2024 International Conference on Unmanned Aircraft Systems (ICUAS)
June 4-7, 2024. Chania, Crete, Greece

Towards Instance Segmentation-Based Litter Collection with Multi-Rotor Aerial Vehicle

Filip Zoric[1], Antonio Franchi[2,3], Matko Orsag[1], Zdenko Kovacic[1], Chiara Gabellieri[2]

***Abstract*— This paper presents a novel aerial robotics application of instance segmentation-based floating litter collection with a multi-rotor aerial vehicle (MRAV). In the scope of the paper, we present a review of the available datasets for litter detection and segmentation. The reviewed datasets are used to train a Mask-RCNN neural network for instance segmentation. The neural network is off-board deployed on an edge computing device and used for litter position estimation. Based on the estimated litter position, we plan a path based on a quadratic Bezier curve for the litter pickup. We compare different trajectory generation methods for the object pickup. The system is verified in a laboratory environment. Eventually, we present practical considerations and improvements necessary to enable autonomous litter collection with MRAV.**

***Index Terms*— instance segmentation, multirotor aerial vehicle, trajectory planning, litter collection**

2024 International Conference on Unmanned Aircraft Systems (ICUAS) | 979-8-3503-5788-2/24/$31.00 ©2024 EU | DOI: 10.1109/ICUAS60882.2024.10557120

I. Introduction

Uncrewed aerial vehicles (UAVs) have entered the mainstream, mostly in the form of lightweight (<3kg) multi-rotor aerial vehicles (MRAVs) for aerial photography. Besides being passive actors, uncrewed aerial manipulator systems actively interacting with the environment have attracted the interest of the research community [1]. As the number of UAVs and registered operators has been growing worldwide, there is an opportunity to explore new and promising applications that our society and economy in general could benefit from. One of the problems of the modern world is directly related to hyperconsumption. As humans' need to consume has heavily grown in the last few decades, mainly due to economic and technological advancements, there are new challenges that pose threats to the environment. One of the major problems is related to packaging, which often ends up in the environment instead of in the waste collection systems. As the European Environmental Agency states, around 40% of plastic production is for product packaging. Some studies identified rivers as a dominant way for plastic accumulation in our oceans [2]. It is well known that plastic pollution has detrimental effects on the environment, mainly due to the endangering effects it has on fish, seabirds, and marine animals (e.g., risk ingestion or remaining entangled) [3].

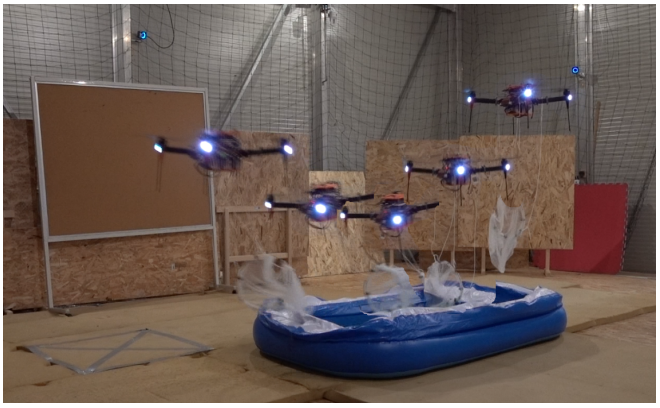

Fig. 1: MRAV executing trajectory planned based on the visual litter position estimation for litter pickup. Video of instance-segmentation based litter collection experiments can be found at `https://youtu.be/605jgRNkcP4`. Video of the FlyFlic proof of concept used in the real canal can be found at `https://youtu.be/_md0IJnaccU?si=75Dwq4RmHvNn9-6a`.

In order to prevent some of the plastic from entering marine systems, many autonomous or semi-autonomous solutions have been proposed. The state of the art of riverine garbage collection is mainly split into two areas: fixed trapping mechanisms and boat-like solutions. Examples of fixed solutions are systems of floaters and nets, such as Interceptor by charity organization The Ocean Cleanup [4], bubble barriers [5], and chains of floating gears to not only stop but also accumulate litter [6]. Moving (semi)autonomous solutions typically rely on floating boat robots, such as RanMarine [7] and SeaVax [8]. We refer the interested reader to the PlasticSoup Foundation webpage [9], which did a great job in collecting many existing solutions.

We propose an autonomous aerial manipulator system for litter collection in rivers. Compared to state-of-the-art solutions, the proposed platform is cost-effective and able to reach non-navigable spots, characterized, e.g., by partly dry sections, low bridges, partially underground portions, dams, and jumps. Moreover, it can be easily deployed in and out of the water and target specific areas on demand.

The main contribution of this paper is the introduction of the first Flyflic prototype (Flying Companion for Floating Litter Collection). Fig. 1 shows Flyflic during litter pickup. To the best of the authors' knowledge, that is the first attempt to create a working concept for instance segmentation-based floating litter collection with MRAVs. During the course

[1]Authors are members of the Laboratory for Robotics and Intelligent Control Systems (LARICS) at the Faculty of Electrical Engineering and Computing, University of Zagreb, Unska ulica 3, 10000 Zagreb, Croatia, e-mail: `filip.zoric@fer.hr`

[2]Authors are members of the Robotics and Mechatronics Group from the University of Twente, e-mail: `c.gabellieri@utwente.nl`, `a.franchi@utwente.nl`

[3]Authors are members of the Department of Computer, Control and Management Engineering, Sapienza University of Rome, 00185 Rome, Italy, `antonio.franchi@uniroma1.it`

This work was partially supported by the H2020 research and innovation programs under agreement no. 101059875 (Flyflic) and the Horizon EU research and innovation programme [grant agreement No. 101120732] AUTOASSESS

of prototype development, a series of practical challenges were identified, encompassing issues related to consistent litter position estimation, trajectory generation and execution, ground effect during object retrieval, and system integration. In response to these challenges, we present herein a collection of prospective solutions aimed at mitigating potential adverse consequences of the aforementioned challenges.

A summary of the content of each section of this paper is the following. In Section II, we present related work on top of which we build. In Section III, we present the system design; In Section IV, we present the methodology used to approach perception and trajectory generation. In Section V, we present experiments done to validate the proposed system. Eventually, closing remarks and future work are in Section VI.

II. Related work

In [10], an autonomous aerial manipulator system for object pick and place is presented. For a fully autonomous system, the authors implemented a multi-task visual recognition system based on convolutional neural networks (CNNs). They also provide an overview of the most important CNNs as well as challenges that arise when developing such a system. In [11] the authors present a fully autonomous aerial manipulator. The system is comprised of an object detection module for perception purposes and an adaptive sliding mode controller based on the coupled dynamics of the aerial manipulator system. Both of the presented papers utilize hexarotors as the flying platform and Jetson TX2 as the onboard computing device.

Computer vision has a lot of potential applications in the waste management industry, especially for recycling. As deep learning became the dominant approach for solving computer vision challenges, there is a vast variety of open datasets available. Trash Annotations in COntext (TACO), as presented in [12], is an open image dataset for litter detection and segmentation that consists of 1500 labeled images with multiple litter categories. Aside from labeled official images, there are over 8000 unofficially annotated images; the authors also proposed a method for augmenting the dataset with synthetic images. In [13], the authors provide a dataset for low-attitude waste detection with UAVs. The dataset consists of more than 700 annotated images which are all classified as waste. Besides the lack of classes, object-level mask annotations are not available, making it only suitable for object detection. TrashCan dataset, presented in [14], consists of more than 7000 annotated underwater images, which make it well-suited for underwater robotics applications. In [15], the authors provide an in-depth analysis of litter segmentation with deep-learning-based instance segmentation methods which was highly informative, especially for our use case.

In [16], the authors present image-based visual servoing (IBVS) for aerial grasping and perching. In the paper, the authors show that it is feasible to execute fast grasping and perching with a micro aerial vehicle (MAV) based on IBVS. The authors draw inspiration from fast-moving birds, such as raptors, that detect and locate the prey and execute maneuvers to catch it. Instead of using IBVS, we opted for position-based visual servoing (PBVS) to provide the ability to include a certain amount of reasoning based on perception. In [17], the authors presented MRAV with a compliant net for autonomous underwater vehicle retrieval. They also presented a mathematical model and controller for such a system. Compared to that work, the net used in our experiments is not designed to be compliant per se, and we did not employ a specially designed controller for it. However, the underlying dynamics of the swinging motion both in flight and during pickup can hold in the proposed scenario.

III. System Modeling and control

In this section, we present system modelling and control.

A. System modelling

For system modelling purposes, we use the quadrotor model as presented in [18]. The quadrotor's pose in the inertial world reference frame (W) is defined as:

$$\mathbf{x}_u = [x \quad y \quad z \quad \phi \quad \theta \quad \psi] \quad \in \mathbb{R}^6. \tag{1}$$

The gravity vector acts in the negative z direction of the inertial reference frame. The quadrotor body-fixed frame B is attached to the center of mass of the quadrotor. The net is modeled as a rigid body with constant offset alongside the inertial reference frame $-z_o$ with negligible roll and pitch angles, and the yaw angle same as the quadrotor. The net dynamics is neglected and observed as a mere disturbance to the system. We use fixed homogeneous transformations to describe the relationship between the MRAV base, the camera, the net, and the inertial coordinate system. To describe spatial transformation between the camera and the MRAV, we use the homogeneous transformation matrix $\mathbf{T}_B^C$. To describe the spatial relationship between the net and the MRAV base, we model it as a constant homogeneous transformation matrix, $\mathbf{T}_B^N$. To represent MRAV pose in the inertial reference frame, $\mathbf{T}_W^B$, the motion capture system measurements are used. The reference frames are shown in Fig 2.

B. System control

Low-level attitude control is implemented on board a standard PX4 autopilot. On top of that, we superimpose a cascaded PID position controller as presented in [19]. As inputs, the controllers receive a desired MRAV pose and calculate the output roll, pitch, yaw, and thrust (ϕ, θ, ψ, f) commands for PX4 autopilot. With the cascaded PID control, it is possible to execute the commanded trajectories to enable the MAV to follow the desired path. Two different trajectory trackers are used, time-optimal path parameterization with reachability analysis (TOPP-RA) [20] and model predictive control (MPC) [21]. Based on the list of points, both trackers output a complete MRAV trajectory with defined positions, velocities, and accelerations. The MPC tracker uses a constant snap model for the MAV model to generate trajectories.

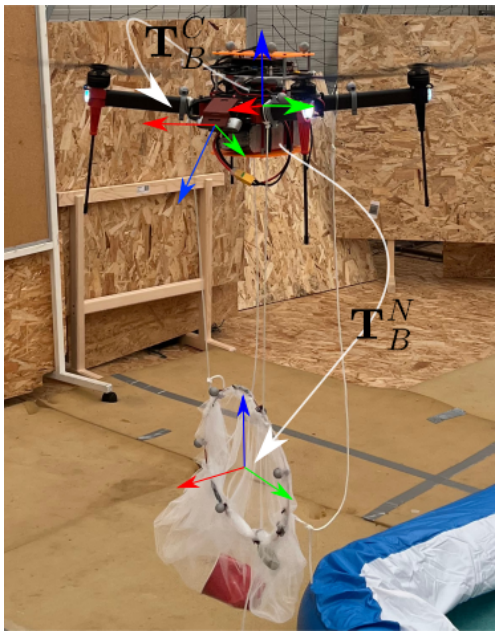

Fig. 2: Hexsoon EDU650 MRAV with the coordinate frames used to describe the spatial relationships between the camera, net, and MRAV base.

Using the presented trajectory planning algorithms makes it possible to specify velocity and acceleration constraints and to generate parameterized trajectories based on the input waypoints.

IV. Methodology

A. Litter perception

For litter perception, a neural network is used to achieve instance segmentation, enabling litter detection in different, changing environments. In the current system, object detection could easily be used instead of instance segmentation; however, instance segmentation provides richer detection information, which can be paired with a point cloud for better MRAV detection during litter pick-up. For the instance segmentation, we used OpenMMLab implementation of the Mask-RCNN [22] with the ResNet101 backbone trained on the full unofficial TACO-10 dataset[1]. After merging the official and unofficial TACO datasets, some data classes were oversampled to mitigate the negative effect of the class imbalance present in the dataset. Although the unofficial TACO dataset includes some noise, the large amount of annotated images and some simple augmentation methods (random crop, scale, and rotate) proved to be enough for the model to work reliably in the real world. The neural network was deployed on the offboard edge computing device. Compressed image transport was realized using ROS and Wi-FI 5 GHz protocol between on-board and off-board PCs. An in-depth description of the used hardware is provided in Section V.

[1]We reduced dataset to the 10 most common classes

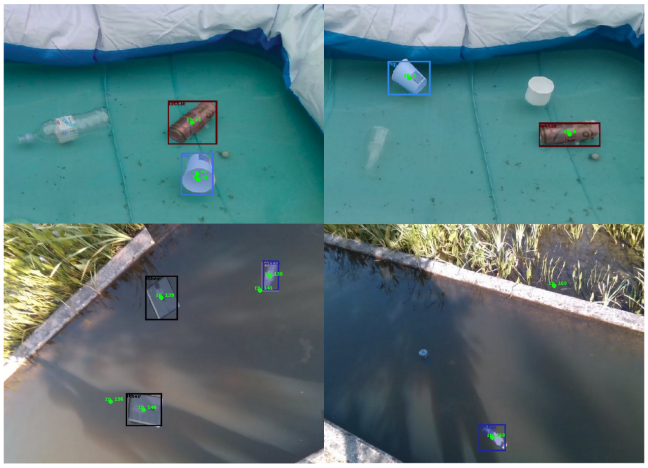

Fig. 3: Instance segmentation of the litter in the laboratory and the outdoor environment. Different colors represent different classes $\mathbf{c_s}$. Polygons represent $\mathbf{P_o}$ and green dots are $\mathbf{C_T}$.

1) Object instance segmentation: The input to the neural network is a tensor made from the image $\mathbf{I} \in \mathbb{R}^{w \times h \times 3}$, where w and h are image width and height. Tensor dimensions are $B \times w_T \times h_T \times 3$. B is the batch size equal to 1, and $w_T, h_T = 1024$. The outputs of the neural network are the detected class IDs $\mathbf{c}$, polygons that describe object masks in the image plane, $\mathbf{P_o}$, and confidence scores $\mathbf{c_s}$. For successful tracking, we filtered detected polygons ($\hat{\mathbf{P}}_\mathbf{o}$, $\hat{\mathbf{c}}_\mathbf{s}$) based on the confidence score threshold as well as class ID that needs to be collected.

From Fig. 3, it is noticeable that instance segmentation had issues with transparent objects such as clear plastic bottles and plastic bags. To mitigate such problems, more examples of such data are needed in the TACO dataset to train the network.

2) Object position estimation: After instance segmentation, we use the detected masks to obtain object-of-interest centroids $\mathbf{C}$ in the image plane. We propagate detected object centroids to the simple centroid tracker that assigns each centroid a unique ID. The simple centroid tracker compares Euclidean distances of the centroids from the current frame $\mathbf{C}_{i+1}$ and from the last frame $\mathbf{C}_i$, and it assigns a unique ID to every centroid as $\mathbf{C}_T \in \mathbb{R}^{2 \times k}$, where k corresponds to the number of tracked object centroids. If a previously detected object is not detected for 5 consecutive frames, it is removed from the $\mathbf{C}_T$. With $\mathbf{C}_T$, we extract positions of the centroids from the point cloud in the camera frame as $\mathbf{P}_T \in \mathbb{R}^{3 \times k}$.

Elements of the $\mathbf{P}_T$ are used to generate a path for the object pick-up, as it will be clear in the following paragraphs. The path is then passed to one of the trajectory planning algorithms.

The complete information propagation pipeline between PCs and the software modules is shown in Fig 4. Passing compressed image $\mathbf{I}$ and tracked centroids $\mathbf{C}_T$ between the off-board and on-board PCs causes a small delay in the system, which depends on the network quality and the distance between PCs. To reduce latency, it is reasonable to

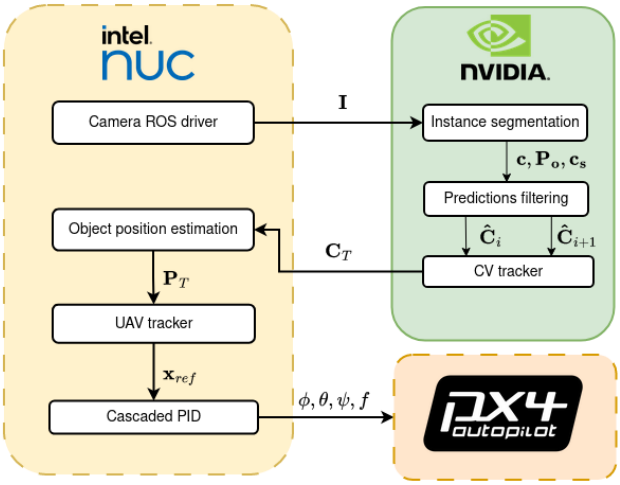

Fig. 4: Information propagation pipeline between multiple PCs for the autonomous litter collection. The green box is the Jetson Xavier AGX. The yellow box is the MRAV onboard control PC, and the orange box is Pixhawk autopilot. Boxes with a dashed border are on board the MRAV. $\mathbf{I}$ and $\mathbf{C}_T$ are sent over wireless network.

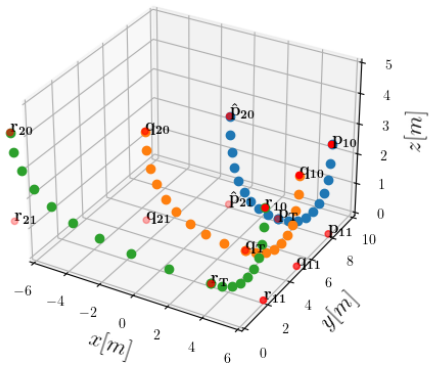

Fig. 5: Planned Bezier curves for the object pick-up with specified control points. Changing control points influences motion profile. Blue dots represent the curve with $\alpha = 1$, orange dots use $\alpha = 2$, and the green curve is for $\alpha = 3$, as defined in eq. 5

design a perception and control system on the same Jetson edge device. Due to MRAV payload restrictions, we had to use Jetson off-board.

B. Target definition

In order to pick up detected litter, we need to plan a trajectory for the net. Based on the detected object positions $\mathbf{P}_T$, we choose specific target $\mathbf{p}_{Tj}$ as the one closest to the camera which corresponds to the column j that has the smallest Euclidean norm.

The estimated object position in the camera coordinate frame is transformed in the inertial reference frame as follows:

$$^W\mathbf{p}_T = \mathbf{T}_W^B \mathbf{T}_B^C \mathbf{p}_{Tj}. \tag{2}$$

After obtaining a single target position in the inertial frame $^W\mathbf{p}_T$, we generate a trajectory for the targeted litter pickup.

C. Trajectory generation

After detecting an object, the trajectory to fetch the object of interest is obtained by merging two quadratic Bezier curves together. The result is provided to the UAV tracker which generates the trajectory.

A quadratic Bezier curve is defined as:

$$\mathbf{B}(t) = (1-t)^2\mathbf{p_0} + 2t(1-t)\mathbf{p_1} + t^2\mathbf{p_2}, \tag{3}$$

where $\mathbf{p_0}, \mathbf{p_1}$, and $\mathbf{p_2}$ represent the start point, control point, and end point, respectively, and $\mathbf{p}_i \in \mathbb{R}^3$. t is time parametrization discretized with N samples of linearly spaced values ranging from zero to one. The start point of the first Bezier curve $\mathbf{p_{10}}$ is the current MRAV position, where we use the first subscript, equal to 1 or 2, to refer to the first or second Bezier curves that we merge together. The control point for the first Bezier curve is defined as:

$$\mathbf{p_{11}} = \mathbf{p_{10}} - \begin{bmatrix} 0 & 0 & z \end{bmatrix}^\top, \tag{4}$$

and z represents the current MRAV height. Effectively, $\mathbf{p_{11}}$ is placed at a zero altitude below the current MRAV position. The endpoint $\mathbf{p_{12}}$ of the first Bezier curve is $^W\mathbf{p}_T$, and it is also the start point of the second Bezier curve. The control point of the second Bezier curve is defined as:

$$\mathbf{p_{21}} = \mathbf{p}_T + \alpha|\mathbf{p_{12}} - \mathbf{p_{11}}|, \tag{5}$$

where α represents a scaling factor. Increasing the scaling factor affects the curve profile as shown in Fig. 5. Increasing the scaling factor α is useful when dealing with light floating objects that can be affected by the MRAV downwash. A larger scaling factor forces the MRAV to dive longer, making sure it collects the target.

The endpoint of the second Bezier curve is defined as:

$$\mathbf{p_{22}} = \mathbf{p_{21}} + \begin{bmatrix} 0 & 0 & z \end{bmatrix}^\top. \tag{6}$$

Fig. 5 shows three exemplary Bezier curves with all the control points. To generate the trajectory for the MRAV, the spatial relationship between the MRAV base and the net has to be considered when defining control points ($\mathbf{p}$). It is possible to do so as follows:

$$\hat{\mathbf{p}} = (\mathbf{T}_B^N)^{-1}\mathbf{p}. \tag{7}$$

When the path for litter pick-up is planned, it is passed as a set of points to the MPC or TOPP-RA trajectory generation.

V. Experimental validation

A. Experimental setup

1) Hardware: For the experiments, we utilized a quadrotor MRAV equipped with a custom-made net for litter collection. For development and experimental purposes, we used Hexsoon EDU650 as a quadrotor. For the quadrotor control, Pixhawk autopilot was used. For the onboard control, an Intel i7 NUC was used. For perception, we utilized NVIDIA Jetson Xavier AGX edge computing device and

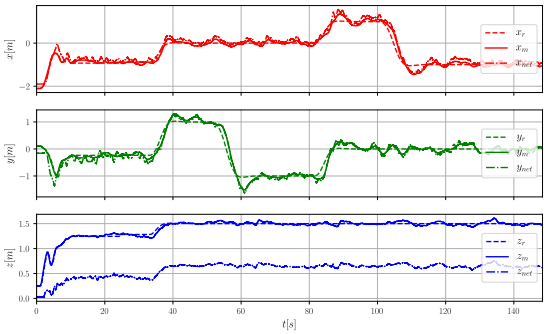

Fig. 6: Position comparison of the MRAV and the net during initial testing phase.

Intel Realsense D415 camera. The net for catching floating objects was created starting from a simple fishing net. The diameter of the net is 0.3 m. The net is connected with four ropes to the MAV arms. The two frontal ropes are shorter and placed on the upper half of the circle frame. Two back ropes are longer than the front ones to distribute the payload of the picked litter with respect to the MRAV center of mass to prevent MRAV from destabilizing. With a too-lightweight net, ropes can get entangled during take-off, which results in the rotation of the net around its z-axis making object pick-up infeasible. An inflatable pool was used to reproduce the water basin indoors. Even though it has reduced dimensions compared to real-world scenarios, the effect of the pool borders may mimic the effects of canals' benches, as the ones shown in Fig. 3 and in the Flyflic proof of concept video referenced in Fig. 1.

2) Software: The software modules are decoupled into the perception and control parts. For the perception part, we developed a custom-made ROS package that serves as OpenMMLab [23] ROS wrapper.[2] It is used as the basis for the aerial manipulator perception and it is used to enable instance segmentation on the vehicle's camera stream. To use an edge-computing device efficiently, we utilized `mmdeploy` toolbox[3] to accelerate and deploy the trained model. For the control part, there is a custom-made ROS stack for on-board MRAV control that consists of the cascaded PID controllers and the MRAV tracker on top of it, as described in [19].

B. Position control

The first experiment was conducted to measure the position of the net with respect to MRAV in motion. The data was recorded while flying in the Optitrack Motion capture system. It can be concluded that the net for litter collection follows MRAV with small horizontal disturbances (noise) caused by the downwash as shown in Fig. 6.

C. Detected centroids position estimation

The estimation of the tracked centroids' positions depends on the object tracker, MRAV movement, and RGBD camera. In order to demonstrate the effect of the MRAV movement on the position estimate in the camera frame, we plotted centroid object position estimates alongside the MRAV movement, as shown and explained in Fig. 7.

D. Trajectory comparisons

To validate the proposed method for the litter pick-up, we executed the planned path based on the target litter detection with MPC and TOPP-RA trajectory planners, showing the differences, advantages, and disadvantages of these two trajectory tracking methods. Trajectories are tested with the zero heading for the sake of simplicity. We assume that MRAV firstly corrects heading towards the detected litter, and then attempts the pick-up procedure.

1) MPC trajectory planning: MPC trajectory planning is mainly used for fast replanning. In this case, we fed the planned path which resulted in the trajectory shown in Fig. 8. As MPC resamples the planned path depending on the constraints and the number of points, when the velocity and acceleration constraints are higher, the tracker smooths the given points which, in this case, result in an almost linear motion. Through experimental procedures, we determined that right combination of the velocity and acceleration constraints ($v < 0.5\ m/s, a < 0.25\ m/s^2$) and distance between MRAV base and the net, can cause downwash to push floating litter in the net during swoop phase as shown in videos referenced in Fig. 1. Note that the effect of the MRAV downwash on the floating object can also be mitigated by increasing the distance between the MRAV and the net. Also, in real-world conditions, such an effect is likely partly counteracted by the water current, too.

2) TOPP-RA trajectory planning: Compared to the MPC trajectory planning, TOPP-RA is slower but unlike MPC, it strictly follows given points, respecting the constraints. In conclusion, TOPP-RA proved more suitable for the current use case. It did not resample the path, which resulted in better reference generation and tracking, as shown in Fig. 9.

VI. Conclusion and Discussion

In this work, we present efforts and practical challenges encountered when developing fully autonomous floating litter pick-up with the MRAV in dynamic and changing laboratory environment. For the perception purposes, we resorted to the instance segmentation Mask-RCNN architecture. We trained it on the custom unofficial TACO-10 dataset, which shows that it is possible to freely train the litter detection model; however, further improvements are necessary. The main perception improvements that are needed for the real-world application are:

1) improved dataset (annotating more images or augmenting dataset with the synthetically generated ones)
2) using a computationally less demanding neural network such as RTMDet [24] for faster inference
3) running perception on-board of the MRAV to reduce the system latency introduced by transporting image from on-board to off-board PC

[2] `https://github.com/larics/mmros_wrapper`
[3] `https://github.com/open-mmlab/mmdeploy`

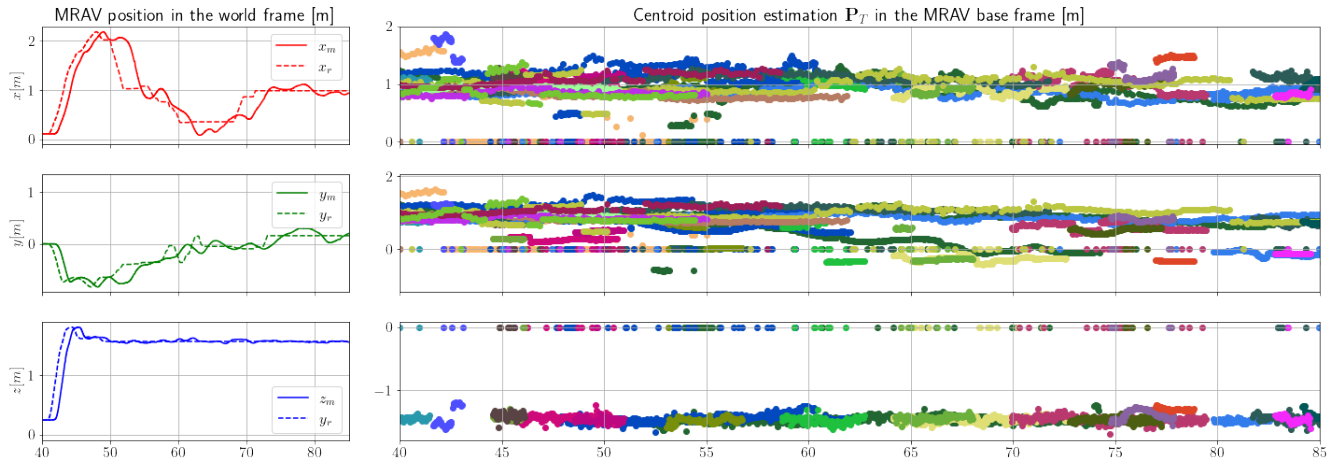

Fig. 7: On the left-hand side subplot, the MRAV motion is shown. On the right-hand side subplot, the tracked centroid position estimates are shown. Different colors represent 'different' tracked objects, due to the tracker logic which is explained in Section IV. Wrong measurements are zeroed and should be neglected. It is possible to notice that consistent position estimation is provided, especially when the MRAV is hovering, as in the last 10 seconds. However, presented object tracker logic is too simple to account for any complex object interaction.

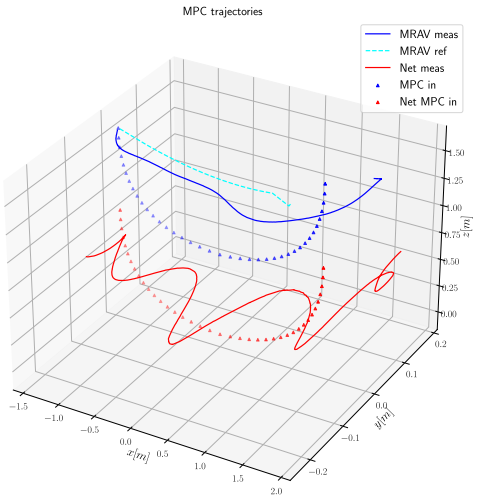

Fig. 8: 3D representation of the commanded points, planned and executed trajectories of MRAV and net with MPC trajectory planner.

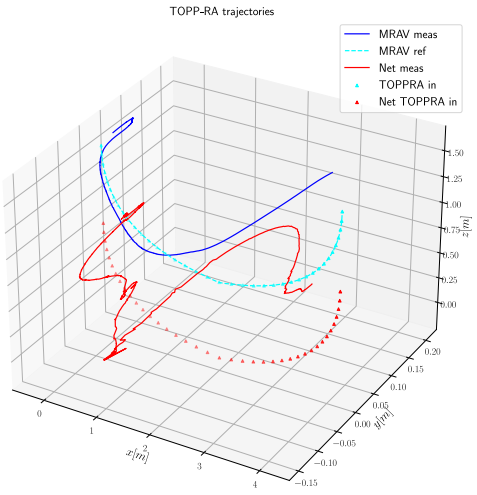

Fig. 9: 3D representation of the commanded points, TOPP-RA planned, and executed trajectories with MRAV and net with TOPP-RA trajectory planner.

4) enabling object position estimation without the use of a depth camera.
5) better centroid tracker with robust object identification/reidentification

With the mentioned improvements, real-time path replanning may be possible, which would result in a much more robust system ready for real-world applications. Besides perception challenges, from the perspective of path planning, using TOPP-RA as a global planner and MPC tracker as a local planner could be useful to mitigate the negative effects of object movement during the pick-up phase. We conclude that it is possible to design an autonomous system for litter pick-up with the MRAV. However, further development and testing are needed, especially in real-world environments. Further work will be oriented to improving the perception capabilities of the MRAV and incorporating both planners for object pick-up. It would also be reasonable to develop a system of MRAVs, where the smaller would be used for litter surveying and mapping, and, after mapping, larger MRAVs could be sent to pick up and retrieve litter. A more complex design of the end-effector embedding mechanisms to actively attract the floating litter inside the net could be considered.

That would help counteract the possible disturbance of the floating litter by the aerial platform.

ACKNOWLEDGEMENT

The authors would like to thank Yaolei Shen, Youssef Aboudorra, Amr Afifi and Lovro Marković for the fruitful discussions. The authors would also like to thank pilots Sander Smith, Jurica Goričanec and Antun Ivanović for the help with experiments. The research work of Filip Zorić is supported by the Croatian Science Foundation under the project "Young Researchers' Career Development Project – Training New Doctoral Students" (DOK-2020-01). This work is also supported by the European Union's Horizon Europe research program Widening participation and spreading excellence, through project Strengthening Research and Innovation Excellence in Autonomous Aerial Systems (AeroSTREAM) - Grant agreement ID: 101071270.

REFERENCES

[1] 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.

[2] O. K. Helinski, C. J. Poor, and J. M. Wolfand, "Ridding our rivers of plastic: A framework for plastic pollution capture device selection," *Marine pollution bulletin*, vol. 165, p. 112095, 2021.

[3] D. Ó. Conchubhair, D. Fitzhenry, A. Lusher, A. L. King, T. van Emmerik, L. Lebreton, C. Ricaurte-Villota, L. Espinosa, and E. O'Rourke, "Joint effort among research infrastructures to quantify the impact of plastic debris in the ocean," *Environmental Research Letters*, vol. 14, no. 6, p. 065001, 2019.

[4] The ocean cleanup. [Online]. Available: https://theoceancleanup.com/rivers/

[5] The great bubble barrier. [Online]. Available: https://thegreatbubblebarrier.com/

[6] River cleaning system. [Online]. Available: https://rivercleaning.com/river-cleaning-system/

[7] Ranmarine. [Online]. Available: https://www.ranmarine.io/

[8] Blue bird electric. [Online]. Available: https://www.bluebird-electric.net/oceanography/Ocean_Plastic_International_Rescue/SeaVax_Ocean_Clean_Up_Robot_Drone_Ship_Sea_Vacuum.html

[9] Plasticsoup. [Online]. Available: https://www.plasticsoupfoundation.org/en/solutions/

[10] A. Kumar, M. Vohra, R. Prakash, and L. Behera, "Towards deep learning assisted autonomous uavs for manipulation tasks in gps-denied environments," pp. 1613–1620, 2020.

[11] R. Jiao, M. Dong, W. Chou, H. Yu, and H. Yu, "Autonomous aerial manipulation using a hexacopter equipped with a robotic arm," pp. 1502–1507, 2018.

[12] P. F. Proença and P. Simões, "TACO: trash annotations in context for litter detection," *CoRR*, vol. abs/2003.06975, 2020. [Online]. Available: https://arxiv.org/abs/2003.06975

[13] M. Kraft, M. Piechocki, B. Ptak, and K. Walas, "Autonomous, onboard vision-based trash and litter detection in low altitude aerial images collected by an unmanned aerial vehicle," *Remote Sensing*, vol. 13, no. 5, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/5/965

[14] J. Hong, M. Fulton, and J. Sattar, "Trashcan: A semantically-segmented dataset towards visual detection of marine debris," *CoRR*, vol. abs/2007.08097, 2020. [Online]. Available: https://arxiv.org/abs/2007.08097

[15] S. Rolf. (2021) Instance segmentation of the multiclass litter: A deep learning model comparison. [Online]. Available: https://liu.diva-portal.org/smash/get/diva2:1546705/FULLTEXT01.pdf

[16] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward image based visual servoing for aerial grasping and perching," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2113–2118.

[17] A. Ivanovic, M. Polic, O. Salah, M. Orsag, and S. Bogdan, "Compliant net for auv retrieval using a uav," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 431–437, 2018, 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318321384

[18] 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)*, 2010, pp. 5420–5425.

[19] L. Markovic, F. Petric, A. Ivanovic, J. Goricanec, M. Car, M. Orsag, and S. Bogdan, "Towards a standardized aerial platform: Icuas'22 firefighting competition," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 3, p. 52, Jul 2023. [Online]. Available: https://doi.org/10.1007/s10846-023-01909-z

[20] H. Pham and Q.-C. Pham, "A new approach to time-optimal path parameterization based on reachability analysis," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 645–659, 2018.

[21] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6753–6760.

[22] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[23] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Mmdetection: Open mmlab detection toolbox and benchmark," *CoRR*, vol. abs/1906.07155, 2019. [Online]. Available: http://arxiv.org/abs/1906.07155

[24] C. Lyu, W. Zhang, H. Huang, Y. Zhou, Y. Wang, Y. Liu, S. Zhang, and K. Chen, "Rtmdet: An empirical study of designing real-time object detectors," 2022.