2021 International Conference on Unmanned Aircraft Systems (ICUAS)
Athens, Greece. June 15-18, 2021

Monocular Vision-based Obstacle Avoidance Scheme for Micro Aerial Vehicle Navigation

Samuel Karlsson[1], Christoforos Kanellakis[1], Sina Sharif Mansouri[1] and George Nikolakopoulos[1]

Abstract— One of the challenges in deploying Micro Aerial Vehicless (MAVs) in unknown environments is the need of securing for collision-free paths with static and dynamic obstacles. This article proposes a monocular vision-based reactive planner for MAVs obstacle avoidance. The avoidance scheme is structured around a Convolution Neural Network (CNN) for object detection and classification (You Only Lock Once (YOLO)), used to identify the bounding box of the objects of interest in the image plane. Moreover, the YOLO is combined with a Kalman filter to robustify the object tracking, in case of losing the boundary boxes, by estimating their position and providing a fixed rate estimation. Since MAVs are fast and agile platforms, the object tracking should be performed in real-time for the collision avoidance. By processing the information of the bounding boxes with the image field of view and applying trigonometry operations, the pixel coordinates of the object are translated to heading commands, which results to a collision free maneuver. The efficacy of the proposed scheme has been extensively evaluated in the Gazebo simulation environment, as well as in experimental evaluations with a MAV equipped with a monocular camera.

I. Introduction

Lately, there has been an increase in the deployment of the Micro Aerial Vehicless (MAVs) in variety applications [1], such as aerial photography [2], [3], infrastructure safety inspection [4], underground mine exploration [5], [6], and delivery of supplies [7].

One of the major challenges, in such application scenarios, is the need to guarantee collision free paths for the MAV navigation, especially in environments that are partially or totally unknown, with the existence of limited or tight open spaces and static or dynamic obstacles, such as humans that requires a dexterous and fine tuned obstacle detection and avoidance scheme. This article proposes a novel scheme based on the Convolution Neural Network (CNN) to detect obstacles by monocular camera. As it will be demonstrated, based on the estimated relative distance and relative speed of the obstacles, the MAV heading is corrected to avoid possible collisions.

A. Background and Motivation

In the related literature, there have been multiple approaches for the detection and avoidance of objects with a monocular camera. In [8] and [9], the You Only Lock Once (YOLO) object detection algorithm has been proposed. In these approaches, the position in a 3D coordinate system and the overall size of the objects are not considered. In order to obtain this information, extra steps, such as the multi view approach [10], [11] or CNN is required. The multi view approach requires for an ego-motion to get multiple views to estimate the distance to different objects, while it fails if the object has moved during the ego-motion, thus the overall approach it not suitable for obstacle avoidance in dynamic environments. The CNN approaches are computationally expensive to obtain the depth and merging it with object detection methods, while usually will not result to a real time performance. For this case, there exist methods that combine the detection and the depth estimation for obstacle avoidance [12], [13], however these methods are mainly designed and tuned for a specific task, such as obstacle avoidance of power transmission lines [12] that results to a not general solution for the MAV navigation.

In the related literature, there are have been many multiple works that consider methods for solving the combined path-planning and obstacle avoidance problem and that have successfully been utilized for MAV's[14], [15]. However most of these methods such as the potential fields [16], [17], or the dynamic graph search methods like $A^\star$ [18] or MSA* [19] consider to know the position of the obstacles and mainly have been studied in environments with static obstacles.

B. Contributions

The main contribution of this article is the development of the reactive planner for obstacle avoidance for the MAVs navigation, based on the forward facing monocular camera. The specific architecture is based on the combination of a CNN detection method (YOLO) with a bounding box processing for correcting the MAV heading commands and thus enabling proper collision avoidance. On top of the object detector, a Kalman filter is implemented to track boxes in the case of loosing them from object detection by estimating the position and by increasing the tracking rate of the object, in order to allow fast bounding box calculations, a capability that is critical for the MAV navigation. The second contribution stems from developing a scheme that is able to provide a weighted risk proposal for all objects in the scene. As such, based on the highest calculated risk of collision, the identified bounding boxes, in pixel coordinates, are

*This work has been partially funded by the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 869379 illuMINEation.

[1]Robotics and Artificial Intelligence Team Department of Computer Science, Electrical and Space Engineering Luleå University of Technology SE-97187, Sweden
Corresponding author's email: samkar@ltu.se

2021 International Conference on Unmanned Aircraft Systems (ICUAS) | 978-1-6654-1535-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICUAS51884.2021.9476793

converted to new heading command using trigonometry.

Finally, the developed architecture has been evaluated in both simulations and experimental studies, where it has been extensively demonstrated the effectiveness in the evaluated scenarios with either static or dynamic pedestrians.

C. Outline

The rest of the article is structured as follows. Section II discusses the proposed methodology for vision based reactive heading regulation, while Section III presents the experimental results performed using the proposed architecture. Finally conclusions are stated in Section V.

II. Vision Based Reactive Planner

Figure 1 depicts the block diagram of the proposed structure, where it can be observed that two components that consist of the reactive planner. The detection and tracking component for the obstacles has the role to transmit, the corresponding bounding boxes B of the tracked obstacles, to the obstacle avoidance component, where the avoidance policy is applied. Moreover, the high level planner is providing the waypoints to the Obstacle avoidance, where a suggested heading ${}^{\mathbb{B}}\alpha_s$ is calculated and is sent to the high level planner. The high level planner is a PD (proportional-derivative) controller, applied on the error between the current heading ${}^{\mathbb{B}}\alpha_c$ and ${}^{\mathbb{B}}\alpha_s$. In the sequel, the output is sent to the high level controller in order to generate the corresponding thrust, roll an pitch commands for the cascade connected low level controller. In the last step, the low level controller generates the proper motor commands for the MAV.

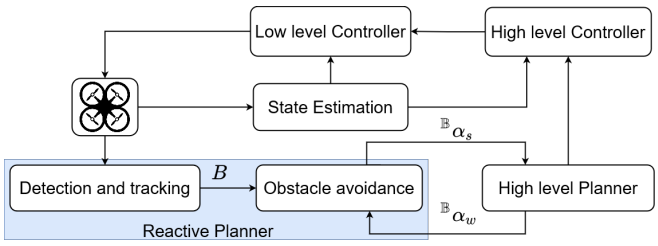


Fig. 1: An overall structure of the proposed method.

Three different frames are considered, as depicted in Figure 2. The inertial frame $\mathbb{I}$, the body fixed frame $\mathbb{B}$, which is fixed on the MAV with ${}^{\mathbb{B}}x$, and the camera frame $\mathbb{C}$. The relation between $\mathbb{C}$ and $\mathbb{B}$ has a fixed rotation as:

$$ {}^{\mathbb{B}}\mathbf{R}_{\mathbb{C}} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \tag{1} $$

A. Detection and tracking

The objects to be avoided are detected using YOLO [9] (the authors pre-trained wights were used). YOLO

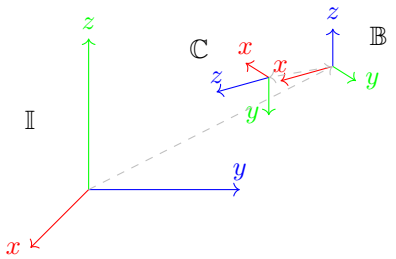


Fig. 2: The coordinate frames used. $\mathbb{I}$ is the inertial frame, $\mathbb{B}$ is the body fixed frame for the MAV, $\mathbb{C}$ is the camera frame. $\mathbb{C}$ has a static relationship to $\mathbb{B}$. Whiles $\mathbb{B}$ moves relative to $\mathbb{I}$.

gives bounding boxes as an output, including the center position of the object ${}^{\mathbb{C}}\begin{bmatrix} x & y \end{bmatrix}^{\top}$, as well as the height (${}^{\mathbb{C}}h$) and width (${}^{\mathbb{C}}w$) of the object, with all the dimensions noted in pixels. In the sequel, the objects are traced between the frames by using the Hungarian algorithm [20], in combination with a Kalman filter [21]. The Kalman filter is used to predict the object's next position to improve the accuracy of the combinatorial optimization Hungarian algorithm.

B. The obstacle avoidance method

1) Angle to obstacle: Object detection methods, like YOLO [9] are able to return the position and size of the detected objects in pixel coordinates ${}^{\mathbb{C}}c_p$. For correcting the heading of the MAV, only the position of the obstacle in 2D coordinate will be sufficient for generating the heading command. In the pixel coordinate, the MAV has an angle of α from its heading to each object and α is zero when the object is in the center of the image. By assuming a set of pixels, based on the resolution of the camera, different α can be extracted for each pixel and the pixel Field of View (FoV) ${}^{\mathbb{C}}p_{fov}$ can be defined. The sum of all the ${}^{\mathbb{C}}p_{fov}$ on the horizontal plane is equal to the camera's FoV in the horizontal plain. By adding the ${}^{\mathbb{C}}p_{fov}$'s of all the pixels, from the center of the image till the pixel corresponding to the objects ${}^{\mathbb{C}}c_p(x, y, 1)$, is the α for that ${}^{\mathbb{C}}c_p$ received. As such, the α between two pixels can be calculated by:

$$ p_{nr} = p_j - p_i \tag{2} $$

$$ {}^{\mathbb{C}}\alpha = {}^{\mathbb{C}}p_{fov} p_{nr}, \tag{3} $$

where ${}^{\mathbb{C}}p_{nr}$ is the number of pixels between pixel i (p_i) and pixel j (p_j). It should be highlighted that the accuracy of the proposed method is highly dependent on the resolution of the camera, while higher resolution results into more accurate angles. The formula to calculate ${}^{\mathbb{C}}p_{fov}$ is:

$$ {}^{\mathbb{C}}p_{fov} = \frac{{}^{\mathbb{C}}w_p}{{}^{\mathbb{C}}I_{fov}}, \tag{4} $$

where ${}^{\mathbb{C}}w_p$ is the image width in pixels and ${}^{\mathbb{C}}I_{fov}$ is the image horizontal FoV. By using a c_p, relative to the

center pixel in the image, ${}^{\mathbb{C}}c_p$ can be transformed into $\mathbb{B}$ by ${}^{\mathbb{B}}\mathbf{R}_{\mathbb{C}}\ {}^{\mathbb{C}}c_p$.

2) Proximity to collision: In this article, the area of the detected objects are used to obtain the time of collision without knowledge of the distance d and the relative speed of the objects. The area A of the detected object's B can be calculated (in pixels). By calculating the area change ΔA between frames, a relative time of arrival can be extracted. Since the objects grow in size, when they come closer, the corresponding growing rate is affected by both distance and speed. Closer objects that move fast have a larger ΔA than objects that are far away and are moving slow. This results to an estimation of the objects that have a higher probability for the collision and thus the MAV should avoid them earlier, when compared to other obstacles in the environment.

3) Avoidance scheme: The avoidance method is implemented by defining a waypoint $\mathbf{W} = {}^{\mathbb{I}}\begin{bmatrix} x & y & z \end{bmatrix}^{\top}$. In this case, ${}^{\mathbb{B}}\alpha_w$ is the yaw angel in $\mathbb{B}$ to the $\mathbf{W}$. To reach the $\mathbf{W}$ safely, the safest heading closest to the ${}^{\mathbb{B}}\alpha_w$ should be chosen. In the case that there are no obstacles, the ${}^{\mathbb{B}}\alpha_w$ is the best choise, at least when the travel d is considered. But if there are obstacles in the way, the heading that is safest and closest to ${}^{\mathbb{B}}\alpha_w$ may be chosen (${}^{\mathbb{B}}\alpha_s$). The ${}^{\mathbb{B}}\alpha_s$ is calculated with a safety marginal s. This is achieved by mapping the obstacles and the free spaces on an risk array ψ, weighted with proximity to collision (ΔA) as presented in the Algorithm 1 and denoted by. The ψ has the same length as the image width in pixels w_p, thus the ${}^{\mathbb{C}}c_p$ for the Bs is equivalent to the index's in ψ. By finding the minimum sum in a span of length $2s$ in ψ as presented in Algorithm 2:

$$min \sum_{i=j}^{j+2s} \psi[i]$$

${}^{\mathbb{C}}c_s = j + s$ can be calculated, thus it is the safest heading found in pixel coordinates. In equation 3 is ${}^{\mathbb{C}}c_s$ transformed to ${}^{\mathbb{B}}\alpha_s$. If there are many possibly spans that have the same collision risk, the one with the heading, closest to ${}^{\mathbb{B}}\alpha_w$, be selected. This method is visualized in Figure 3 and in Figure 4 where an actual frame is visualized. By updating ${}^{\mathbb{B}}\alpha_s$ regularity, the MAV will turn back towards the $\mathbf{W}$ when the obstacle disappears from ${}^{\mathbb{B}}\alpha_w$.

III. Simulation and Experimental Results

This Section describes the simulation and experimental evaluation of the proposed method, while a video summary of the obtained results can be reached in the following link [1].

Initially, the proposed method is evaluated in the Gazebo simulation environment and in closed loop with a MAV in a laboratory environment. In both cases, the Parrot Bebop 2 [22] is used for the evaluation of the

[1]https://drive.google.com/file/d/136C2AU5uKKRjEaH-niPfGdd-lMCilRIQ/view?usp=sharing

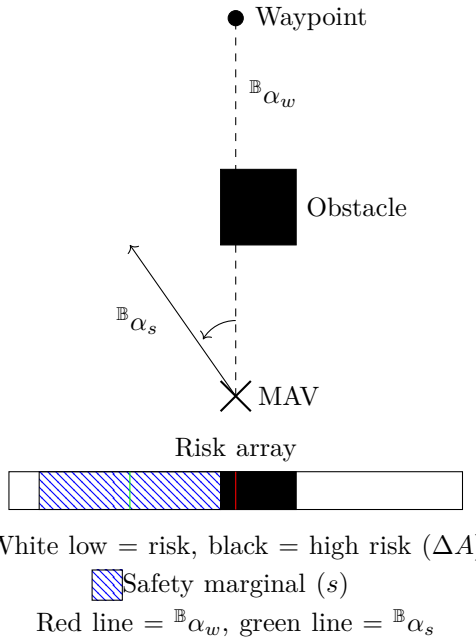


Fig. 3: A schematic view of how the heading selection.

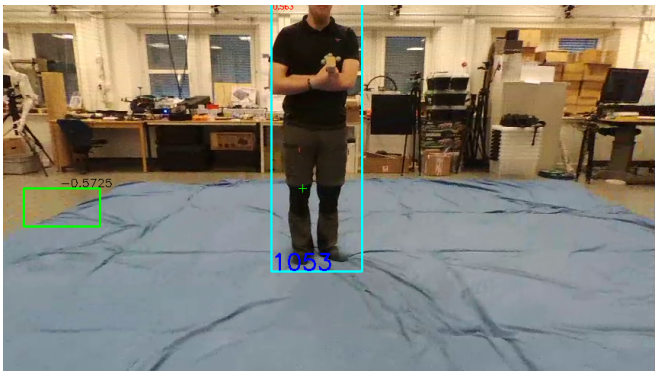


Fig. 4: A visualization of the MAV's vision frame. The light blue box is the detected bounding box. In the top left corner of the bounding box can ΔA be seen. At the bottom of the bounding box is the box's unique ID printed in dark blue. The green box to the left is the ${}^{\mathbb{C}}\alpha_s$. In this example, the ${}^{\mathbb{C}}\alpha_s$ is -0.5725 radians. And ${}^{\mathbb{C}}\alpha_w$ is visualized with a green cross

Algorithm 1 Map bounding boxes to risk Array

Input: Bounding boxes
Output: Risk array ψ

$\psi \leftarrow$ new array$[w_p : 0]$
for all Bounding boxes as B do
  for all $B.C_{py} - w_p/2$ to $B.C_{py} + w_p/2$ as i do
    if $B.\Delta A > \psi[i]$ then
      $\psi[i] \leftarrow B.\Delta A$
    end if
  end for
end for

Algorithm 2 Find minimum span
The method waypointToPixels() is (3) in reverse given an angle it returns the corresponding ${}^{\mathbb{C}}c_p$).

Input: Risk array ψ
Output: Safe heading ${}^{\mathbb{B}}\alpha_s$
Requires: s

$\alpha_s \leftarrow$ waypointToPixels()
$min \leftarrow 2s$
for $o \leftarrow 0 : o < w_p/2 : o \leftarrow o + 1$ do
  $current \leftarrow \sum_{i=waypointToPixels()+o-s}^{waypointToPixels()+o+s} \psi[i]$
  if $current < min$ then
    $\alpha_s \leftarrow o+$waypointToPixels()
    $min \leftarrow current$
  end if
  $current \leftarrow \sum_{i=waypointToPixels()-o-s}^{waypointToPixels()-o+s} \psi[i]$
  if $current < min$ then
    $\alpha_s \leftarrow$waypointToPixels()$-o$
    $min \leftarrow current$
  end if
end for

proposed method. Parrot Bebop 2 weighs 0.5 kg, offering 25 mins of flight time and it is equipped with a forward-looking camera, optical flow and sonar sensor looking down as depicted. The platform provides a WiFi link and all the computations are performed on the ground station computer, equipped with an Intel i7 processor and a Nvidia GTX1060 graphics card, on which the algorithm ran in 8 Hz. The Bebop-Autonomy [2] is used for the estimation of the states and controlling the platform, while the heading of the MAV is corrected based on the proposed method. The looking forward camera horizontal FoV is 80°. The images are still slightly distorted, however this will not affect the overall proposed method. The small distortion can be seen around the edges in Figure 4, which is hard to spot and it should be highlighted that the obstacles with a higher probability for the collisions are usually not in the edges of FoV.

A. Gazebo Simulations

In the simulation evaluations, a single pedestrian was placed in front of the MAV. The goal set point for the MAV is in the back of the pedestrian, where without the correction of the heading, the collision is not avoidable. Figure 5 depicts the movement of the MAV and the location of the pedestrian.

Moreover, Figure 6 depicts the distance between the MAV and a pedestrian in first case. As it can be seen from the obtained results, the MAV successfully avoids the collision and a minimum distance of 0.88m is obtained.

[2]https://bebop-autonomy.readthedocs.io/

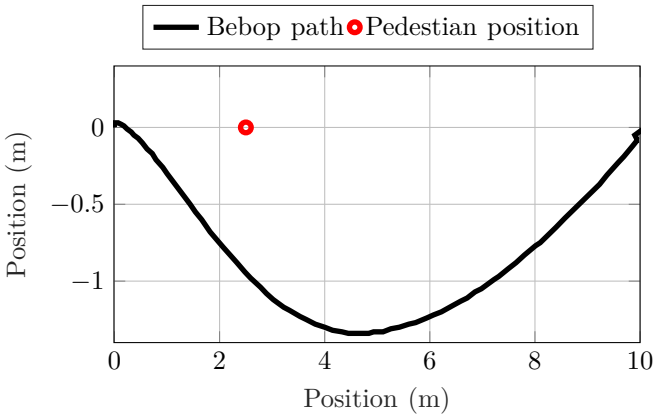


Fig. 5: The obstacle avoidance path (black line) that the MAV took when one pedestrian (red circle) was standing still in the ${}^{\mathbb{B}}\alpha_w$.

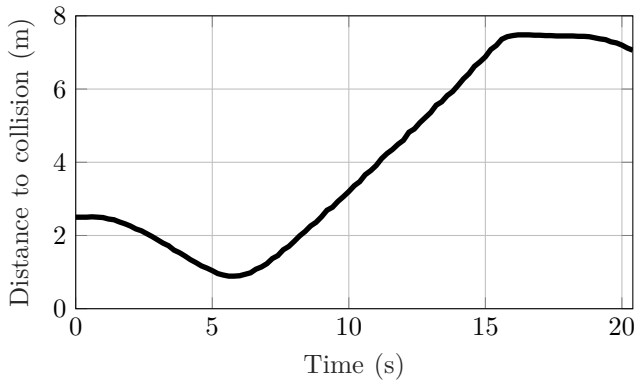


Fig. 6: The distance to the colision d with one pedestrian standing still. In this scenario, the minimum diastase were 0.88m

B. Experimental Evaluation

1) MAV navigation with static obstacle: In the experimental evaluation, the same scenario, as in the Gazebo simulations was considered. Figure 7 depicts the sequence of the obtained images, from the forward looking camera, while the MAV navigates to the desired set point. The center of the green rectangle is the direction of the generated heading command. As it can be seen, the heading is corrected in a way that the pedestrian in the pixel coordinates moves from the center to the left, which results to a collision free navigation.

The distance between the MAV and the pedestrian is depicted in Figure 8, where it is also observed that same performance, as in the case of the simulation results, is obtained and the minimum distance achieved was 0.87m, between the pedestrian and MAV.

2) Time of collision with multiple dynamic obstacles: In this case, it is assumed that the MAV is static and the objects are moving towards the MAV. The main reason of this scenario, is to evaluate the proposed approach for detecting the time of the collision, based on the ΔA of

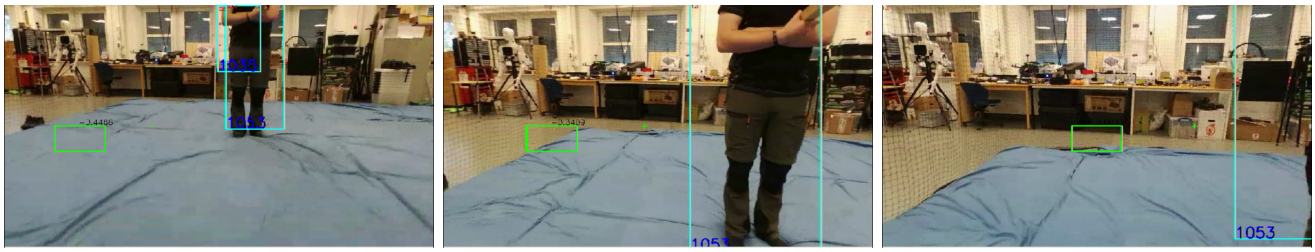


Fig. 7: Sequential onboard images from the MAV, during the real flying test with one pedestrian standing still, where the center of the green rectangle is the direction of the generated heading command.

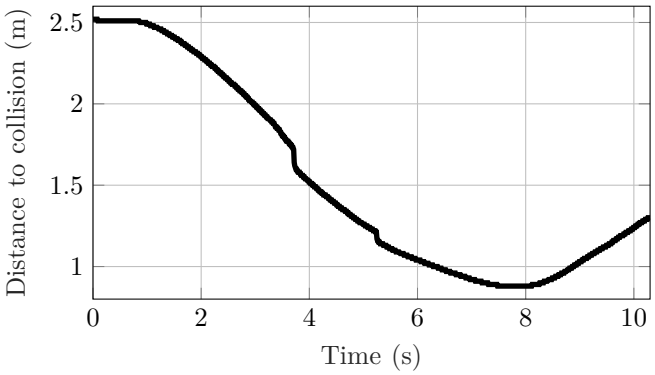


Fig. 8: The distance d between the MAV and the pedestrian durign the experimental flying test, while the pedestrian was standing still.

the objects. The accuracy of the ΔA is highly dependent on the quality of the Bs precision. The Bs precision is affected by both the detection algorithm YOLO and the Kalman filter.

In Figure 10 it is possible to observe how the ΔA is changing when two pedestrians approach a stationary MAV. ΔA is generally small, in comparison to what it could be in extreme cases, as depicted in Figure 10a. The 0.5 steps occur when some part of the detected pedestrian goes outside the image (a safety feature to avoid dangerous behavior).

Figure 10b is a zoom-in on the first part of graph (a). The pedestrians were standing still for about 16 frames, before they started to move. The pedestrians that were standing still are not completely motionless, and this results to small changes in ΔA. When the pedestrians had started to move, there are many factors that affect how the ΔA is changing. From the obtained curves, it is possible to determine which pedestrian has higher probability to reach to the MAV earlier and higher chance of collision. The pedestrian in the left side of the image (Pedesterian 1) has almost always a higher ΔA and as a result it is the first one to reach the MAV.

IV. Discussion

A safety distance ≈ 0.87m is a rather small in most cases, when compared to the 1m distance used for near-collision [23]. When the pedestrian's size is taken into account, the real distance to a collision is closer to 0.5m. This distance is likely to be independent of the pedestrian's size, because the avoidance is based on the edge of the detected obstacle.

The ΔA potential was not tested for scenarios with multiple obstacles, so if ΔA and more areas with dense population of pedestrians/obstacles should be considered. Figure 10 suggests that such effective operation could be possible, however, this needs to be further verified with future work and related experimentation. To increase the passing safety distance, a memory could be also considered that should track the past obstacles that are outside the FoV. At the moment, it is only the pedestrians that have been detected in this article and therefore avoided. This means that any other type of obstacles (walls for example) will not be avoided. One challenge when addressing different objects is the different size and the corresponding different impact on ΔA that this could have. This effect has been also noted in the presented simulations as small pedestrians (e.g children) is at higher risk to be collided with.

V. Conclusions

This work presented a vision-based reactive avoidance planner focusing on performing heading avoidance maneuvers. The method combined state-of-the-art CNN object detection with Kalman filter for object tracking purposes in order to accomplish a higher speed on the object identification. The core component, of the proposed architecture, relies on the translation of the extracted bounding box on the image plane for the heading commands of the aerial platform, by using the camera properties and trigonometric operations. The proposed scheme has been evaluated in the simulation environment Gazebo, as well as in real experimentations with the commercially available MAV Parrot Bebop 2. In all the cases, the MAV successfully avoided collisions with the pedestrians, based on the generated heading commands.

References

[1] D. Joshi. (2019) Drone technology uses and applications for commercial, industrial and military drones in 2020 and the future. Last visited 2020-07-03. [Online]. Available: https://www.businessinsider.com/drone-technology-uses-applications?r=US&IR=T

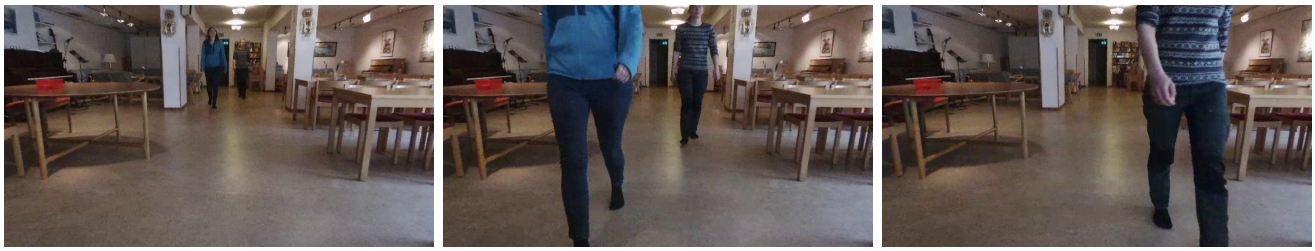


Fig. 9: An image sequence from the experiment performed in Figures 10.

[2] S. Mansouri, C. Kanellakis, G. Georgoulas, D. Kominiak, T. Gustafsson, and G. Nikolakopoulos, "2d visual area coverage and path planning coupled with camera footprints," Control Engineering Practice 75, 1-16, 2018.

[3] S. S. Mansouri, C. Kanellakis, G. Georgoulas, D. Kominiak, T. Gustafsson, and G. Nikolakopoulos, "2d visual area coverage and path planning coupled with camera footprints," Control Engineering Practice, vol. 75, pp. 1–16, 2018.

[4] S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," Control Engineering Practice 74, 118-131, 2018.

[5] S. S. Mansouri, C. Kanellakis, D. Kominiak, and G. Nikolakopoulos, "Deploying mavs for autonomous navigation in dark underground mine environments," Robotics and Autonomous Systems, vol. 126, p. 103472, 2020.

[6] S. S. Mansouri, C. Kanellakis, E. Fresk, B. Lindqvist, D. Kominiak, A. Koval, P. Sopasakis, and G. Nikolakopoulos, "Subterranean MAV navigation based on nonlinear MPC with collision avoidance constraints," IFAC-PapersOnLine, 2020.

[7] M. Eichleay, E. Evens, K. Stankevitz, and C. Parker, "Using the unmanned aerial vehicle delivery decision tool to consider transporting medical supplies via drone," Global Health: Science and Practice, 2019.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99, last visited 2020-09-22.

[9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv, 2018.

[10] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2366–2374, last visited 2020-09-22.

[11] R. T. Collins, "A space-sweep approach to true multi-image matching," in Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996, pp. 358–363.

[12] J. H. Yoo, C. Kim, and D. H. Kim, "Mono-camera based simultaneous obstacle recognition and distance estimation for obstacle avoidance of power transmission lines inspection robot," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 6902–6907.

[13] A. M. Ibrahim, R. M. Hassan, A. E. Tawfiles, T. Ismail, and M. S. Darweesh, "Real-time collision warning system based on computer vision using mono camera," in 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), 2020, pp. 60–64.

[14] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous uav guidance," Journal of Intelligent and Robotic Systems, vol. 57, no. 1-4, p. 65, 2010.

[15] S. M. LaValle, Planning algorithms. Cambridge university press, 2006.

[16] D. Droeschel, M. Nieuwenhuisen, M. Beul, D. Holz, J. Stückler, and S. Behnke, "Multilayered mapping and navigation for autonomous micro aerial vehicles," Journal of Field Robotics, vol. 33, no. 4, pp. 451–475, 2016.

[17] S. S. Mansouri, C. Kanellakis, D. Kominiak, and G. Nikolakopoulos, "Deploying MAVs for autonomous navigation in dark underground mine environments," Robotics and Autonomous Systems, 2020.

[18] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Autonomous obstacle avoidance and maneuvering on a vision-guided mav using on-board processing," in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 2472–2477.

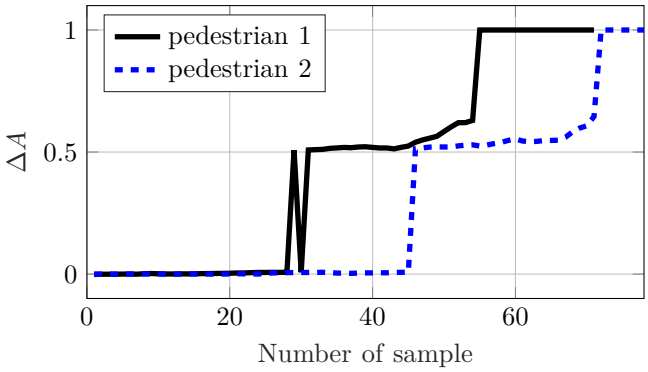
[19] P. P.-Y. Wu, D. Campbell, and T. Merz, "Multi-objective four-dimensional vehicle motion planning in large dynamic environments," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 41, no. 3, pp. 621–634, 2010.

[20] H. W. Kuhn, "The hungarian method for the assignment problem," Naval Resertch Logistic, 2004.

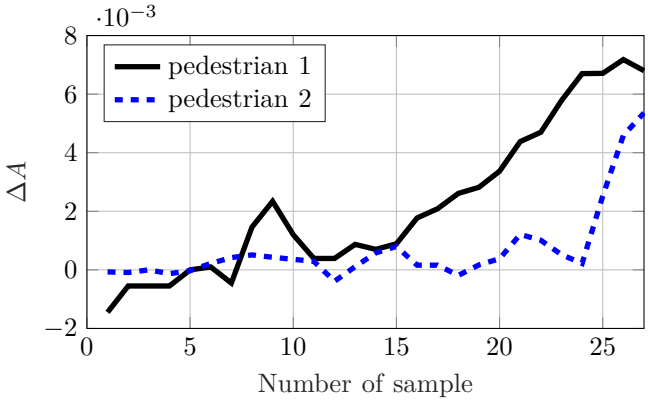
[21] L. V. Santana, A. S. Brandão, M. Sarcinelli-Filho, and R. Carelli, "A trajectory tracking and 3d positioning controller for the ar.drone quadrotor," in 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 2014, pp. 756–767.

[22] Parrot Bebop 2 user manual, Parrot, last visited 2020-09-22. [Online]. Available: \url{https://www.parrot.com/files/s3fs-public/firmware/bebop-2_user-guide_uk_2_0.pdf}

[23] A. Manglik, X. Weng, E. Ohn-bar, and K. Kitani, "Forecasting Time-to-Collision from Monocular Video: Feasibility, Dataset and Challenges," arXiv:1903.09102, 2019, last visited 2020-09-22. [Online]. Available: \url{https://arxiv.org/pdf/1903.09102.pdf}



(a) Normalised area change ΔA from a test with two pedestrians walking towards the MAV from a slightly different distance d.



(b) The first subsection of (a), before the first step (samples 0-26).

Fig. 10: A test of the calculated area change ΔA to show how it changes when two pedestrians are waking towards the MAV. The pedestrians started at slightly different distance d. An image sequence related to the depicted data set can be seen in Figure 9. The left pedestrian in Figure 9 is the pedestrian 1.