Autonomous MAV Navigation in Underground Mines Using Darkness Contours Detection*

S. S. Mansouri[1], M. Castaño[2], C. Kanellakis[1], and G. Nikolakopoulos[1]

[1] Robotics Team
Department of Computer, Electrical and Space Engineering
Luleå University of Technology
Luleå SE-97187, Sweden
[2] eMaintenance Group
Division of Operation, Maintenance and Acoustics
Department of Civil, Environmental and Natural Resources Engineering
Luleå University of Technology
Luleå SE-97187, Sweden

Abstract. This article considers a low-cost and light weight platform for the task of autonomous flying for inspection in underground mine tunnels. The main contribution of this paper is integrating simple, efficient and well-established methods in the computer vision community in a state of the art vision-based system for Micro Aerial Vehicle (MAV) navigation in dark tunnels. These methods include Otsu's threshold and Moore-Neighborhood object tracing. The vision system can detect the position of low-illuminated tunnels in image frame by exploiting the inherent darkness in the longitudinal direction. In the sequel, it is converted from the pixel coordinates to the heading rate command of the MAV for adjusting the heading towards the center of the tunnel. The efficacy of the proposed framework has been evaluated in multiple experimental field trials in an underground mine in Sweden, thus demonstrating the capability of low-cost and resource-constrained aerial vehicles to fly autonomously through tunnel confined spaces.

Keywords: Micro Aerial Vehicles (MAVs) · Vision-based Navigation · Autonomous Drift Inspection · Otsu's Theshold · Moore-Neighborhood Tracing

1 Introduction

The deployment of Micro Aerial Vehicles (MAVs) is gaining more attention in different applications, such as infrastructure inspection [6], underground mine inspection [4], subterranean exploration [9]. etc. since in general MAVs can reduce service and inspection procedures and increase the overall safety of the personnel

* This work has been partially funded by the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 730302 SIMS. Corresponding author's email: sinsha@ltu.se

by navigating and exploring unattainable, complex, dark and dangerous mining underground locations in production areas.

In underground mine tunnels, incidents such as rock falls, blasting, drift expansion, fire detection or propagation, etc. require a frequent inspection for collecting information e.g. images, gas levels, dust levels, 3D models, etc. that can be used for ensuring the safety of the human workers and the overall increase of productivity and the reduction of the related down times in production. However, underground mine tunnels are challenging environments for deploying MAVs due to the lack of illumination, the existence of narrow passages, dust, wind gusts, water drops, conductive dust, etc. Additionally, during the autonomous mission of the MAV it should detect obstacles and avoid collisions for providing a successful and safe autonomous navigation. In general, the MAVs are equipped with high-end and expensive sensor suites to provide stable autonomous navigation, nonetheless, in the case of harsh environments inside mines, there is an ongoing trend to utilize low cost hardware and give emphasis on the algorithmic part of the aerial platform in order to enable the consideration of the MAVs as consumable platforms for the completion of specific missions.

Inspired by the mining needs, a low computational complexity image processing method is proposed for correcting the heading of MAVs in dark underground tunnels. In the proposed method, the image stream from the on-board forward looking camera is utilized for extracting the darkest contour of the image. In the sequel, the center of this area is extracted in order to correct the heading of the MAV towards the tunnel axis. In general, due to the lack of natural and external illumination of the environment, as depicted in Figure 1, the obstacles such as walls are brighter especially when compared to the open areas, a fact that this results to the correction of the MAV's heading towards the open area. Moreover, due to uncertainties in the position estimation, the platform navigates as a floating object with v_x and v_y velocity commands, while the position information on the x and y axes are not used, moreover the potential fields method [4] provides desired velocities $v_{d,x}$, $v_{d,y}$ for avoiding collisions to the walls.

1.1 Background & Motivation

Autonomous navigation in unknown environments is highly coupled with the need for collision avoidance and obstacle detection. Moreover, obstacle detection and navigation, based on vision based techniques for MAVs has received a significant attention the latest years and with a big variety of application scenarios [5]. At the same time, visual stereo or monocular camera systems are able to provide depth measurements for obstacle avoidance, while the obstacle detection methods, based on a monocular camera, in the corresponding literature, are based mainly either on computer vision algorithms or on machine learning methods.

In [10], a mathematical model to estimate the obstacle distance to the MAV was implemented for collision avoidance. However, the method provided poor results at high velocity and low illumination environments. In [13] an obstacle avoidance scheme for MAVs was presented, consisting of three stereo cameras for 360° coverage of the platform's surroundings in the form of pointclouds. How-

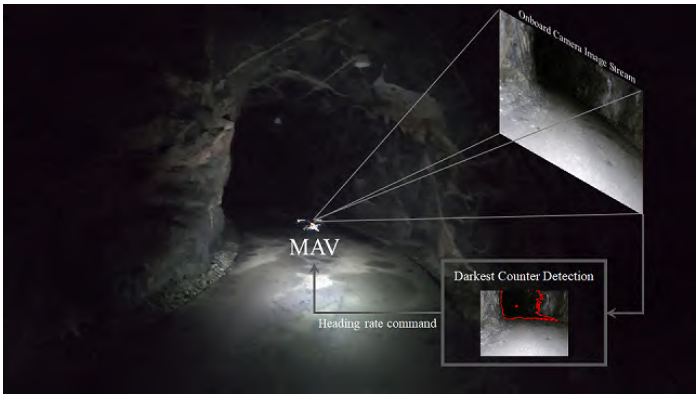

Fig. 1: The proposed approach for extracting the center of the darkest contour from the forward looking camera during autonomous navigation in underground mines. *Supplementary Video:* `https://youtu.be/KNWE0BTpALU`

ever, the proposed method relied on sufficient illumination, landmark extraction and high on-board processing power. In [2], random trees were generated to find the best branch on-line, while building the occupancy map of the perceived environment. This method requires in general a high computation power to process the images.

Moreover, few works using Convolutional Neural Network (CNN) for navigation, such as [1], [12], utilized the image frame of on-board camera to feed the CNN for providing heading commands. These works have been evaluated and tuned in out-door environments and with good illumination. Furthermore, preliminary and limited studies of MAV navigation in an underground mine using CNN was presented in [7], however the method was evaluated in off-line collected data-sets from two underground tunnels, without the MAV in the loop.

In general, the performance of the computer vision-based algorithms mainly relies on the surrounding environment with good distinctive features, good illumination and lighting conditions [10] and on a high computation power, factors that could limit the usage of these methods in real-life underground mine applications. Furthermore, when using machine learning techniques a large amount of data and high computation power for the off-line training of a CNN are required. However, the online use of the trained CNN has lower computation power demands and is applicable for the navigation task.

1.2 Contributions

Based on the aforementioned state of the art, the main contributions of this article are provided in this section. The first and major contribution of this work is the development of the low computational complexity method for providing heading rate commands. The proposed method does not require training data-sets, which is a major limitation of most machine learning methods. To the best

of our knowledge, this is the first work towards MAV navigation in underground tunnels based on centroid extraction of the darkest contour from on-board looking forward image stream. The method requires low computation power and enables online MAV autonomous navigation.

The second contribution, stems from the evaluation of the proposed method and the low-cost MAV in a dark underground tunnel in Sweden, while accurate pose estimation is not available and the platform operates as a floating object. The experimental results demonstrate the performance of the proposed method in underground tunnels without a natural illumination, while the following link provides a video summary of the system in `https://youtu.be/KNWE0BTpALU`.

1.3 Outline

The rest of the article is structured as follows. Initially, Section 2 presents the system architecture. Then, the algorithm for the darkest contour centroid extraction is presented in Section 3. Later, in Section 4 the experimental setup and the extended experimental evaluation of the proposed method in an underground tunnel in Sweden are presented. Finally, the article concludes by summarizing the article and future works in Section 5.

2 System Architecture

The MAV body-fixed frame is $\mathbb{B}$ and the world frame is denoted by $\mathbb{W}$ in the North-West-Up (NWU) frame. The forward looking camera frame is $\mathbb{C}$ and the image frame is $\mathbb{I}$ with unit vector $\{x^{\mathcal{I}}, y^{\mathcal{I}}\}$. The $p_x \in \mathbf{Z}^+$ and $p_z \in \mathbf{Z}^+$ are the pixel coordinates of the image I, and $\mathbf{Z}$ is the integer set of numbers $\mathbf{Z} = \{-\infty, \ldots, -1, 0, 1, \ldots \infty\}$. Figure 2 depicts the coordinate system.

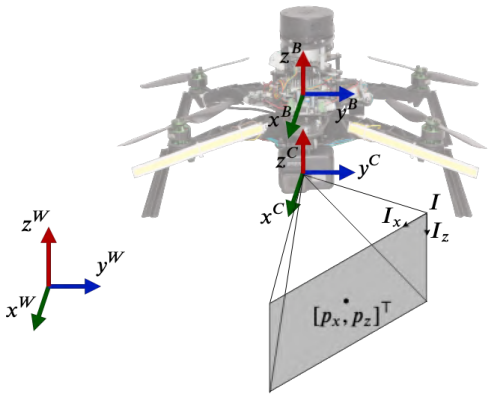

Fig. 2: Coordinate frames, where $\mathbb{W}$, $\mathbb{B}$, $\mathbb{C}$ and $\mathbb{I}$ denote the world, body and camera and image coordinate frames respectively.

The MAV is considered as a floating object, while the state of the system is $X = [z, v_x, v_y, v_z, \phi, \theta]^\top$. The Inertial Measurement Unit (IMU) measurements, which are a_x, a_y, a_z, w_x, w_y, and w_z for the linear and angular accelerations along each axis are passing through an Extended Kalman Filter (EKF) and provides the ϕ and θ. The down-ward optical-flow sensor provides v_x and v_y and the single beam lidar provides altitude z estimation. The image stream from the looking forward camera is denoted by I and the estimated states is indicated with $\hat{}$. Additionally, the potential fields method is implemented to generate proper velocity commands $[v_{d,x}, v_{d,y}]^\top$ to avoid collisions to the walls or obstacles from range measurements $R = \{r_i | r_i \in \mathbf{R}^+, i \in \mathbf{Z} \cap [-\pi, \pi]\}$ of the 2D lidar placed on top of the MAV. The heading rate commands $\dot{\psi}_d$ are provided from the centroid extraction from the darkest contours of the image stream to move towards open spaces. Furthermore, for tracking the desired velocity and altitude commands $[z_{d,x}, v_{d,x}, v_{d,y}]^\top$ the Nonlinear Model Predictive Control (NMPC) [11] is implemented to generate the corresponding thrust and attitude commands $[T_d, \phi_d, \theta_d]^\top$ for the low level controller. The low level controller generates the motor commands $[n_1, \ldots, n_4]^\top$ for the MAV. The overall control structure is presented in Figure 3.

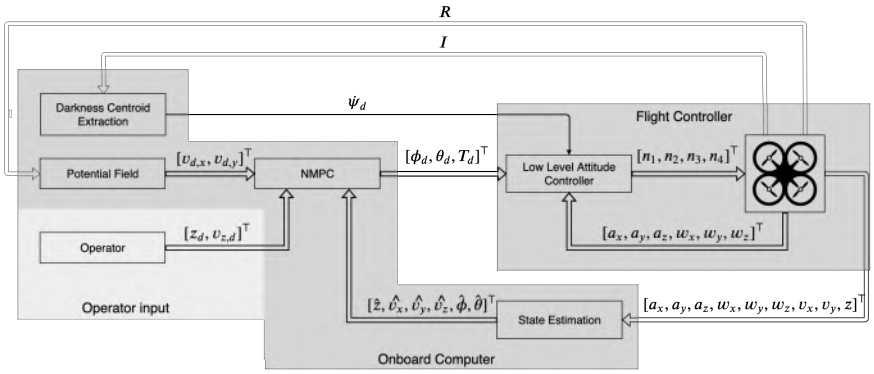

Fig. 3: Control scheme of the proposed navigation module with potential fields for desired velocities generation, while the heading commands are geenerated from the contour extraction method. The NMPC generates thrust and attitude commands, while the low level controller generates motor commands $[n_1, \ldots, n_4]^\top$. The state estimation is based on IMU measurements, optical flow and single beam lidar.

3 Methodology

This section describes the darkness centroid extraction method (see Fig. 3) that receives a camera image as input and delivers the heading correction. The input

to the algorithm is an image I taken by the MAVs forward looking camera. The method can be described as follows.

Darkness Centroid Extraction:

– Input: RGB image acquired by the forward looking camera.
Start
- Step 1. Convert the RGB image to binary image using Otsu's threshold.
- Step 2. Extract contour of background objects using Moore-Neighborhood tracing algorithm.
- Step 3. Identify the darkness in the tunnel as the background object with largest area.
- Step 4. Calculate the heading rate correction based on the centroid of the darkness.

End
– Output: Heading correction.

It is considered that the images are composed by a set of physical objects (walls, pipes, lights) as well as a background. Step 1 is described in Subsection 3.1 and deals with the separation of such physical objects from the background, which is done by finding a threshold that separates the data and uses that threshold to create a binary image. Tracing the contour of the dark background objects is performed by using the Moore-Neighborhood tracing method described in Subsection 3.2. Later, the largest set of background ojects is identified and assumed to be the darkness in the tunnel as described in Subsection 3.3. Finally, the position of the center of the darkness is determined and converted from pixel coordinates to heading rate commands, described in Subsection 3.4.

3.1 Step 1. Converts the RGB image to a binary image using Otsu's threshold

Otsus method for the threshold selection was introduced in [8] and is widely used to reduce a grayscale image to a binary image. It is therefore necessary that the images have been priory converted to a grayscale. In a grayscale image, a single intensity value is assigned to each pixel and the input pictures are represented by a matrix of intensities $I(p(p_x, p_z))$, where $p(p_x, p_z)$ represents a pixel at image coordinates (p_x, p_z).

Otsus method assumes that the image is composed by two classes of pixels, namely the foreground pixels and the background pixels. Otsus method targets the separation of these two classes of pixels that will be named as:

– $\mathcal{T}_1$: class of pixels belonging to the bright foreground.
– $\mathcal{T}_2$: class of pixels belonging to the dark background.

Otsu's threshold is calculated independently for each image, due to the fact that each image has different properties, such as brightness, and they require therefore separate processing for separation between foreground and background. A threshold τ is sought, which determines the belonging of the pixels $p(p_x, p_z)$ to either class. Being a pixel $p(p_x, p_z)$ belonging to the background if the intensity of the pixel $\mathcal{I}(p(p_x, p_z))$ is smaller than the threshold τ.

$$\mathcal{T}_1 = \{p(p_x, p_z) : \mathcal{I}(p(p_x, p_z)) \geq \tau\} \quad ; \quad \mathcal{T}_2 = \{p(p_x, p_z) : \mathcal{I}(p(p_x, p_z)) < \tau\}$$

The goal of Otsu's method is to find the value of the separation threshold τ, which minimizes the intra-class variance $\sigma_w^2(\tau)$:

$$\sigma_w^2(\tau) = \omega_1 \sigma_1^2(\tau) + \omega_2 \sigma_2^2(\tau) \tag{1}$$

where ω_1 and ω_2 are the number of pixels in each class and where $\sigma_1^2(\tau)$ and $\sigma_2^2(\tau)$ are the variance of the elements of each class. Minimizing the intra-class variance is equialent to maximizing the separation betweeen classes.

The calculation of the variances $\sigma_1^2(\tau)$ and $\sigma_2^2(\tau)$ requires the previous calculation of the means of each class. However, Otsu showed that minimizing the intra-class variance in (1) is equivalent to maximizing the inter-class variance $\sigma_b^2(\tau)$, which only depends on the means of the classes μ_1 and μ_2:

$$\sigma_b^2(\tau) = \omega_1(\tau) \cdot \omega_2(\tau) \cdot (\mu_1(\tau) - \mu_2(\tau))^2 \tag{2}$$

The search of the value of τ, which maximizes the inter-class variance can be performed by considering all the different values of the intensity $\mathcal{I}(p(p_x, p_z))$ and choosing the one which maximizes (2). The threshold τ synthesized using Otsu's method is then used to create a binary image $I_b(p(p_x, p_z))$. In this binary image, the pixels, which correspond to elements in the foreground class $\mathcal{T}_1$ get assigned the value 1 and the pixels that belong to the background class $\mathcal{T}_2$ get assigned the value 0. This binary image therefore classifies the pixels as belonging to the set of foreground pixels or to the set of background pixels.

$$I_b(p(p_x, p_z)) = \begin{cases} 1 \; if \; p(p_x, p_z) \in \mathcal{T}_1 \\ 0 \; if \; p(p_x, p_z) \in \mathcal{T}_2 \end{cases} \tag{3}$$

Step 1 is therefore executed as follows:

Step 1. Convert the RGB image to a binary image using Otsu's threshold:

– Input: RGB image acquired by the forward looking camera.
Start

- Step 1.1 Convert the RGB image to a gray scale image represented by an intensity matrix $I(p(p_x, p_z))$. This is performed by eliminating the hue and saturation information from the image, while retaining the luminance.

- Step 1.2 Find Otsu's threshold τ by searching through the different intensity values in $I(p(p_x, p_z))$ until the value that maximizes the interclass variance in (2).
- Step 1.3 Use Otsu's threshold τ to convert the image to binary as stated in (3).

End

– Output: Binary image $I_b(p(p_x, p_z))$.

3.2 Step 2. Finding Boundaries of the darkness using the Moore-Neighbor tracing algorithm

The goal of using Moore-Neighbor tracing is to find all the sets of the interconnected pixels by using a search based on the Moore-neighborhood concept. The details of Moore-Neighborhood tracing algorithm are omitted here for being widely available in the literature. For details, the reader can refer to [3].

The input to the Moore-Neighbor tracing algorithm is the binary image $I_b(p(p_x, p_z))$. Notice that the usual formulation of the Moore-Neighbor algorithm traces pixels with a binary value of 1 (foreground pixels), but in this example the dark objects (value of 0) has to be traced. This can be done e.g. by simply performing the binary negation of $I_b(p(p_x, p_z))$ before applying Moore-Neighbor tracing.

The output of the algorithm is sequence of sets of ordered points $\mathcal{F}_k$ related to each dark background object, being $\mathcal{F}_k = \{P_{k,1}, P_{k,2}, ..., P_{k,N}\}$, where each point $P_{k,i}$ has as coordinates $(x_{k,i}, z_{k,i})$.

3.3 Step 3. Identify the darkness in the tunnel as the background object with the largest area.

The area of each set $\mathcal{F}_k$ can then be calculated using the shoelace formula as:

$$area(\mathcal{F}_k) = \frac{1}{2} \sum_{i=1}^{N-1} (x_{k,i} \cdot z_{k,i+1} - x_{k,i+1} \cdot z_{k,i}) \tag{4}$$

where the last vertex given by the coordinates (x_N, z_N) is the same vertex as the original vertex (x_1, z_1).

The set of pixels with the largest area is considered to enclose the darkness in the tunnel. Last, the abscissa $\hat{x}_k$ of the centroid of the largest dark area is calculated as:

$$\hat{x}_k = \frac{1}{6 \cdot area(\mathcal{F}_k)} \sum_{i=1}^{N-1} (x_{k,i} + x_{k,i+1})(x_{k,i} z_{k,i+1} - x_{k,i+1} z_{k,i})] \tag{5}$$

3.4 Step 4. Calculate the heading rate correction based on the centroid of the darkness

Finally we map the value of $\hat{x}$ to heading rate command.

Step 4. Calculate the heading rate correction based on the centroid of the darkness:

– Input: Centroid of the darkness $(\hat{x}, \hat{z})$

Start

- Step 4.1. $\overline{x} = \frac{\hat{x}}{n}$, $\overline{x} \in [0, 1]$ // Linear mapping $\hat{x} \rightarrow [0, 1]$
- Step 4.2. $\dot{\psi}_d = \frac{\overline{x}-0.5}{2.5}$ // Linear Mapping $[0, 1] \rightarrow [-0.2, 0.2]$ rad/sec

– Output: Heading rate correction $\dot{\psi}_d$.

4 Results

This section describes the experimental setup and the experimental evaluation of the proposed method for sending heading rate commands to the MAV for autonomous navigation in an underground tunnel. The following link provides a video summary of the overall results: `https://youtu.be/KNWE0BTpALU`.

4.1 Experimental Setup

A low-cost quadcopter has been utilized for underground tunnel navigation, which was developed at Luleå University of Technology based on the ROSflight flight controller. The ROSflight Based quad-copter weights 1.5 kg and provides a $10-14$ mins of flight time with a 4-cell 3.7 hA LiPo battery. The Aaeon UP-Board is the main processing unit, incorporating an Intel Atom x5-Z8350 processor and 4 GB RAM. The operating system running on the board is the Ubuntu Desktop 18.04, while Robot Operating System (ROS) Melodic has been also included. The platform is equipped with the PX4Flow optical flow sensor, a single beam Lidar-lite v3 for altitude measurement, two 10 W LED light bars in front arms for providing additional illumination for the looking forward camera and four low-power LEDs looking down for providing illumination for the optical flow sensor. Figure 4 presents the platform, highlighting it's dimensions and the overall sensor configuration.

4.2 Experimental Evaluations

The performance of the proposed method is evaluated in an underground tunnel located at Luleå Sweden with lack of natural and external illumination in the tunnel. The tunnel did not have corrupting magnetic fields, while small particles were in the air. The tunnel morphology resembled an S shape and the dimensions of the area where the MAV navigates autonomously were $3.5(\text{width}) \times 3(\text{height}) \times 30(\text{length})\text{m}^3$. The platform is equipped with a PlayStation Eye camera with a resolution of 640×480 pixels and 10 fps. The front LED bars provide illumination

Fig. 4: The developed ROSflight based quad-copter equipped with 2D and one beam lidars, optical flow, PlayStation camera and LED bars.

of 460 lux from 1 m distance, while for the optical flow sensor low-power LEDs looking down are provided. The desired altitude and velocities for the MAV were set to 1 m, and $v_{d,x} = 0.5\,\mathrm{m/s}$, $v_{d,y} = 0.0\,\mathrm{m/s}$ respectively.

In Figure 5 some examples from the on-board image stream during the autonomous navigation are depicted, while the centroids of the darkest contours are shown. Moreover, it is observed that in case of branches in the tunnel the proposed method cannot recognize them and select the darkest branch as darkest contour or combine both branches.

Fig. 5: Sample images of the on-board forward looking camera, while the boundary of the darkest contour is shown by red color.

5 Conclusions

This article presented a darkness contours detection for MAV navigation in underground tunnels using a forward looking camera. In the proposed method, the image is first converted to grayscale and later a threshold of the image stream from the on-board camera is obtained by Otsu's method, then the image is converted to binary image and boundaries of the darkness areas are extracted. Later on, the largest dark area is selected and the center of this area is used to generate heading commands for the platform. The proposed method is evaluated in dark

underground tunnels in Sweden and provides autonomous navigation, while the heading is corrected towards the center of the tunnel axis.

As expected the method fails to obtain correct heading rate for the platform, in the tunnel with external illumination, as in these cases the tunnel center is not the darkest area.

References

1. Adhikari, S.P., Yang, C., Slot, K., Kim, H.: Accurate natural trail detection using a combination of a deep neural network and dynamic programming. Sensors **18**(1), 178 (2018)
2. Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., Siegwart, R.: Receding horizon next-best-view planner for 3d exploration. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1462–1468 (2016)
3. Blanchet, G., Charbit, M.: Digital signal and image processing using MATLAB, vol. 4. Wiley Online Library (2006)
4. Kanellakis, C., Mansouri, S.S., Georgoulas, G., Nikolakopoulos, G.: Towards Autonomous Surveying of Underground Mine Using MAVs. In: International Conference on Robotics in Alpe-Adria Danube Region. pp. 173–180. Springer (2018)
5. Kanellakis, C., Nikolakopoulos, G.: Survey on computer vision for uavs: Current developments and trends. Journal of Intelligent & Robotic Systems pp. 1–28 (2017). https://doi.org/10.1007/s10846-017-0483-z
6. Mansouri, S.S., Kanellakis, C., Fresk, E., Kominiak, D., Nikolakopoulos, G.: Cooperative coverage path planning for visual inspection. Control Engineering Practice **74**, 118–131 (2018)
7. Mansouri, S.S., Kanellakis, C., Georgoulas, G., Nikolakopoulos, G.: Towards MAV navigation in underground mine using deep learning. In: IEEE International Conference on Robotics and Biomimetics (ROBIO) (2018)
8. Otsu, N.: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics **9**(1), 62–66 (1979)
9. Rogers, J.G., Sherrill, R.E., Schang, A., Meadows, S.L., Cox, E.P., Byrne, B., Baran, D.G., Curtis, J.W., Brink, K.M.: Distributed subterranean exploration and mapping with teams of uavs. In: Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VIII. vol. 10190, p. 1019017. International Society for Optics and Photonics (2017)
10. Saha, S., Natraj, A., Waharte, S.: A real-time monocular vision-based frontal obstacle detection and avoidance for low cost uavs in gps denied environment. In: 2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology. pp. 189–195. IEEE (2014)
11. Small, E., Sopasakis, P., Fresk, E., Patrinos, P., Nikolakopoulos, G.: Aerial navigation in obstructed environments with embedded nonlinear model predictive control. In: 2019 European Control Conference (ECC). IEEE (2019)
12. Smolyanskiy, N., Kamenev, A., Smith, J., Birchfield, S.: Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness. arXiv preprint arXiv:1705.02550 (2017)
13. Valenti, F., Giaquinto, D., Musto, L., Zinelli, A., Bertozzi, M., Broggi, A.: Enabling computer vision-based autonomous navigation for unmanned aerial vehicles in cluttered gps-denied environments. 2018 21st International Conference on Intelligent Transportation Systems (ITSC) pp. 3886–3891 (2018)