IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Design, Localization, Perception, and Control for GPS-Denied Autonomous Aerial Grasping and Harvesting

Ashish Kumar†, Laxmidhar Behera†, *Senior Member, IEEE*

***Abstract*—In this paper, we present a comprehensive UAV system design to perform the highly complex task of off-centered aerial grasping. This task has several interdisciplinary research challenges which need to be addressed at once. The main design challenges are GPS-denied functionality, solely onboard computing, and avoiding off-the-shelf costly positioning systems. While in terms of algorithms, visual perception, localization, control, and grasping are the leading research problems. Hence in this paper, we make interdisciplinary contributions: (i) A detailed description of the fundamental challenges in indoor aerial grasping, (ii) a novel lightweight gripper design, (iii) a complete aerial platform design and in-lab fabrication, and (iv) localization, perception, control, grasping systems, and an end-to-end flight autonomy state-machine. Finally, we demonstrate the resulting aerial grasping system *Drone-Bee* achieving a high grasping rate for a highly challenging agricultural task of apple-like fruit harvesting, indoors in a vertical farming setting (Fig. 1). To our knowledge, such a system has not been previously discussed in the literature, and with its capabilities, this system pushes aerial manipulation towards 4^{th} generation.**

***Index Terms*—Aerial Systems: Applications; Agricultural Automation; Grasping, Dynamic Payloads, Aerial Fruit Harvesting**

I. Introduction

Precision agriculture and vertical plantation are the future of farming. It aims to obtain higher crop yields in a small area of land by performing plantation onto stacked structures, forming tall assemblies (Fig. 1). Due to the structured cultivation, we foresee a huge potential of aerial grasping in the harvesting process to increase crop yields and to reduce production costs.

However, aerial grasping is quite difficult [1], and is still in its infancy as compared to robotic grasping [2], [3] due to its convoluted challenges. Some are specific to aerial harvesting, while others are specific to aerial manipulation. For instance, [4] extensively reviews aerial manipulators and enlists unsolved challenges. The most crucial ones are *accuracy* of the platform to perform a task and the *decisional autonomy*. Achieving task accuracy is quite essential in the presence of wind gusts or nearby surface perturbations, which are challenging to model and can result in collision [4]. Then, for fully autonomous operations, decisional autonomy requires several algorithms to work in conjunction to enable a UAV to make its decision onboard only. However, with the task under consideration, i.e., in-air grasping, the complexity involved is too high, making developing the autonomy engine a difficult process. Finally, cross effects between challenges also exist, e.g. more algorithms require high on-board computational power, which directly affects onboard power, size, and flight time. Since aerial grasping is a subset of aerial manipulation, these unsolved challenges also apply to our case. In this paper, we target them and push the frontiers in this area while targeting aerial grasping. Now, we discuss challenges specific to aerial harvesting and our contributions to advancing the state-of-the-art (SOTA).

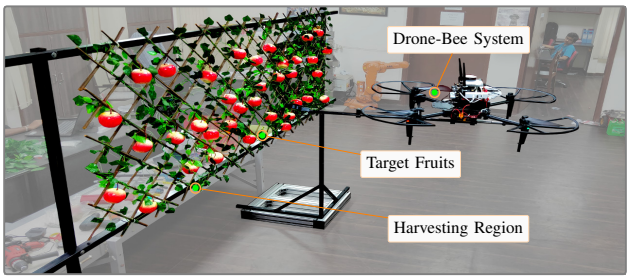

Figure 1: Left: proposed *indoor harvesting setup* inspired by indoor/outdoor orchards and vertical farming to facilitate round-the-clock testing. Right: proposed *Drone-Bee* aerial grasping system.

1) Task-centric Design: Hardware design for aerial grasping requires interdisciplinary expertise, e.g. UAV design, sensor system, electronics, etc. Also, the gripper design is task-specific, which is constrained by the power, weight and size of the UAV, turning into an iterative and costly process.

Our Contribution: Although, multi-sensor retractable magnetic gripper [1], fixed gripper [5], multi-finger gripper [6], pipe holding gripper for perching [7], multi-DoF arm [8] exist, works on off-center aerial grasping are lacking in the literature. In this sector, we propose off-center grasping and a gripper design, which are quite novel from an aerial grasping perspective. It is one of the main components of our system, inspired by human-like behaviour for grasping or harvesting tasks.

2) High-Speed and Accurate Visual Perception: Visual perception makes aerial grasping feasible in the first place, with deep learning-based detectors [9], [10] being the modern choice. However, they have complex training and testing phases, with their runtime and accuracy sensitive to postprocessing steps and hyperparameters. Further, detecting fruit-like tiny objects ($<\sim 15 \times 15$ pixels) from low-resolution CNN features is a challenging research problem. Generally, it is handled via multi-scale detection [11] at the cost of increased runtime. However, aerial grasping needs high-speed detection to approach the target and counter the disturbances quickly.

Our Contribution: Although post-processing free Transformer-based detectors [12] are now available, they have large size, slow convergence, and compute and memory-intensive Transformer attention modules which

†EE, Indian Institute of Technology (IIT), Kanpur, India. {ashishkumar822@gmail.com, lbehera@iitk.ac.in}
Supplementary: https://github.com/ashishkumar822/DroneBee

exhaust the low-power devices. In this sector, keeping the challenge of limited onboard computing in mind, we develop a novel single-scale detector free of post-processing and outperforms recent SOTA detectors [13].

3) High-Speed and High-Accuracy Localization: Accurate positioning is crucial in UAV control and grasping. For GPS-free operations, using VICON-like motion capture systems or off-the-shelf positioning sensors is not viable, mainly due to their high cost. Also, they lack in building and reusing maps which is a must-have attribute of robotic autonomy. Optical flow sensors are also an option, but they have limited accuracy. Thus, ready-made sensors increase the cost and the number of sensors. Visual markers are unsuitable from a production standpoint due to their installation and algorithmic dependency on them. In contrast, visual SLAM [14] offers high metric accuracy, mapping, and customization but is quite compute-intensive. Visual odometry [15] is not viable due to the accumulated drift over time and lack of mapping.

Our contributions: We address the above challenges and develop a GPU-accelerated stereo visual SLAM design that is the fastest available open-source SLAM system, outperforming state-of-the-art SLAM algorithms.

4) Off-centered Dynamic Payload: It is a critical issue in control systems since off-center grasping requires the gripper to be mounted only along the roll axis, inducing off-center loading. Therefore, as soon as the target is grasped, its weight exerts sudden torque about the pitch axes, resulting in unwanted motion or collision. Although low-weighted centered payload systems exist [16], [17], control systems for off-centered payload are not explicitly studied, which is of our concern. Moreover, existing controllers are tested in controlled scenarios, e.g. VICON-based positioning, but the evaluation in real scenarios is far more complex.

Our Contributions: We develop a new thrust microstepping via acceleration feedback thrust controller that is mass and gravity independent. The thrust controller forms the control system of Drone-Bee and outperforms recent state-of-the-art direct acceleration feedback thrust controller [17].

5) Precision In-Air Grasping: This challenge is listed as a future challenge in [4] and is a challenging research problem as it demands reaching and flying near the target with a small tolerance. Also, it relies on vision, control, localization, and depth measurements. Therefore, any error in them can cause grasp failure and collision. There exist many works, such as single arm [18], dual arm [19], parallel manipulator [20], human-robot handover task [21], brick pick-transport-place [1], however, none tackles in-air off-center grasping while focusing on design, localization, perception, control and the autonomy engine simultaneously. Although [22] addresses the design, localization, and grasping of magnetic objects, it relies on costly GPS-RTK for localization, and deals with relatively simple scenarios from a vision and control perspective.

Our Contributions: As this problem directly translates to the autonomy engine, we address it via our detection, SLAM, and control systems. We develop an advanced state machine based on our SOTA algorithms, visual servoing and grasping techniques to achieve precision air-grasping.

6) Imprecise Depth Measurements: The grasping phase heavily relies on depth measurements, acquired from Intel Realsense D435i-like devices. Such devices have a tolerance of $\pm$3cm that is enough to trigger a grasp failure or unwanted grasping of nearby items, thus making it a critical issue.

Our Contributions: Since it is a sensor limitation, we handle it via autonomy engine, i.e. approaching the target slowly when it is near because these devices are relatively accurate at smaller depths. Nonetheless, noise remains in the picture, thus, the probability of losing a target is still not zero.

7) Co-existing Subsystems and Only Onboard Computations: Another challenge is avoiding off-the-shelf positioning systems, only onboard computations, and having GPS-free capability. This requires several co-existing algorithms/sub-systems that mainly include object detection, positioning, control, and grasping. The ones such as object detection and positioning systems, are highly compute-intensive, which prevents concurrent execution of all sub-systems at desired rates without exhausting the onboard computer. Hence, simply deploying existing algorithms is not viable in aerial grasping.

Our Contributions: This challenge poses restrictions on the level of autonomy achievable onboard. We address it via our high-speed detection, SLAM, and control systems. Particularly, these subsystems have been designed in such a way that they can be deployed onboard without computationally taxing the onboard computer. Notably, this is the most critical challenge pointed out by [4]. We address this challenge to push aerial manipulation towards 4^{th} generation.

8) Autonomous Robotic Harvesting: It is a recent research area to improve crop yields [23], [24], [25]. Initial efforts are seen in the ground vehicle for crop harvesting by using existing algorithms. However, developing such systems is very difficult due to the complex nature of the harvesting tasks, varying from crop to crop. This also applies to UAV-based harvesting, which is even more convoluted due to the previously discussed challenges and complex onboard flight autonomy. This is one of the primary reasons why an end-to-end UAV-based harvesting system is still not visible in the literature.

Our Contributions: Hence, in this paper, we have developed a complete hardware and algorithmic solution by pushing the state-of-the-art in multiple domains, called *Drone-Bee*. We demonstrate it performing the complicated task of aerial grasping indoors and outdoors. To our knowledge, this is the first open-source aerial harvesting system in the literature.

9) Lack of Realistic Experimental Setup for Harvesting: Despite many robotic harvesting solutions, none addresses the issue of experimental setup. Since developing a robotic harvesting system is itself a challenging research area, it is not possible to go into the fields or farms every time to conduct experiments. This brings up the challenge of accelerating system development without going into the fields and round-the-clock experimentation.

Our contributions: Hence, we develop an easy-to-build in-lab harvesting setup for researchers to perform experiments round-the-clock, accelerating the system development from multiple fronts, e.g. hardware and software, autonomy, etc.

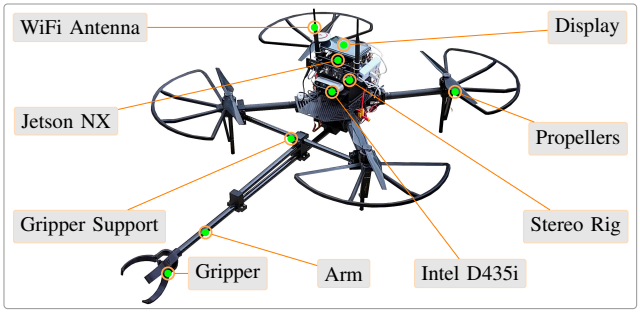

Figure 2: The proposed Drone-Bee aerial grasping system.

Table I. UAV platform specifications.

Attribute	Specification	Attribute	Specification
Size	0.90m× 0.90m× 0.45m	Operating Voltage	22V
Size w/ Gripper	1.5m × 0.90m × 0.45m	Hover time	20 minutes
Weight	2.5Kg	Hover time w/ Gripper	15 minutes
Weight w/ Gripper	3.4Kg	Onboard Computer	NVIDIA Jetson-NX
Rotor diameter	0.46m	Communication	UART @921600 Baud
Rotor-to-Rotor distance	0.35m	Stereo-Rig	2× @432 × 240, 30Hz

II. Hardware Design

In this section, we address the hardware design challenges of aerial grasping and describe our contributions. Our motivations are low-cost hardware, ease of component accessibility, in-lab fabrication, and rapid prototyping to accelerate research.

A. UAV Platform

This is the major source of cost which can be lowered by developing a customized platform. Although our solution is UAV-agnostic, in this work, we use DJI Matrice-100 quadrotor (Table I) while focusing more on the aerial grasping motive. The system can carry a payload of 1.2Kg@70% motor spin. More weight is avoided to prevent motor heating. We control the UAV via roll, pitch, yaw, and thrust commands, which are generated by our control system (Sec. III-B). We use DJI Onboard-SDK (OSDK) to send these commands to the flight controller and to receive IMU data from it.

B. Gripper Design

Our gripper design is primarily inspired by two reasons; *First,* fruits in orchards or trees lie vertically, thus approaching them from the top and performing centered grasping [22], [6] is not feasible. *Second*, humans perform grasping by extending their multi-DoF arm horizontally in similar cases. However, the multi-DoF arm in aerial platforms has considerable stability issues [8]. This gives rise to our novel idea of off-center grasping to approach the fruits horizontally.

To this end, we propose a lightweight and compact, fixed-length, off-center gripper having a human-like three-finger hand jaw with a precisely controllable opening (Fig. 3). Our design reduces gripper complexity and platform instability while increasing the target reachability. In terms of lifting capacity, it can easily hold apple-like spherical items having a radius in the range $\sim 3-6$cm, and weight up to 2.5 kg.

The gripper mainly has four parts: *fingers*, *wrist*, *arm*, and *actuator* (Fig. 4). Each finger consists of two hinges, one attached to the wrist and the other attached to a *finger coupler*

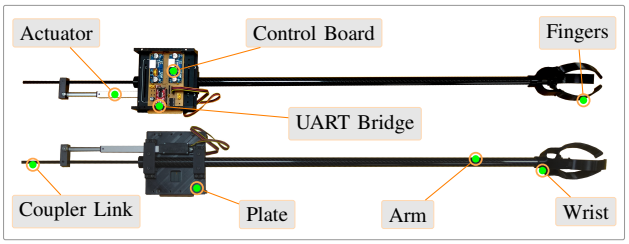

Figure 3: Realized version of the proposed gripper design.

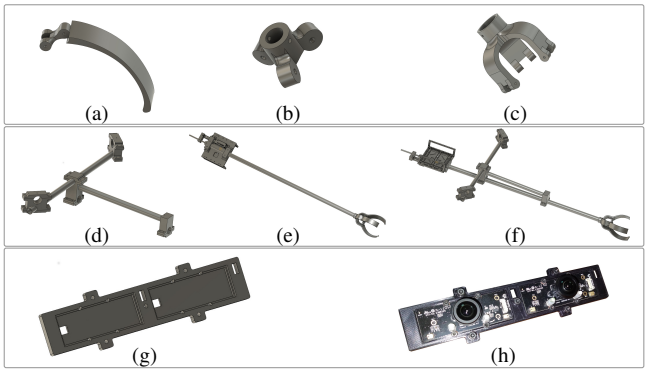

Figure 4: CAD models. (a) finger, (b) finger coupler, (c) wrist, (d) gripper support, (e) gripper without support, and (f) gripper with support, (g) stereo rig, and (h) stereo rig with cameras.

(Fig. 4b). A push action on the coupler produces outward torque on the fingers, leading to the gripper opening, while a pull action produces inward torque, leading to the gripper closure. The push and pull actions are realized by linking the coupler with a carbon-fiber tube of outer diameter 6mm, called *coupler link*. The coupler link runs inside another carbon-fiber tube of outer diameter 16mm, called *arm*. At one end of the arm, the wrist is mounted, whereas another end is attached to a *plate*, on which the actuator is mounted. The actuator provides linear motion to the coupler link that precisely controls the gripper opening or closure. The actuator and the coupler link are coupled via a 3D printed part (Fig. 3).

The actuator comes from Actuonix P16-P with 10cm stroke length and voltage feedback to obtain the current extruded length. The control circuit is developed in-lab, consisting of a PIC18F2550 microcontroller used for ADC conversion of the feedback signal and to drive the actuator. The microcontroller communicates with the onboard computer via a serial link.

We mount the gripper at the bottom of the UAV. The arm of the gripper is quite long (84cm from the UAV center), which introduces severe vibrations because it is left open at the wrist end. To suppress them, we design a gripper support structure (Fig 4d) which holds the arm firmly (Fig. 4e, 4f). Please see the video for a visual illustration of the vibrations.

Our gripper fulfils the size, weight and power constraints, and has fairly easy assembling, thanks to its modular design.

C. Vision Sensors

Our positioning system is based on Stereo visual SLAM; Hence, from a cost and customization standpoint, we take two Lenovo 300 FHD cameras and extract their PCB. Then we mount the PCBs onto a 3D-printed part to form a stereo-rig (Fig. 4g, 4h). We use OpenCV [26] for calibrating the rig.

D. Communication Device and Emergency Kill-Switch

We use ASUS GT-AX-11000 wireless router to link the ground station and the UAV for real-time data logging and monitoring. We develop a *software kill-switch* that is available on the ground station and halts the UAV in case of fatality.

E. Onboard Computing Infrastructure

We use one NVIDIA Jetson Xavier NX, a low-powered 10|20W embedded computer with 6 CPU cores, 384 GPU cores for parallel computing, 8GB RAM and a dual-band WiFi, and four USB 3.0 ports for interfacing peripherals such as USB-Serial link, Intel Realsense D435i depth sensor. The flight controller communicates with the Jetson NX via a UART available at `/dev/ttyTHS0` in the operating system.

F. Harvesting Region and Target Fruit

Harvesting farms may be unavailable near the research lab (Sec. I-9). We address this challenge and propose a harvesting setup resembling plants in vertical farming or orchards. We use a wooden trellis that is attached to the harvesting setup skeleton and is wrapped with artificial leaves to simulate plants. The setup stands tall at 1.50m, while the harvesting region begins at 0.80m and measures 2.20m $\times$0.70m (Fig. 1).

Our harvesting setup offers several benefits: *First*, it can be kept indoors, allowing the testing of the algorithms and dataset collection. *Second*, it allows the iterative system design process (Sec. I-1), i.e. build-modify-test without going into the fields and waiting for favourable weather conditions. These attributes speed up the overall system design and testing.

We use apple fruit as the target object, weighing 300gms and having radii $\sim$6cm. In this work, we keep only one type of target to avoid too much complexity in the system design, because the UAV-based harvesting is already quite complex.

III. Flight Autonomy Sub-Systems

Flight Autonomy is the most fundamental and state-of-the-art challenge to realize 4^{th} generation of aerial manipulators [4]. In this work, we propose an advanced flight autonomy engine based on our high-speed algorithms, which outperforms state-of-the-art algorithms while catering for the constraint of limited onboard computing power. Below, we describe our algorithm which we develop for the system autonomy.

A. High-Speed Accurate Localization

A 4^{th} generation system would require localization without visual markers or costly off-the-shelf GPS-RTK sensors [22], leading to challenging research problems since speed, metric accuracy, and precision are critical for control and grasping (Sec. I-3) while having high speed. Hence, we base our localization module on stereo visual SLAM due to its high metric accuracy, however, its high computing costs [14] becomes a bottleneck for Jetson NX-like devices in the presence of other algorithms.

Hence, we develop a GPU-accelerated, resource-efficient and accurate stereo visual SLAM system (Fig. 5, [27]) reaching beyond 60Hz @432×240 even at eight scales on Jetson NX

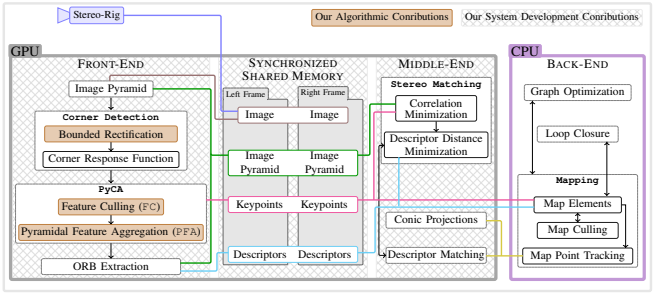

Figure 5: Our SLAM pipeline with our contributions highlighted.

alongside other algorithms. It is based on our novel components of (*i*) Bounded Rectification to prevent tagging of non-corners as corners, and (*ii*) Pyramidal Culling and Aggregation (`PyCA`) to obtain robust features at high speeds by harnessing a GPU device. PyCA is based on our novel techniques of feature culling, pyramidal feature aggregation, efficient GPU warp allocation via multi-location per thread culling and thread-efficient warp allocation. (*iii*) We also develop synchronized shared memory that turns our SLAM system resource-efficient. Our SLAM system is the fastest available accurate and GPU-accelerated system (Sec. V-A).

Further, to obtain a high-frequency state estimation of the UAV, we use Extended Kalman Filter (EKF) and fuse the onboard Inertial-Measurement-Unit (IMU) data with 6-DoF state $(x, y, z, \phi, \theta, \psi)$ provided by the SLAM system.

B. Modelling and Control System

Our UAV platform is shown in Fig. 2. We define a *world frame* $\mathcal{F}_W = [\boldsymbol{x}_W, \boldsymbol{y}_W, \boldsymbol{z}_W]$ fixed at the take-off point, and a UAV *body frame* $\mathcal{F}_B = [\boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B]$ coinciding with the UAV's Center-of-Gravity (CoG). $\mathcal{F}_B$ is described relative to $\mathcal{F}_W$ via a position vector $\boldsymbol{p}_B = [x, y, z]^T \in \mathbb{R}^3$ and a rotation matrix $\boldsymbol{R}_B \in SO3$. Both $\mathcal{F}_W$ and $\mathcal{F}_B$ follows *Front-Left-Up* convention, denoting their $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ axes, respectively. The UAV's velocity, acceleration, and angular velocity are denoted as $\boldsymbol{v}_B, \boldsymbol{a}_B, \boldsymbol{\omega}_B \in \mathbb{R}^3$, expressed in $\mathcal{F}_W$, except $\boldsymbol{\omega}_B$ is in $\mathcal{F}_B$.

The UAV motion is governed by the *thrust* $f_B \in \mathbb{R}$ applied parallel to $\boldsymbol{z}_B$, and *torques* $\boldsymbol{\tau}_B \in \mathbb{R}^3$ operating in $\mathcal{F}_B$. These are controlled via four commands: namely *thrust, roll, pitch, yaw* or simply $\{f_B, \phi, \theta, \psi\}$. The above model description leads to the following rigid body dynamics of the UAV:

$$\dot{\boldsymbol{p}}_B = \boldsymbol{v}_B, \quad \dot{\boldsymbol{v}}_B = \boldsymbol{a}_B \tag{1}$$

$$\boldsymbol{f}_B = m\boldsymbol{a}_B + mg - \boldsymbol{f}_e \tag{2}$$

$$\dot{\boldsymbol{R}}_B = \boldsymbol{S}(\boldsymbol{\omega}_B)\boldsymbol{R}_B, \quad \boldsymbol{J}_B\dot{\boldsymbol{\omega}}_B = -\boldsymbol{S}(\boldsymbol{\omega}_B)\boldsymbol{J}_R\boldsymbol{\omega}_B + \boldsymbol{\tau}_B S \tag{3}$$

where, $\boldsymbol{f}_e \in \mathbb{R}^3$ is external disturbance, m is the UAV mass, and J_B is the inertia matrix. $\boldsymbol{S}(\cdot)$ is a skew-symmetric matrix. A '$\star$' superscript denotes the desired value of any variable.

The thrust controller in the above model (Eq. 2) poses a challenge in aerial grasping due to the dynamic payloads, wind drafts, and battery issues (Sec. I-4). We address them simultaneously via our novel Thrust Microstepping via Accelerometer Feedback [28] technique. It accurately estimates the thrust in the position controller of the quadrotor control, even without the knowledge of UAV mass and gravity, outperforming the existing

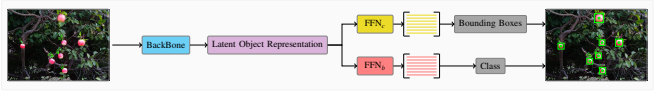

Figure 6: Our object detection pipeline, free of post-processing steps.

thrust controllers (Eq. 2) [17]. In that case, the definition of our thrust controller becomes:

$$\boldsymbol{f}_B^\star = \alpha \boldsymbol{e_a} + \beta \dot{\boldsymbol{e}}_{\boldsymbol{a}} + \boldsymbol{f}_{B_{t-1}}^\star \tag{4}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are the tunable parameters, $\boldsymbol{e_a} = \boldsymbol{a}_B^\star - \boldsymbol{a}_B$, and $\boldsymbol{f}_B^\star$ is the desired thrust vector to attain $\boldsymbol{a}_B^\star$. See [28] for details.

Further, aerial grasping necessitates a velocity controller for visual servoing tasks. Hence, we developed a position controller based on the proposed thrust controller. Based on the desired position or velocity, the desired acceleration $\boldsymbol{a}_B^\star$ is obtained that is fed to the thrust controller. The obtained thrust is fed to [29] to calculate desired orientation. Thus, the overall control system outputs $\{f_B^\star, \phi^\star, \theta^\star, \psi^\star\}$ which is sent to the attitude controller via OSDK (Sec. II-A). Our control system offers independent control in all three axes, i.e. an axis can operate in position or velocity mode irrespective of the others, facilitating precise navigation and visual servoing. This is the major novelty of our control system design, along with its mass and gravity-agnostic nature. See Sec. V-A for the comparison with the state-of-the-art.

C. High-Speed Object Detection

Object detection is a critical component in aerial grasping because any error in it can cause incorrect target grasp or even failure. Hence, we use deep learning-based detectors for their accuracy, but they suffer from many limitations in our context (Sec I-2); thus, we redesign the detection head in CNN-based detectors because it consists of most of the hyperparameters and post-processing steps. Our detector FFD [30] is exceptionally lightweight, single-stage, single-scale, free of anchor-box and NMS while having much-simplified training and testing phases (Fig 6). Thus, FFD reaches 100FPS@FP32-precision on Jetson NX while co-existing with the other time-critical sub-systems, such as localization, control, and grasping.

In FFD, an image is forwarded through a CNN backbone with a progressive reduction in the spatial size. The backbone is customized VGG [31], having five stages with $\{2,2,3,3,4\}$ layers and $\{16,32,64,128,256\}$ neurons per stage, operating at a stride of two. The backbone output is fed to our novel Latent Object Representation (`LOR`) module, which aggregates global information via Cross Channel Global Context (`CCGC`), and outputs queries similar to Transformer-based detectors but without using Transformers. Each query represents an object which refined using Query Transformation (`QT`). The queries are finally passed through two Feed Forward Neural-Networks (FFN), producing classification scores and bounding boxes.

Further, we adapt synthetic scene synthesis to generate vast training data. This is a scalable approach in real time as collecting and labelling fruit images is a time-consuming task since these images consist of many instances.

Our FFD outperforms many state-of-the-art multi-scale detectors in various aspects, i.e. speed and accuracy. See Sec. V-A for the evaluation.

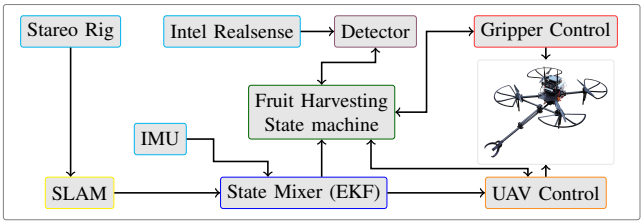

Figure 7: Connectivity between different harware (□, □) and software (□, □, □, □, □, □) components.

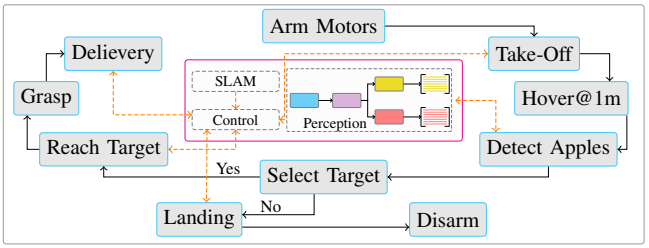

Figure 8: The proposed state machine

IV. State-Machine for Aerial Grasping Autonomy

Despite developing the subsystems, designing a state-machine for the aerial grasping task is complicated due to switching between many phases [22] i.e. detection, visual servoing, grasping, target delivery etc. Hence, we develop a novel end-to-end state machine based on a few novel techniques. It is the final component of our aerial grasping system, which communicates with all the subsystems (Fig. 7) to enable the UAV to perform the challenging fruit harvesting task autonomously indoors. The state-machine diagram is shown in Fig. 8. Our software is written in C++ and uses Robot Operating System (ROS) for inter-process communication.

A. Hovering

Precise hovering is critically important, otherwise, it results in camera motion blur, leading to object detection and tracking failure. Our control system achieves precise hovering via thrust microstepping and accelerometer feedback.

B. Target Selection and Associated Information

After achieving stable hovering ($\pm$5cm), the detector detects fruits in the RGB image acquired from the D435i. In our experiments, we choose the target as the fruit instance closest to the UAV in the line of sight. Now we compute the target's 3D centroid using the D435i depth measurements.

C. Target Tracking and Visual Servoing

The target centroid is used to navigate near the targets, achievable in two ways: (`i`) open loop, and (`ii`) closed loop or visual servoing. In the open loop, the target's location is sent only once to the control system, and then navigation is executed in position control mode blindly. For this reason, if UAV drifts due to wind drafts during navigation, the final UAV position may differ from the desired target's location.

Whereas in the closed loop, the target is continuously tracked in the RGB frames while updating its 3D centroid. In this case,

Figure 9: Drone-Bee performing harvesting in different scenarios. *Top:* Indoor fruit plucking. *Bottom:* Outdoor fruit plucking. A '▭' denotes UAV.

the control system is set to velocity mode in x (front), while position mode in y (side) and z (vertical). This significantly boosts the system's accuracy. We use color-based tracking to avoid algorithmic complexity. This may fail during overlapped instances but can be easily tackled via depth information and/or deep learning-based tracker, which we leave for future work.

D. Object Localization via Instance Mapping

In visual servoing, noisy target centroid is a critical issue. It originates due to noisy depth measurements (Sec. I-5, I-6) and also when UAV can not maintain absolute zero error from the set point. Therefore, we develop *instance mapping* in 3D, which generates a 3D map containing the locations of all the instances detected so far. In this strategy, the target location from the object tracker is not directly used; instead, first, the map is updated with the tracker output, and then the location is obtained from the map. Hence, even if the tracker fails during visual servoing, we still have access to the target's location to continue approaching the target.

Note: The UAV localization is independent of the number of apples since it is carried out by our SLAM-based localization. Hence it should not be confused with the apple localization.

E. Grasp Synthesis

Grasp synthesis refers to generating a grasp pose to grasp a target successfully. In our case, after reaching near the target ($\pm$2cm), the gripper is closed, and the UAV performs a backward motion to separate the fruit from its stem. To execute a grasp sequence, the gripper is opened early, i.e. before the commencement of the visual servoing. The UAV now reaches back to the hovering spot and releases the target. The UAV can also be programmed to deliver the fruit to a prefixed location. However, we only have demonstrated the grasping capabilities. See Fig. 9 for a visual evolution of the fully autonomous execution of the proposed state machine.

V. System Performance

We extensively evaluate our *Drone-Bee* system. We perform indoor and outdoor experiments to incorporate the effect of rotor draft, wind turbulence, constrained workspaces, and lighting. The UAV takes off and lands 1.8m far from the harvesting setup, measured horizontally. Since this is a system paper, we evaluate the overall system performance while discussing only the key results of our subsystems. See [27], [30], [28] for a detailed sub-system evaluation.

Table II. SLAM evaluation on KAIST-VIO sequences [32].

Approach	KAIST-VIO Sequence											
	circle			infinite			square			rotation		
	normal	fast	head	normal	fast	head	normal	fast	head	normal	head	
• KIMERA-VIO [33]	0.12m	0.07m	0.28m	0.05m	0.14m	1.08m	0.17m	0.19m	1.57m	0.17m	0.74m	
• VINS-Fusion + GPU [34]	0.09m	0.13m	0.11m	0.09m	0.05m	0.14m	0.12m	0.11m	0.15m	0.12m	0.11m	
• ORB-SLAM2 [14]	0.09m	0.11m	0.13m	0.08m	0.10m	0.12m	0.09m	0.09m	0.16m	0.17m	0.21m	
• Our Pipeline	**0.014**m	**0.017**m	0.12m	**0.017**m	**0.016**m	**0.09**m	**0.016**m	**0.017**m	**0.04**m	**0.07**m	**0.09**m	

Table III. Detection Performance of FFD. AP average precision. 'S' and 'M' denotes small ($< 30 \times 30$) and medium-sized ($< 90 \times 90$ pixels) objects.

Detector	AP	AP_S	AP_M	Per Iteration Train-time	Inference time@FP32
SSD @multi-scale [10]	38.0	20.1	39.1	4.60s	32ms
DETR @single-scale [12]	40.2	24.9	43.0	6.80s	25ms
Faster-RCNN @multi-scale [9]	45.9	28.7	51.5	5.10s	49ms
YOLO-v8 @multi-scale [13]	46.3	29.2	51.5	2.65s	29ms
YOLO-v8 @single-scale [13]	35.2	23.1	41.4	0.95s	24ms
FFD @single-scale	**46.6**	**31.2**	**52.1**	**0.40**s	**11**ms

Figure 10: A few indoor and outdoor detection results. Boxes in red are the predictions while the ones in green are groundtruth.

A. Localization, Perception and Control

Our localization system [27] is a quite fast and accurate system. Table II shows metric accuracy of our method against state-of-the-art systems on the recent KAIST-VIO [32] dataset for UAVs. Our SLAM system can reach beyond 60FPS on Jetson-NX at 432×240 resolution despite in stereo mode, which is the most unique feature of our localization system.

Our detector [30] outperforms prominent [9] and recent SOTA [13], [12] detectors by having faster training, inference, and promising results on small objects without multi-scale detection (Table. III, Fig. 10). Our detector achieves this performance due to the novel design of its head (Sec. III-C).

Our control system [28] offers precise hovering, navigation and handles dynamic payloads via accurate thrust estimation. We compare our thrust controller against SOTA thrust controller [17], which we outperform. See Figure 11.

Notably, the individual achievement of each subsystem makes it possible to realize Drone-Bee, which can make its own decisions onboard. Hence, running the entire stack into a limited computing budget is the major novelty and contribution of this paper, which is also listed as a critical bottleneck in [4]. By addressing this bottleneck, we push aerial manipulation towards 4^{th} generation.

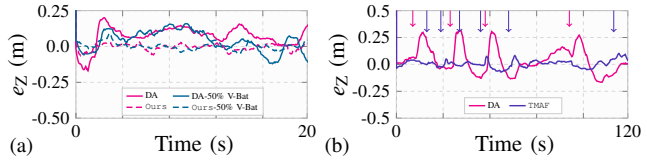

Figure 11: Hovering performance of our control system *vs* DA [17] in different settings. (a) DA achieves an RMSE of 0.16m but only 0.03m with TMDC. While for 50% battery discharge, we observe RMSE of 0.15m with DA and 0.02m with our control system. (b) Test for external disturbance rejection of 15N during hovering. The arrows ↓ and ↓ indicate disturbance introduction. Here DA shows a peak offset of 0.25m whereas ours only of 0.07m.

B. Evaluation Metrics & Grasping Performance

We define three error metrics to quantify the grasping performance, as reported in robotic manipulation tasks [3].

1) Grasp Success Rate: A successful grasp occurs when an item is gripped fully or partially by the gripper. A partial grasp occurs when the target does not lie in the grasping cavity of the gripper, i.e. lies outside of the jaw. While full grasping occurs when the object lies completely inside the jaw. To evaluate the grasp success rate, we vary the number of fruits

As shown in Table IV, Exp-1, our system grasps items accurately. A few times, the system was observed to fail a grasp. As per our analysis, depth measurements ($\pm$0.03m) are quite noisy, which resulted in incorrect grasp point. It led to the gripping of the trellis while attempting a grasp. In practice, such kind of trellis is not present except small stems, which do not pose any restriction in front of the system.

2) Average Time per Instance: It is the average time taken to execute a grasp sequence, measured from target selection to target delivery at a specified spot. To evaluate the average time per instance, we perform two experiments, each with 5 trails: (*i*) change the number of fruits, and (*ii*), we fix the number of fruits to 5 but vary their position. Exp-2 and Exp-3 in Table IV shows the corresponding results. It can be seen that the average time varies from $\sim 3-8$ seconds. The smaller time is required when the fruits lie nearly in the line-of-sight, while the larger time is required when the target lie near the boundaries of the harvesting region and the UAV is flying in the middle.

3) Error Rate: It is the ratio of a number of grasped items dropped before reaching the delivery spot, and a total number of grasped items. Based on Exp-$1-3$, we did not observe any event which could contribute to the error rate.

C. Visual Servoing Performance

We evaluate visual servoing due to its key role in grasping. To setup the experiments, we adhere a fruit target to a stick which can be freely moved. The UAV is instructed to stay 0.20m before the target while pointing towards it. Now, we move the stick in x, y, z axis simultaneously. We report the deviation of UAV from the desired configuration. We find that the system can perform visual servoing precisely (Table V).

D. Frame-Rate & Computing Resource Occupancy

Deploying complex autonomy systems onboard is an unsolved challenge [4] in transitioning towards 4^{th} generation aerial manipulators. Hence, we evaluate our flight and decisional autonomy in terms of computing resources. Table VI shows the maximum frame rates of various autonomy subsystems and the computer resource profiles, e.g. power consumption and temperature. Notably, despite a tremendous load, 30% of computational space is still left. This clearly caters for the goal of successfully deploying the autonomy engine onboard, indicating the major achievement of this work.

Table IV. Grasping Performance Evaluation.

Exp-1		Exp-2 @ Variable #Fruits		Exp-3 @ Variable Position	
#Fruits	Grasp Success rate (%)	#Fruits	Average Grasping Time	#Fruits	Average Grasping Time
2	100%	2	4s	5	9s
3	66%	3	6s	5	11s
6	80%	6	5s	5	8s
8	87%	8	4s	5	7s

Table V. Visual servoing performance.

Metric	Δx	Δy	Δz	$\Delta\psi$
mean (μ)	0.021m	0.029m	0.018m	2^o
standard-deviation (σ)	0.019m	0.023m	0.020m	3^o

Table VI. Frame processing rates and computing resource utilization.

Algorithmic Component	Frame rate	Computing Resource	Utilization
Stereo Image Acquisition	30 FPS @432×240	RAM	40%
Image Rectification	1000 FPS @432×240	CPU	70%
SLAM	60 FPS @432×240	GPU	45%
Detector	100 FPS @320×240	Power	20W
Realsense Image Acquisition	60 FPS @320×240	Temperature	55^oC

E. Prototype Hardware Assessment

We examined gripper durability by continuously opening and closing it for ~ 5 hours. Although it is a short duration, it is sufficient for a 3D printed part. Post the experiment, no wear-and-tear was seen in the gripper assembly, indicating that our mechanical designs are stable for proof-of-concept and can be affordably reprinted by researchers interested in this area. We use PLA material for 3D printing; however, other suitable materials can be chosen for greater strength.

VI. Hidden Challenges, Solutions and Future

Apart from the core aerial grasping challenges (Sec. I), we also introduce the hidden challenges faced during real-time implementation. We also discuss the prospects of the paper.

1) A Coaxial Rotor UAV: In our UAV, the long arm introduces off-center load intensively. The long arm was required to maintain a safety gap of 30cm between the rotors and the gripper, given the large size of the UAV used. If not accounted for, the propellers may hit nearby structures while executing a grasp sequence. Thus a short arm is more beneficial, but it requires the UAV to span a smaller area. This issue can be resolved via a coaxial rotor UAV.

2) Gripper Improvements: Although the gripper developed is sufficient for this task, we observed that if the fruit instance is attached to the trellis firmly, the UAV needs to exert a pull effort to detach the target. This issue can be resolved by improvising the gripper with a rotatable wrist.

3) Depth Sensing: One major cause of grasp failures was inaccurate depth sensing. Intel D435i provides noisy depth, and due to which, even hovering at the exact location, the depth of the target instance is measured $\sim 2-3$cm in or out of the harvesting region, leading to the gripping of trellis woods. A potential solution can be to use stereo triangulation.

4) Hardware Synchronized Stereo-Rig: We strongly recommend a hardware synchronized stereo-rig where both of the cameras are triggered at the same time. It is so

because, sometimes, due to the operating system overhead or consumption of computing resources, it is not possible to grab images from each camera in a time-synchronized manner. Hence, two images may have different capture-time which negatively affects rectification and stereo matching, thus causing errors in the 6DoF pose estimation.

5) Depth Measurements Rate: We used registered depth which is aligned w.r.t. to the RGB camera. By default, the registered depth is computed on the host CPU instead of the ASIC of D435i, which can provide the depth images at a rate of only $\sim 2-3$ FPS on Jetson-NX. We solved this issue by compiling Intel Realsense SDK with CUDA to enable GPU operations. By doing so, we achieve a rate of ~ 30 FPS.

VII. Conclusion

This work introduces comprehensive hardware design and flight autonomy engine for off-center aerial grasping in GPS-denied environments. The contributions of the paper are: to present aerial harvesting challenges, intricate details of the hardware designs and in-lab fabrication, sensor selection and integration, experimental setup realization, and system integration along with developing crucial sub-systems such as object detection, positioning system, and control system. The flight autonomy engine can run on a $\sim$ 10W NVIDIA Jetson-NX embedded computer which is the representative achievement of this paper to push aerial manipulators state-of-the-art towards 4^{th} generation. We call the resulting system as Drone-Bee, which is demonstrated for a challenging task of fruit harvesting. We evaluate Drone-Bee by conducting several experiments and show that it is capable of grasping desired items precisely. With its capabilities, the Drone-Bee system opens new doors to the future of autonomous aerial agriculture.

References

[1] A. Kumar, M. Vohra, R. Prakash, and L. Behera, "Towards deep learning assisted autonomous UAVs for manipulation tasks in gps-denied environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1613–1620, IEEE, 2020.

[2] A. Kumar and L. Behera, "Semi supervised deep quick instance detection and segmentation," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8325–8331, IEEE, 2019.

[3] D. Morrison, A. W. Tow, M. McTaggart, R. Smith, N. Kelly-Boxall, S. Wade-McCue, J. Erskine, R. Grinover, A. Gurman, T. Hunn, D. Lee, A. Milan, T. Pham, G. Rallos, A. Razjigaev, T. Rowntree, K. Vijay, Z. Zhuang, C. F. Lehnert, I. D. Reid, P. Corke, and J. Leitner, "Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge," 2017.

[4] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 626–645, 2021.

[5] M. Beul, M. Schwarz, J. Quenzel, M. Splietker, S. Bultmann, D. Schleich, A. Rochow, D. Pavlichenko, R. A. Rosu, P. Lowin, *et al.*, "Target chase, wall building, and fire fighting: Autonomous UAVs of team nimbro at MBZIRC 2020," *arXiv preprint arXiv:2201.03844*, 2022.

[6] L. Y. Lee, O. A. Syadiqeen, C. P. Tan, and S. G. Nurzaman, "Closed-structure compliant gripper with morphologically optimized multi-material fingertips for aerial grasping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 887–894, 2021.

[7] A. McLaren, Z. Fitzgerald, G. Gao, and M. Liarokapis, "A passive closing, tendon driven, adaptive robot hand for ultra-fast, aerial grasping and perching," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5602–5607, IEEE, 2019.

[8] F. Ruggiero, M. A. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Peréz, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor UAVs equipped with a servo robot arm," in *International conf. on robotics and automation (ICRA)*, pp. 4014–4020, IEEE, 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *CVPR*, 2017.

[12] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conf. on Computer Vision*, pp. 213–229, Springer, 2020.

[13] "YOLO-v8," in *https://github.com/ultralytics/ultralytics*.

[14] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[15] L. Von Stumberg and D. Cremers, "DM-VIO: Delayed marginalization visual-inertial odometry," *IEEE Robotics and Automation Letters*, 2022.

[16] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "Stability of small-scale UAV helicopters and quadrotors with added payload mass under pid control," *Autonomous Robots*, vol. 33, no. 1, pp. 129–142, 2012.

[17] M. Hamandi, M. Tognon, and A. Franchi, "Direct acceleration feedback control of quadrotor aerial vehicles," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5335–5341, IEEE.

[18] G. Heredia, A. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. Acosta, and A. Ollero, "Control of a multirotor outdoor aerial manipulator," in *2014 IEEE/RSJ international conference on intelligent robots and systems*, pp. 3417–3422, IEEE, 2014.

[19] A. Suarez, P. R. Soria, G. Heredia, B. C. Arrue, and A. Ollero, "Anthropomorphic, compliant and lightweight dual arm system for aerial manipulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 992–997, IEEE, 2017.

[20] B. Stephens, L. Orr, B. B. Kocer, H.-N. Nguyen, and M. Kovac, "An aerial parallel manipulator with shared compliance," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11902–11909, 2022.

[21] G. Corsini, M. Jacquet, H. Das, A. Afifi, D. Sidobre, and A. Franchi, "Nonlinear model predictive control for human-robot handover with application to the aerial case," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7597–7604, IEEE, 2022.

[22] G. Loianno, V. Spurny, J. Thomas, T. Baca, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho, *et al.*, "Localization, grasping, and transportation of magnetic objects by a team of mavs in challenging desert-like environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1576–1583, 2018.

[23] N. Häni, P. Roy, and V. Isler, "Minneapple: a benchmark dataset for apple detection and segmentation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 852–858, 2020.

[24] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3626–3633, IEEE, 2017.

[25] P. Roy and V. Isler, "Surveying apple orchards with a monocular vision system," in *2016 IEEE international conference on automation science and engineering (CASE)*, pp. 916–921, IEEE, 2016.

[26] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[27] A. Kumar, J. Park, and L. Behera, "High-speed stereo visual slam for low-powered computing devices," *IEEE Robotics and Automation Letters*, 2023.

[28] A. Kumar and L. Behera, "Thrust microstepping via acceleration feedback in quadrotor control for aerial grasping of dynamic payload," *IEEE Robotics and Automation Letters*, 2023.

[29] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on se (3)," in *49th IEEE conference on decision and control (CDC)*, pp. 5420–5425, IEEE, 2010.

[30] A. Kumar and L. Behera, "High-speed detector for low-powered devices in aerial grasping," *IEEE Robotics and Automation Letters*, 2024.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[32] J. Jeon, S. Jung, E. Lee, D. Choi, and H. Myung, "Run your visual-inertial odometry on nvidia jetson: Benchmark tests on a micro aerial vehicle," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5332–5339, 2021.

[33] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1689–1696, IEEE, 2020.

[34] "Vins-fusion-gpu," in *https://github.com/pjrambo/VINS-Fusion-gpu*.