Automated Multimodal Data Capture for Photorealistic Construction Progress Monitoring in Virtual Reality

Harvey Stedman
University College London

Ziwen Lu
University College London

Vijay M. Pawar*
University College London

Figure 1: *Left:* Husky robot platfrom with mounted FARO Focus M70 scanner. *Middle:* Rendering from Ricoh Theta Z1 360° camera in the Unity interface. *Right:* Rendering of FARO pointcloud in the Unity interface. In the proposed framework, photorealistic is captured and localised autonomously and can readily be reconstructed in the Unity interface.

Abstract

Construction monitoring is vital for the timely delivery of projects. However, manual data collection and fusion methods are arduous. We propose a framework for autonomous multimodal data collection and VR visualisation. Based on “work-in-progress” results, we demonstrate its capabilities in-the-lab and validate its functionality on a real site. We explore how such a framework could complement construction-centric deep learning and 4D as-built datasets to aid human decision-making using VR.

Keywords: Autonomous mobile inspection, deep learning construction monitoring, laser scanning, human-robot decision making, virtual reality.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality; Computing methodologies—Artificial intelligence—Computer vision—Reconstruction; Computer systems organization—Embedded and cyber-physical systems—Robotics—Robotic autonomy

1 Introduction

The construction industry is important to the world’s economy. It forms 9% of the world’s GDP [4], 6% of the total workforce of the UK [26], and will continue to grow over the coming years. However, the industry faces many challenges in its working practices. Due to the complexity of projects, it can be difficult to know the state of ground conditions, impacting decision making and driving up costs significantly [31]. Safety is also a concern, with the fatality rate of UK construction workers approximately four times the average and the non-fatality rate almost twice the average of all industries [24]. Additionally, productivity in construction is amongst the lowest in industry [11], and it accounts for 11% of the world’s carbon footprint [16]. As the industry evolves, these issues remain key challenges that must be tackled.

The adoption of new technologies and innovations, including artificial intelligence, the internet of things, big data, robotics and virtual reality (dubbed ”Industry 4.0”) presents new opportunities to improve workflows, safety and efficiency. [22]. A key focus of digitising the industry is to improve the management and monitoring of projects. Effective progress monitoring has been identified as critically important in ensuring that projects finish on time and within budget [10]. However, this remains to be a largely manual process [21] and modern working practices require trained professionals to collect data with a variety of cameras and terrestrial laser scanners (TLS), and significant processing time to integrate as-built information from different data sources into Building Information Modelling (BIM) to track progress [15]. Therefore, in practice, the frequency of data collection on site has been reduced to minimise the time and cost associated [18].

As an extension of the above, there are additional questions around how best to visualise and present collected datasets for progress monitoring [17]. The majority of construction monitoring processes are still completed with traditional means of 2D imagery and project reports [25]. However, as construction projects are intrinsically linked to 3D space and professionals in the industry rely heavily on imagery for communication, Virtual Reality (VR) technologies have been identified as an intuitive alternative for progress monitoring [5]. For example, VR applications have shown potential benefits for improving collaboration [6], reducing inspection time of civil infrastructure projects [23] and found as an effective medium for comparing images captured onsite to pre-generated models for progress monitoring [12]. Despite this, VR systems are rarely applied in practice as generating VR visualisations requires significant manual processing [5], which is compounded by the difficulty of gathering site data as detailed above. To remedy these issues, previous works have developed robotic systems integrated with SLAM and navigation and TLS systems as a means to increase the regularity of high-quality data capture on-site [13, 14], and explored combining VR visualisations with data collected from mobile robotic systems [7, 9, 19]. However, there has been limited work

*e-mail: v.pawar@ucl.ac.uk

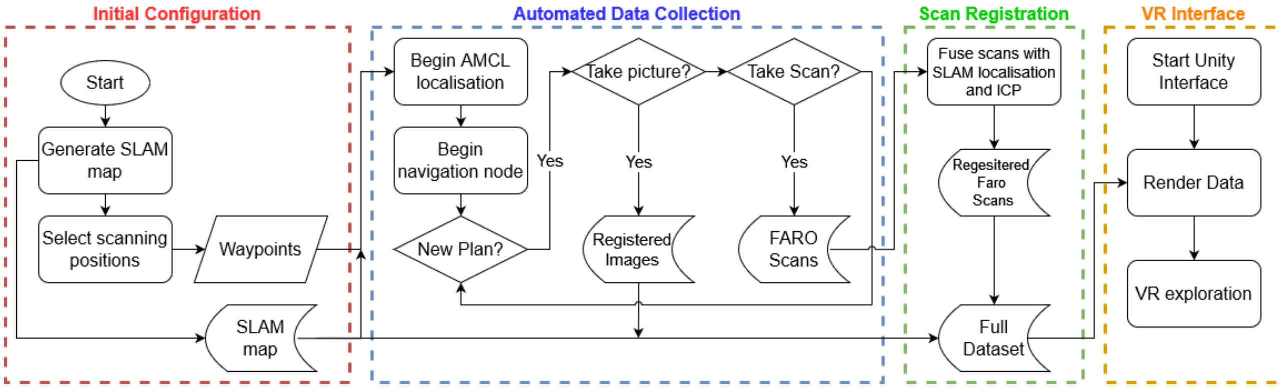


Figure 2: Flowchart of the proposed framework for automated data collection and visualisation in a VR interface. In this work, the Initial Configuration, Automated Data Collection and VR Interface are presented.

investigating how VR interfaces can be integrated with robotic platforms to provide a near real-time immersive automated construction monitoring pipeline.

To address the outlined issues, we present a framework for autonomously capturing multimodal photorealistic datasets, with 360° images from a high-resolution camera and coloured pointcloud scans from a terrestrial laser scanner, using a robotic platform.

The robotic platform localises and navigates around a construction site using a pre-generated 2D laser map and collects data with the onboard sensor payload. Collected data is then spatially registered within the map on capture to simplify post-processing. We also present a Unity interface to visualise datasets as a virtual environment for immersive interaction and exploration with VR devices, enabling immersive construction project monitoring (see Figure 1). We demonstrate the functionality of the robot platform in both lab and onsite conditions and present examples of visualisations from the interface. Additionally, we explore how the full framework can be used to enable further research within progress monitoring, including the generation of 4D datasets of as-built construction progress and the enhancement of collected datasets through the use of construction-centric deep learning models.

2 System Design

The proposed framework comprises a ROS-enabled mobile platform mounted with a survey-grade terrestrial laser scanner, a 360° camera, and a Unity interface built with SteamVR support. The mobile base localises itself in an environment through 2D SLAM, moves to predefined waypoints on the SLAM map and collects data from its onboard sensors payload. Data from the sensor payload is localised and saved within the active SLAM map, simplifying post-processing and dataset organisation. Datasets are then loaded into the Unity interface and redendered at a 1:1 scale, allowing operators to explore the 2D map, images and pointclouds as a digital twin of the construction site using VR devices. An overview of the proposed framework and process flow chart is shown in Figure 2. We present a realised prototype with the initial configuration, automated data collection and VR interface sections of the framework. Although the information required for scan registration has also been collected, currently, the pointcloud alignment is completed manually.

2.1 Hardware

The Clearpath Husky A200 was selected as the mobile base for its ability to traverse the difficult terrain of construction sites. It has a payload capacity of 75kg, a standard battery runtime of 3 hours, and a rugged design capable of off-road travel. The platform has an onboard computer running Ubuntu 18.04 with ROS Melodic, which was used as the master computer of the system, and a SICK LMS1XX 2D lidar (SICK) for SLAM. The default ROS gmapping, AMCL (Advanced Monte Carlo Localisation) algorithm and `move_base` were used for mapping, localisation and navigation.

To collect detailed pointcloud data of the environment, the FARO Focus M70 laser scanner (FARO) was mounted to the platform. With a range of up to 70m and errors of ±3mm below 25m, this style of device is already used in scan to BIM processes [28] and has been integrated into other projects for automated construction progress monitoring [3]. A separate machine (FARO computer) is mounted on the platform as a base station for the FARO during operation.

The FARO scanner must be stationary during data capture, meaning that it cannot be used continuously during navigation. Therefore an additional Ricoh Theta Z1 360° (Z1) camera was also mounted onto the platform to capture high-resolution 360° pictures and collect site data more regularly on the fly than if relying on the FARO scanner alone. Additionally, the Z1 camera allows real-time video data to be analysed during navigation and streamed to remote operators, although currently, this is not implemented on the platform.

2.2 Software Architecture

Both the Z1 and the FARO scanner were integrated into the ROS framework to allow ROS messages to trigger data capture. To communicate with the FARO scanner, the `rosbridge_server` package was adapted to parse messages over WebSocket to the FARO's base computer, which then triggered the FARO SDK to collect a scan and save recorded data.

2.2.1 Localisation and Mapping

Localisation and mapping for robot navigation are achieved through the `ros_navigation` package. A 2D occupancy grid map is created using gmapping by manually controlling the mobile platform traversing through the site. Gmapping was selected due to its robust CPU performance and low map error compared to other 2D SLAM systems [27,30]. The map data is then saved as a YAML file containing data about the generated map, and a PGM file of the actual map. AMCL was used to load the generated map from gmapping and localise the robot within it. After using an initial position provided by the operator, Odometry and SICK LIDAR data from the Husky are used to update the position estimation during operation.

2.2.2 Automated Navigation and Data Capture

Currently, waypoints are required to be defined manually in a config YAML file for the robot to navigate to the desired destination. A custom ROS node was created to organise sequential navigation to waypoints and trigger commands to collect data from the Z1 and FARO, Once AMCL is started to localise the platform within the required map, the node loads the waypoints defined in the config file as an array of robot goal points and then loops through each one sequentially. For each pose, a plan is generated using `move_base`,

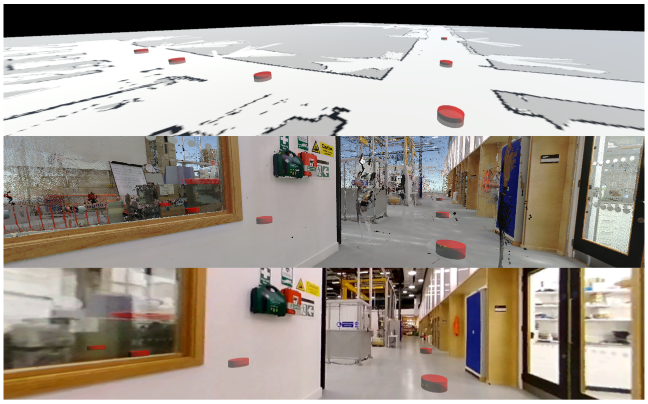


Figure 3: Example of a dataset rendered in the Unity scene and construction map. *Top:* Rendered ImageNodes (red circles) placed on the SLAM map using registered image positions. *Middle:* Pointcloud data rendered over the map. *Bottom:* Image sphere with inverted normals rendered over the map. At all points, the relative position of ImageNodes are represented spatially in the image.

Figure 4: The generated SLAM map and positions of data collection on the construction site. *Green Circles:* Scan positions loaded into the ROS node. *Red Circles:* Positions of 360° images. *Dashed Line:* Trajectory of the robot during navigation

and the current heading difference between the robot's orientation and the desired plan is checked and corrected. Once the orientation is corrected, the Z1 camera starts to take images with the registered pose in the map frame until the next waypoint is reached. Then a FARO laser scan prompt is sent to the FARO computer, and a scan is conducted with the pose of the vehicle registered with the scan. Once the scan finishes, the FARO SDK triggers the robot to proceed to the next waypoint.

As the system uses `move_base`, there are built-in processes in place to address if the platform fails to navigate to the desired waypoints. These have been left as the default, meaning that recovery behaviours are set as clearing the local cost map and rotating on the spot to gather new information. If these fail, then the ROS node will skip to the next waypoint and attempt planning again.

2.3 Unity Interface

Once collected, datasets can be transferred and rendered within the Unity interface. To simplify this process, a standardised data format has been defined in which every dataset contains a separate directory for generated SLAM maps, 360° images and pointcloud scans. Datasets are then placed within a common directory local to the Unity interface to allow for easy retrieval at runtime. When a dataset is loaded in the interface, three separate processes render the corresponding data formats. Figure 3 presents an example visualisation of a dataset rendered in the Unity interface.

MapRenderer The SLAM map's YAML and PGM data are parsed into Unity to generate a visualisation in the scene. The map's origin and resolution are obtained from the YAML file, whilst the height and width of the map are taken directly from the PGM data. A new texture is generated from the PGM data and the scale and position of the texture's GameObject are set based on the map's size, resolution and scale, with ros-sharp being used to swap between ROS and Unity's coordinate systems. The resulting visualisation of the SLAM map has a 1:1 scale and its origin point is centred on Unity's origin (0,0,0).

ImageRenderer The images are retrieved from the data directory and loaded into an array. Looping through each, the image's ID and relative position and orientation are retrieved from metadata. Prefab GameObjects called ImageNodes are instantiated for each and assigned to the corresponding positions on the rendered map. The image is then loaded as a texture and assigned to a field within the ImageNode object to make them easily retrievable at runtime.

After all nodes are generated, the first image (corresponding to the lowest image number) is selected and rendered on a sphere GameObject with inverted texture normals. Using Unity's layer mask, ImageNodes are then rendered over the sphere to represent their location within the image, allowing users to select and load new images with ray tracing from VR controllers.

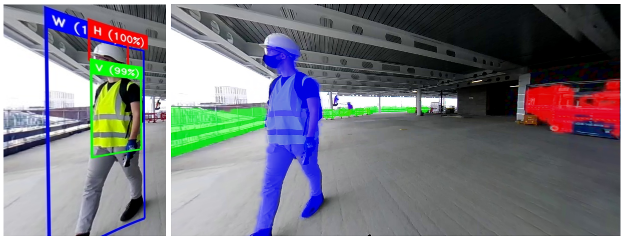
PointcloudRenderer Pointclouds in the dataset are parsed and rendered through a custom pointloud renderer. As both the map and the FARO pointclouds are generated through calibrated LIDAR systems, they both have the correct scale and will therefore align as long as the localisation of the scans is accurate. Currently, this alignment is completed manually. However, automated alignment using the robot's localisation system is possible with the collected data and left for future work. As with the 360° image sphere, ImageNodes are rendered on top of the pointcloud data to intuitively represent where other data has been collected within the digital environment.

Virtual Reality Interface Using the SteamVR Unity plugin, the interface is compatible with common VR devices such as HTC Vive. Simple UI was also created to enable users to swap between image and pointcloud visualisations by enabling and disabling corresponding GameObjects in the interface. Additionally, teleport functionalities were developed to allow users to visualise different ImageNodes and re-position themselves in the pointcloud environment. The full interface allows users to immersively explore collected image and pointcloud data, experiencing a digital 1:1 representation of the construction project at the time of data capture.

3 Preliminary Results

3.1 Construction Site Use Case

After confirming the system's capabilities within lab conditions and a controlled large indoor environment, the robot platform was trialled on a real construction site - an 18-storey office building post phase 3 of construction. Following the same procedure, the robot platform was first manually driven around the environment to build a SLAM map with the gmapping algorithm. Then, specific scanning positions were defined and loaded into the custom ROS node. The platform then navigated around the site and collect data autonomously. The floorplan of the environment was approximately 2500 square meters, and the platform captured 8 FARO scans, totalling 55681124 points, and 44 360° images over a duration of approximately 46 minutes, broken down into 6 minutes total navigation time and 5 minutes per



(a) Pictor-v3 YOLO-V3-A1 estimation.
(b) Deeplabv3 semantic segmentation. *Red*: machinery, *Green*: fence, *Blue*: people

Figure 5: Example of image visualisations from the same perspective in different stages of the construction project.

scan. An overview of the SLAM map and data collection points are presented in Figure 4, and examples of registered images and pointclouds in the dataset are presented in Figure 1.

For performance reference, we used a PC of Intel Xeon E5-2697 V3 with 64GB of RAM and NVIDIA TITAN X (Pascal) GPU for the Unity interface. Each scan was loaded and rendered in seperate threads with a mean size of 9280188 ± 205386 points per scan. Each scan was rendered in the interface with a mean processing time of 159.08 ± 17.92 seconds, and during operation the unity interface had a mean FPS of 86.23 ± 35.34 with performance varying depending on the number of points in view of the camera GameObject.

3.2 Deep Learning Models for Construction Analytics

To demonstrate how the presented framework can complement the applied research of machine learning for construction site analytics, we present example applications of two deep learning models trained and applied to data collected from the robotic platform. In the first instance, we apply the YOLO-v3-A1 model trained with the Pictor-PPE dataset to detect workers and PPE usage [20]. This model is applied directly on the raw collected 360° images to generate 2D bounding boxes over construction workers, hard hats and high visibility safety vests detected in the image, as shown in Figure .

Additionally, we apply a DeeplabV3+ model trained for semantic segmentation and visual understanding of construction sites [29]. This dataset has 859 images containing 1720 instances of 12 classes to generate semantic masks of humans, fences, and 9 classes for construction vehicles and machinery, including excavators, trucks and cranes. For our application, we merge all construction vehicles into a singular class called "machinery" to give a total of 3 classes; humans, fences and machinery. We also augment the dataset with 324 new labelled images collected from our onsite experiments. An example of the model output is presented in Figure 5b.

Through using the robot platform to capture regular datasets on site it is hypothesised that the presented framework will reduce the manual time required to gather training data and provide a means of readily generating new representations of model classes. Further work can also integrate such models into the collected datasets and interface directly by comparing and updating as-built and as-planned BIM models [25]. Additionally, the models can be applied in the planning framework of the platform in real-time to ensure that waypoint trajectories avoid areas with heightened risks based on the robot's perception of its environment [1].

3.3 Generation of 4D As-Built Datasets

Additionally, by reusing the same waypoint missions onsite, the presented framework allows for the simple generation of co-located datasets at different points in time. In future, it is hoped that these can then be integrated into full 4D datasets [2], allowing construction professionals to explore and compare datasets spatiotemporally with VR devices and track as-built progress with project scheduling.

Figure 6: Example of image visualisations from the same perspective in different stages of the construction project.

Example images collected by the robot platform from the same perspective at different stages of the project is presented in Figure 6.

4 Conclusion and Future Work

In this work, a framework for autonomous construction progress monitoring in virtual reality was proposed, and a prototype pipeline was presented. The system was based upon a Clearpath Husky ground-based vehicle with a mounted FARO Focus M70 terrestrial laser scanner for collecting detailed pointcloud data and a Z1 camera for collecting high-resolution 360° images and videos in real-time. The mounted sensors were integrated into the ROS framework, and a custom ROS node was created that was integrated into the ROS `move_base` framework to move to defined waypoints autonomously and trigger data capture on the mounted sensors. Collected datasets were then rendered in a novel VR-enabled Unity interface to allow users to explore and inspect the immersive virtual environments. The proposed system was tuned in a lab environment and tested on a real construction site, and further work on the presented framework could be applied for enhancing research around construction progress monitoring, including using the collected datasets for generating new training data for deep learning models and creating 4D as-built progress visualisations for managing purposes.

It is anticipated that this work will act as a starting point for further work around immersive construction progress monitoring, and there are many exciting areas for future directions. Firstly, the proposed pipeline will be implemented in full, with automatic registration and alignment of pointcloud data. System optimisations will be implemented, including distance and frustum culling during pointcloud rendering to increase FPS during operation. More work is expected in improving Velodyne-type lidar mapping accuracy matching expectations in civil engineering and improving mapping efficiency. Also, legged robots or even drones can be investigated for improving the mobility of the platform on complex terrain in the construction site. The presented interface will be expanded to include further data representations, such as 4D BIM integration with project scheduling to aid with project management [2], and to provide real-time remote connection and teleoperated control of the platform [8]. Additionally, the robotic platform will be improved with better computational power and trained deep learning models will be configured to run in real-time, allowing for waypoints trajectories to be updated based upon the contextual understanding of the environment as investigated in [1]. Further work will explore how construction managers use the presented interface, analysing the effectiveness of virtual reality systems for detecting change and exploring collected datasets.

Acknowledgments

We gratefully acknowledge the support of Oliver Dawkins, Daniel Rennie, The Bartlett Centre for Advanced Spatial Analysis (CASA) and the MACE group for helping orchestrate and organise onsite trials, and valuable comments from Simon Julier on the paper. We acknowledge the funding of ESPRC award no. EP/R026173/ and EP/S031464/1.

References

[1] K. Asadi, H. Ramshankar, H. Pullagurla, A. Bhandare, S. Shanbhag, P. Mehta, S. Kundu, K. Han, E. Lobaton, and T. Wu. Vision-based integrated mobile robotic system for real-time applications in construction. *Automation in Construction*, 96(April):470–482, 2018. doi: 10.1016/j.autcon.2018.10.009

[2] F. Bosché. Plane-based registration of construction laser scans with 3D/4D building models. *Advanced Engineering Informatics*, 26(1):90–102, 2012. doi: 10.1016/j.aei.2011.08.009

[3] W. Cheng, H. Shen, Y. Chen, X. Jiang, and Y. Liu. Automatical acquisition of point clouds of construction sites and its application in autonomous interior finishing robot. *IEEE International Conference on Robotics and Biomimetics, ROBIO 2019*, (December):1711–1716, 2019. doi: 10.1109/ROBIO49542.2019.8961394

[4] D. Crosthwaite. The global construction market: a cross-sectional analysis. *Construction Management and Economics*, 18(5):619–627, July 2000. doi: 10.1080/014461900407428

[5] J. M. Davila Delgado, L. Oyedele, T. Beach, and P. Demian. Augmented and Virtual Reality in Construction: Drivers and Limitations for Industry Adoption. *Journal of Construction Engineering and Management*, 146(7):04020079, July 2020. Publisher: American Society of Civil Engineers. doi: 10.1061/(ASCE)CO.1943-7862.0001844

[6] J. Du, Y. Shi, Z. Zou, and D. Zhao. CoVR: Cloud-Based Multiuser Virtual Reality Headset System for Project Communication of Remote Users. *Journal of Construction Engineering and Management*, 144(2), 2018. doi: 10.1061/(asce)co.1943-7862.0001426

[7] S. Halder and K. Afsari. Real-time Construction Inspection in an Immersive Environment with an Inspector Assistant Robot. pp. 389–379. doi: 10.29007/ck81

[8] S. Halder and K. Afsari. Real-time Construction Inspection in an Immersive Environment with an Inspector Assistant Robot. 3:389–379, 2022. doi: 10.29007/ck81

[9] S. Halder, K. Afsari, J. Serdakowski, and S. DeVito. A Methodology for BIM-enabled Automated Reality Capture in Construction Inspection with Quadruped Robots. *Proceedings of the International Symposium on Automation and Robotics in Construction*, 2021-Novem(November 2021):17–24, 2021. doi: 10.22260/isarc2021/0005

[10] K. C. Iyer and K. N. Jha. Factors affecting cost performance: Evidence from Indian construction projects. *International Journal of Project Management*, 23(4):283–295, 2005. doi: 10.1016/j.ijproman.2004.10.003

[11] Juergen Maier. Made Smarter Review. Technical report, Jan. 2017.

[12] H. Kim and N. Kano. Comparison of construction photograph and VR image in construction progress. *Automation in Construction*, 17(2):137–143, 2008. doi: 10.1016/j.autcon.2006.12.005

[13] P. Kim, J. Chen, and Y. K. Cho. SLAM-driven robotic mapping and registration of 3D point clouds. *Automation in Construction*, 89(December 2017):38–48, 2018. doi: 10.1016/j.autcon.2018.01.009

[14] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf. A flexible and scalable SLAM system with full 3D motion estimation. *9th IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2011*, pp. 155–160, 2011. doi: 10.1109/SSRR.2011.6106777

[15] M. Kopsida and P. A. Vela. A Review of Automated Construction Progress Monitoring and Inspection Methods Geometric Digital Twin Generation for Railway Infrastructure View project A Review of Automated Construction Progress Monitoring and Inspection Methods. (October), 2015.

[16] Laura Cozzi and Tim Gould. World Energy Outlook 2018. Technical report, 2018.

[17] S. Lee and F. Pena-Mora. Visualization of construction progress monitoring. *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, pp. 2527–2533, 2006.

[18] R. Maalek, J. Ruwanpura, and K. Ranaweera. Evaluation of the State-of-the-Art Automated Construction Progress Monitoring and Control Systems. pp. 1023–1032, 2014. doi: 10.1061/9780784413517.105

[19] S. Mortezapoor, C. Schonauer, J. Ruggeberg, and H. Kaufmann. Photogrammabot: An Autonomous ROS-Based Mobile Photography Robot for Precise 3D Reconstruction and Mapping of Large Indoor Spaces for Mixed Reality. *Proceedings - 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, VRW 2022*, pp. 101–107, 2022. doi: 10.1109/VRW55335.2022.00033

[20] N. D. Nath, A. H. Behzadan, and S. G. Paal. Deep learning for site safety: Real-time detection of personal protective equipment. *Automation in Construction*, 112(July 2019):103085, 2020. doi: 10.1016/j.autcon.2020.103085

[21] R. Navon. Automated project performance control of construction projects. *Automation in Construction*, 14(4):467–476, 2005. doi: 10.1016/j.autcon.2004.09.006

[22] C. Newman, D. Edwards, I. Martek, J. Lai, W. D. Thwala, and I. Rillie. Industry 4.0 deployment in the construction industry: a bibliometric literature review and UK-based case study. *Smart and Sustainable Built Environment*, 10(4):557–580, Jan. 2020. Publisher: Emerald Publishing Limited. doi: 10.1108/SASBE-02-2020-0016

[23] M. Omer, L. Margetts, M. Hadi Mosleh, S. Hewitt, and M. Parwaiz. Use of gaming technology to bring bridge inspection to the office. *Structure and Infrastructure Engineering*, 15(10):1292–1307, 2019. doi: 10.1080/15732479.2019.1615962

[24] I. Polanowski. Construction statistics in Great Britain, 2021. p. 24, 2021.

[25] F. Pour Rahimian, S. Seyedzadeh, S. Oliver, S. Rodriguez, and N. Dawood. On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning. *Automation in Construction*, 110(August 2019):103012, 2020. doi: 10.1016/j.autcon.2019.103012

[26] C. Rhodes. Construction industry: statistics and policy. BRIEFING PAPER 01432, House of Commons Library, Dec. 2019.

[27] J. M. Santos, D. Portugal, and R. P. Rocha. An evaluation of 2d slam techniques available in robot operating system. In *2013 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pp. 1–6. IEEE, 2013.

[28] Q. Wang, J. Guo, and M. K. Kim. An application oriented scan-to-bim framework. *Remote Sensing*, 11(3), 2019. doi: 10.3390/rs11030365

[29] Z. Wang, Y. Zhang, K. M. Mosalam, Y. Gao, and S. L. Huang. Deep semantic segmentation for visual understanding on construction sites. *Computer-Aided Civil and Infrastructure Engineering*, 37(2):145–162, 2022. doi: 10.1111/mice.12701

[30] R. Yagfarov, M. Ivanou, and I. Afanasyev. Map comparison of lidar-based 2d slam algorithms using precise ground truth. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1979–1983. IEEE, 2018.

[31] X. Zhang, N. Bakis, T. C. Lukins, Y. M. Ibrahim, S. Wu, M. Kagioglou, G. Aouad, A. P. Kaka, and E. Trucco. Automating progress measurement of construction projects. *Automation in Construction*, 18(3):294–301, 2009. doi: 10.1016/j.autcon.2008.09.004