IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 16, 2019, accepted January 12, 2020, date of publication February 5, 2020, date of current version February 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2971938

Onboard Detection and Localization of Drones Using Depth Maps

ADRIAN CARRIO[1], (Member, IEEE), JESUS TORDESILLAS[3], (Student Member, IEEE), SAI VEMPRALA[2,4], (Student Member, IEEE), SRIKANTH SARIPALLI[2], (Senior Member, IEEE), PASCUAL CAMPOY[1], (Member, IEEE), AND JONATHAN P. HOW[3], (Fellow, IEEE)

[1]Centre for Automation and Robotics, Universidad Politécnica de Madrid, 28006 Madrid, Spain
[2]Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA
[3]Aerospace Controls Laboratory (ACL), MIT, Cambridge, MA 02139, USA
[4]AI and Research Division, Microsoft Corporation, Redmond, WA 98052, USA

Corresponding author: Adrian Carrio (adrian.carrio@upm.es)

This work was supported in part by a grant by Lockheed Martin and by the Spanish Ministry of Economy and Competitiveness (Complex Coordinated Inspection and Security missions by UAVs in cooperation with UGVs) under Project RTI2018-100847-B-C21.

ABSTRACT Obstacle avoidance is a key feature for safe drone navigation. While solutions are already commercially available for static obstacle avoidance, systems enabling avoidance of dynamic objects, such as drones, are much harder to develop due to the efficient perception, planning and control capabilities required, particularly in small drones with constrained takeoff weights. For reasonable performance, obstacle detection systems should be capable of running in real-time, with sufficient field-of-view (FOV) and detection range, and ideally providing relative position estimates of potential obstacles. In this work, we achieve all of these requirements by proposing a novel strategy to perform onboard drone detection and localization using depth maps. We integrate it on a small quadrotor, thoroughly evaluate its performance through several flight experiments, and demonstrate its capability to simultaneously detect and localize drones of different sizes and shapes. In particular, our stereo-based approach runs onboard a small drone at 16 Hz, detecting drones at a maximum distance of 8 meters, with a maximum error of 10% of the distance and at relative speeds up to 2.3 m/s. The approach is directly applicable to other 3D sensing technologies with higher range and accuracy, such as 3D LIDAR.

INDEX TERMS Drone, detection, collision avoidance, depth map.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are a popular choice for robotic applications given their advantages such as small size, agility and ability to navigate through remote or cluttered environments. Collision avoidance is a key capability for autonomous navigation, which typically involves four stages [1]: detection (obstacle perception), decision (whether the detected drone is an actual threat and how to avoid it), action (the actual collision maneuver execution) and resolution (which determines whether the navigation can be safely resumed).

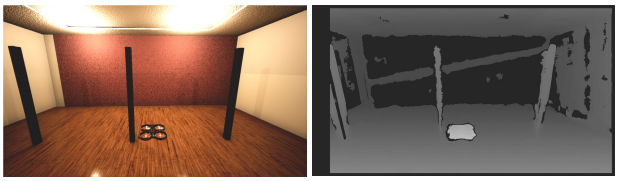
The detection stage typically involves the use of sensing technologies to determine the presence of obstacles and gather information which can be useful for preventing a potential collision, such as the relative position and/or the speed of the obstacle.

Several sensing technologies have been proposed for drone detection, such as radar [2] and other RF-based sensors [3], acoustic sensors [4] and LIDAR [5]. Hybrid approaches have also been researched [6]. However, some of these technologies have limitations for being integrated onboard small drones due to various factors such as: high power consumption, weight and/or size requirements, and cost. As opposed to the aforementioned technologies, electro-optical sensors provide a small, passive, low-cost and low-weight solution for drone detection. These sensors are therefore particularly suitable for small drones.

In the literature, visible-spectrum imaging sensors have been widely proposed for flying object detection. [7]–[13]. Thermal infrared imaging has also been considered for the same purpose [14], [15]. The most relevant difference

The associate editor coordinating the review of this manuscript and approving it for publication was Zhen Li.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/



(a) (b)

FIGURE 1. Simulated indoor scene showing a flying drone (left) and corresponding depth map (right). Any drone or flying object in a depth map generates a blob which contrasts with the background. This happens as a flying object is typically isolated in 3D space, with no contact with other objects having the same depth. In other words, a flying object typically generates a discontinuity in the depth map, which can be used as a distinct visual feature for drone detection.

between both imaging technologies is that, while the sensors in thermal infrared cameras typically have lower spatial resolutions, they have the advantage that they can be operated at night.

Image-based detection approaches typically rely either on background subtraction methods [16], or on the extraction of visual features, either manually, using morphological operations to extract background contrast features [17] or automatically using deep learning methods [18], [19]. Rozantsev *et al*. [7] present a comparison between the performance of various of these methods. The aforementioned detection techniques rely on the assumption that there is enough contrast in the visible spectrum between the drone and the background. The use of depth maps, which can be obtained from different sensors (stereo cameras, RGB-D sensors or 3D LIDAR), relies mostly on the geometry of the scene and not so much on its visual appearance.

3D point clouds have been recently proposed for having drones autonomously execute obstacle avoidance maneuvers using an RGB-D camera [20], but focusing on the detection of static obstacles only. An alternative representation for point clouds are depth maps, which have been investigated for general object detection [21] and human detection [22], providing better detection performance compared to RGB image-based methods. In the context of drone detection, a key concept that explains the usefulness of depth maps is that any drone or flying object in a depth map appears as a blob which contrasts with the background. The reason for this is that a flying object is typically isolated in 3D space, with no contact with other objects having the same depth. In other words, a flying object typically generates a discontinuity in the depth map, which can be used as a distinct visual feature for drone detection. This concept is depicted in Fig. 1. An additional advantage of detection using depth maps is that, while data from monocular image sensors can generally provide relative altitude and azimuth of the object only, depth maps can provide full 3D relative localization of the objects. This is particularly useful in the case of obstacle avoidance for drones, since the 3D position of the drone can be exploited to perform effective collision-free path planning.

In this paper, we build on our previous work for drone detection using depth maps obtained from a stereo camera [23] and introduce several improvements. Instead of training the detector directly with ground truth depth maps generated with Microsoft AirSim, we now perform stereo matching of simulated RGB image pairs to generate more realistic depth maps for training. Additionally, in terms of implementation, we propose the use of a shared memory block to synchronize image data between the image acquisition and the inference processes, implemented in C++ and C/CUDA, respectively. This is an efficient alternative to using code wrappers or middlewares, which typically introduce processing overheads. Additionally, we fully integrate the detection system onboard a small drone, allowing for drone detection during navigation. To the best of the authors' knowledge, this is the first time that depth maps are used onboard a drone for the detection of other drones. The proposed detection method has been evaluated in a series of real flight experiments with different collision scenarios.

The remainder of this paper is as follows. Firstly, in Section II, we present our framework for drone detection and localization. Secondly, in Section III, the proposed hardware setup is presented. Thirdly, in Section IV, we describe the evaluation methodology followed. In Section V, we present the results of the experiments and finally, in Section VI, we present the conclusions and future work.

II. PROPOSED DETECTION AND LOCALIZATION FRAMEWORK

The objective of this work is to develop a system which can operate onboard a small drone, performing drone detection and 3D localization with respect to the drone carrying the detection system. In order to avoid confusion, we will onwards refer to the drone that carries the camera as the *detector drone*, and the drones which we aim to detect and localize with the proposed system as *target drones*. The purpose of gathering data about the target drone (the degree of certainty about its presence and its relative position) can be exploited for collision-free path planning in later stages. This section will focus on the proposed software system, including algorithmic and implementation details.

A. PROCESSING PIPELINE

The software system has been designed to efficiently perform various tasks: depth image acquisition, detection using a state-of-the-art object detection model and 3D localization of the target drone. The system is depicted in Fig 2.

Synthetic RGB image pairs of a flying drone, obtained from a simulated stereo camera, and the segmentation maps corresponding to the images from the left camera, are generated using Microsoft AirSim in order to build a dataset for training the object detection model. The segmentation maps are RGB images in which drone pixels have a characteristic color which facilitates segmentation. The synthetic RGB image pairs are processed using a stereo matching algorithm in order to obtain depth maps, and the labels for each depth map are obtained straightforwardly by extracting the minimum area rectangles enclosing the drone blobs in the segmentation maps. The resulting dataset is used to train a modified version of a state-of-the-art object detection model.

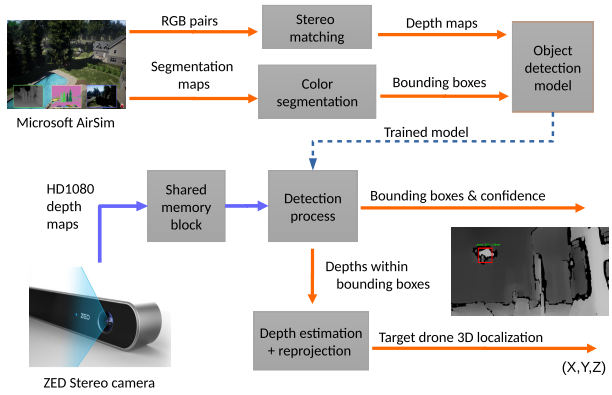


FIGURE 2. Detection system overview. Synthetic RGB image pairs of a flying drone and segmentation maps are generated using Microsoft AirSim in order to train the object detection model. During operation, depth maps are acquired from the Stereolabs ZED Stereo camera onboard and fed to the trained model. The model outputs bounding boxes and confidence values for detected drones. The minimum depth in each of the bounding boxes is extracted and used to obtain a robust estimate of the 3D position of the detected drone.

During operation, depth maps are acquired from a Stereolabs ZED Stereo camera onboard and copied to a shared memory block from which they are fed to the trained model. This allows to share the data directly between both processes (image acquisition and inference) in an efficient way, without the need of a wrapper interface which could affect overall performance. Once the model is trained, it outputs bounding boxes and confidence values for detected drones. In order to filter possible outliers corresponding to background points that fall within the bounding box, the centroid of all 3D points whose depth is within 500mm of the minimum depth in the bounding box is used as a robust estimate of the relative 3D position of the detected drone.

B. OBJECT DETECTION MODEL

The aforementioned dataset is used to train a modified version of the YOLOv2 object detection model [24], implemented in C and CUDA. This model is the continuation of the original YOLO model [25], a detection architecture named after the fact that a single forward pass through its convolutional neural network (CNN) is enough for object detection. This architecture has gained a lot of popularity both in the industry and in academia because of its high accuracy and speed. Although an extensive description falls out of the scope of this article, the most relevant details and particularly, the modifications to adapt the model for drone detection in depth maps will be presented next.

In terms of architecture, we rely on the model depicted in Fig. 3, with an input image size of 672×672, featuring 9 convolutional layers with batch normalization and leaky rectified linear units, plus 6 intermediate max-pooling layers.

Our model incorporates the following improvements from YOLOv2:

- **3-stage training**: the model is trained initially as a classifier using 224×224 pictures from ImageNet 1000 class classification dataset [26]. Secondly, the classifier is retuned with 448×448 pictures using much fewer epochs. Finally, the fully connected layers are removed and a convolutional layer is added to obtain the definitive architecture, which is retrained end-to-end for object detection. This procedure provides the classifier with greater spatial resolution, which can be helpful for detecting small or distant drones.
- **Bounding box priors**: if the model predicts bounding boxes, the initial training steps are susceptible to unstable gradients due to the fact that the predictions might work very well for some objects and very bad for others, causing steep gradient changes. However, the strategy followed here consists of predicting offsets to bounding box priors, also known as anchors, as opposed to predicting bounding boxes. When the training is performed by using a limited number of diverse guesses that are common for real-life objects, each prediction focuses on a specific shape, leading to a much more stable training. This makes even more sense when training the detector for a single object, a drone in our case. Following the methodology proposed in the [24], we run a k-means clustering algorithm, using intersection over union as distance metric, in order to find the 5 bounding boxes that have the best coverage for the drone images in our training data.
- **Fine-grained features**: in order to improve the detection of smaller objects (i.e. small and/or distant drones), feature maps are rearranged by concatenating layers with different feature map sizes during inference. Adjacent high resolution and low resolution features in the last layers are stacked together into different channels, allowing the detector to have accesses to both types of features.
- **Multi-scale training and inference**: in order to improve the performance across different input image sizes, the model is randomly scaled every 10 batches during training, forcing the network to generalize across a variety of input dimensions. Since the model only uses convolutional and pooling layers, this is a straightforward process, which also provides the model with a lot of flexibility to balance accuracy and speed.

The last layer corresponds to a 1×1 convolution, leading to a final tensor size of $21\times21\times30$. The first two values come from the division of the image in a 21×21 grid. The value of 30 comes from the fact that each cell in the grid makes a prediction using each prior box, in our case 5 predictions, each with 6 values: 4 offset values for the prior box, a value of confidence score (objectness) and a value for the class probability, in our case, the probability that the object is a drone given that there is an object in the bounding box. All the proposed predictions are filtered using non-maxima suppression and threshed by their confidence values in order to remove false positives.

Most YOLO implementations available perform multi-frame detection averaging in order to make video predictions visually smoother. However, this produces mismatches between the input depth maps and their corresponding

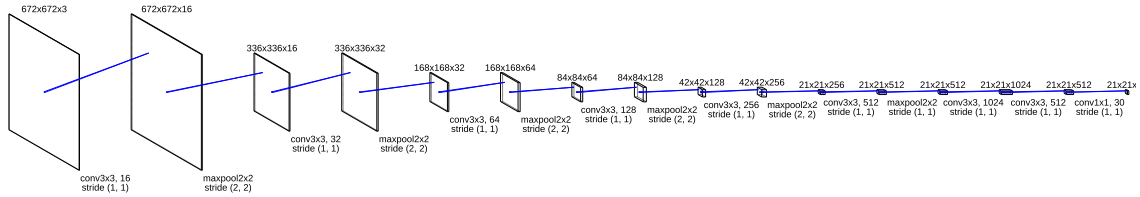


FIGURE 3. Proposed CNN architecture for drone detection. The model has been customized taking into account the hardware constrains onboard a small UAV to achieve real-time performance.

bounding box predictions, which become specially noticeable when detected objects move fast. For this reason, we remove this feature in our implementation to ensure correct mapping of the predicted boxes to their corresponding depth maps.

C. SYNTHETIC DATASETS AND TRAINING METHODOLOGY

In order to effectively generate large datasets for training the model previously discussed, we opted to use synthetic images. We utilized the high-fidelity UAV simulator Microsoft AirSim [27] for this purpose. Microsoft AirSim runs as a plugin for Unreal Engine, which is a popular game engine routinely used for AAA videogames. Unreal Engine makes it possible to render photorealistic environments, while providing features such as high-resolution textures, photometric lighting, dynamic soft shadows and screenspace reflections. Using these features, environments can be modeled in Unreal Engine to be very close to real life scenes, which makes it a good choice for computer vision related applications. On top of Unreal Engine's base features, AirSim provides the capabilities to create and instantiate multiple drone models, simulate their physics, allow for basic flight control and also create and access onboard camera streams.

In order to generate a synthetic dataset, we used AirSim and Unreal Engine with a few enhancements. In Unreal Engine, we created a custom outdoor urban environment. We chose the quadrotor model resembling a Parrot AR.Drone provided by AirSim and equipped it with two cameras so as to simulate a stereo vision system. The parameters for the cameras, such as baseline between cameras and the field of view were configured to match the specifications of the ZED camera, as that was our test platform of choice.

Within the environment, we simulated two drones and observed one drone through the stereo camera of the other. As Unreal Engine keeps track of all object meshes and materials present in the scene, it can easily generate ground truth segmentation images, which we used to obtain pixel-wise segmentation images of the drone by removing all other objects from the segmentation images. At every instant, both left and right views of the onboard RGB camera were recorded, along with a segmentation image indicating the pixels corresponding to the target drone, which was used as ground truth. In this way, we recorded images of the target drone from various distances, viewpoints and angles in 3D space, attempting to simulate the observation of a detector drone hovering as well as in motion. Our dataset contains images where the detector drone is in motion as well as while it is static whereas the target drone always moves. In our previous work [23], we created a dataset that contained images of multiple resolutions and simpler environments as we experimented with various parameters to create the best configuration. For this work, we have generated images with a single resolution, 1920×1080 pixels and we chose to record left and right RGB images to perform stereo matching separately, instead of using the default depth maps provided by AirSim. While AirSim provides depth maps along with RGB and segmentation images, these depth maps do not have a realistic appearance, as they are obtained directly from rendering the scene meshes, without any imperfections. In order to closely match real life stereo matching and allow for correspondence errors, we run the left and right RGB images separately through a block matching algorithm to compute correspondences. We use the OpenCV implementation of the stereo block matching algorithm to generate disparity images, which are filtered using a weighted least squares (WLS) filter and then used for training. Bounding boxes for the target drones were easily extracted from the corresponding segmentation images by finding the minimum area rectangles enclosing the target drones.

The synthetic dataset used for training the detection model used in the experiments contained 470 depth maps, from which 80% of the images were used for training and the rest for validation. As it will be mentioned later on, data augmentation provides an actual number of training images which is several orders of magnitude higher. The model was trained for 380k iterations, following an early stopping strategy.

D. TARGET DRONE LOCALIZATION

The more information available about the obstacle (e.g. speed, relative position, detection uncertainty), the higher the likelihood of successful planning and executing collision avoidance maneuvers [28]. This makes the proposed solution advantageous with respect to monocular detection methods, which typically provide information about the obstacle only in the image plane. In order to estimate the 3D relative position of the drone, first its depth is estimated. Given a ROI with certainty of containing a drone above a given threshold, the minimum depth in the ROI, $\min(Z_{i,j})$, is computed. Given a depth margin for the drone, δ, its depth estimation will be given by Eq. 1.

$$Z_{est} = \overline{Z_{i,j}} \mid Z_{i,j} \in [\min(Z_{i,j}), \min(Z_{i,j}) + \delta] \tag{1}$$

(a) (b) (c)

FIGURE 4. Sample images from the dataset. In (a), the RGB image from the detector drone's perspective is shown for reference, where it views a 'target' drone, a quadrotor. The corresponding depth map is shown in (b), and (c) shows the segmentation image that isolates only the target drone.

FIGURE 5. Detector drone platform used in the experiments. It is equipped with a ZED camera, a Jetson TX2 (used for perception) and a Snapdragon Flight board (used for control).

If u, v are the pixel coordinates of the pixel with minimum depth in the ROI, X and Y coordinates are estimated through 3D point reprojection using intrinsic camera parameters (f_x, f_y, c_x, c_y) and Z_{est}, assuming a pinhole camera model.

$$X_{est} = Z_{est}\frac{u - c_x}{f_x}$$
$$Y_{est} = Z_{est}\frac{v - c_y}{f_y} \quad (2)$$

III. HARDWARE SETUP

A. AERIAL PLATFORMS

The drone shown in Fig. 5 is the one that has been used as the detector drone in the indoor experiments. It is equipped with a Jetson TX2, a ZED camera and a Snapdragon Flight, with a takeoff weight under one kilogram.

In order to show the generalization capability of our detection algorithm, and its ability to detect and localize multiple drones, three different target drones have been used, shown in Fig. 6. One of them is the Parrot AR.Drone, while the other two ones are custom ones: one medium UAV, and a small UAV. The sizes of these drones (with the battery included) is shown in Table 1.

The small UAV is used to test the generalization of our algorithm when the size is much smaller compared to the AR Drone, while the medium UAV is used to test the algorithm when the shape (closer to a spherical one) is different from the one of the AR Drone. The hardware design and control of the small and medium UAVs is very similar to the drone that carries the ZED camera, with the only difference being that they do not carry a Jetson TX2 and a ZED camera.

TABLE 1. Dimensions of the target drones used in the experiments.

	Width (mm)	Length (mm)	Height (mm)
Parrot AR.Drone	517	517	95
Small UAV	350	285	115
Medium UAV	430	430	205

FIGURE 6. Three drones were used as target drones in the experiments in order to evaluate the generalization capability of the model to different drone modalities.

B. DEPTH SENSOR

A ZED Stereolabs camera has been used for the experiments. This stereo camera based on two 4MP sensors features a 120 mm baseline. Although it is capable of computing 4416×1242 depth maps at 15Hz, it allows for different sensing modes and resolutions to balance computational speed and depth map quality.

C. ONBOARD PROCESSING UNIT

The perception and detection algorithm runs onboard on a Jetson TX2, while a Snapdragon Flight is used for state estimation and control. Both devices run independently, with no connection between them. The state estimation is performed by fusing IMU measurements with an external motion capture system. A cascade control architecture is used to generate the commands to the motors: the outer loop controller receives

the desired trajectory and the estimated position and velocity, and sends the desired orientation and angular rates to the inner loop controller. This controller compares them with the estimated attitudes and rates, and generates the commands that are sent to the motors. More details related to the state estimation and the control of the drone can be found in [29].

A joystick controlled manually by an operator was used to generate the desired trajectories for the experiments.

An external motion capture system has also been used for obtaining the ground truth relative position between the camera and the target drone. This position was recorded using a ground station computer. All three computing devices: Jetson TX2, Snapdragon Flight and the ground station computer had synchronized clocks, allowing to compare position estimations based on detections, computed on the TX2, with position estimations based on the Vicon system and recorded on the ground station computer.

IV. EVALUATION METHODOLOGY

A. EVALUATION METRICS

As it has been mentioned earlier, the proposed system performs first a drone detection in the image plane and then uses the information from this detection to perform localization of the target drone in the scene. For this reason we propose a double quantitative evaluation: one for target drone detection and one for 3D localization of the target drone.

1) DRONE DETECTION METRICS

Accuracy of drone detection is mainly assessed by means of the average intersection over union, a standard metric for object detection. This metric evaluates the average area of overlap between a predicted and a ground truth bounding box. The area of overlap is measured using Eq. 3, where B_{gt} is a ground truth bounding box and B_p represents a predicted bounding box. The average intersection over union is computed considering for each ground truth box the predicted box with the highest area of overlap and an objectness value above 50%.

$$a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{3}$$

Additionally, classification metrics have been computed for different detection threshold values. Specifically, precision p and recall r, defined by Eqns. 4, 5, respectively have been used. In these equations, TP represents the number of true positives, FP represents the number of false positives and FN represents the number of false negatives.

$$p = \frac{TP}{TP + FP} \tag{4}$$

$$r = \frac{TP}{TP + FN} \tag{5}$$

2) TARGET DRONE LOCALIZATION METRICS

The accuracy of the target drone localization has been measured in indoor experiments using a Vicon motion capture system. In this way, the timestamped, relative ground truth positions reported by the system have been compared with those reported by the onboard system and RMS errors have been computed. Additionally, the largest depth reported by the system will be considered as an indicator of the detection range of the system. This range depends mostly on the sensor technology used and could be enlarged with other sensing technologies, such as 3D LIDAR.

3) COMPUTATION SPEED

Computation speed will also be assessed, as it is a critical metric for algorithms running in embedded systems and particularly onboard UAVs. Eq. 6 describes the total time it takes the system to detect and localize a drone, which can be measured as the sum of the durations of each of the processes involved: Δt_{acq} (acquisition), Δt_{dm} (depth map generation), Δt_i (inference) and Δt_{repr} (reprojection).

$$\Delta t_{total} = \Delta t_{acq} + \Delta t_{dm} + \Delta t_i + \Delta t_{repr} \tag{6}$$

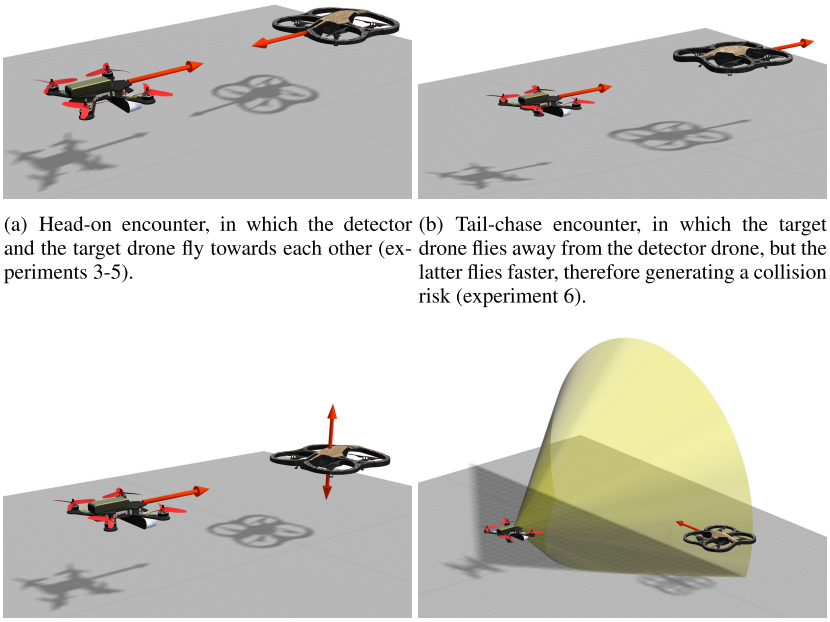
While the duration of the acquisition process depends mainly on the sensor and the CPU resources available, the depth map generation and the detection model inference run on GPU. Both processes compete for GPU resources, but since the depth estimation algorithm in the ZED camera is pre-compiled, it is not possible to synchronize the processes and the only way to balance the GPU resources is by choosing an adequate depth sensing mode. Higher quality depth sensing modes will produce depth maps with higher resolution at the cost of a slower detection model inference and viceversa. As for the reprojection process, it runs fully on CPU.

B. EXPERIMENTS

Eight flight experiments were run in order to assess the reliability and robustness of the system quantitatively in different scenarios. An outdoor experiment (experiment 1) was run in an open field in College Station, Texas. This experiment was conceived to evaluate the impact of outdoor illumination conditions and distant image backgrounds which may affect the appearance of the generated depth maps. The remaining experiments, 2 to 8, were run indoors, in the Aerospace Controls Laboratory at MIT.

In experiments 1 and 2, the target drone detection stage was quantitatively evaluated. For experiment 1, a 3D Robotics Solo quadrotor acted as target drone, while the detecting camera was moved manually for simplicity. In experiment 2, both the detector and the target drone, a Parrot AR.Drone, were flying. In both experiments, the depth maps generated onboard were stored in order to be manually labelled for the posterior evaluation.

Finally, target drone localization has been evaluated in experiments 3 to 8 by capturing ground truth data from both detector and target drones using a motion capture system. In these experiments, summarized in Fig. 7, a Parrot AR.Drone was used as target drone. Experiments 3 to 5 correspond to head-on encounters, in which the detector and the target drone fly towards each other. The purpose of these experiments is to evaluate the performance of the



(a) Head-on encounter, in which the detector and the target drone fly towards each other (experiments 3-5).

(b) Tail-chase encounter, in which the target drone flies away from the detector drone, but the latter flies faster, therefore generating a collision risk (experiment 6).

(c) The detector drone moves forward while the target drone moves vertically (experiment 7).

(d) The target drone crosses the FOV limits of the camera onboard the detector drone (experiment 8).

FIGURE 7. Graphical explanation of the performed experiments.

system when drones in collision course approach each other at different speeds. Three relative speeds were chosen for the evaluation: low (max relative speed of 1.1 m/s), medium (max relative speed of 1.7 m/s) and high (max relative speed of 2.3 m/s). Experiment 6 corresponds to another common potential collision scenario: a tail-chase encounter. Here, the target drone flies away from the detector drone, but the latter flies faster, therefore generating a collision risk.

In experiments 3 to 6, both drones, detector and target, fly at relatively similar altitudes. In experiment 7, we allow for altitude variations of the target drone while keeping the detector drone at a constant altitude to evaluate the potential impact of non-zero relative vertical speed between the drones in collision course. Finally, in experiment 8, the target drone crosses the FOV limits of the stereo camera onboard the detector drone. This allows to assess the actual detection FOV of the system.

Besides these experiments for quantitatively assessing the performance of the system, more flights were perfomed for qualitative evaluation using all three different drones shown in Fig. 6. In these additional flight experiments the system performs the onboard detection of up to three three target drones simultaneously.

V. RESULTS

As mentioned in Section IV, two types of quantitative results are presented: drone detection and target drone localization results. Additionally, a video showing some qualitative results from the indoor environment with up to three flying target drones can be found online [1]. While the video shows how the system is capable of detecting multiple drones, localizing more than one drone simultaneously requires a visual tracker which can deal with the data association problem, which is left for future work.

A. DRONE DETECTION PERFORMANCE

The trained model achieved an IoU of 84.86% and a recall rate of 100%. The results are good despite the limited amount of training images as we benefit from YoLo's implementation which incorporates data augmentation, enabling the generation of an unlimited number of samples through variation of saturation and exposure, and random cropping and resizing.

The performance of the model for image classification is summarized in Fig. 14. From this curve, the resulting mAPs were 0.7487 and 0.6564, for experiments 1 and 2, respectively. The gap in precision can be explained by the different complexity of the environments: while the indoor environment is cluttered with objects which may be considered false positives, the outdoor experiment is run in an open field with the camera facing the sky and no ground objects in the background for most of the frames.

These results are consistent with the performance of this model in the much more extensive COCO dataset (20 classes, 40k labelled images), where it achieves 57.1 mAP. The model is able to detect drones in depth images, with no relevant impact of the motion blur or the changes in illumination in the

[1] https://vimeo.com/277984275

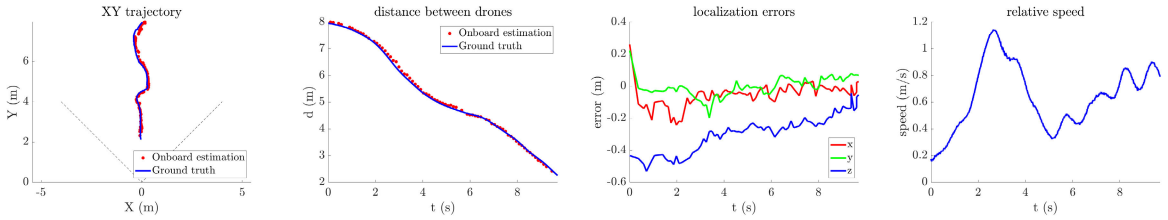


FIGURE 8. Results for experiment 3, a head-on encounter at low speed. Dashed lines indicate the FOV of the detector drone.

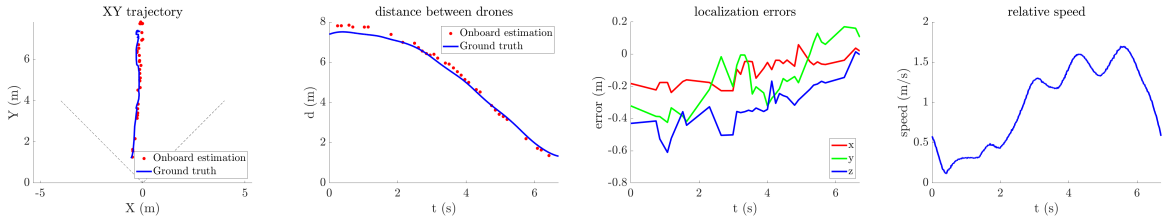


FIGURE 9. Results for experiment 4, a head-on encounter at medium speed. Dashed lines indicate the FOV of the detector drone.

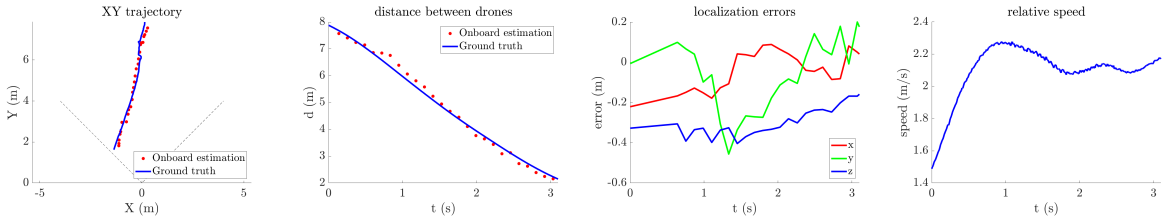


FIGURE 10. Results for experiment 5, a head-on encounter at high speed. Dashed lines indicate the FOV of the detector drone.

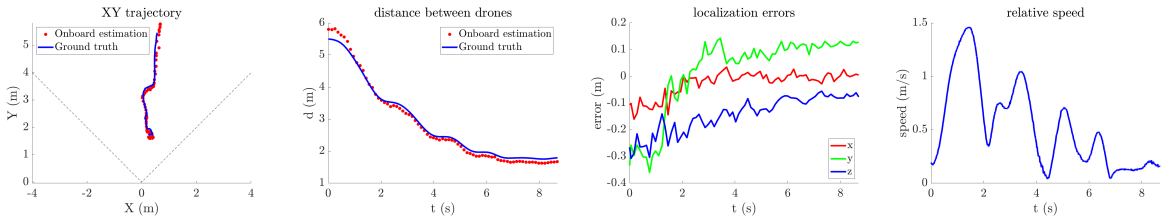


FIGURE 11. Results for experiment 6, a tail-chase encounter. Dashed lines indicate the FOV of the detector drone.

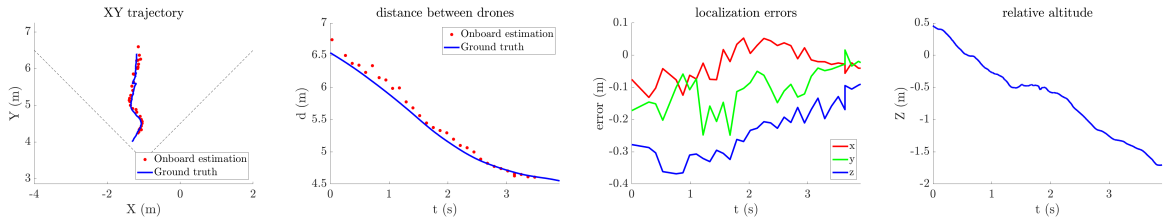


FIGURE 12. Results for experiment 7, a head-on encounter with relative altitude changes. Dashed lines indicate the FOV of the detector drone.

performance of the model. Furthermore, the model is able to generalize correctly to different drone modalities, even with a low number of training samples obtained from a single type of drone model.

B. TARGET DRONE LOCALIZATION ACCURACY

Localization results show the implemented system is able to detect and track drones robustly, even at high speeds, as shown in Figs. 8 to 13. In experiments 3 to 5, the target

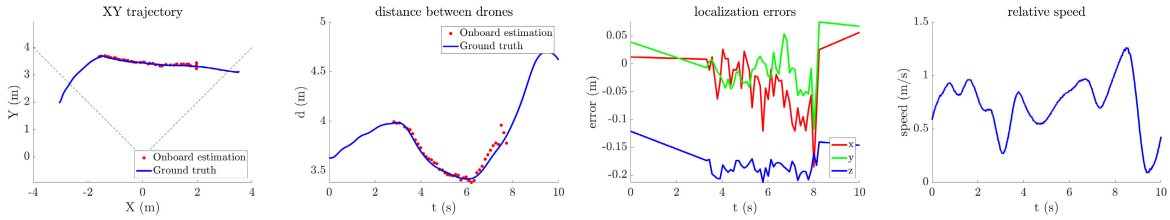


FIGURE 13. Results for experiment 8, an encounter while flying in and out of the field of view. Dashed lines indicate the FOV of the detector drone.

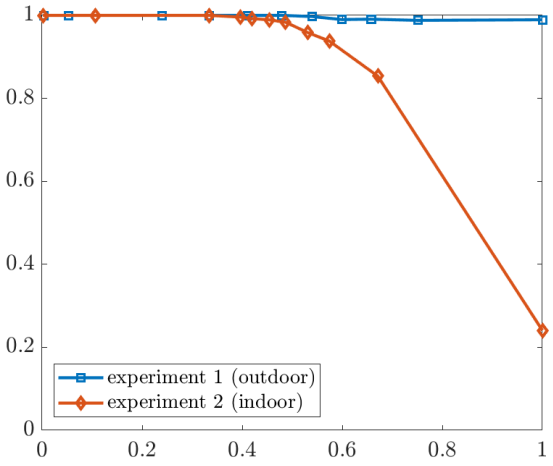


FIGURE 14. Precision-recall curve of the drone detection model for the indoor and outdoor experiments.

drone is localized at distances of up to 8 meters with a maximum error below 10% of the distance. Localization errors are consistent over all the experiments and up to 2.3 m/s, we find the localization accuracy to be consistent over all experiments and up to velocities of 2.3 m/s. We also find the localization to be independent of the relative speeds and the angle from which the target was captured (for example, viewing it from the front versus the back). Similarly, changes in altitude do not produce noticeable effects and the target drone trajectory is tracked accurately.

With respect to experiment 8, Fig. 13 shows how the effective FOV of the system, i.e. the FOV at which detections happen, is smaller (60 degrees) than the camera FOV (90 degrees). This might have been caused by the fact that drones in the training labels were fully contained within the image.

C. COMPUTATION SPEED

As mentioned earlier, the different processes involved in the detection and localization of drones share the CPU and GPU resources onboard. Choosing a resolution of 1080p, both the image acquisition and depth map computation from the ZED stereo camera can simultaneously reach 30 Hz. The model inference, when fed with pre-stored depth frames, runs at 20 Hz. However, as mentioned earlier, acquiring depth maps and running the model in parallel reduces the performance

TABLE 2. Execution times.

	Acquisition and depth map generation	Drone detection	3D reprojection
Time [ms]	45.2 ± 8.3	9.4 ± 1.4	7.1 ± 0.6

of each algorithm individually. Best results were obtained when choosing to acquire 1920×1080 pixel depth maps, with the sensing mode providing the highest depth map quality. With this configuration, the system acquires frames, computes depth maps, detects drones and localizes them at 16 Hz. Table 2 indicates the average execution time and its standard deviation over 600 frames for each of the sub-processes.

VI. CONCLUSION AND FUTURE WORK

Obstacle avoidance for drones is currently an active field of research as it is a desired capability for safe drone navigation. While many commercial drones already incorporate obstacle avoidance systems, they are designed mainly for avoiding structures and not specifically for avoiding dynamic obstacles, such as drones.

The integration of such capabilities in small drones with constrained takeoff weights is extremely challenging, due to the efficient perception, planning and control capabilities required. Perception-wise, obstacle detection systems should be capable of running in real-time, with sufficient field-of-view and detection range, and ideally being capable of providing relative position estimates of potential obstacles.

In this work, we provide a high-performance solution for small drones based on our previous approach for drone detection [23]. Here, we propose a novel strategy to perform drone detection and localization using depth maps which has been adapted to run at 16 Hz onboard a small drone using a stereoscopic camera. Experiments successfully demonstrate that the system can simultaneously detect and localize drones of different sizes and shapes at a maximum distance of 8 meters, with a maximum error of 10% of the distance and at relative speeds up to 2.3 m/s. This is a remarkable achievement, given the payload limitations of the small platform used in the experiments.

A relevant aspect of this research is the ability of the object detection model to generalize from simulated to real images. While the results of the proposed solution indicate this generalization is possible, further research will be conducted to provide exhaustive information about how each of

the simulation variables (i.e. lighting, background, etc.) affect the detection performance of the model.

Other future works include the integration of our system with planning and navigation algorithms, such as reciprocal velocity obstacles (RVO) or dynamic potential fields. Also the use of filtering and prediction techniques will be considered, to both minimize the effect of bad detections and to provide smoothed velocities of the obstacles, which are generally required by planning algorithms. A strategy combining detection and tracking will also be studied to provide continuity of the detections and simultaneous localization of multiple drones.

ACKNOWLEDGMENT

The authors would like to thank MISTI-Spain and A. Goldstein in particular for the financial support received through the project entitled "Drone Autonomy". Special thanks to B. Lopez (ACL-MIT) and A. Ripoll (TU Delft) for their contributions in the early stages of this work. They would also like to thank J. Yuan (MIT) for his help in the experiments and P. Tordesillas for his help with the figures in the article.

REFERENCES

[1] T. Hutchings, S. Jeffryes, and S. Farmer, "Architecting UAV sense & avoid systems," in *Proc. Inst. Eng. Technol. Conf. Auton. Syst.*, Nov. 2007, pp. 1–8.

[2] J. Drozdowicz, M. Wielgo, P. Samczynski, K. Kulpa, J. Krzonkalla, M. Mordzonek, M. Bryl, and Z. Jakielaszek, "35 GHz FMCW drone detection system," in *Proc. 17th Int. Radar Symp. (IRS)*, May 2016, pp. 1–4.

[3] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, "Investigating Cost-effective RF-based Detection of Drones," in *Proc. 2nd Workshop Micro Aerial Vehicle Netw., Syst., Appl. Civilian Use (DroNet)*, 2016, pp. 17–22, doi: 10.1145/2935620.2935632.

[4] J. Mezei, V. Fiaska, and A. Molnar, "Drone sound detection," in *Proc. 16th IEEE Int. Symp. Comput. Intell. Informat. (CINTI)*, Nov. 2015, pp. 333–338.

[5] M. U. De Haag, C. G. Bartone, and M. S. Braasch, "Flight-test evaluation of small form-factor LiDAR and radar sensors for sUAS detect-and-avoid applications," in *Proc. IEEE/AIAA 35th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2016, pp. 1–11.

[6] F. Christnacher, S. Hengy, M. Laurenzis, A. Matwyschuk, P. Naz, S. Schertzer, and G. Schmitt, "Optical and acoustical UAV detection," *Proc. SPIE*, vol. 9988, 2016, Art. no. 99880B.

[7] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4128–4136.

[8] T. Zsedrovits, A. Zarandy, B. Vanek, T. Peni, J. Bokor, and T. Roska, "Collision avoidance for UAV using visual detection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 2173–2176.

[9] Y. Wu, Y. Sui, and G. Wang, "Vision-based real-time aerial object localization and tracking for UAV sensing system," *IEEE Access*, vol. 5, pp. 23969–23978, 2017.

[10] F. Gökçe, G. Üçoluk, E. Şahin, and S. Kalkan, "Vision-based detection and distance estimation of micro unmanned aerial vehicles," *Sensors*, vol. 15, no. 9, pp. 23805–23846, Sep. 2015. [Online]. Available: http://www.mdpi.com/1424-8220/15/9/23805

[11] A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. Beyerer, "Deep cross-domain flying object classification for robust UAV detection," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.

[12] E. Unlu, E. Zenou, and N. Riviere, "Ordered minimum distance bag-of-words approach for aerial object identification," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.

[13] E. Unlu, E. Zenou, and N. Riviere, "Using shape descriptors for UAV detection," *Electron. Imaging*, vol. 2018, no. 9, pp. 128-1–128-5, Jan. 2018.

[14] P. Andraši, T. Radišić, M. Muštra, and J. Ivošević, "Night-time detection of uavs using thermal infrared camera," *Transp. Res. Procedia*, vol. 28, pp. 183–190, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352146517311043

[15] A. Carrio, Y. Lin, S. Saripalli, and P. Campoy, "Obstacle detection system for small UAVs using ADS-B and thermal imaging," *J. Intell. Robot. Syst.*, vol. 88, nos. 2–4, pp. 583–595, Dec. 2017, doi: 10.1007/s10846-017-0529-2.

[16] S. R. Ganti and Y. Kim, "Implementation of detection and tracking mechanism for small UAS," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2016, pp. 1254–1260.

[17] J. Lai, L. Mejias, and J. J. Ford, "Airborne vision-based collision-detection system," *J. Field Robot.*, vol. 28, no. 2, pp. 137–157, Mar. 2011.

[18] M. Saqib, S. Daud Khan, N. Sharma, and M. Blumenstein, "A study on detecting drones using deep convolutional neural networks," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–5.

[19] C. Aker and S. Kalkan, "Using deep networks for drone detection," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.

[20] B. T. Lopez and J. P. How, "Aggressive 3-D collision avoidance for high-speed navigation," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2017, pp. 5759–5765.

[21] S. Song and J. Xiao, "Sliding shapes for 3D object detection in depth images," in *Proc. Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 634–651.

[22] L. Xia, C.-C. Chen, and J. K. Aggarwal, "Human detection using depth information by Kinect," in *Proc. CVPR WORKSHOPS*, Jun. 2011, pp. 15–22.

[23] A. Carrio, S. Vemprala, A. Ripoll, S. Saripalli, and P. Campoy, "Drone detection using depth maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1034–1037.

[24] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2016, *arXiv:1612.08242*. [Online]. Available: https://arxiv.org/abs/1612.08242

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[27] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Proc. Field Service Robot. Conf.* Zürich, Switzerland: Springer, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[28] Y. I. Jenie, E.-J. Van Kampen, C. C. De Visser, and Q. P. Chu, "Velocity obstacle method for non-cooperative autonomous collision avoidance system for UAVs," in *Proc. AIAA Guid., Navigat., Control Conf.*, Jan. 2014, p. 1472.

[29] B. T. Lopez, "Low-latency trajectory planning for high-speed navigation in unknown environments," Ph.D. dissertation, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Cambridge, MA, USA, 2016.

ADRIAN CARRIO (Member, IEEE) received the B.S. degree in industrial engineering from the University of Oviedo, Spain. He is currently pursuing the Ph.D. degree in automation and robotics with the Technical University of Madrid, Spain. His thesis work focuses on the development of vision-based collision avoidance systems for unmanned aerial vehicles. His research interests include robot perception using computer vision and machine learning, and autonomous robot navigation.

JESUS TORDESILLAS (Student Member, IEEE) received the B.S. degree in electronic engineering and robotics from the Technical University of Madrid, Spain, in 2016. He is currently pursuing the master's degree with the Aeronautics and Astronautics Department, Massachusetts Institute of Technology. He was a Researcher with the Center for Automation and Robotics (CAR), from 2015 to 2016. He is a member with the Aerospace Controls Laboratory. His research interests include optimal control and estimation, path planning, computer vision, and deep learning.

SAI VEMPRALA (Student Member, IEEE) received the B.Tech. degree from JNT University, India, in 2011, the M.S. degree from Arizona State University, in 2013, and the Ph.D. degree from Texas A&M University, in 2019. He is currently a Researcher with Microsoft Corporation. He is also a Roboticist with specific interest in unmanned aerial vehicles. His Ph.D. thesis was on the topic of collaborative localization and path planning for unmanned aerial vehicles. His research interests include robotic localization, planning, computer vision, and machine learning.

SRIKANTH SARIPALLI (Senior Member, IEEE) received the B.E. degree (Hons.) from the Birla Institute of Technology and Sciences, Pilani, India, in 1999, and the M.S. and Ph.D. degrees from the University of Southern California, in 2002 and 2007, respectively. He is currently an Associate Professor of mechanical engineering with Texas A&M University. He is also a Roboticist with research interests in unmanned systems in general and aerial vehicles in particular. His research interest includes robotic exploration particularly in air and ground vehicles and necessary foundations in perception, planning, and control for this domain. He was the Program Chair with the International Conference on Unmanned Aerial Systems 2015 and a member of AIAA.

PASCUAL CAMPOY (Member, IEEE) is currently a Full Professor of automation with the Universidad Politecnica de Madrid (UPM), Spain, and a Visiting Professor with TUDelft (The Netherlands). He has also been a Visiting Professor with Tong Ji University, Shanghai, China, and Q.U.T., Australia. He also lectures on Control, Machine Learning and Computer Vision. He leads the Computer Vision and Aerial Robotics Group, UPM within the Centre for Automation and Robotics (CAR), whose activities are aimed at increasing the autonomy of Unmanned Aerial Vehicles (UAVs). He has led over 40 Research and Development projects, including Research and Development European projects, national Research and Development projects and over 25 technological transfer projects directly contracted with the industry. He is the author of over 200 international scientific publications and nine patents, three of them registered internationally.

JONATHAN P. HOW (Fellow, IEEE) received the B.A.Sc. degree from the University of Toronto, in 1987, and the S.M. and Ph.D. degrees in aeronautics and astronautics from MIT, in 1990 and 1993, respectively. He then studied for two years at MIT as a Postdoctoral Associate for the Middeck Active Control Experiment (MACE) that flew onboard the Space Shuttle Endeavour, in March 1995. Prior to joining MIT in 2000, he was an Assistant Professor at the Department of Aeronautics and Astronautics, Stanford University. He is currently the Richard C. Maclaurin Professor of aeronautics and astronautics with the Massachusetts Institute of Technology. Prof. How is a Fellow of AIAA. He was a recipient of the 2002 Institute of Navigation Burka Award, the Boeing Special Invention Award, in 2008, the IFAC Automatica Award for Best Applications Paper, in 2011, the AeroLion Technologies Outstanding Paper Award for the Journal Unmanned Systems, in 2015, received the IEEE Control Systems Society Video Clip Contest, in 2015, and the AIAA Best Paper in Conference Awards, in 2011, 2012, and 2013. He is the Editor-in-Chief of *IEEE Control Systems Magazine* and an Associate Editor for the AIAA *Journal of Aerospace Information Systems*.