Heterogeneous Multiple Vehicles Cooperation Approach for Smart Roads

Abdulla Al-Kaff [1]*, Ángel Madridano[1], Ahmed Radwan[2], Francisco Miguel Moreno[1], Ahmed Hussein[3] and Arturo de la Escalera[1]
[1]Intelligent Systems Lab (LSI), Universidad Carlos III de Madrid, Madrid, Spain
[2]Multi-Robot Systems Lab (MRS), German University in Cairo, Cairo, Egypt
[3]IAV GmbH, Berlin, Germany

Abstract

Intelligent vehicles are equipped with multiple on-board sensors for environment perception. Moreover, with the increasing number of these vehicles on the roads, the more cooperation and coordination among them is becoming more crucial. Accordingly, this paper presents multiple heterogeneous vehicles cooperation approaches, to be used in smart roads to improve driving safety. The heterogeneity aspect is based on the use of Unmanned Aerial Vehicle (UAV) to scout the surrounding of the Unmanned Ground Vehicle (UGV), thus increasing the perception efficiency. Two approaches were proposed for this cooperation, vehicle and pedestrian detection. The algorithms are implemented in the on-board computers. In order to evaluate the proposed approaches, different scenarios were selected and multiple experiments were carried out. The obtained results show the high performance of the algorithm in almost real-time detection and classification, moreover the ability to communicate the outcomes to the UGV, thus improving the automated navigation process for out of the line of sight pedestrians.

1 Introduction

The advances in Intelligent Transportation Systems (ITS) are exponentially improving over the last century. The objective is to provide intelligent and innovative services for the different modes of transportation, towards better, safer, coordinated and smarter transport networks. The ITS focus is divided into two main categories; improve existing components of the transport networks, and develop intelligent vehicles which facilitate the transportation process [1]. In recent years, interest in self-driving vehicles has significantly increased. Accordingly, the necessity of cooperation with all road entities becomes more crucial. The ITS consists of three main entities: vehicles, infrastructure and pedestrians [2].

Accordingly, an intelligent vehicle on the road must cooperate with all road entities, to ensure road safety, especially the safety of pedestrians and other Vulnerable Road Users (VRU). Therefore, the VRU recognition and avoidance in intelligent vehicles are essential tasks. However, due to sensor limitations and several blind spots surrounding the vehicles, researchers are studying different possibilities improving the perception to detect out of the line of sight obstacles.

This paper presents a heterogeneous cooperative approach to tackling the problem of obstacle detection and avoidance with intelligent vehicles. In particular, an Unmanned Aerial Vehicle (UAV) is used to help an autonomous vehicle detect pedestrians located in blind or low visibility areas for the car. To do this, it embarks on the UAV, a monocular camera and a computer, which can process visual information and determine both the position of the vehicle and that of pedestrians. The information generated by the vision algorithms is shared with the vehicle to be incorporated into its perception of the environment through intervehicular communication. In this way, a method is presented that allows providing the terrestrial system with safer navigation.

The remainder of this paper is organized as follows; Section 2 presents the background overview of previous carried-out work in this field. Afterwards, Section 3 introduces the proposed algorithms for detecting and tracking pedestrians and vehicles. In Section 4, the experimental work is illustrated with the selected platforms and scenarios, followed by the discussion of the obtained results in Section 5. Finally, in Section 6 conclusion and future work are summarized.

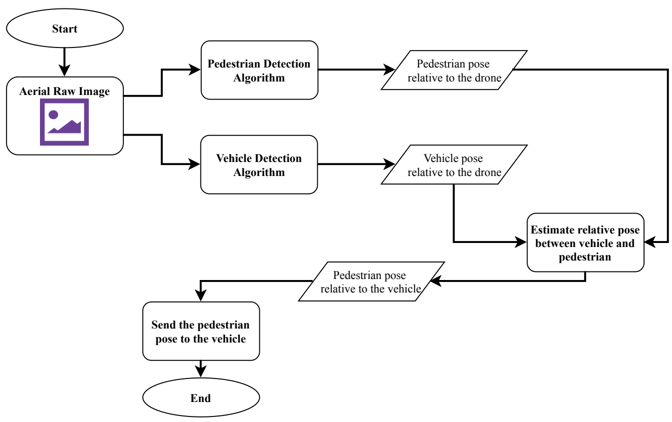


Figure 1: Proposed Approach

*Email address(es): akaff@ing.uc3m.es

2 BACKGROUND OVERVIEW

Pedestrian detection using computer vision is considered as a challenging problem in traffic environments, and most of the solutions presented in this field are based on a common approach, which uses a Histogram of Oriented Gradients (HOG) descriptor, and a Support Vector Machine (SVM) classifier [3]. In [4], a monocular camera is used to detect pedestrians from a UAV by applying a HOG descriptor. Thereafter, based on three image sequences, the distance to the pedestrian is estimated. Other works, such as [5], prefer a Haar-Like based algorithm for pedestrian detection, followed by a template-based tracking.

Furthermore, a detecting and tracking feature-based method, from UAVs was presented in [6]. First, the features are extracted using Harris detector, then the pyramidal Lucas-Kanade (LK) optical flow model and Least Median Square Estimator (LMedS) are used; to classify the movement of the detected features. Then, a Kalman filter and a template matching algorithm are used to track the targets.

Lately, with the advances in deep learning, new methods for object detection and classification are used. For instance, in [7], a deep Convolutional Neural Network (CNN) is trained to classify moving vehicles, showing promising results. In addition, the heterogeneous cooperation between ground and aerial vehicles has been explored in applications such as search and rescue. Recently, a heterogeneous robot collaboration of UGV-UAV has been presented in [8]; in order to collect observations in cluttered urban environments. In this approach, the robot team is able to map the environment while following predefined waypoints. First, the UGV builds the 3D map of the environment using a LIDAR, then, the UAV performs the data gathering. Moreover, the UAV estimates its location by detecting and tracking the UGV.

Furthermore, authors in [9] introduced a method for pedestrians detection and localization based on perception for cooperation between a team of UAV and UGV. The geographic information systems localization system considered that the UGV as a moving landmark for a perspective transformation; to convert the image locations of the targets.

3 PROPOSED APPROACH

In this section, the proposed approach is divided into two algorithms: vehicle and pedestrian detection, as it is shown in Figure 1. These algorithms are explained below.

3.1 Vehicle Detection

The main objective at this point is to be able to detect, by computer vision and in real time, a characteristic pattern located on the roof of an autonomous vehicle, and knowing its position with respect to the UAV while it flies over an area, in which is located the Vehicle and a set of pedestrians.

The procedure of detecting and estimating the position of the autonomous UGV consists of analyzing each frame captured by the camera equipped in the UAV. Once the pattern is located, the algorithm estimates the position of the UGV with respect to the UAV. Once the position of the UGV and the pedestrians with respect to the UAV is estimated, the relative position of pedestrians to the UGV is estimated.

In this work, the UGV is equipped with a pattern, placed on the roof, as it is shown in Figure 2. The detection of this pattern will make it possible to know the position of the vehicle with respect to the UAV, and to be able to know the relative position of the vehicle with respect to the pedestrians located in the vehicle environment.

In the case of the circumference, it is defined by Equation 1 and is described by three parameters: coordinates x and y to the center of the circumference (a, b) and the radius (r).

$$(x-a)^2 + (y-b)^2 = r^2 \tag{1}$$

In this algorithm, Hough is used to detect the circumference of the pattern. Since the flight altitude of the UAV is variable, it can be found in different sizes depending on the height, at which it is found, so no search size will be set for the radius. The only parameter established will be the distance between centres of the different circumferences to be found within the image, so that in each image only one circle is detected, given that in the proposed scenarios there will only be one UGV.

As it is shown in Figure 2, the pattern is formed by a circle of $790mm$ diameter, and X-shape inside.

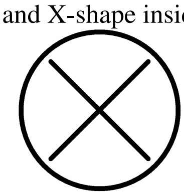


Figure 2: Landing Pattern

The procedure used to try to detect and know the position of the autonomous vehicle consists of analyzing each frame captured by the camera and, through computer vision algorithms, detecting the described pattern on the roof of the vehicle. Once located, it will be possible to know the position of the land vehicle with respect to the UAV.

Once the position of the autonomous vehicle with respect to the UAV and the position of pedestrians with respect to the air vehicle is known, the relative position of persons with respect to the autonomous vehicle may also be known.

The algorithm is based on the localization of circles within the image and, in the subsequent analysis of the regions of interest (ROIs) generated from the circles detected in the frame. The algorithm will be taking frame to frame by trying to carry out positive locations of the model in each of the captured images, and performing a tracking process in case there is no detection in the consecutive frame to a positive detection. The steps within this algorithm for the detection process are detailed below:

Circle detection: it is carried out using the transformation of Hough.

Creating a region of interest: to create a new image from the original frame, extracting the section of the frame in which the circle has been detected, provided that the pattern is completely inside the image. Next, a filtering by size is carried out to ensure the successful completion of the remaining steps of the algorithms. The location of the small-sized pattern would negatively affect all other tasks.

Resetting pattern size: an image of the model to be searched is created that matches the size of the circle being analyzed at this time.

Match between model and detection: Once the pattern is reset, a correlation is established between the model and the ROI created around the detected circle. If the correlation value is above a threshold, which has been set experimentally, it is considered that the detected circle corresponds with the pattern placed on the vehicle, so that it is proceeded to accept that detection (Figure 3) and to calculate the position with respect to the UAV.

If the correlation value is below the set value, the algorithm starts the tracking process as long as the pattern has been detected in the previous frame.

Before moving to a new detection or to carry out the tracking process it is verified that the detection is not incorrect by finding the model rotated with respect to the original, thus repeating this step by making turns of 30^o in the reset pattern. This process is repeated as much 2 times before leaving this step, because when you reach the third iteration the pattern will have rotated 90^o and therefore reach its original position.

Figure 3: Detection of the Landing-Pad

Position calculation: Finally, the calculation of the position is performed. For this, as it is collected in the following equations, it is necessary to know the size of the pattern, both real and in the image, as well as the focal length of the camera. In addition, the value of the x and Y coordinates of the image, as well as the value of the x and Y coordinates of the center of the detected circle, must be used, all in pixels

$$z[m] = \frac{RealPatternSize * FocalLength}{ImagePatternSize * 1000} \quad (2)$$

$$x[m] = \frac{|CenterImgX - CenterDetecX| * z}{Focal\_LengthX * 1000} \quad (3)$$

$$y[m] = \frac{|CenterImgY - CenterDetecY| * z}{Focal\_LengthY * 1000} \quad (4)$$

As indicated above, in the case that there is no detection after a correct location, a tracking process of the pattern is started using the position of the pattern in the previous image. The tracking is carried out using the OpenCV library and follows the steps as follow:

Tracker initialization: to carry out the creation and initialization of the tracker. To do this, it is necessary to pass to the function an image and an area or region of the element to be followed during the tracking process, so this operation is performed by passing to the function the previous frame and the region of interest in which the pattern has been correctly detected. Each time the detection algorithm concludes with a positive result, the MedianFlow tracker is initialized.

Tracker update: to update the tracker; in order to carry out the detection of the model in the current frame, in which the detection has not achieved a satisfactory result. In this way, the tracking updates the location of the pattern from the last known position of the pattern in the new captured image.

Checking the tracking: It establishes a new region of interest obtained from the frame being analyzed and the position of the pattern obtained by the tracker. The pattern is readjusted to the size of the region of interest and again a matching process is performed where the correlation between the ROI obtained from the tracking process and the model being sought is checked. If the correlation is above the established threshold, the tracking is correct and the position of the last detected pattern is updated, whereas if the correlation result is below the threshold, the tracker is considered to have failed and the pattern is lost, so that the algorithm will loop back the detection process until a good location (Figure 4) is obtained.

From the results obtained from the tracking algorithm, it has been decided to set a threshold value in the tracking algorithm lower than the threshold value in the detection algorithm. Which resulting the increament the number of frames in which the location of the vehicle is known.

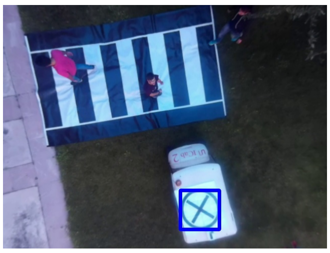


Figure 4: Tracking of the Landing-Pad

Position calculation: If the tracking is correct, the position is calculated using the Equations 2, 3, and 4 previously set.

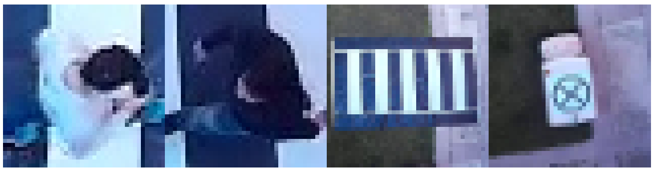
3.2 *Pedestrian Detection*

The main objective at this point is the detection of pedestrian, by HOG in real time, and to locate their positions with respect to the UAV.

HOG descriptor uses a global feature to represent an object rather than a collection of local features. An entire object is represented by a single feature vector, as opposed to many individual vectors representing smaller parts of the object. Typically, HOG descriptor converts an RGB image of size $(width \times height \times 3)$ to a single feature vector n.

Pedestrian detection is done by a camera housed on a UAV. The procedure is to process each frame from the camera stream to detect pedestrians within the image. Detection steps within this algorithm for the detection process are as follows:

1. ***Training HOG descriptor:*** the descriptor training is performed using linear SVM. The training set is a balanced one; number of positive set equal number of negative set. The positive set consists of two hundred 40x40 pixels images of the object of interest (Figure 5a and 5b). The negative set consists of two hundred 40x40 images from the background of the object of interest (Figure 5c and 5d).



(a) (b) (c) (d)

Figure 5: Positive and Negative Training Set

2. ***Setup HOG descriptor:*** as stated earlier in subsection (3.2), the HOG descriptor detection depends on several parameters that were tuned through trial and error. WinStid size is set to 8x8 pixel step between each sliding window location. Padding of size 8x8 was used in the detection. The scale parameter was set to 1.01, this value provided a sufficient factor by which the image is resized at each layer and number of levels in the image pyramid. The hit-threshold proved to be the most important parameter in HOG detection. In the results section, the performance of HOG detection under three different hit-threshold values will be discussed.

3. ***Detections Filtering:*** the detection suffered from noise at the edges of the frame to be processed. A small number of false positives appeared at the edges of the processed frame. These false detection are filtered and omitted from the detection. Furthermore, the detection was filtered against size. Detections with size greater than 60x60 pixels are not counted.

4. ***Position Calculation:*** the positions of detected pedestrians is calculated using Equations 2, 3, and 4. When the positions of both pedestrians and the vehicle are calculated relative the UAV, the relative position between pedestrians and vehicle can be calculated.

3.3 Inter-Vehicular Communication

In this work, an approach for inter-vehicular communication for the broad off-road environment is proposed. The approach objective is to maintain a continuous connection among the vehicles in the system. Accordingly, a Virtual Private Network (VPN) is created, which requires secure connection via the use of authentication keys and certificates. The platforms connect to the VPN via any suitable internet connection using the proper authentication credentials. In reference to platform ROS software architecture, the approach utilizes the multi-master presented in [10]. This enables the platform to have a separate ROS core, thus it is self-dependent and does not operate in a centralized paradigm. The proposed scheme allows the platform to access two networks. One for the vehicular communication schemes, and another for any other types of communication.

Accordingly, for the cooperation among the UAV and the intelligent vehicle, the proposed communication approach assures continuous connection among the vehicles, which was verified in the previous work for cooperative driving in [11].

4 EXPERIMENTAL WORK

The approach presented in previous sections has been validated by performing real tests with UAV and UGV platforms. The following subsections will describe the research platforms used, the scenario designed for the experiments.

4.1 Platforms

On the one hand, the experiments have been carried out on a ground autonomous vehicle under the iCab project (Intelligent Campus Automobile). This vehicle consists of an electric golf cart, which has been modified mechanically and electronically to satisfy the goal of autonomous navigation from one point to another within campus vicinity, as shown in Figure 6a.

(a) (b)

Figure 6: Research Platforms

On the other hand, the UAV platform used in the experiments consists of a 3D printed quadcopter as shown in Figure 6b, with a total weight of 1.5Kg. The autopilot used with this quadcopter is the *Pixhawk*, equipped with GPS, magnetometer, IMU and barometer sensors. For the perception purposes, SJCAM SJ4000 camera is used and mounted on Walkera G-2D gimbal, which provides 640×480 RGB images at 30 frames per second. All the processing is performed on-board by an ODROID-XU4 embedded computer. Finally, both platforms are running Ubuntu 16.04 operating system, and the software architecture has been integrated into ROS middleware. Moreover, in order to avoid a centralized approach and guarantee their operations in an independent way, each vehicle has its own ROS master.

4.2 Scenario

The tests have been performed in outdoor environments, emulating a zebra crossing area, with an autonomous vehicle approaching and several pedestrians crossing. A UAV is hovering while the pedestrian is crossing, detecting both the vehicle and the pedestrians in the area. The relative distance from the vehicle to each pedestrian is computed in the UAV and shared with the UGV to perform safer navigation tasks.

5 Results

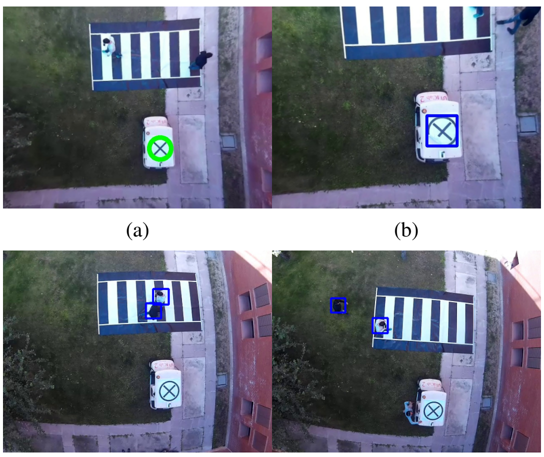
In this section, the results from both vehicle and pedestrian detection are discussed.

5.1 Vehicle Detection Results

Table 1 collects the results obtained in terms of detections of the vehicle are concerned (Figures 7a and 7b). Each of the tables corresponds to each of the two sequences used for the test of the algorithm.

In the first sequence, the lighting conditions are adverse for the perception systems, as the UAV is flying over areas with shadows and light changes. In the second sequence, the tests are performed in a shaded area, where the level of illumination is low, but without light changes.

Both sequences have been tested with three different correlation threshold values, where the vehicle appears is constant and what varies is the number of detections of the vehicle and the false positives.



(a) (b)

(c) (d)

Figure 7: Vehicle and Pedestrain Detection results

In the case of false positives, a precise knowledge of the altitude at which the UAV is flying, would allow applying a filter of size of the circle of the pattern. Knowing the height at which you are flying, and since you know the size of the actual model, you can know the size with which the pattern should be detected in the image, which would allow to carry out a filtering by the size of the detected circles, thus decreasing the likelihood of false positives appearing.

It is shown that slightly decreasing the threshold significantly increases the percentage of correct detection, but also increases the number of false positives, that is, there are times when it is detected as good something it is not the vehicle. In addition, it can be seen how the algorithm improves as the light conditions become more suitable for the vision system.

As for the value of the threshold, if you want to avoid false positives, and whenever a detection is known to the 100% that is being detected the vehicle, it will be necessary to establish a threshold of at least 0.94. This may interest you if you want to carry out control actions on the UAV's flight depending on the detection, either to follow the vehicle or to carry out landing maneuvers on it. If instead, what you want is to get a greater number of detections even if you lose reliability, you can reduce the threshold value below the 0.94 set above.

Although sequence 2 was performed under better lighting conditions, the percentage of detections decreased. This is because the number of frames in which the vehicle appears is very low, which causes any detection failure to be penalized more than in the case of sequence 1. Even so, it can be seen that the number of times the pattern is located is over 90%.

5.2 Pedestrian Detection Results

Table 1shows the results obtained in pedestrian detection (Figures 7c and 7d). Both sequences have been tested with 3 different hit-threshold values. In each sequence the number of frames in which the pedestrian pass zebra crossing area is constant. The table illustrates the variation of the detection rate and number of false positives, depending on the threshold with which the detection works. Detection rate is the ratio of pedestrian detected correctly to the total number of pedestrians detected in the frame, and is defined by Equation 5. False positives can be filtered knowing the height at which the UAV is flying and average pedestrian size as stated earlier.

$$DetectionRate = \frac{TruePositive}{TruePositive + FalseNegative} \quad (5)$$

A slight decrement in the hit-threshold value significantly increases the detection rate, but also increases the number of false positives detected. Furthermore, the results in this table support the relation between algorithm detection improvement and light conditions. In sequence 2, the UAV is maintained at a high altitude which provided a challenge in training and detection. The training file size had to be increased in order to maintain a good detection rate.

6 Conclusions

This paper presents a heterogeneous vehicles cooperation approach to cope with cutting-edge UAVs technology

Table 1: Vehicle and Pedestrian Detection Results

Vehicle Detection				
Sequence	**Threshold**	**Vehicle frames**	**Detection rate %**	**False positive**
Seq. 1	0.92	301	100	65
	0.93	301	100	26
	0.94	301	100	0
Seq. 2	0.92	36	100	7
	0.93	36	91.6	2
	0.94	36	91.6	0

Pedestrian Detection				
Sequence	**Threshold**	**Pedestrian frames**	**Detection rate %**	**False positive**
Seq. 1	0.73	301	98	71
	0.9	301	94.7	32
	0.98	301	89	0
Seq. 2	0.73	308	96.75	43
	0.9	308	94.48	15
	0.98	308	87.01	0

for smart roads. This approach consists of real-time pedestrian and UGVs detection and tracking. These algorithms are studied as a complex and essential task for intelligent vehicles in transportation systems. The proposed algorithms take the advantages of on-board camera for sensing and detecting the two important components in the road (pedestrian and vehicle), and providing information about its position and velocity; to increase the safety in the smart roads.

Different scenarios are evaluated and difficulties have been successfully overcame by means of monocular camera on-board processing, where the pedestrians and UGVs are detected with a total accuracy (>92%).

Future works will focus on the increasing of the environment understanding; which refers to detecting more components in the road, and the combinations of all this information with on-line context information; such as digital maps.

ACKNOWLEDGEMENT

This work was supported by the Comunidad de Madrid Government through the Industrial Doctorates Grants (GRANT IND2017/TIC-7834), and the Spanish Government through the CICYT projects (TRA2016-78886-C3-1-R and RTI2018-096036-B-C21), and the Comunidad de Madrid through SEGVAUTO-4.0-CM (P2018/EMT-4362)

REFERENCES

[1] Fei-Yue Wang, Daniel Zeng, and Liuqing Yang. Smart cars on smart roads: an ieee intelligent transportation systems society update. *IEEE Pervasive Computing*, pages 68–69, 2006.

[2] George Dimitrakopoulos and Panagiotis Demestichas. Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, 5(1):77–84, 2010.

[3] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, Cheng Chen, et al. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, 2011.

[4] Kim Insu and Yow KinChoong. Object location estimation from a single flying camera. In *Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM), 2015 9th International Conference on*, pages 82–88. IARIA, 2015.

[5] Chen Yan-yan, Chen Ning, Zhou Yu-yang, Wu Ke-han, and Zhang Wei-wei. Pedestrian detection and tracking for counting applications in metro station. *Discrete dynamics in nature and society*, 2014, 2014.

[6] Mennatullah Siam and Mohamed ElHelw. Robust autonomous visual detection and tracking of moving targets in UAV imagery. In *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, volume 2, pages 1060–1066. IEEE, 2012.

[7] Y. Qu, L. Jiang, and X. Guo. Moving vehicle detection with convolutional networks in uav videos. In *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*, pages 225–229, April 2016.

[8] Christopher Reardon and Jonathan Fink. Air-ground robot team surveillance of complex 3d environments. In *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*, pages 320–327. IEEE, 2016.

[9] Sara Minaeian, Jian Liu, and Young-Jun Son. Vision-based target detection and localization via a team of cooperative uav and ugvs. *IEEE Transactions on systems, man, and cybernetics: systems*, 46(7):1005–1016, 2016.

[10] Sergi Hernandez Juan and Fernando Herrero Cotarelo. Multi-master ros systems. *Institut de Robotics and Industrial Informatics*, pages 1–18.

[11] Ahmed Hussein, Pablo Marín-Plaza, David Martín, Arturo de la Escalera, and José María Armingol. Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 104–109. IEEE, 2016.