Towards a Mixed-Reality framework for autonomous driving*

Imane Argui[1], Maxime Gueriau[1], Samia Ainouz[1]

***Abstract*— Testing autonomous driving algorithms on mobile systems in simulation is an essential step to validate the models and train the system for a large set of (possibly unpredictable and critical) situations. Yet, the transfer of the model from simulation to reality is challenging due to the reality gap (*i.e.*, discrepancies between reality and simulation models). Mixed-reality environments enable testing models on real vehicles without taking financial and safety risks. Additionally, it can reduce the development costs of the system by providing faster testing and debugging for mobile robots. This paper proposes a preliminary work towards a mixed-reality framework for autonomous navigation based on RGB-D cameras. The aim is to represent the objects in two environments within a single display using an augmentation strategy. We tested a first prototype by introducing a differential robot able to navigate in its environment, visualize augmented objects and detect them correctly using a pre-trained model based on Faster R-CNN.**

I. INTRODUCTION

Autonomous driving (AD) is expected to contribute to the emergence of safe, cost-effective and efficient alternatives to existing transportation systems. AD-enabled vehicles have been used in a wide range of applications: urban driving (cars, taxis, trucks), healthcare industry (ambulances), farming (tractors, irrigators, buggies), industries (forklifts, car crash testing vehicles). However, deploying a fully Autonomous Vehicle (AV) remains challenging as it requires prior training in various and repeated testing conditions. Deep learning algorithms have been used in order to adapt to different situations (including unpredictable or risky scenarios). Such approaches allow to train intelligent vehicles to autonomous navigation, obstacle avoidance, mapping, object detection, etc. Ideally, a fully autonomous vehicle should be able to adapt its actions continuously in response to the changes occurring in its environment. The process of learning and training for autonomous vehicles can be summarized in three steps:

1) Pre-training of the learning model in simulation;
2) Implementation of the pre-trained model on real-life robots/cars;
3) Fine-tuning of model parameters on real vehicles.

Running tests in simulation is an essential step for model verification and validation. During this stage, the vehicle also needs to be trained to face critical, possibly dangerous and/or expensive situations. Even with the latest developments on simulation tools proposing high graphics and numerous options to reproduce real-world conditions, one classical problem persists: the reality gap during the transfer of models from simulation to reality. This is generally caused by discrepancies between conditions in simulation and reality (vehicle dynamics, sensor data, road and weather conditions). On the other hand, experiments on real vehicles and environments are necessary to evaluate and validate the model and obtain realistic results. Nonetheless, real world experiments can be very cost-consuming as they may require expensive materials or special facilities.

A promising alternative for this problem is to combine the advantages of simulation with real-world conditions. This technique is referred to in literature as Mixed-Reality (MR) testing. It was first introduced in [7], where the authors defined the mixed-reality environment as an environment where real world and virtual world objects belong together to a single environment. Figure 1, known as Milgram's Continuum, illustrates a taxonomy for the change of real/virtual environments to yield a new world where real and virtual elements can co-exist. MR can be defined as the space between a purely physical world and purely virtual world.

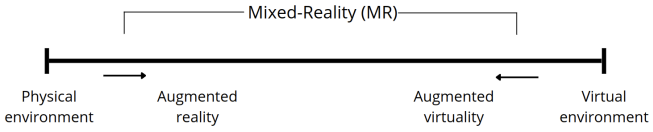


Fig. 1: Virtuality continuum [7]

Training vehicles to autonomous navigation in a mixed-reality environment allows to run experiments repeatedly to train an agent to take the convenient action for a particular situation. Moreover, it can contribute to reducing most of the safety risks by replacing actual obstacles with virtual ones. It also simplifies debugging by testing directly on a real vehicle. Virtual information such as maps and sensor data can be mixed with information collected from the physical environment and visualized in a unique world. In this work, we focus on vision-based autonomous systems as they rely on the detection and recognition of complex roadside information (traffic lights, traffic signs, etc.) and moving users (cars, pedestrians, etc.). The idea is to enable these systems to be trained to face risky situations in a safer environment where virtual elements can be added to their perception. This paper describes a research effort towards the development of a MR framework based on vision. We introduce a strategy to augment a vehicle agent's perception by adding objects modeled in a second (virtual)

*This work was supported by the ANR RAIMo under grant reference ANR-20-CHIA-0021

[1]The authors are with Normandie Univ, INSA Rouen, UNIROUEN, LITIS, 76801 Saint Etienne du Rouvray, France firstname.lastname@insa-rouen.fr

environment. The potential of the proposed framework as well as the results of the developed augmentation strategy are demonstrated in simulation.

II. Related work

Recently, Mixed-Reality simulations have been explored to tackle research problems in different domains. The ambition behind this interest is to leverage existing visual and spatial skills to expand user interaction possibilities. In one of the first applications of MR [3], the authors proposed a MR interface for a user-experience based on a real book. Using an augmented-reality display, the book is augmented with virtual elements, allowing the users to view a shared Mixed Reality object. Later on, MR has been applied for different purposes. Several studies review the use of MR: its applications and current challenges [12], the applications of MR in healthcare higher education [14], and the opportunities of using MR simulations in teaching and learning [9]. In robotics, MR was used first to facilitate prototyping for virtual humanoids amongst real obstacles [13]. MR was applied for mobile robotics initially in Chen *et al.* [4]. The aim of this research was to integrate virtual obstacles in the real world in order to provide a safe simulation environment. Using LiDAR, the authors mixed the data received from the virtual and the real world. More recent works focused on introducing dynamic obstacles for self-driving vehicles: (i) in [15], the authors used LiDAR to detect a virtual pedestrian. The experiments were performed in the real environment augmented by virtual elements. (ii) In [8], the mobile robot is trained in a multi-vehicle multi-lane environment. The real robot was learning autonomous driving with 16 virtual agents simulating a background traffic. (iii) Another application of introducing virtual elements was presented in [2], where the agent followed a simulated bus using ArUco markers. The real robot performed the same actions as its virtual twin.

To the best of our knowledge, MR environments developed and proposed for self-driving vehicles rely on augmenting data received from LiDAR. Although LiDAR has been considered lately as an essential part of self-driving vehicles proving its accuracy and efficiency, it remains unable to interpret roadway information like landmarks and drivable paths, to face difficulties in bad weather conditions, and has expensive initial and maintenance costs. On the other hand, the depth cameras have gained more importance lately. They are less expensive, and create high-definition mapping data by identifying target object shape, appearance, and texture. They not only capture objects but also the landmarks, drivable paths among other data making it a reliable sensor. Moreover, they can gain a high-resolution image of distant objects, thus making them able to see objects that can't be perceived by low-resolution LiDAR. Therefore, the main focus of this work is to develop a mixed-reality framework based on RGB-D cameras where an agent navigates autonomously in an environment and perceives objects present in both virtual and real worlds.

III. Mixed-Reality framework

This paper presents a preliminary work that aims at demonstrating the potential of the framework. The architecture of the framework is illustrated in Figure 2. The main idea is to control a mobile robot in environment A (real world), and enable it to perceive the fusion of two environments (real and virtual). In this paper, we present the first steps of testing on simulation by having two virtual worlds. The tests were established on simulation. To obtain two distinct environments in ROS, we created namespace groups to run in parallel two simulations in the environments A and B. The robot in the environment A is controlled by the differential driving plugin in Gazebo, and its digital twin in the environment B reproduced the same behavior in this environment (section III-B). The augmentation strategy described in section III-C is executed on the images received from camera A and camera B before applying the object detection algorithm to the output result.

A. Simulation tools

The simulation environment used for the framework is the Open-Source robot simulation Gazebo [5], known for its high flexibility and its seamless integration with the middleware ROS [10]. It also already supports a large selection of mobile robots and manipulators, as well as their sensors and control algorithms. Two environments were created, where the robot and its twin can navigate. The communication between different nodes is set up by ROS.

B. Control and digital twining

In order to control the robot, we used existing Gazebo plugins. They give a greater functionality to the robot model and can link to ROS messages and service calls for sensor output and motor input. The plugin used for driving provides a basic controller for 2-wheeled robot in Gazebo. In order to mirror the robot's behavior in the second environment, we used a simplified digital twining method by sending the linear and angular velocity of robot A to robot B.

C. Augmentation strategy

The characteristics of the depth camera are described in URDF files (Unified Robotic Description Format) that can be interpreted in Gazebo. The images received from the camera were converted to `OpenCv` format using the `CvBridge` library. To enable the robot to perceive the fusion of two environments, it is necessary to compare the two images, calculate the difference between them, and add this difference to the initial image the robot will perceive. To achieve this augmentation, we designed an algorithm that enables us to combine the output of the camera A (the initial image where we will add the difference I_A) and the output of the camera B (I_B). union between the two images is applied pixelwise. Since I_B is the output of the camera in the virtual world, we are only interested in the obstacles and objects, the background is irrelevent, therefore, it's equal 0. I_r is the result the robot is going to perceive, as computed in Equation 1.

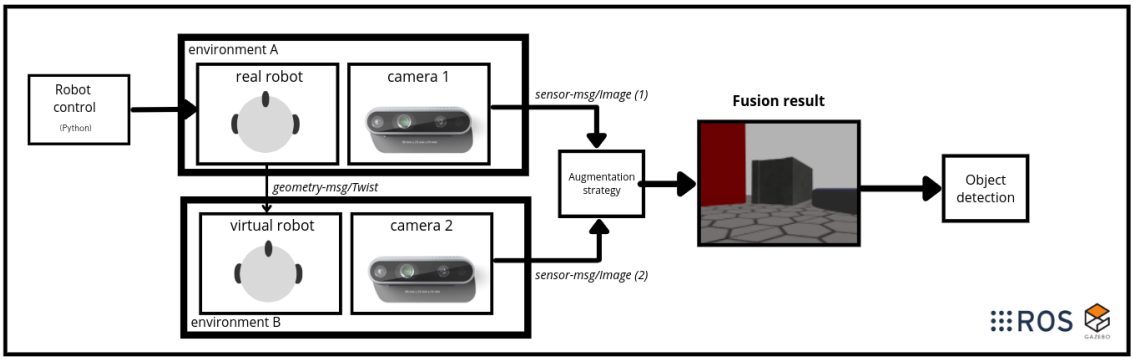


Fig. 2: Overview of the proposed framework enabling the physical coupling of two virtual robots and the fusion of their perception using a vision-based strategy.

$$I_r = I_A \cup (I_B > 0) * I_B \tag{1}$$

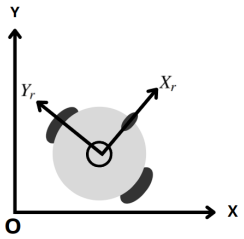
This strategy allows the robot to see, in the same image, objects appearing in the two environments. The pixelar position is assigned to the center of the objects to determine their location in the scene. At this stage of the work, occlusions and light conditions are still not considered.

IV. Experimentations

The MR framework developed in this paper is based on a differential wheeled robot integrating a depth camera. The tools and elements used as well as the results of the tests are described in this section.

A. Experimental setup

The framework presented in this paper is based on a differential wheeled robot illustrated in Figure 3. Its movement is based on two separately driven wheels placed on either sides of the robot. The direction can be controlled by varying the rate of rotation of the wheels. The kinematic model for the robot can be expressed in Equation 2.



$$\begin{pmatrix} \dot{x_r} \\ \dot{y_r} \\ \dot{\theta_r} \end{pmatrix} = \begin{pmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_r \\ \omega_r \end{pmatrix} \tag{2}$$

Fig. 3: Kinematic model of a differential drive mobile robot

$P = (x_r, y_r, \theta_r)$ is the pose of the robot, v_r is the linear velocity and ω_r is the angular velocity of the robot. As mentioned in section II, this work is mainly based on depth cameras, enabling the agent to get an additional understanding of a scene that doesn't require a human intervention. This is especially important for tasks like autonomous navigation and obstacle avoidance. The camera used during this work is the Intel Realsense D435. It's an active stereo depth camera that can get up to 848x480 at up to 90 FPS [1]. It provides distance measurement along with RGB data. This sensor has beneficial characteristics in terms of resolution, frames per second, form factor, weight and price range.

In order to verify and complete the framework, we decided to test object detection algorithms on the results of the fusion. The algorithm used for this task is the FASTER R-CNN [11]. This network is an extension of FAST R-CNN, and as its name suggests, it is faster due to the region proposal network (RPN). The architecture of this network is a combination of two modules: RPN is in charge of generating region proposals and FAST R-CNN ensures the multiple object detection in the proposed regions. Since the goal of this implementation was to test the augmentation strategy proposed, we used the pre-trained network on MS-COCO 2017 [6], a large-scale object detection, segmentation, and captioning dataset. It contains 328.000 images of everyday objects and humans.

B. Tests of the augmentation strategy

In order to verify the augmentation strategy presented in section III-C, we carried out a first test in an offline setup (*i.e.*, using various pre-recorded images). Figure 4 presents the fusion of two images acquired from simulation (image 1 – left, and image 2 – center) and the result of the fusion – right. In the initial images, we place different elements (pedestrian, car, stop sign, etc.) in different positions (each row in Figure 4 represents a different setup). After the application of the algorithm, we can observe that the objects present in the initial images (1 & 2) are placed in their exact position within a single image (result of the fusion). Since these images have the same background, and are both virtual, we decided to go a step further, and test the fusion of one real image and one virtual image. The real images represent a traffic road. Results of this second test are presented in Figure 5. The most-right images (resulting from the fusion) seem to accurately combine the original images (1 & 2). The virtual elements were added in the real images in the same

shape, size and position. The results of the offline testing

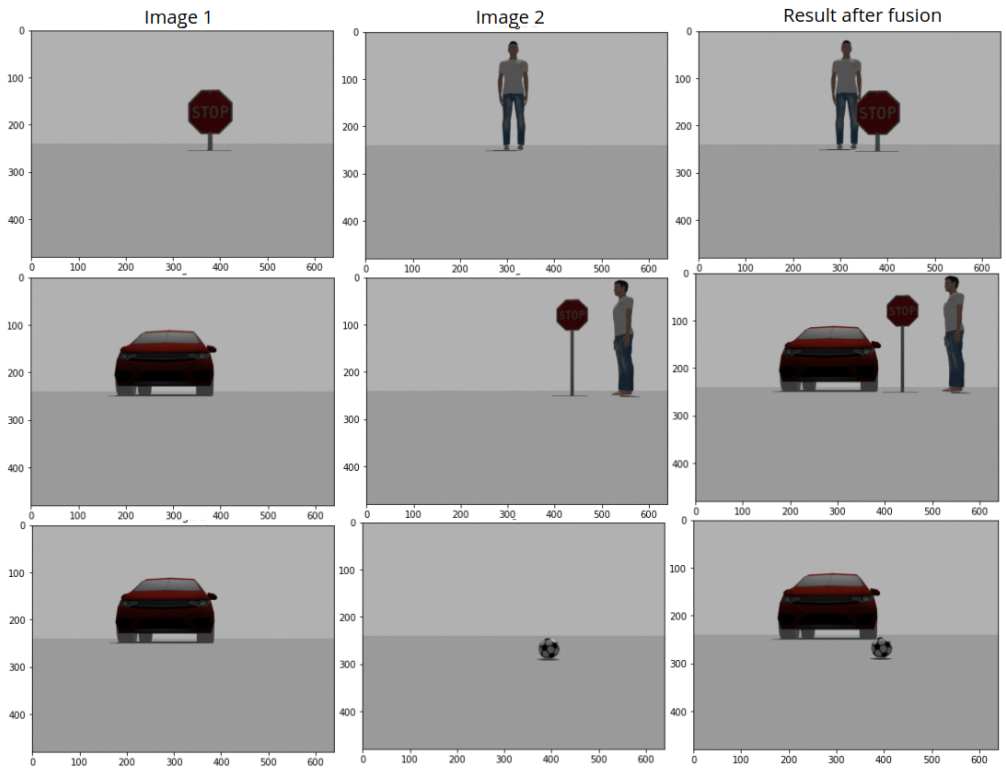


Fig. 4: Results of the fusion of Image 1 (virtual) and Image 2 (virtual)

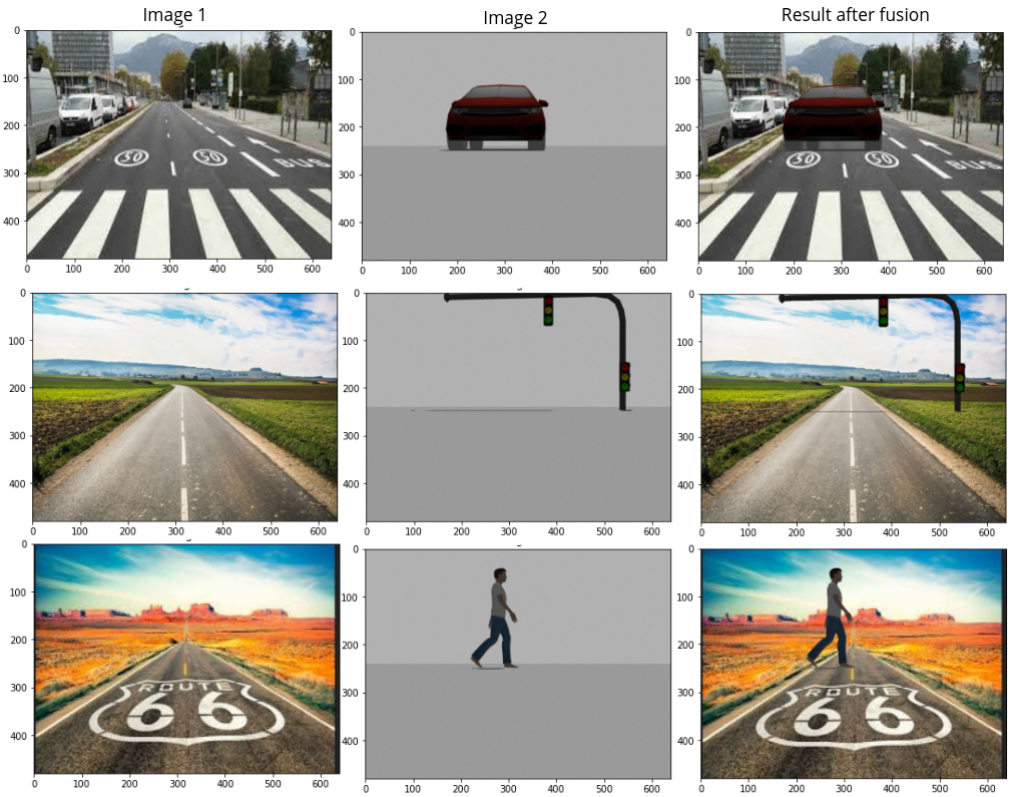


Fig. 5: Results of the fusion of Image 1 (real) and Image 2 (virtual)

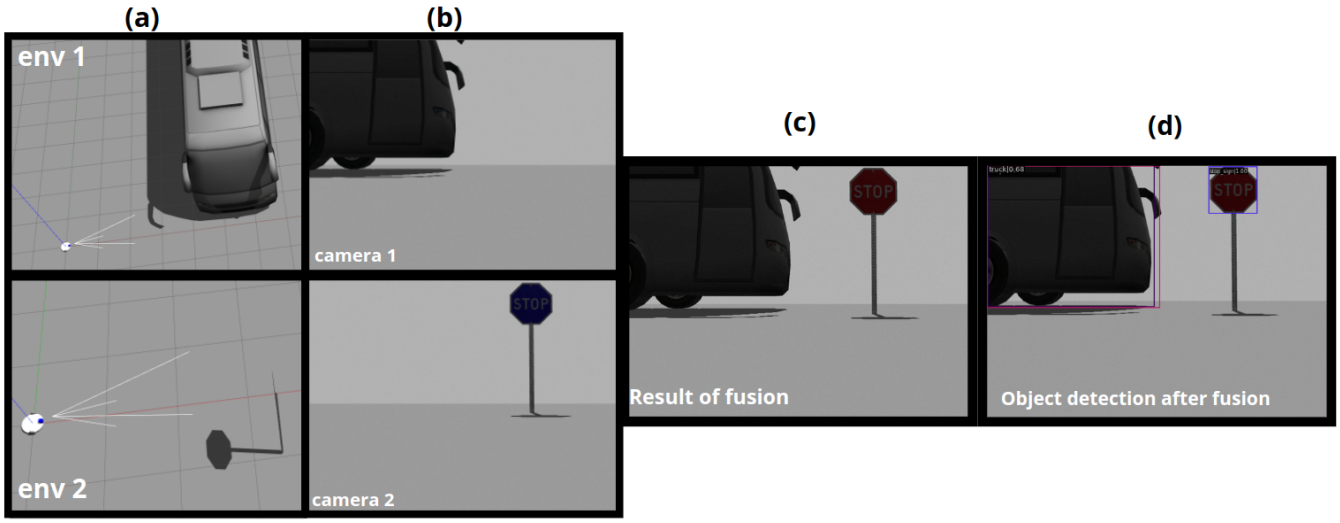


Fig. 6: Overview of the simulation in the RM framework (a) robot's environment (b) robot's perception (c) output of the fusion (d) object detection applied after the fusion.

confirmed that our augmentation strategy works when objects are located at the same position (and possibly overlapping). However, this version of the fusion algorithm does not cast shadows from objects present in one environment to close objects located in the second one, but it does keep the shadows casted by all objects on the ground. The results of the offline tests were encouraging to start online testing.

C. Application of the detection algorithm

Once the environment was set-up and coupled to our RGB-D augmentation strategy, the next step was to complete the framework by implementing an object detection algorithm and run additional on-line tests to check the validity of the fusion. The robot will navigate in the environment A, perceive the fusion of the two environments and start detecting the objects surrounding it (*e.g.*, stop sign, buses, stop light, etc.).

We started testing the algorithm online using Gazebo and ROS. We created a ROS node that subscribes to the data received from camera *A* and camera *B*, applies the fusion strategy, and sends the output to the robot. The object detection algorithm was tested on the resulting images after the fusion. The robot navigates in the environment A, perceives the fusion of the two environments (*A* and *B*), and detects the objects in this image as shown in the figure 6 where it recognizes the bus and the stop sign. To test the robustness of the fusion and the detection algorithm, we decided to overlap two elements. The results are represented in figure 7. Image (a) and image (b) are the initial images. Image (c) is the result of the fusion of the images. The pedestrian and the truck are overlapped. The image (d) is the result of the object detection algorithm applied. It detected the pedestrian and the truck.

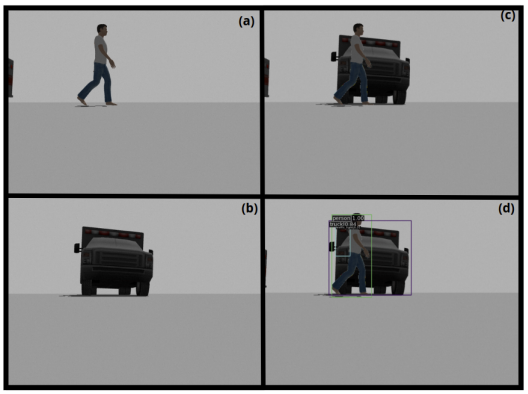


Fig. 7: Results of the fusion and object detection algorithm

V. Conclusion

In this paper, we proposed a preliminary work towards a MR framework for autonomous navigation based on RGB-D cameras. The main motivation behind this research is to fill the gap between tests in simulation and real-world, to limit safety risks and to be able to run an experiment several times in order for the agent to learn and train on a given task. A fusion algorithm has been proposed to mix two virtual environments so that the robot can perceive one single environment with different obstacles. After numerous tests in offline and online settings, an object detection model has been introduced to complete the framework. The particularity of this framework is that it relies on depth-cameras, enabling the robot to detect objects, but also landmarks and drivable paths. We aim in the future to introduce a real robot able to navigate in a real world and interact with physical and virtual obstacles. Additional improvements will be considered for future work: the improvement of digital twining by sending velocity and position to avoid drifts, the introduction of

dynamic obstacles in the virtual environment could help to evaluate the agent's behavior and train it to avoid critical situations like unexpected pedestrian crossing or accidents. Learning-based approaches require a substantial amount of data for learning and training the agent to autonomous navigation and obstacle avoidance, the training of these models in the MR framework will lead to a faster learning by generating a diverse amount of data. Finally, the introduction of a virtual learning agent would enable us to investigate the communication between the agents and the interaction in a multi-agent MR environment.

REFERENCES

[1] M. S. Ahn, H. Chae, D. Noh, H. Nam, and D. Hong. Analysis and noise modeling of the intel realsense d435 for mobile robots. In *2019 16th International Conference on Ubiquitous Robots (UR)*, pages 707–711, 2019.

[2] R. Baruffa, J. Pereira, P. Romet, F. Gechter, and T. Weiss. Mixed reality autonomous vehicle simulation: Implementation of a hardware-in-the-loop architecture at a miniature scale. 10 2020.

[3] M. Billinghurst, H. Kato, and I. Poupyrev. The magicbook: a transitional ar interface. *Computers Graphics*, 25(5):745–753, 2001. Mixed realities - beyond conventions.

[4] I. Y.-H. Chen, B. MacDonald, and B. Wunsche. Mixed reality simulation for mobile robots. In *2009 ICRA*, pages 232–237, 2009.

[5] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004.

[6] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2014.

[7] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Trans. Information Systems*, vol. E77-D, no. 12:1321–1329, 12 1994.

[8] R. Mitchell, J. Fletcher, J. Panerati, and A. Prorok. Multi-vehicle mixed-reality reinforcement learning for autonomous multi-lane driving, 2020.

[9] K. M. Murphy, J. Cash, and J. J. Kellinger. Learning with avatars: Exploring mixed reality simulations for next-generation teaching and learning. In *Handbook of research on pedagogical models for next-generation teaching and learning*, pages 1–20. IGI Global, 2018.

[10] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[11] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[12] S. Rokhsaritalemi, A. Sadeghi-Niaraki, and S.-M. Choi. A review on mixed reality: Current trends, challenges and prospects. *Applied Sciences*, 10(2), 2020.

[13] M. Stilman, P. Michel, J. Chestnutt, K. Nishiwaki, S. Kagami, and J. Kuner. Augmented reality for robot development and experimentation. 01 2005.

[14] T. Stretton, T. Cochrane, and V. Narayan. Exploring mobile mixed reality in healthcare higher education: A systematic review. *Research in Learning Technology*, 26, Nov. 2018.

[15] M. R. Zofka, M. Essinger, T. Fleck, R. Kohlhaas, and J. M. Zollner. The sleepwalker framework: Verification and validation of autonomous vehicles by mixed reality LiDAR stimulation. In *2018 SIMPAR*, pages 151–157.