RoboticsXR: Extended Reality for Robotics Visual Navigation

RoboticsXR: Extended Reality for Robotics Visual Navigation

Petre Ricioppo[1] and Riccardo Enrico[1†] and Jean-Luc Sarvadon[1] and Dario Ruggiero[1] and Elisa Capello[1]

***Abstract*— Virtual Environments and Extended Reality (XR) have transformed artificial intelligence and robotics research by enabling realistic simulations for training and testing. XR enables the seamless integration of digital and physical spaces, offering new opportunities for training and testing autonomous systems. In this work, we propose RoboticsXR, an XR-based framework where a robot operates in a physical lab while images are captured from a virtual environment. This approach fuses real-world motion with synthetic visual data to develop and validate visual navigation algorithms efficiently. RoboticsXR feasibility is assessed using a convolutional neural network trained on synthetic images from NVIDIA Isaac. Performance is compared across different hardware platforms, and using virtual and real camera images. Real-time performance is evaluated for a UAV moving in a confined test area, while virtual images from a synthetic mountain environment are shared with the robotic platform. Results highlight RoboticsXR's potential to extend laboratory testing capabilities, reduce costs, and improve the reliability of vision-based navigation in autonomous robotics systems.**

I. Introduction

Autonomous robotic systems increasingly rely on vision-based algorithms for navigation, obstacle avoidance, and target tracking. Validating these algorithms requires extensive testing under diverse environmental conditions, yet traditional development workflows face a fundamental trade-off: real-world testing is expensive, time-consuming, and constrained by laboratory size, while purely simulated environments, despite recent progress in photorealistic rendering, cannot fully capture the dynamics of a physical platform, including actuator behavior, communication latencies, and onboard computational constraints. This gap is particularly critical for embedded systems, where algorithm performance can differ significantly from workstation-based evaluations.

Extended Reality (XR) offers a compelling middle ground. By allowing a physical robot to operate in a real test area while receiving synthetic visual input from a virtual environment, XR-based frameworks retain the authentic platform dynamics that pure simulation can only approximate, while simultaneously enabling rapid reconfiguration of the perceived surroundings without modifying the physical testing ground. This hybrid approach allows researchers to evaluate vision-based navigation pipelines on actual hardware, under realistic computational loads, across arbitrarily diverse virtual scenarios from indoor laboratories to mountainous terrain, all within a confined physical space.

Virtual production, a growing field within XR, leverages real-time imaging techniques to create dynamic and interactive content. Through high-resolution camera tracking and real-time rendering, virtual production has been widely adopted in the entertainment industry to generate immersive backgrounds and interactive scenes. The fusion of real-world elements with virtual data enables more efficient workflows, cost savings, and enhanced creative possibilities. Virtual environments have been extensively explored for artificial intelligence training, but their application in robotics remains relatively less popular, and most studies refer to telemanipulation or telemanufacturing, and Digital Twin, as shown in [1], [2].

Virtual production has expanded beyond entertainment into industrial and research applications. Industries such as automotive design, architectural visualization, and healthcare training leverage virtual production tools for real-time prototyping and simulation. Technologies such as LED walls, volumetric capture, and physics-based rendering enhance the realism and interactivity of these virtual environments, making them valuable for immersive training and predictive modeling, as shown in [3], [4], [5]. Similarly, Artificial Intelligence (AI) training in virtual environments allows for rapid experimentation, reinforcement learning, and generalization across diverse scenarios. Prominent frameworks such as OpenAI Gym, DeepMind's AlphaFold, and NVIDIA's Isaac Sim enable researchers to train models in highly realistic and customizable virtual spaces before deploying them in the real world. These AI models benefit from large-scale synthetic data generation, which improves their adaptability and robustness when transitioning to physical environments.

Robotic visual navigation is a crucial aspect of autonomous systems, enabling robots to perceive, interpret, and interact with their surroundings in known and unknown environments. Using a combination of computer vision, sensor fusion, and machine learning techniques, robots can identify objects, detect obstacles, and plan optimal paths within complex environments. Advances in deep learning and neural networks have significantly improved scene understanding and real-time decision-making, making visual navigation more robust and efficient. Among the various sensors available, camera-based navigation offers significant advantages, including lower cost, reduced weight, and lower energy consumption compared to infrared, ultrasonic, thermal, or LIDAR sensors. Indeed, as shown in [6], [7], camera-based algorithms provide valuable data for visual navigation. However, to fully leverage this technology and

*This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 117/2023 and DM 630/2024.

*Additional material is available at the RoboticsXR Webpage (https://streamroboticslab.github.io/projects/RoboticsXR/RoboticsXR.html)

[1]The authors are with the Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Torino, 10129, Italy.

[†]R.Enrico is the corresponding author for this submission riccardo.enrico@polito.it.

validate its effectiveness, it is required to have intensive testing. Traditional robotic development often faces a trade-off: either rely on real-world physical testing, which is expensive and time-consuming, or use simulators that lack photorealistic rendering. This lack of realism is a significant hurdle, especially for computer vision applications where accurate visual data is paramount. With advancements in XR, a novel approach emerges that bridges the gap between physical limitations of laboratories and virtual environments. By leveraging new gaming engines (i.e., Unreal Engine), Furthermore, it facilitates rapid iteration and optimization of guidance and control algorithms, streamlining the development process and accelerating the time needed to validate the system in real-world conditions.

To address these challenges, we propose *RoboticsXR*, an XR-based framework in which a robot operates in a confined physical test area while receiving visual input from a virtual environment. This approach enables the fusion of real-world robot motion with synthetic visual data, providing a framework for developing and validating robotic perception, decision-making, and navigation algorithms under realistic hardware constraints.

Unlike existing approaches such as [8], which rely on large projection systems to reproduce the environment, the proposed method directly streams synthetic images to the robotic platform for real-time processing. This eliminates the need for extensive physical infrastructure and enables a more flexible and scalable testing setup, particularly suited for embedded systems.

The RoboticsXR framework is based on a bidirectional communication architecture between a workstation, responsible for rendering the virtual environment, and one or more robotic platforms. The workstation streams images to the platform, while the robot's position and orientation are fed back to update the virtual camera in real time, ensuring dynamic consistency between physical motion and virtual perception. Image transmission performance is evaluated to characterize the system's real-time capabilities.

The main contributions of this work are summarized as follows:

- We propose an XR-based framework that fuses real robot motion with synthetic visual data for hardware-in-the-loop validation of visual navigation algorithms.
- We develop a real-time architecture enabling synchronized interaction between a rendering workstation and embedded robotic platforms.
- We provide a cross-platform evaluation highlighting the impact of computational constraints on visual navigation performance.
- We experimentally compare visual navigation performance using real and virtual camera inputs under different environmental conditions.

The proposed framework is validated in a virtual environment powered by NVIDIA Isaac. A convolutional neural network (CNN), trained on synthetic images, is used to estimate the distance-to-dimension ratio of a target object. Visual navigation performance is evaluated across different hardware platforms and lighting conditions, considering a simple spherical target to isolate perception performance. Results demonstrate the feasibility of extending laboratory testing capabilities for vision-based navigation systems, enabling realistic, repeatable, and resource-efficient experimentation. The proposed approach allows testing under conditions that may be impractical or unsafe to reproduce in real environments, while maintaining real-world execution constraints.

The paper is organized as follows. The simulation and experimental setup is detailed in Section II. In Section III the AI algorithm for distance estimation is described, while results are given in Section IV. Finally, conclusive remarks are reported in Section V.

II. Simulation and Experimental Setup

The RoboticsXR framework (see Figure 1) is based on the integration of a virtual environment with real-time data exchange between a workstation and a robotic processing platform. The workstation runs the virtual environment and controls a virtual camera that simulates realistic imaging conditions. It continuously exchanges data with the robotic platform, which processes the captured image data. The platform shares real-time updates on its position and orientation back to the workstation, allowing for dynamic camera state adjustments. This closed interaction ensures accurate simulation of navigation scenarios in the real environment, enabling precise performance assessment of the visual navigation algorithm.

The images are generated as PNG, we convert them to JPEG format to expedite transmission by reducing file size while preserving adequate quality. A single 1280×720 frame occupies 657.1 kB as PNG and 94.7 kB as JPEG file. The images are captured by the workstation and shared via SSH. Streaming is carried out using the Image Transport ROS 2 library at 30 Hz over WLAN (802.11ac) or Ethernet, with an average received rate of about 29.9 Hz. Image transmission via WLAN shows a delay of 0.025 s and via Ethernet 0.008 s, as observed using ROS 2 CLI Tools. Frame rate, bandwidth, and latency metrics validate the protocol's performance. Then, the image is elaborated by the visual navigation algorithm. In this work, a CNN is trained for the evaluation of target object distance/dimension (d/L) ratio. The visual navigation algorithm is adapted as a ROS2 node and tested on three platforms (specifications are given in Table I): a laptop workstation, a UGV equipped with a workstation-class CPU (without a dedicated GPU), and an embedded computer for UAV (Jetson Orin Nano). Importantly, the algorithm remains identical across all platforms, with no modifications required for any specific hardware.

TABLE I
Platform Specifications

Target Platform	CPU	GPU
Laptop Workstation	Intel Ultra 7	NVIDIA RTX 2000 Ada
Workstation-Class CPU	AMD Ryzen 5 5600G	Integrated - AMD Radeon Graphics
Jetson Orin Nano	Arm Cortex-A78AE	NVIDIA Ampere

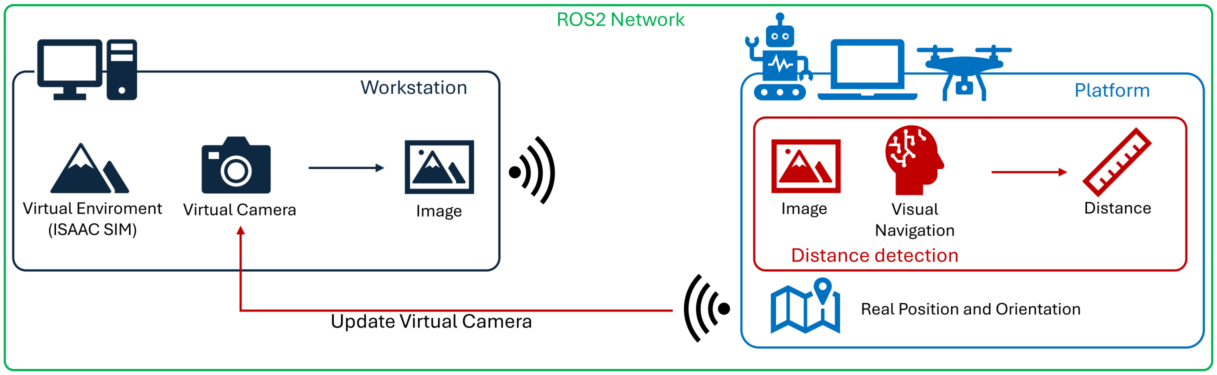

Fig. 1. RoboticsXR framework.

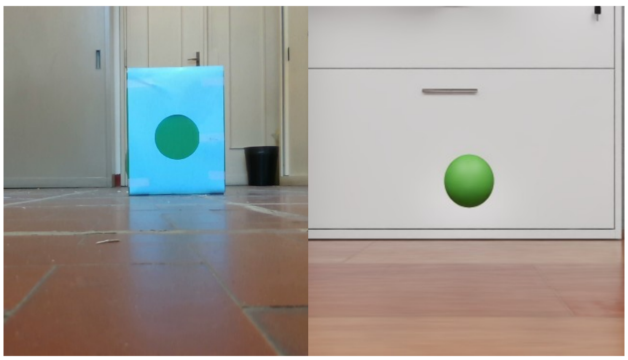

Fig. 2. Real and virtual laboratory environment for visual navigation algorithm performance comparison.

Fig. 3. RoboticsXR for UAV target inspection in mountain environment (see video at the `RoboticsXR Webpage`).

The main purpose of this study is the verification of the RoboticsXR framework real-time implementation. For this reason, simple shaped objects are considered. Indeed, the feasibility of the framework is carried out for the evaluation of d/L ratio of a spherical random object in two virtual environments. The first replicates a robotic laboratory, with the objective to compare static UGV visual navigation performance using real and virtual images (see Figure 2). The UGV's position within its local frame is determined through rotary encoder data and through onboard computations [9]. Position estimation is then utilized to update the camera pose within the Isaac Sim simulator. The second involves a real-time implementation of the RoboticsXR framework, where a UAV operates within a confined test area, while virtual images simulate motion in a mountain environment (see Figure 3). UAV position within its local inertial frame is primarily provided by the PX4 Autopilot's EKF2, which fuses data from multiple sensors. This is augmented with Time-of-Flight infrared distance sensor data. The estimated position is then utilized for updating the virtual camera pose within the Isaac Sim virtual environment, enabling synchronized virtual and physical motion.

III. Visual Navigation Algorithm

This section delineates the methodology employed in developing a machine learning model to evaluate d/L ratio of spherical objects in images. The approach is designed to address the challenges inherent in real-world scenarios, where variations in lighting, perspective, and background complexity can significantly affect image quality.

A. Dataset Description

The training dataset was generated synthetically using the Isaac Sim Replicator extensions, a simulation tool that has been widely recognized in the literature for its ability to accurately reproduce complex environmental conditions, including variations in lighting, atmospheric effects (i.e. fog), material textures, and object physics [10], [11].

In the present study, a unit-diameter sphere served as the target object, rendered against a uniformly textured background. To enhance the dataset's robustness and diversity, several parameters were randomized during generation: the ambient light's intensity and color temperature, the background color, and the camera's relative position with respect to the sphere. Each generated instance is annotated with a two-dimensional bounding box and the corresponding distance of the object from the camera. In Figure 4 is presented an example of the dataset's element.

B. Model Architecture

The machine learning model presented in this study is composed of two separate CNNs configured in a cascade manner. Conventional techniques, such as stereo vision and triangulation, depend on precisely calibrated camera systems

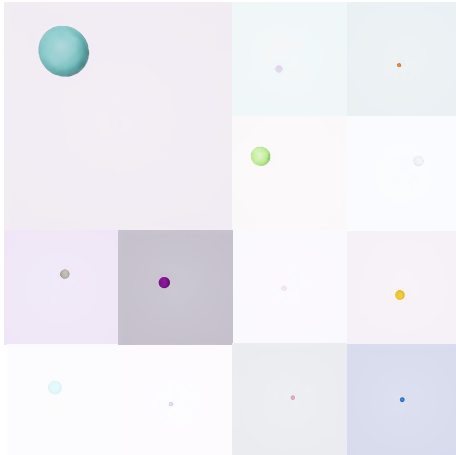

Fig. 4. Training dataset elements example.

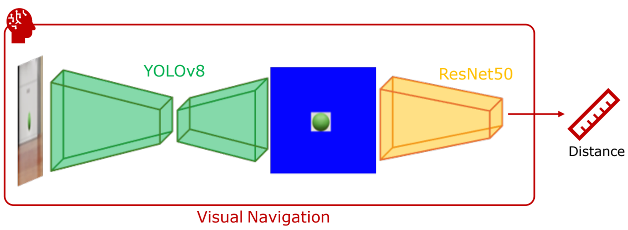

Fig. 5. Visual navigation architecture.

and the exact matching of the corresponding features across the images. These traditional approaches often face challenges under fluctuating lighting conditions, occlusions, or when dealing with objects that lack clear, distinctive features, thereby affecting their overall reliability and accuracy, as seen in [12]. In contrast, CNN-based approaches leverage deep learning to automatically extract hierarchical features from raw image data, thereby overcoming many of the limitations of conventional methods, as discussed in [13], [14].

The architecture of the d/L ratio evaluation machine learning model utilized in this paper is illustrated in Figure 5. The model process consists of three primary steps:

1) A YOLOv8 [15] network is employed to detect the target object within the raw input image.
2) The image undergoes a series of preprocessing steps, wherein all extraneous regions outside the identified bounding box are replaced with a uniform color, thereby effectively isolating the object from its surroundings.
3) The processed image is then passed through a ResNet50-based CNN [16], specifically trained to estimate the object's d/L ratio from the camera.

This preprocessing step not only localizes the target within the image but also reduces the influence of background noise, thereby facilitating more accurate distance prediction.

C. Model Training

The training procedure consists of two main stages: first, training the YOLOv8 network for object detection, and then using the trained YOLOv8 model to preprocess the dataset for training the ResNet50 network for distance prediction.

The dataset used for training comprised 3,000 synthetic images. Initially, the YOLOv8 model is trained to detect the target object in the images. The training is conducted using a batch size of 8, a learning rate of 0.001, and a total of 50 epochs, while all other hyperparameters are set to their default values. Once trained, the YOLOv8 model is used to process the dataset by detecting the target object and replacing all image regions outside the identified bounding box with a uniform color. This step ensures that only the relevant object features are preserved, minimizing the influence of background variations.

After this preprocessing stage, the modified dataset is used to train the ResNet50-based CNN for d/L ratio estimation. The machine learning model is trained with a batch size of 16 and a learning rate of 0.0001 for 100 epochs. The ground truth labels for training are the actual distances of the target object from the camera and the mean squared error is used as the loss function. The Adam optimizer handles weight updates, and early stopping prevents overfitting.

IV. Results

This section presents the results obtained from the proposed methodology, highlighting training effectiveness for accurate evaluation of the d/L ratio of a spherical object in different lighting conditions. Virtual environments offer significant advantages in testing algorithm performance, particularly in generating realistic imaging and diverse lighting conditions. These environments enable precise correlation between measurements and environment conditions, ensuring a high level of accuracy in performance assessment. Additionally, virtual testing allows for the rapid generation of large datasets in a short time, facilitating a thorough evaluation of visual navigation algorithms. This accelerated testing process enhances the reliability of the algorithm while reducing the costs and limits associated with real-world experimentation. In real-world scenarios, visual navigation performance can be influenced by various factors, including moving shadows, reflections, and variations in ambient lighting. These elements can introduce noise and affect the accuracy of d/L ratio evaluation. Therefore, it is crucial to assess the system's performance across different lighting conditions to ensure its robustness. By analyzing the statistical behavior of errors, we establish the error bounds that the real system satisfies during real-world operating conditions. The analysis compares the performance of the system using real and virtual cameras. This evaluation provides insights into the algorithm's reliability and adaptability when deployed in dynamic environments. Then, the RoboticsXR framework performance is evaluated for real-time motion considering a UAV platform.

A. Computational cost

The visual navigation algorithm is tested on three hardware platforms (see Table I). The image processing times reveal that both hardware capabilities and GPU acceleration play a critical role in the algorithm's computational performance. In our tests, the desktop workstation with CUDA achieves the fastest mean processing time of 0.027 s (with a standard deviation of 0.012 s), underscoring the benefits of effective GPU acceleration combined with high-end hardware. In contrast, the UGV system, operating without CUDA on a desktop-grade CPU, yields a mean detection time of 0.131 s with a markedly lower variability (standard deviation of 0.004 s). Interestingly, the Jetson Orin Nano, although employing GPU acceleration, exhibits the slowest performance among the three platforms with a mean detection time of 0.154 s and a standard deviation of 0.038 s. The reduced performance of the Jetson Orin Nano is attributable to its embedded, energy-efficient design. Specifically, its CPU operates at lower clock speeds and comprises fewer cores and threads compared to those found in workstation-class processors.

These findings are significant because they demonstrate that performance evaluations conducted solely on high-powered workstations may lead to overly optimistic estimates of the algorithm's real-time viability. Conversely, when deployed on mobile or embedded platforms and operating at the reference publishing rate for lower-level control, the detection algorithm—though slower than on a workstation—still achieves acceptable performance.

B. Visual navigation in XR environments

Visual navigation performance is evaluated considering the Percentual Error (PE), defined as

$$PE = \frac{|(d/L)_e - (d/L)_t|}{(d/L)_t}, \tag{1}$$

where $(d/L)_e$ and $(d/L)_t$ are the evaluated and the true value, respectively. PE measures the relative deviation between the evaluated and the actual d/L values, providing insight into the accuracy of the navigation system. By leveraging the virtual environment, we efficiently generated a large volume of measurements in a controlled setting, allowing for precise correlation with real-world conditions. This approach facilitates a thorough assessment of the algorithm's performance under varying distances, lighting conditions, and environmental complexities. Figure 6 shows the distribution of PE as a function of the real d/L ratio. The analysis is conducted in the virtual laboratory (see Figure 2) over 6 lighting conditions across 100 different d/L cases. Each data point represents the mean PE over 150 measurements at each d/L ratio, with a standard deviation of around 0.3 %. For large values of the d/L ratio, the evaluation error starts increasing, and also the evaluation accuracy is reduced. For 50 Lx and $d/L > 60$ the algorithm is no longer able to identify the object due to the lack of light. These extensive results enable a comprehensive evaluation of the accuracy of visual-based distance detection algorithm, ensuring statistical robustness and reliability.

Then, performance is compared with real-world imaging, as shown in Figure 7. The evaluation is conducted for the determination of the object distance, considering a dimension

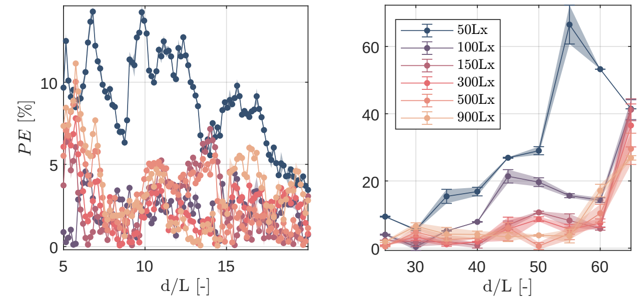

Fig. 6. Visual navigation performance as function of d/L ratio, considering images from the virtual environment shown in Figure 2, and evaluated for different light conditions.

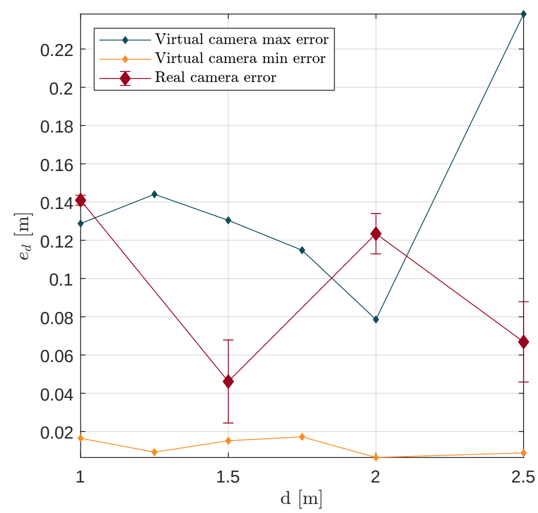

Fig. 7. Comparison between virtual and real camera distance evaluation errors for laboratory scenario.

of $L = 10$ cm. The virtual camera's distance evaluation error (e_d) is determined based on the analysis in Figure 6. The error is modeled using the mean and standard deviation across a discrete set of distances. It accounts for all the lighting conditions, and is expressed as minimum and maximum error bounds. This approach ensures a more precise representation of real-world algorithm uncertainties, which are influenced by factors such as reflections, shadows, and image contrast. The errors observed with the real camera are bounded and closely align with those obtained using virtual images. However, the accuracy of error evaluation may be affected by uncertainties in measuring the true position of the target distance, as well as low contrast between the target and its surrounding background.

Real-time behavior is evaluated considering the integration of the RoboticsXR framework with a UAV in a mountain environment (see Figure 3). The distance evaluation is performed considering a target with $L = 1$ m and $L = 0.1$ m, light condition is daylight, and it allows a good visibility of the target. The error is determined by the evaluated distance and the real distance at the publishing time (accounting for image transmission and computation delay). Distance evaluation error is within the expected range, proving real-

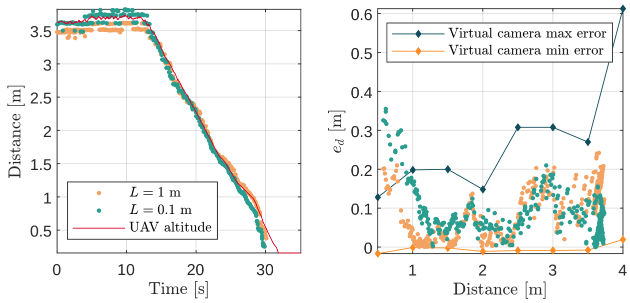

Fig. 8. Visual navigation performance in RoboticsXR framework with real-time UAV motion in a realistic mountain environment.

time feasibility of the RoboticsXR framework in real-time testing.

V. Conclusions

In this paper, we introduced RoboticsXR, an Extended Reality framework that integrates real-world robot motion with synthetic visual data from a virtual environment. This approach enables efficient testing and validation of visual navigation algorithms, reducing the need for costly and time-consuming physical experiments. By leveraging a virtual environment powered by NVIDIA Isaac, we demonstrated the feasibility of using synthetic imaging for distance estimation and navigation, evaluating performance across different hardware platforms and lighting conditions. Comparison with real camera data underlines the need to consider various lighting conditions to assess algorithm robustness against real-world imaging noise.

The results highlight the potential of RoboticsXR to enhance laboratory testing capabilities, providing a controlled yet flexible environment for assessing vision-based navigation systems during real-time operations. This framework facilitates large-scale experimentation, accelerates algorithm development, and improves the transition from simulation to real-world deployment. Future work will focus on expanding the system's adaptability to more complex scenarios. The integration of real and virtual environment elements, creating a hybrid testing framework, will enhance the evaluation of robotic perception and navigation, allowing for more realistic and adaptable test scenario, and bridging the gap between controlled virtual environments and real-world deployment.

References

[1] Y.-P. Su, X.-Q. Chen, C. Zhou, L. H. Pearson, C. G. Pretty, and J. G. Chase, "Integrating virtual, mixed, and augmented reality into remote robotic applications: a brief review of extended reality-enhanced robotic systems for intuitive telemanipulation and telemanufacturing tasks in hazardous conditions," *Applied Sciences*, vol. 13, no. 22, p. 12129, 2023.

[2] B. A. Talkhestani, N. Jazdi, W. Schlögl, and M. Weyrich, "A concept in synchronization of virtual production system with real factory based on anchor-point method," *Procedia Cirp*, vol. 67, pp. 13–17, 2018.

[3] J. Bennett and C. Carter, "Adopting virtual production for animated filmmaking," in *Proceedings of the 7th Annual International Conference on Computer Games, Multimedia and Allied Technology*. Global Science and Technology Forum (GSTF), 2014, pp. 81–86.

[4] J. Carvill, "Integrating actors with digital elements: How technological advancements in the mandalorian (2019) and his dark materials (2019) demonstrate the effectiveness of combining digital and physical elements during the production process," Ph.D. dissertation, Institute of Art, Design+ Technology, 2024.

[5] W. Kerr, C. Jicol, T. S. Haines, and W. Li, "Camera chameleon—the creative impact of tracked tangible interfaces for virtual film pre-production," in *2024 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2024, pp. 1–6.

[6] S. I. Nikolenko, "Synthetic data for deep learning," *Preprint*, September 2019, preprint.

[7] R. Enrico, P. Ricioppo, M. Mancini, and S. Primatesta, "Computer Vision-Based Autonomous Navigation for UAV Vineyard Row Following," in *2024 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*. IEEE, 2024. [Online]. Available: https://iris.polito.it/handle/11583/2995272

[8] M. Peterson, M. Du, B. Springle, and J. Black, "Spacedrones 2.0—hardware-in-the-loop simulation and validation for orbital and deep space computer vision and machine learning tasking using free-flying drone platforms," *Aerospace*, vol. 9, no. 5, p. 254, 2022.

[9] F. Faliero, E. I. Trombetta, and E. Capello, "Data-driven identification based model predictive control for a ground robot," in *2024 11th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE, June 2024, pp. 543–548. [Online]. Available: https://ieeexplore.ieee.org/document/10591559

[10] E. Bonetto, C. Xu, and A. Ahmad, "Grade: Generating realistic and dynamic environments for robotics research with isaac sim," 2024.

[11] R. Singh, J. Liu, K. V. Wyk, Y.-W. Chao, J.-F. Lafleche, F. Shkurti, N. Ratliff, and A. Handa, "Synthetica: Large scale synthetic data for robot perception," 2024.

[12] P. Pinggera, D. Pfeiffer, U. Franke, and R. Mester, "Know your limits: Accuracy of long range stereoscopic object measurements in practice," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 96–111.

[13] M. Vajgl, P. Hurtik, and T. Nejezchleba, "Dist-yolo: Fast object detection with distance estimation," *Applied Sciences*, vol. 12, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1354

[14] Y. Shi, B. Chen, R. Wang, and Y. Zhang, "Structured deep learning based object-specific distance estimation from a monocular image," *International Journal of Machine Learning and Cybernetics*, vol. 14, pp. 1–11, 06 2023.

[15] R. Varghese and S. M., "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 2024, pp. 1–6.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385