mathcal{L}_1$ and $\mathcal{L}_2$ can also be derived. For brevity the detailed derivation is omitted here.

By adding the virtual state $c\beta$, the augmented state of the system is therefore defined as $\boldsymbol{\xi}_{aug} = (\boldsymbol{\xi}, c\beta)$. The augmented state equation can therefore be obtained as

$$\dot{\boldsymbol{\xi}}_{aug} = f_{aug}(\boldsymbol{\xi}_{aug}, \mathbf{u}) := \begin{bmatrix} f_B(\boldsymbol{\xi}, \mathbf{u}) \\ f_\beta(\boldsymbol{\xi}, \boldsymbol{\omega}_B) \end{bmatrix}. \tag{20}$$

B. MPC Design

The objective of the MPC is to ensure that the state error converges while meeting necessary state, input, and perception constraints. Given the reference state trajectory of the UAV ${}^W_B\mathbf{R}_r$, ${}^W_B\mathbf{p}_r$, ${}^W_B\mathbf{v}_r$, the objective function in the MPC is designed as

$$\begin{aligned} Q(t) = K_R tr(\mathbf{I} - \mathbf{E}_R) + K_p\|\mathbf{E}_p\|^2 + K_v\|\mathbf{E}_v\|^2 + \\ K_\omega\|\boldsymbol{\omega}_B\|^2 + K_C(c\beta_C - 1)^2 + K_L(c\beta_L - 1)^2 \end{aligned} \tag{21}$$

where the tracking errors are defined as $\mathbf{E}_R = {}^W_B\mathbf{R}_r^T\ {}^W_B\mathbf{R}$, $\mathbf{E}_p = {}^W_B\mathbf{p}_r - {}^W_B\mathbf{p}$, $\mathbf{E}_v = {}^W_B\mathbf{v}_r - {}^W_B\mathbf{v}$, and the positive constants K_R, K_p, K_v, K_ω, K_C, K_L represent the weights which can be adjusted.

Given the objective function, the perception constraints, and the admissible input and state set, the optimal control problem at each time step t_j in the MPC is now ready,

$$\begin{aligned} &\min_{\mathbf{u}(t)} \int_{t_j}^{t_j+\mathit{\Gamma}} Q(t)dt \\ &\text{s.t.} \begin{cases} \dot{\boldsymbol{\xi}}_{aug} = f_{aug}(\boldsymbol{\xi}_{aug}, \mathbf{u}) \\ {}^W_B\mathbf{p}(t) \in \mathcal{X},\ {}^W_B\mathbf{v}(t) \in \mathcal{V}, \\ {}^W_B\mathbf{R}(t) \in \mathcal{A},\ \boldsymbol{\omega}_B(t) \in \mathcal{W}, \\ \mathcal{L}_1 \leq c\beta_C(t) \leq \mathcal{L}_2, F \leq \mathcal{L}_3, t \in [t_j, t_j + \mathit{\Gamma}) \end{cases} \end{aligned} \tag{22}$$

where $\mathcal{X}$, $\mathcal{V}$, $\mathcal{A}$, $\mathcal{W}$ represent the admissible set of position, velocity, attitude, and angular velocity respectively, positive constant $\mathcal{L}_3$ defines the maximum net thrust, and $\mathit{\Gamma}$ is the time horizon.

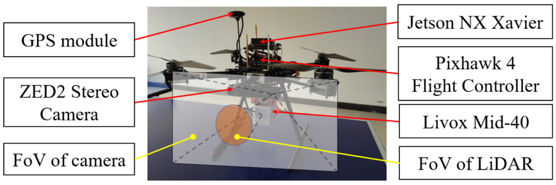

Fig. 4. Detector UAV used in the experiments. Our method can also be applied to UAVs with different FoV distributions, as shown in Fig. 1.

V. Experiments

In order to validate our algorithm and demonstrate its robustness to various environments, we conducted experiments using an onboard UAV platform to actively detect another independent UAV in real time.

A. Aerial Platform

As shown in Fig. 4, the detector was equipped with a Livox Mid-40 LiDAR (38.4° × 38.4° FoV, 100 Hz), a ZED2 Stereo camera (10 Hz) whose left monocular camera was used for our algorithm, a Pixhawk 4 flight controller, a GPS module, and a Jetson Xavier NX onboard computer. It is noted that both the solid-state LiDAR and the camera are forward-looking and their FoVs overlap.

Sensor calibration is fundamental for the multi-sensors system. We adopted the Kalibr calibration tool[1] to calibrate the camera's intrinsic parameters and the extrinsic pose ${}^C_B\mathbf{T}$ between the camera and IMU. For the extrinsic pose ${}^C_L\mathbf{T}$ of the camera and LiDAR sensor, we applied the Livox Camera-LiDAR-Calibration tool.[2] All of the calibration work was executed offline, and the calibration results were regarded as true values in our experiments.

The software architecture of our system is divided into three processes: the YOLO detection process, the MPC process, and the Preprocessing process. The ROS infrastructure is used to exchange messages between different processes. ACADO [3] is adopted as the solver of the MPC. To process data from different sensors with varying computational requirements, the Preprocessing process adopts a multithreading architecture, including

[1] [Online]. Available: https://github.com/ethz-asl/kalibr

[2] [Online]. Available: https://github.com/Livox-SDK/livox_camera_lidar_calibration

[3] [Online]. Available: http://www.acadotoolkit.org.

TABLE I
TIME CONSUMPTION OF EACH COMPONENT

Process	**Thread/Description**	**Time cost [ms]**
Pre-processing	The vEKF prediction thread	0.82
	The vEKF update thread	11.56
	pc_pre (**Reordered** + **Discretized**[$\gamma = 0.1$])	**65.98**
	pc_pre (**Reordered** only)	83.35
	pc_pre (the method in [24])	647.42
YOLO detection	Image size 512 × 512	322.58
MPC		<10

TABLE II
DETAILS OF THE RGBXYZ DATASET

Set		**GL-D**	**GL**	**LL**	**CD**	**TOTAL**
Training & Valid	Labeled	/	2029	492	311	2832
	Negative Samples	240	287	54	256	837
Test		327	272	166	105	870

the vEKF prediction thread, the vEKF update thread, and a point cloud preprocessing thread. The point cloud preprocessing thread performs motion compensation, accumulation, projection, and up-sampling. Table I shows the time consumption of the main components of our proposed system when executed on the onboard computer with $k = 40$ and a kernel size of $l = 5$. The point cloud preprocessing thread is denoted as **pc_pre**, and the up-sampling methods mentioned in Section III-A2 are compared in terms of their time consumption. It is important to mention that according to the results in Table I, while the **pc_pre** method requires 647.42 ms, our improvements in reordering and discretizing data reduce the computational time roughly by a factor of 10, resulting in a reduced processing time of only 65.98 ms.

B. Dataset and Networks Training & Evaluation

1) Dataset: We constructed a dataset capturing a hovering UAV in different scenarios and illumination conditions. The dataset, outlined in Table II, consists of 4,539 RGBXYZ images with a resolution of 672 × 376 pixels, and it contains two types of UAVs: the target UAV and the distractor UAV. A snapshot of our dataset is presented in the attached video. The detector UAV as shown in Fig. 4 is used to collect the raw data (RGB images, point clouds, and pose and velocity information) and generate the dataset.

We partitioned the dataset into several subsets based on the lighting conditions, namely good light (GL), low light (LL), and complete darkness (CD). Each subset contains labeled images that feature the target UAV, as well as negative samples consisting of background images without the target UAV. Specially, we designed the negative samples that include the distractor UAV in good light as another subset, called GL-D. To ensure unbiased evaluation, we generated the test set using real-life experimental data and additional datasets that were completely independent of the training or validation sets.

TABLE III
NETWORK EVALUATION WITH mAP-50 ON THE TEST SET

Test Set	Vision only	LiDAR only	LiDAR-camera fused			
			smooth	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 1$
GL	84.64	98.68	98.51	98.51	97.44	97.77
LL	94.29	98.74	97.59	97.59	97.58	97.44
CD	39.61	99.00	99.86	99.86	98.85	98.37
SUM	80.66	98.76	98.50	98.50	97.75	97.79
GL-D+SUM	76.09	89.6	94.6	94.6	x	x

2) Training: We retain the Darknet-53 backbone network of Gaussian YOLOv3 by applying early fusion. In order to load 6-channel tensors (RGBXYZ) during the training and testing process of the multi-modal YOLO, we modified the interface of YOLO. Additionally, we trained two separate 3-channel YOLO models, the vision-only and LiDAR-only model. The vision-only model was trained exclusively using the RGB images from our dataset, while the LiDAR-only model utilized the upsampling XYZ images obtained from the RGBXYZ dataset. In terms of the model architecture and other settings, we maintained consistency among the different models and the original YOLO framework.

3) Evaluation: The above networks were tested on the test set and evaluated using the mean average precision at 50% intersection over union (mAP-50) metric. The result, presented in Table III, shows that the vision-only network performs poorly in complete darkness, while the fused method achieves excellent performance by integrating point cloud information. The LiDAR-only network struggles to handle the distractor UAV which has a similar upsampling image as the target UAV. In contrast, both the vision-only and fused networks demonstrate robustness in handling the distractor UAV. Additionally, we observed that the performance of the fused network is influenced by the discretization resolution of the discretized upsampling images. Lower resolutions negatively impact the performance, while higher resolutions, such as $\gamma = 0.1$ pixel, effectively eliminate this impact. Moreover, we noticed that the vision-only YOLO performs poorly on the GL set, which contains many challenging scenarios where distinguishing the target from the background (e.g. both are black) is difficult. On the other hand, the LL set, collected near a street lamp at night, benefits from the light reflection on the target, which helps it stand out from the darker surroundings.

C. Real-World Experiment Results

We tested the proposed algorithm in four series of experiments with various trajectories in different lighting conditions: straight-line flight (LF) in good lighting (GL) conditions, square flight (SF) in low lighting (LL) conditions, and circular flights (CF) in both GL conditions and complete darkness (CD) conditions. The NOKOV motion capture system was utilized to localize UAVs during indoor flights, while GPS/IMU sensors were employed for outdoor flights. Noted that the reference yaw input to the MPC was set to zero for all experiments. A video of the experiments is attached. The detailed parameters are also presented in the attached video.

1) LF in GL Conditions: We designed a two-step flight experiment as shown in Fig. 5(a). The detector UAV hovered at point C, while the target UAV flew from point A to B and hovered there. The detector UAV then moved to point D and back to C. Fig. 6(a) shows the MPC tracking results and the estimated target position in the world frame. The yaw and cosine curve demonstrates that the active vision algorithm adjusts the yaw angle of the detector UAV to keep the target in its view. Moreover, the result demonstrates that the proposed algorithm can handle moving targets with low velocity (approximately

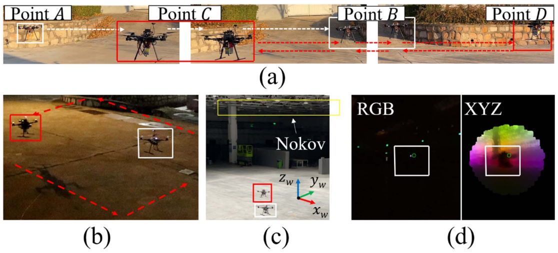

Fig. 5. Real-world experiments. Red box: the detector UAV and white box: the target UAV. (a) LF. (b) SF in LL conditions. (c) Indoor experimental environments. (d) The first-person view of the detector UAV of the circular flight in CD conditions.

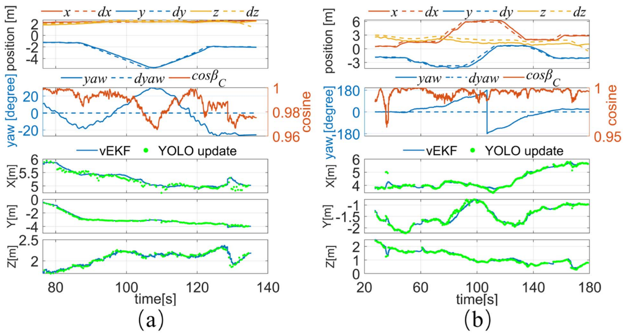

Fig. 6. From top to bottom: the evolution of the position, yaw angle and cosine of β_C, and target UAV positions (expressed in the world frame) estimated by the vEKF and Algorithm 1 (labeled as YOLO update). (a) LF results. (b) SF results.

0.3 m/s), even though we designed the vEKF with the assumption of a static target object.

2) SF in LL Conditions: In this experiment, depicted in Fig 5(b), the detector UAV flew a square trajectory with the target UAV remaining hovering at the center. Fig 6(b) shows that the detector UAV tracked the square trajectory and adjusted the yaw to face the target in low light conditions. It is worth noting that at around 40 seconds, there are some erroneous measurements estimated by Algorithm 1 (YOLO update). However, the vEKF incorporates these measurements while considering the measurement uncertainty, rather than blindly trusting them. This highlights the capability of the vEKF to generate high-frequency and smooth results while also being resilient to incorrect detections, thereby enhancing the overall robustness of the system.

3) CF in GL Conditions: In this experiment, the detector UAV tracked two circular trajectories with radii of 3 m and 2 m under good light conditions. The target UAV remained centered during lap one but moved outside the circle during lap two. As shown in Fig. 7(b), on the second lap, the distance between the UAVs ranged from 2 to 4.5 m.

4) CF in CD Conditions: We performed the second circle flight experiment with a radius of 2 m in complete darkness, rendering the camera sensor nearly non-functional, as evidenced by the RGB image shown in Fig. 5(d). Using the active multi-modal fusion strategy, the detector UAV successfully detected the target and adjusted the yaw angle while tracking the circle reference trajectory, as shown in Fig. 8.

5) Analysis: It is seen that the active multi-modal detection algorithm achieves success in all the above four scenarios. Table IV summarises the Root Mean Square Error (RMSE) and Standard Deviation (STD) of the tracking error for the desired trajectory under the MPC, and the estimation error of the target UAV position obtained by the vEKF, respectively. The MPC tracking error increases as the reference trajectory speed rises. The vEKF demonstrates consistent tracking accuracy across different lighting levels. Specially, it is worth noting that even in challenging conditions (CD conditions), the detector UAV can effectively detect the target and adjust the yaw angle while tracking the reference trajectory. This emphasizes the advantages of our multi-modal framework compared to methods that rely solely on vision in challenging environments [5], [6], [21]. However, it is also noticed that there is a limitation of our proposed approach. Tracking a fast-moving target is a challenge using our approach, as the vEKF is designed based on the assumptions of static targets. Moreover, our proposed approach suffers from the detection network's latency which also degrades our system's performance on a moving target.

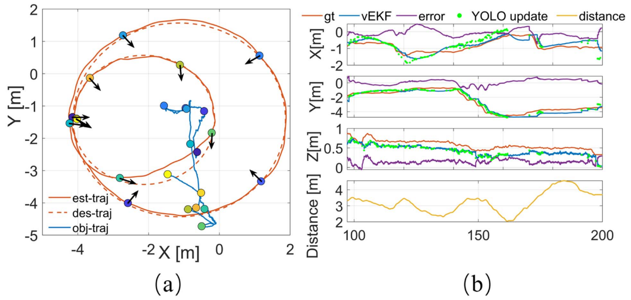

Fig. 7. CF results in GL conditions. (a) Trajectories of two UAVs in the xy plane. The blue line: the estimated target UAV trajectory. The arrows represent the heading direction of the detector UAV. The circles, each with a different color, depict the positions of the detector and target UAVs at various time instants. Two circles with the same color correspond to the same time instant. (b) The estimated target positions. The NOKOV mocap system provides ground truth for evaluation. Yellow line: the ground-truth distance between two UAVs.

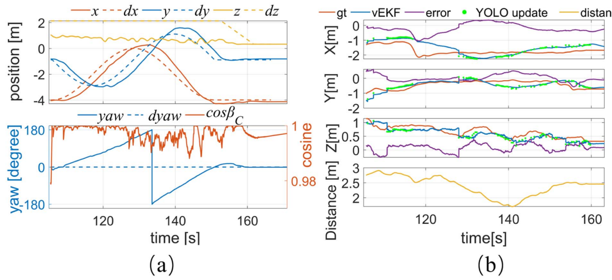

Fig. 8. CF results in CD conditions. (a) The position, yaw angle, and cosine of β_C. (b) The estimated target positions.

TABLE IV
POSITION TRACKING ERROR OF DETECTOR UAV UNDER MPC, AND ESTIMATION ERROR OF TARGET UAV FROM vEKF, EXPRESSED IN THE WORLD FRAME

		MPC				vEKF	
	Metrics [m]	LF (GL)	SF (LL)	CF (GL)	CF (CD)	CF (GL)	CF (CD)
x	STD	0.149	0.395	0.619	0.475	0.359	0.649
	RMSE	0.082	0.344	0.504	0.415	0.239	0.398
y	STD	0.224	0.283	0.61	0.565	0.452	0.304
	RMSE	0.16	0.273	0.453	0.429	0.391	0.248
z	STD	0.145	0.327	0.284	0.449	0.062	0.156
	RMSE	0.218	0.595	0.994	1.212	0.165	0.162

VI. Conclusion

In this letter, we present a multi-modal data fusion framework for active vision detection on small-scale UAVs. The real-time problem has been investigated by carefully designing the algorithms to fuse the point cloud, image, and pose/velocity. The proposed system still requires further improvements in tracking fast-moving targets. It has been verified with real-life experiments in different illumination scenarios. Extensive experiments show that our method efficiently improves the real-time performance by reducing the computational time of data pre-processing by a factor of 10. Also, the fusion network shows better performance than the single-sensor network in challenging scenarios, which reaches 94.6 mAP on the test set. Since our approach can be used for a generic LiDAR camera fusion domain, our future work is to apply our approach to other 3D perception tasks, such as semantic segmentation.

References

[1] R. Bajcsy, "Active perception," in *Proc. IEEE*, vol. 76, no. 8, pp. 966–1005, Aug. 1988.

[2] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "PAMPC: Perception-aware model predictive control for quadrotors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–8.

[3] Z. Zhang and D. Scaramuzza, "Beyond point clouds: Fisher information field for active visual localization," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 5986–5992.

[4] J. Ji, N. Pan, C. Xu, and F. Gao, "Elastic tracker: A spatio-temporal trajectory planner for flexible aerial tracking," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 47–53.

[5] E. Çintaş, B. Özyer, and E. Şimşek, "Vision-based moving UAV tracking by another UAV on low-cost hardware and a new ground control station," *IEEE Access*, vol. 8, pp. 194601–194611, 2020.

[6] W. Zhang, K. Song, X. Rong, and Y. Li, "Coarse-to-fine UAV target tracking with deep reinforcement learning," *IEEE Trans. Automat. Sci. Eng.*, vol. 16, no. 4, pp. 1522–1530, Oct. 2019.

[7] H. Zhong, H. Wang, Z. Wu, C. Zhang, Y. Zheng, and T. Tang, "A survey of LiDAR and camera fusion enhancement," in *Procedia Comput. Sci.*, vol. 183, pp. 579–588, 2021.

[8] J. Hosang, M. Omran, R. Benenson, and B. Schiele, "Taking a deeper look at pedestrians," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4073–4082.

[9] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv.1804.02767*.

[10] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LiDAR–camera fusion for road detection using fully convolutional neural networks," *Robot. Auton. Syst.*, vol. 111, pp. 125–131, 2019.

[11] L. Caltagirone, M. Bellone, L. Svensson, M. Wahde, and R. Sell, "LiDAR–camera semi-supervised learning for semantic segmentation," *Sensors*, vol. 21, no. 14, 2021, Art. no. 4813.

[12] D. Falanga, E. Mueggler, M. Faessler, and D. Scaramuzza, "Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5774–5781.

[13] B. Zhou, J. Pan, F. Gao, and S. Shen, "RAPTOR: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1992–2009, Dec. 2021.

[14] L. Bartolomei, L. Teixeira, and M. Chli, "Perception-aware path planning for UAVs using semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5808–5815.

[15] Z. Zhang and D. Scaramuzza, "Perception-aware receding horizon navigation for MAVs," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2534–2541.

[16] R. Tallamraju et al., "Active perception based formation control for multiple aerial vehicles," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4491–4498, Oct. 2019.

[17] M. Jacquet and A. Franchi, "Motor and perception constrained NMPC for torque-controlled generic aerial vehicles," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 518–525, Apr. 2021.

[18] M. Jacquet, G. Corsini, D. Bicego, and A. Franchi, "Perception-constrained and motor-level nonlinear MPC for both underactuated and tilted-propeller UAVs," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4301–4306.

[19] Z. Liu, F. Zhang, and X. Hong, "Low-cost retina-like robotic LiDARs based on incommensurable scanning," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 1, pp. 58–68, Feb. 2022.

[20] L. Qingqing, Y. Xianjia, J. P. Queralta, and T. Westerlund, "Adaptive LiDAR scan frame integration: Tracking known MAVs in 3D point clouds," in *Proc. 20th Int. Conf. Adv. Robot.*, 2021, pp. 1079–1086.

[21] P. Zhang, G. Chen, Y. Li, and W. Dong, "Agile formation control of drone flocking enhanced with active vision-based relative localization," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6359–6366, Jul. 2022.

[22] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 502–511.

[23] J. Schlosser, C. K. Chow, and Z. Kira, "Fusing LiDAR and images for pedestrian detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 2198–2205.

[24] C. Premebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian detection combining RGB and dense LiDAR data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 4112–4117.

[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 226–231.

[26] Y. Yu and X. Ding, "A global tracking controller for underactuated aerial vehicles: Design, analysis, and experimental tests on quadrotor," *IEEE/ASME Trans. Mechatron.*, vol. 21, no. 5, pp. 2499–2511, Oct. 2016.

[27] Y. Yu, C. Shi, D. Shan, V. Lippiello, and Y. Yang, "A hierarchical control scheme for multiple aerial vehicle transportation systems with uncertainties and state/input constraints," *Appl. Math. Modelling*, vol. 109, pp. 651–678, 2022.