Detection of Unmanned Air Systems Using Multi-Camera Architectures

Ana Veiga
ana.margarida.veiga@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2021

Abstract

The number of lightweight Unmanned Aerial Vehicles (UAVs) available on the market is increasing. Due to limited payload, small UAVs are restricted in the sensors they can carry, and many make use of monocular or depth cameras, since they are lightweight and power efficient. These types of UAV are suitable for operating in cluttered environments, where they are at a high risk of collisions. Therefore, the ability to detect obstacles with camera sensors is essential. Additionally, because of their versatility and availability of access, these types of UAV can be exploited for dangerous or criminal activities. Being able to detect and localize malicious UAVs is then very important.

This thesis will focus on evaluating the capabilities of monocular and stereo depth fusion for obstacle detection. Depth predictions from a neural network will be combined with measurements of a depth camera, in order to obtain a more accurate and dense depth map. The second focus of this thesis will be to evaluate the possibility of utilizing a group of UAVs equipped with monocular cameras to localize an intruder UAV. An object detector network will be employed for the task of detecting the target, and then the location of the target will be found by triangulation. Three distinct triangulation algorithms will be evaluated and compared.

Keywords: Sensor Fusion, Triangulation, UAV Detection, Sense and Avoid, Obstacle Detection

1. Introduction

Nowadays, there is a large number of affordable, lightweight quadrotors available on the market. Due to their small size, they are particularly suited for operating at low altitude in cluttered environments [18], where the risk of colliding against unknown obstacles is much higher. Therefore, the detection of potential obstacles as well as of other UAVs play a key role in the safety of these types of UAVs.

Considering the limitations of smaller UAVs, many reported studies make use of lightweight and power efficient sensors like monocular cameras and depth cameras for obstacle detection. In Yang et al. [25], a probabilistic Convolutional Neural Network (CNN) was designed for monocular depth prediction, with the goal of obstacle avoidance. Since only a monocular camera was used, the depth is predicted up to a scale factor. Deep learning was also used by Wang et al. [24], where a convolutional neural network was used in combination with a depth camera for obstacle avoidance. First, the CNN was employed to obtain the obstacle's classification and bounding box. Then, the obstacle's profile and 3D spatial information are extracted from the depth map provided by the depth camera.

While depth cameras provide metric information about the localization of the obstacles, their range is very limited, since the types of cameras possible on a quadrotor have necessarily a small baseline. When it comes to monocular cameras, their main advantage compared to stereo vision is that since only one view is considered, theoretically their only range limitation is imposed by the image resolution. On the other hand, in contrast to the case with depth cameras, depth estimation methods that make use of monocular cameras are typically unable to provide metric information, and instead rely on additional sensors for absolute depth retrieval. For example, in Teixeira et al. [22], a neural network that performs depth completion takes as input the RGB image captured by a monocular camera as well as LiDAR measurements.

With a depth camera it is possible to take advantage of both the depth map and the RGB image as complementary information sources, and mitigate some of the limitations of these two sensors individually. In Fácil et al. [5], a method for the fusion of single- and multi-view depth estimates was developed, and in Martins et al. [13], a method for the

fusion of stereo and monocular depth estimates was presented. In Zhang et al. [27], monocular depth estimation was used to complete the depth channel of an RGB-D image, by making use of surface normal and occlusion boundaries.

The detection and localization of intruder UAVs is also an important task, again made especially challenging by the limitations in the type of sensors that lightweight UAVs can carry.

In Huang et al. [9], a method for the distance detection of a UAV using only a monocular camera was proposed. First, You Only Look Once (YOLO) object detector was used to detect the UAV in the image captured by the camera. Then, a convolutional neural network was employed to estimate the distance to the target. In Zahedi et al. [26], two neural networks were utilized for accurate mobile target localization and tracking. More traditional methods were used in Husodo et al. [10] and Laurito et al. [11], were algebraic expressions and prior knowledge of the target's dimensions were used to calculate its position. Husodo et al. [10] also proposes a method for following the target UAV, in order to obtain better results.

The use of multiple cooperative UAVs for the detection of another has also been studied in the literature. In Shinde et al. [19] a YOLO network was used to detect the target in images captured by the group of cooperative UAVs, and subsequently its position was ascertained, while in Arnold et al. [1] different methods of swarm formation for the purposes of malicious UAV tracking are evaluated.

The first objective of this thesis is to study whether it is feasible to resort to monocular depth estimation strategies, making use of the RGB image, to improve the depth camera measurements for obstacle avoidance. The second objective is to explore the use of triangulation algorithms for the localization of uncooperative UAVs, and compare the performance of three different triangulation methods: the linear, midpoint and L2 triangulation methods.

2. Object Detection

In order to use the information provided by the RGB image to complement the depth camera measurements, it was decided to use an architecture similar to the one described in Martins et al. [13], which is exemplified in Figure 1.

This architecture can be divided into three blocks: the stereo estimate (blue), the monocular estimate (orange) and the merging of both estimates (green). The stereo estimate is given directly by the depth camera, and may have missing values for some pixels. On the other hand, the monocular estimate is given by a convolutional neural network, which receives the RGB image as an input and outputs the estimated distance of each pixel to the camera. Because the monocular depth predictions in sky areas are unreliable, a sky segmentation step is included, where sky areas are removed from the monocular estimate. The adopted procedure for sky segmentation was the one described in Mashaly et al. [14], due to its fast execution speed.

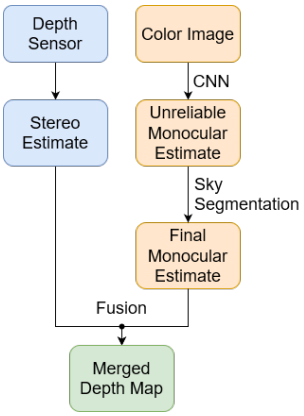


Figure 1: Proposed architecture for the monocular and stereo fusion.

Then, both estimates are merged, resulting in a more complete and accurate depth map than the previous ones.

2.1. Monocular Depth Estimation

To perform the monocular depth estimation, the MiDaS v2.1 Small [16] neural network was chosen, due to its robustness to various types of environments and fast execution speed. This network is a convolutional neural network that estimates the distance of each pixel to the camera from an RGB image. Additionally, to evaluate results, the DIODE dataset [23] was used, since it includes varied environments and types of obstacles and has dense ground truth measurements over a large range of distances (from 0.6 to 350m).

The output of the CNN corresponds to a relative and inverted depth map. Therefore, before the two depth estimates can be merged, the monocular depth estimate has to be scaled according to the depth camera measurements. The adopted procedure corresponds to the following steps:

1. Invert the depth camera measurements.
2. Align the monocular estimate with points from the inverted depth camera measurements, using the least squares method.
3. Invert the aligned monocular estimate in order to obtain the depth in meters.

2.2. Fusion of Stereo and Monocular Depth Estimates

The algorithm for the fusion of the two individual estimates was adapted from Martins et al. [13]. The changes made arise from taking into account

that in this case the two estimates do not have the same range of distances, since the stereo estimate is bound by the depth camera limitations in terms of range, while the monocular estimate is not. This algorithm can be summarized by the following points:

1. When the stereo estimate for a given pixel is considered reliable, it is preserved.
2. When the stereo estimate for a given pixel is missing, the monocular estimate is preserved.
3. When the stereo estimate for a given pixel isn't considered reliable:
   (a) if the two depth estimates are dissimilar then the monocular estimate is trusted more,
   (b) otherwise the stereo estimate is trusted more.

In practice, these rules correspond to the following equation

$$Z_{(x,y)} = W_{c_{(x,y)}} \times Z_{s(x,y)} + \left(1 - W_{c_{(x,y)}}\right) \cdot \left[\left(1 - W_{s_{(x,y)}}\right) \times Z_{m(x,y)} + W_{s_{(x,y)}} \times Z_{s(x,y)}\right] \quad (1)$$

where $Z_{(x,y)}$ is the final depth estimate of pixel (x, y), $Z_{m(x,y)}$ and $Z_{s(x,y)}$ are the monocular and stereo estimates of pixel (x, y), respectively, $W_{c_{(x,y)}}$ is a weighting factor dependent on the confidence of the stereo map at pixel (x, y), and $W_{s_{(x,y)}}$ is a weighting factor dependent on the ratio between the monocular and stereo estimates at pixel (x, y).

The weighting factor $W_{c_{(x,y)}}$ is given by

$$W_{c_{(x,y)}} = \frac{1}{1 + e^{0.25 \times d}} \quad (2)$$

where d corresponds to the distance in pixels between pixel (x, y) and the closest edge detected in the image, until a maximum distance of 5. The weighting factor $W_{s_{(x,y)}}$ is simply given by

$$W_{s_{(x,y)}} = \begin{cases} \dfrac{Z_{m(x,y)}}{Z_{s(x,y)}} & \text{if } Z_{s(x,y)} > Z_{m(x,y)} \\ \dfrac{Z_{s(x,y)}}{Z_{m(x,y)}} & \text{if } Z_{s(x,y)} < Z_{m(x,y)} \end{cases} \quad (3)$$

3. Target UAV Localization

The monocular and stereo vision fusion method presented in the previous section does not yield good results for the distance estimation of an intruder UAV, since "floating objects" were not contemplated in the training of the monocular depth estimation network. Therefore, a different method is needed to localize intruder UAVs. Since the type of depth cameras that can be installed in UAVs have severe limitations in terms of range, it was decided to use several cooperative UAVs, each equipped with a monocular camera, for this task. Figure 2 illustrates the proposed architecture for the localization of a target UAV.

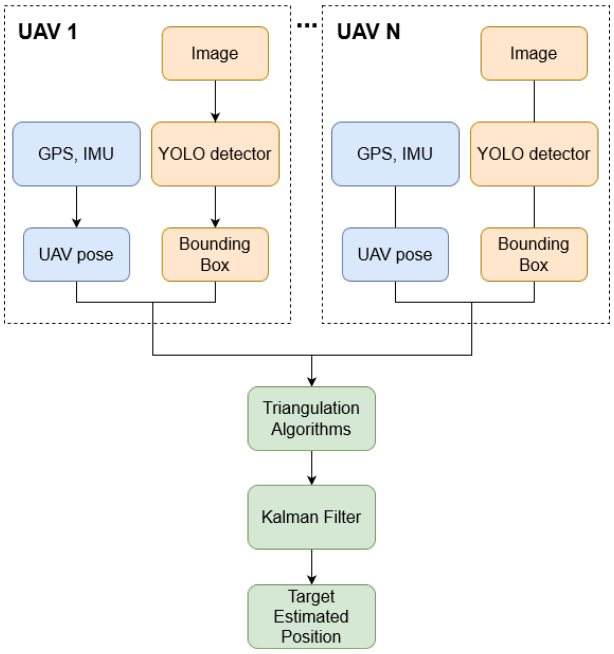


Figure 2: Proposed architecture for the target localization method.

First, for each cooperative UAV that sees the target, the YOLOv3 [17] algorithm provides the respective target's image coordinates. Then that information, along with the position and orientation of each cooperative UAV, is used by a set of triangulation algorithms in order to estimate the target's location. To improve the estimated intruder UAV position, a Kalman filter was used after the triangulation step, resulting in the final estimated target position.

3.1. Visual Object Detection

Two criteria were used to select the visual detection algorithm to perform the target UAV's detections: accuracy and processing speed. To construct a real-time solution, the latter is required. As previously mentioned, the YOLOv3 algorithm was chosen because it offered a good compromise between both factors [28].

The Detfly dataset [29], which consists of more than 13,000 labeled images of a flying UAV (DJI Mavic2), was chosen for training. This dataset was selected because it includes a variety of realistic scenarios with an assortment of background scenes, viewing angles, relative distances, flying altitudes, and lightning conditions. The images were divided into training and validation sets in an 80/20 split.

3.2. Pinhole Projection Model

The pinhole camera projection model [6] describes the projection of points in three-dimensional space onto a two-dimensional image plane. It does so by considering that the camera aperture can be described as a single point (called pinhole), and that all the light captured by the camera must pass though it before reaching the optical sensor. Despite its approximations, it is a reasonable description of how a camera depicts a 3D scene, and is widely used in computer vision applications [20].

The camera matrix $P \in \mathbb{R}^{3\times4}$ describes the relation between a point in the world reference frame and its projection in the camera reference frame [7]. The projection $\tilde{\mathbf{u}}$ of a three-dimensional point $\tilde{\mathbf{x}}$ onto the two-dimensional image plane is then given by

$$\tilde{\mathbf{u}} = P\tilde{\mathbf{x}} \tag{4}$$

where $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{x}}$ are both expressed in homogeneous coordinates.

The camera matrix P consists of two camera elements. Firstly, the camera intrinsic parameters are described by matrix $K \in \mathbb{R}^{3\times3}$, and include information about camera specifics like its focal length and principal point. Second, the camera extrinsic parameters express how a point in the world coordinate system is transformed into the camera coordinate system. This transformation, described by matrix $E \in \mathbb{R}^{3\times4}$, is characterized by a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. Matrix E is then determined as

$$E = [R|\mathbf{t}] \tag{5}$$

and the camera matrix P, defined as $P = KE$, thus corresponds to

$$P = K[R|\mathbf{t}] \tag{6}$$

3.3. Triangulation Algorithms

Triangulation algorithms deal with the problem of finding the position of a point $\mathbf{x} \in \mathbb{R}^3$ given its projection $\mathbf{u_1}, \ldots, \mathbf{u_n} \in \mathbb{R}^2$ in n images taken with cameras with known calibration and pose, that is, with known camera matrices $P_1, \ldots, P_n \in \mathbb{R}^{3\times4}$. For the triangulation to be possible, at least two non-collinear detections are necessary. In this thesis, three triangulation methods will be discussed: the linear method, the midpoint method and the L2 method.

The linear triangulation algorithm, described in Hartley et al. [8], is a simple and efficient method to solve the triangulation problem.

The projection of a point in space into an image plane is can be expressed in homogeneous coordinates by

$$\tilde{\mathbf{u}} = w(u, v, 1)^\mathsf{T} \tag{7}$$

where (u, v) are the observed point coordinates in the image and w is an unknown scale factor.

Each detection, given by the projection equation for the pinhole camera $\tilde{\mathbf{u}} = P\tilde{\mathbf{x}}$, results in two linearly independent equations. All these equations can be combined in the form

$$A\tilde{\mathbf{x}} = 0 \tag{8}$$

with $A \in \mathbb{R}^{2n\times4}$.

In the presence of noise, equation (8) does not have an exact solution. A common approach to find an approximate solution is to use the Homogeneous method [7], which minimizes $||A\tilde{\mathbf{x}}||$ subject to the condition $||\tilde{\mathbf{x}}|| = 1$. This problem can be solved using single value decomposition (SVD) [2].

The midpoint triangulation algorithm is described in Beardsley et al. [3] for the two-views case, and further extended for the general case of n-views in Ramalingam et al. [15].

For every point of view $i \in \{1, \ldots, n\}$, we can construct a detection ray that starts at the camera position $\mathbf{c_i} = -R_i^\mathsf{T}\mathbf{t_i}$ and passes through a point $\mathbf{v_i} \in \mathbb{R}^3$ given in homogeneous coordinates by $\tilde{\mathbf{v}}_\mathbf{i} = \tilde{P}_i^{-1}\tilde{\mathbf{u}}_\mathbf{i}$. We can write this detection ray as

$$\mathbf{r_i}(t_i) = \mathbf{c_i} + t_i\mathbf{d_i} \tag{9}$$

The midpoint triangulation algorithm determines the point $\hat{\mathbf{x}}$ which is closest on average to all rays, that is

$$\hat{\mathbf{x}} = \underset{\hat{\mathbf{x}}}{\operatorname{argmin}} \sum_{i=1}^{n} \mathrm{d}(\hat{\mathbf{x}}, \mathbf{r_i})^2 \tag{10}$$

where $\mathrm{d}(*,*)$ denotes the Euclidean distance between a point and a line. In the two-view case, $\hat{\mathbf{x}}$ corresponds to the midpoint of the common perpendicular to the two rays.

Ramalingam et al. [15] provides a closed-form solution for $\hat{\mathbf{x}}$ via

$$\hat{\mathbf{x}} = \frac{1}{n}(I_3 + DD^\mathsf{T}A)\sum_{i=1}^{n}\mathbf{c_i} - A\sum_{i=1}^{n}\mathbf{d_i}\mathbf{d_i}^\mathsf{T}\mathbf{c_i} \tag{11}$$

where n is the number of detections, I_3 is the 3×3 Identity matrix, $D = [\mathbf{d_1}|\ldots|\mathbf{d_n}] \in \mathbb{R}^{3\times n}$ is a matrix than contains the direction vectors of the detections, and matrix $A \in \mathbb{R}^{3\times3}$ is given by

$$A = (nI_3 - DD^\mathsf{T})^{-1} \tag{12}$$

The two previous triangulation algorithms presented do not take into account the projective properties of pinhole cameras. In contrast, the L2 triangulation method, outlined in Sturm et al. [21] and

Chen et al. [4], operates in the two-dimensional space of the image planes. The goal of the L2 algorithm is to find the $\hat{\mathbf{x}}$ that minimizes the reprojection error, which is corresponds to

$$\hat{\mathbf{x}} = \underset{\hat{\mathbf{x}}}{\operatorname{argmin}} \sum_{i=1}^{n} \mathrm{d}(\mathbf{u_i}, \hat{\mathbf{u}}_\mathbf{i})^2 \tag{13}$$

where $\mathrm{d}(*,*)$ denotes the Euclidean distance between two points, and $\hat{\mathbf{u}}_\mathbf{i} = P_i\hat{\mathbf{x}}$.

Equation (13) corresponds to a non-linear least squares problem, therefore is not solvable in a trivial matter. A common approach to solve this problem is the Levenberg-Marquardt method [12].

4. Results

In this section, the results related to the monocular and stereo depth fusion method will be presented first, in subsection 4.1, while the results pertaining to the intruder UAV localization will be presented in subsection 4.2.

4.1. Obstacle Detection

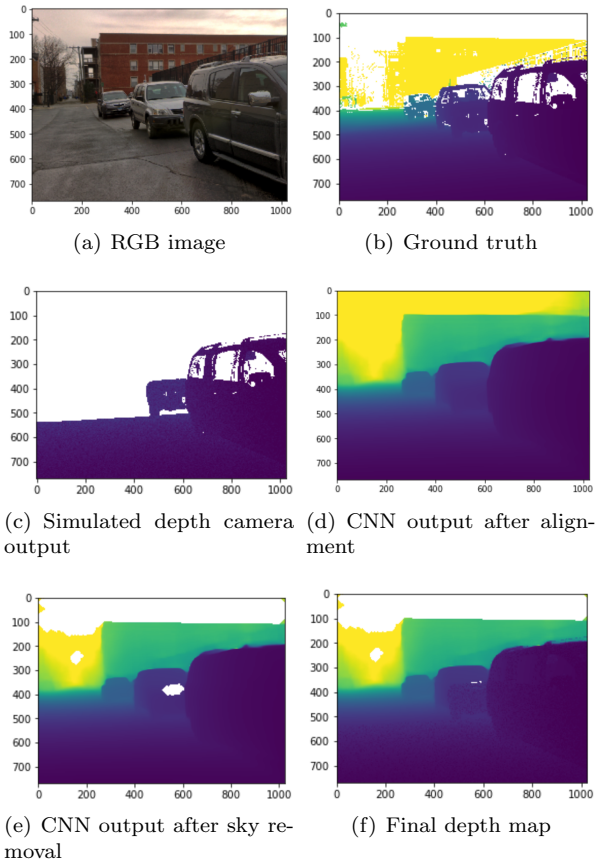
The output of a depth camera was simulated from the ground truth of the DIODE dataset. For this the ground truth measurements smaller than 10 meters, since it is a common range for depth cameras commonly used in UAVs, were considered as captured by a depth camera, and error was introduced. The results obtained for one image are shown in Figure 3, as an illustration of the method.

From the comparison of Figure 3(d), which represents the monocular estimate, with the ground truth depicted in Figure 3(b), it can be exemplified that the monocular estimate has a tendency to underestimate the distance to faraway objects, like in this case the building behind the cars. Moreover, in Figure 3(e) it can be observed that the sky segmentation step removed two regions from the estimate wrongfully. One of the "holes" in the monocular estimate was filled in Figure 3(f) since there were depth camera measurements for that area of the image, however the other "hole" remained. These situations could probably be avoided with the use of a more precise sky segmentation algorithm. Finally, it can be seen from the qualitative comparison of Figures 3(c) and 3(f), depicting the depth camera simulated measurements and the final depth estimate respectively, that the final depth map is much more full than the stereo depth map.

Table 1 summarizes the results obtained for the validation set of the Diode Dataset. Overall, there was a 21.16% increase in the number of pixels with depth information.

4.2. Target UAV Detection

The method mentioned in Section 3 for the detection and localization of an intruder UAV was evaluated with data collected with the AirSim simulation environment. The triangulation algorithms were executed at each time step, and to improve results a Kalman filter was used as well.



(a) RGB image (b) Ground truth

(c) Simulated depth camera output (d) CNN output after alignment

(e) CNN output after sky removal (f) Final depth map

Figure 3: Representation of the steps taken for the obtainment of the final estimated depth map.

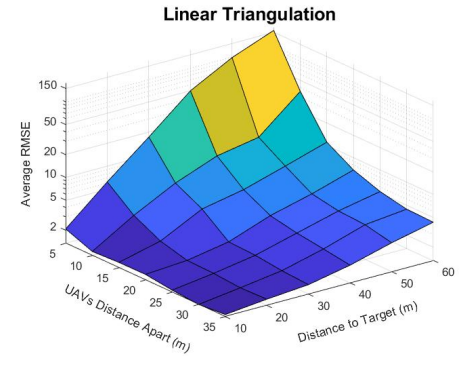
For the results presented in Figure 4, two cooperative UAVs and one target were considered, and the impact of the distance between the cooperative UAVs, as well as the distance between the cooperative UAVs and the target on the average position error obtained was studied.

As we can see from the figure, when the distance between the two cooperative UAVs is small in comparison to the distance between the two UAVs and the target, the error obtained is very large. Therefore, to accurately position a target that is far away, the cooperative UAVs should be as far away from each other as possible, in order to capture varied points of view.

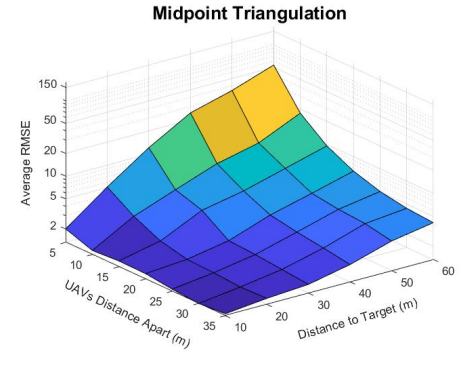
In almost all simulations, the midpoint triangulation algorithm obtained the best results. The linear triangulation algorithm was the second best, closely followed by the L2 triangulation algorithm. However, the difference is the most relevant in the yellow portions of the graphs, where the target is located far away from the cooperative UAVs and these are positioned close together. Here, the midpoint algorithm clearly outperformed the other two. In the

Table 1: Average error metrics of the final predicted depth maps.

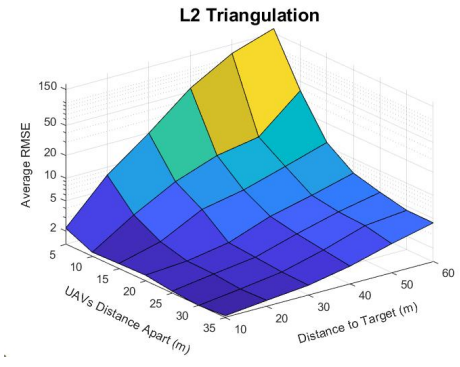
RMSE	Absolute Relative Error	Threshold Accuracy		
		$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
9.51	0.385	0.524	0.648	0.762



(a)



(b)



(c)

Figure 4: Average RMSE obtained as a function of the distance to the target, and the distance between the two cooperative UAVs, for each algorithm: (a) linear triangulation, (b) midpoint triangulation and (c) L2 triangulation.

worst case, with the cooperative UAVs separated by only 5m, and 60m away from the target, the error obtained with the midpoint algorithm (53.4m) was less than half the error obtained with the linear (158.4m) or L2 (172.9m) algorithms. Although these errors are too large for any real application in these conditions, they serve to illustrate some of the differences between the three triangulation strategies.

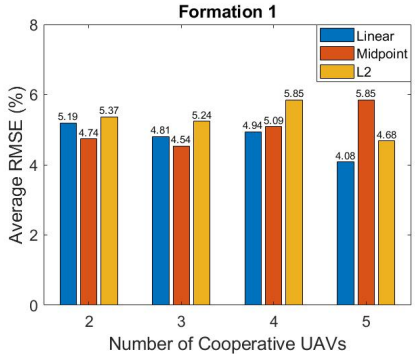
In the conditions corresponding to the blue areas of the graph, an application of these algorithms is much more realistic. In the best case scenario, the linear, midpoint and L2 algorithms obtained an average RMSE of 1.43m, 1.42m and 1.41m respectively.

The effect of the number of cooperative UAVs on the obtained results was also studied, and configurations with a different number of cooperative UAVs, positioned according to two different formations, were examined. The target was at an average distance of 30m from the cooperative UAVs. In *Formation 1*, the cooperative UAVs position themselves close to each other, so that the effect of adding more viewpoints without increasing the maximum parallax angle between detections can be studied.

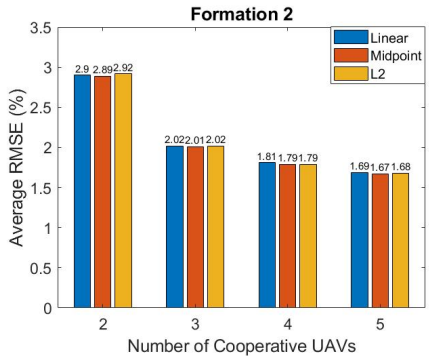
In the case of *Formation 2*, the cooperative UAVs position themselves equally spaced along a circle centered around the target's trajectory. This configuration is intended to study the effect that additional varied points of view have in the accuracy of the estimated location of the target.

The results in Figure 5(a) seem to indicate no meaningful performance increase as a result of the extra UAVs. A closer inspection revealed that the benefit of having more sensors is outweighed by the higher probability of having some detections that are not perfectly centered around the target, since when the detections are close to parallel, even small inaccuracies in the detection coordinates can lead to considerable errors in the estimated target position.

In the case of Figure 5(b), as expected, with the increase in the number of cooperative UAVs the RMSE decreases. Even though with more cooperative UAVs there is a higher probability that at a certain time step, there will be some detections that are not perfectly centered around the target, the additional points of view are sufficient to improve results. Since in this circumstance the detection directions are not close to being parallel with each other, small errors in the detection coordinates do not affect the final estimated position as signifi-



(a) Formation 1



(b) Formation 2

Figure 5: Influence of the number of cooperative UAVs on the results obtained, with the cooperative UAVs disposed according to two different formations.

cantly.

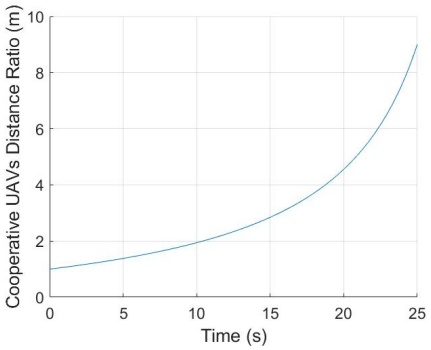
Nonetheless, it can be infered that increasing the number of cooperative UAVs seems to have a diminishing return, since the biggest performance gain happens with the increment from two to three cooperative UAVs, and after that any additional increases lead to smaller improvements. Additionally, it can also be concluded that in this case the discrepancies observed between the three triangulation algorithms are not very significant.

The effect of having some cooperative UAVs much closer to the target than others was also studied. Triangulation algorithms that minimize the reprojection error, such as the L2 triangulation algorithm, should theoretically perform better in this case. For this study, a trajectory for two cooperative UAVs was created where they change their distance to the target, one moving closer to it and the other moving in the opposite direction.

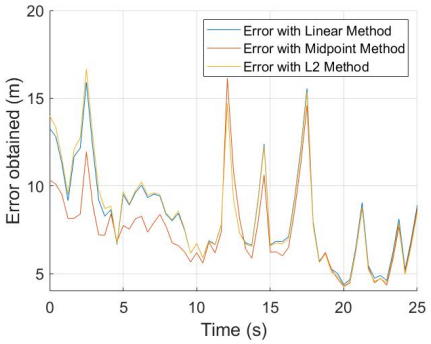
If we consider that UAV number 1 is moving towards the target and UAV number 2 is moving away, we can define the ratio of their respective distances as

$$D_{ratio} = \frac{d_2}{d_1} \tag{14}$$

where d_1 and d_2 are the distance from UAVs 1 and 2, respectively, to the target. This ratio quantifies how much closer one UAV is to the target than the other. The change of D_{ratio} over the course of the trajectory is plotted in Figure 6(a). At the start of the simulation, both cooperative UAVs are at the same distance from the target, therefore $D_{ratio} = 1$. As the simulation progresses, this difference in respective distances increases until UAV 1 is 9 times closer to the target than UAV 2.



(a) Cooperative UAVs Distance Ratio



(b) Error obtained

Figure 6: Results obtained for the *trajectory 3* simulation.

Figure 6(b) depicts the error of each of the three triangulation algorithms throughout the simulation. At the start of the trajectory the midpoint algorithm obtained the best results, which is in agreement with the results previously discussed. However, as D_{ratio} increases, the difference in performance between the algorithms becomes less apparent, and at the end of the trajectory all algorithms have a very comparable performance.

The maximum value of D_{ratio} is limited by the ability of the furthest away cooperative UAV to detect the target. It would be expected that the L2 triangulation algorithm would outperform the other two when some sensors are much closer to the target than others, since it minimizes the reprojection error of the detections. However, the maximum value of D_{ratio} does not seem to be high enough to see

this effect.

5. Conclusions

This thesis provides a multi-sensor methodology for UAV obstacle detection, as well as for the localization of an intruder UAV, taking advantage of current deep learning algorithms.

The examined results suggest that there is a benefit in using the presented method for the fusion of monocular and stereo depth, in order to use monocular depth estimation to complement the measurements of a depth camera for UAV obstacle detection. This method is specially usefull when it comes to filling in the gaps in the depth map provided by the depth camera due to the obstacles being out of range. This improves the information available to be used by an obstacle avoidance algorithm, for instance. Having information about potential obstacles at greater distances can allow the UAV to fly faster, for example.

In regards to the target UAV detection, the results also look promising in that monocular cameras are suitable sensors to detect and localize an intruder UAV. This research also evaluated three different triangulation algorithms in different scenarios. These are the linear triangulation, the midpoint triangulation, and the L2 triangulation methods. It was concluded that the midpoint triangulation algorithm is the most appropriate for the task from among those considered. It also presented suggestions of how to minimize the target position error obtained, both in terms of improved YOLO training and cooperative UAV positioning.

In regards to the monocular and stereo depth fusion method, future steps would include more thorough evaluation, which could be performed with data from the AirSim simulation environment, as well as evaluation onboard a UAV in real time. Further work could also include the integration of this method with an obstacle avoidance algorithm.

When it comes to the intruder UAV localization, potential future work includes improvements in the YOLO detector training, both in terms of increasing its detection capabilities below the horizon and also in augmenting its bounding box precision, in order to reduce the error introduced in the triangulation algorithms. Additionally, the creation of a dynamic region of interest where the target is expected to be in the captured images, based on previous localization results, would allow YOLO to run faster and therefore increase the results precision.

Further work for both methods could also include the algorithms optimization for performance on an onboard computer, and proving their real time capabilities.

References

[1] C. Arnold and J. Brown. Performance evaluation for tracking a malicious uav using an autonomous uav swarm. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0707–0712. IEEE, 2020.

[2] K. E. Atkinson. *An Introduction to Numerical Analysis*. Wiley, 1989.

[3] P. A. Beardsley, A. Zisserman, and D. W. Murray. Navigation using affine structure from motion. In *European Conference on Computer Vision*, pages 85–96. Springer, 1994.

[4] J. Chen, D. Wu, P. Song, F. Deng, Y. He, and S. Pang. Multi-view triangulation: Systematic comparison and an improved method. *Ieee Access*, 8:21017–21027, 2020.

[5] J. M. Fácil, A. Concha, L. Montesano, and J. Civera. Single-view and multi-view depth fusion. *IEEE Robotics and Automation Letters*, 2(4):1994–2001, 2017.

[6] D. A. Forsyth and J. Ponc. *Computer Vision, A Modern Approach*. Prentice Hall, 2003.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2000.

[8] R. I. Hartley and P. Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[9] Z.-Y. Huang and Y.-C. Lai. Image-based sense and avoid of small scale uav using deep learning approach. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 545–550. IEEE, 2020.

[10] A. Y. Husodo, G. Jati, N. Alfiany, and W. Jatmiko. Intruder drone localization based on 2d image and area expansion principle for supporting military defence system. In *2019 IEEE International Conference on Communication, Networks and Satellite (Comnetsat)*, pages 35–40. IEEE, 2019.

[11] G. Laurito, B. Fraser, and K. Rosser. Airborne localisation of small uas using visual detection: A field experiment. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1435–1443. IEEE, 2020.

[12] K. Madsen, H. B. Nielsen, and O. Tingleff. Methods for non-linear least squares problems. 2004.

[13] D. Martins, K. Van Hecke, and G. De Croon. Fusion of stereo and still monocular depth estimates in a self-supervised learning context. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 849–856, 2018.

[14] A. S. Mashaly. Performance assessment of sky segmentation approaches for uavs. *International Journal of Image and Graphics*, 19(04):1950023, 2019.

[15] S. Ramalingam, S. K. Lodha, and P. Sturm. A generic structure-from-motion framework. *Computer Vision and Image Understanding*, 103(3):218–228, 2006.

[16] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2020.

[17] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[18] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert. Learning monocular reactive uav control in cluttered natural environments. In *2013 IEEE international conference on robotics and automation*, pages 1765–1772. IEEE, 2013.

[19] C. Shinde, R. Lima, and K. Das. Multi-view geometry and deep learning based drone detection and localization. In *2019 Fifth Indian Control Conference (ICC)*, pages 289–294. IEEE, 2019.

[20] P. Sturm. Pinhole camera model, 2014.

[21] P. Sturm and R. Hartley. Triangulation. In *Image Understanding Workshop*, volume 11, pages 957–966, 1994.

[22] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli. Aerial single-view depth completion with image-guided uncertainty estimation. *IEEE Robotics and Automation Letters*, 5(2):1055–1062, 2020.

[23] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.

[24] D. Wang, W. Li, X. Liu, N. Li, and C. Zhang. Uav environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution. *Computers and Electronics in Agriculture*, 175:105523, 2020.

[25] X. Yang, J. Chen, Y. Dang, H. Luo, Y. Tang, C. Liao, P. Chen, and K.-T. Cheng. Fast depth prediction and obstacle avoidance on a monocular drone using probabilistic convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[26] R. Zahedi, E. Ceh-Varela, R. Selje II, H. Cao, and L. Sun. Neural network based approaches to mobile target localization and tracking using unmanned aerial vehicles. In *AIAA Scitech 2020 Forum*, page 0392, 2020.

[27] Y. Zhang and T. Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao. Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning. *IEEE Robotics and Automation Letters*, 6(2):1020–1027, 2021.

[29] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao. Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning. *IEEE Robotics and Automation Letters*, 6(2):1020–1027, 2021.