Real-Time Process-Level Digital Twin for Collaborative Human-Robot Construction Work

X. Wang[a], C. J. Liang[a], C. C. Menassa[a,b], and V. R. Kamat[a.b]

[a]Department of Civil and Environmental Engineering, University of Michigan, USA
[b]Robotics Institute, University of Michigan, USA

wangix@umich.edu, cjliang@umich.edu, menassa@umich.edu, vkamat@umich.edu

Abstract -

Widespread use of autonomous robots in on-site construction has been limited because it is impractical to preprogram robots to perform quasi-repetitive tasks due to the relatively loose work tolerances and deviations of as-built work from the project design. Robotization of field construction work must thus be conceived as a collaborative human-robot endeavor capable of planning and improvising during the performance of construction tasks. Although humans can control robot motion through teleoperation, it is often impractical due to the range of a robot's motion and associated safety issues arising from heavy or large construction materials. An intuitive and safe bi-directional interface is thus needed to enable construction robots to seamlessly interact with and partner with human co-workers. This paper proposes a framework that allows human-robot interaction and collaboration within a real-time, process-level, immersive virtual reality (VR) digital twin that is created by combining the as-designed BIM model and the evolving as-built workspace geometry obtained from on-site sensors. Humans can use the digital twin to remotely demonstrate a task plan to the robot. The robot understands the communicated objectives and plans its motion to complete the task, which is communicated back through the system for human evaluation and approval before the robot executes the task. A case study involving imperfect rough carpentry (i.e., stud framing) and a 6DOF KUKA drywall-installing robot arm is conducted to demonstrate and evaluate the digital twin system.

Keywords -

Improvisation; Digital twin; Virtual reality; Human-robot interaction

1 Introduction

The construction industry is one of the largest sectors of the economy, accounting for up to 13% of GDP worldwide [1]. However, as one of the most labor-intensive industries, the construction industry is suffering from shortage and aging of the labor force [2, 3]. On one hand, the construction site is unstructured and dynamic. On the other hand, the construction work imposes considerable physical demands on workers. These facts lead to high fatality and injury rates in construction workers [4, 5]. In addition, the productivity of the construction industry has barely increased over the past few decades [6]. More recently, the outbreak of the Covid-19 pandemic has caused serious economic impact and schedule delays on construction projects since it is hard to maintain social-distancing while working in close proximity on construction sites [7]. This has highlighted the need for construction techniques that can allow workers to perform tasks remotely, allowing for reduction in the number of on-site workers or their physical separation while on site.

Robots can manipulate heavy objects and could potentially relieve construction workers from excessive physical demand, alleviate labor shortage, increase productivity, and promote remote construction. Although robots have already boosted the productivity of several industries, some attributes of the construction industry inhibit the wide application of construction robots [8]. First, the unique and static nature of the construction product requires robots being able to move to the workspace, accurately localize themselves, and conduct a series of different actions on the product [9, 10]. Second, the unstructured construction site limits the workspace of the robot and adds to the difficulty of robot motion planning and localization [11, 12]. Third, the moving workers, components, and construction equipment require robots to be able to comprehensively perceive the environment and make quick responses [13].

In addition, construction work has relatively loose tolerances [14, 15]. The evolving as-built structure and some construction materials may deviate from designed geometry, which requires adjustment of high-level task plans accordingly [16]. Although the recent development of artificial intelligence algorithms allows robots to be programmed with adaptability, it is not cost-effective or practical to equip and program construction robots with such high perception ability and adaptivity to cope with all potential issues on construction sites [17]. Human-robot collaboration (HRC) combines human beings' cognitive ability with robots' competency in power, speed, and accuracy, and has thus become a promising solution for robotizing construction work.

Several HRC methods have been adopted in the construction industry. An intuitive method for collaborative human-robot construction is to lead the robot by directly applying forces to the robot or the object carried by the robot through physical contacts, such as MULE135 (Material Unit Lift Enhancer) [18] and curtain wall installation robot [13]. It relieves construction workers from high physical stress while retains their operation agility. However, it still requires human workers to be present alongside the robot. Considering the needs of performing construction work remotely, several teleoperation techniques have been proposed for construction robotics, such as joysticks [19], haptic devices [20], wearable sensors [21], and vision detection systems [22]. Although teleoperation can protect workers from potential dangers on-site, operating robots with multiple degrees of freedom (DOFs) requires expertise. The robot is moving at the same time as human operation and the human needs to figure out and lead the robot through the full manipulation path. There are also safety issues caused by the limited perception of working environments [13]. Recently, the emergence of commercial head-mounted devices promoted the application of immersive virtual reality (VR), augmented reality (AR), and mixed reality (MR) in HRC. For example, VR has been used to study worker reactions while sharing workspaces with robots and AR has been used to give worker instructions to cooperate with robots [23, 24]. Therefore, a safe and intuitive HRC interface for construction robots that takes advantage of immersive technologies and allows remote operation is proposed.

The objective of this paper is to propose a real-time, process-level, immersive VR digital twin for intuitive and remote human-robot collaborative construction work. The human worker performs high-level decision making and supervision in an immersive VR digital twin of the construction site. The robot is responsible for detailed motion planning and task execution on-site. The detailed motion plan and robot status information are visualized in VR for human approval before actual execution. A case study involving imperfect rough carpentry (i.e., stud framing) and a 6DOF KUKA drywall-installing robot arm is conducted to demonstrate and evaluate the digital twin system. The construction site and robot arm are emulated in the Gazebo simulator that allows rapid prototyping of robotic tasks and direct subsequent transfer of the methods to the corresponding real robotic platforms [25].

2 Collaborative Human-Robot Construction System

Figure 1 gives an overview of the proposed collaborative human-robot construction framework. The human worker interacts with the robot through an immersive VR interface developed in Unity3D. The Oculus Rift S VR headset and the Oculus Touch controllers are used to create the VR experience. The immersive VR interface is connected to the robot operation environment (i.e. the construction site environment in which the robot performs the task) via the Robot Operating System (ROS) as the computational core. The computational core is responsible for computation and data processing. It also acts as the communication tool between the human and the robot. In this section, the immersive VR interface and the computational core are discussed in detail. The operation environment is discussed later in the case study.

2.1 Immersive VR interface

2.1.1 Immersive VR environment

The immersive VR environment is the digital twin of the construction environment. There are two common methods of developing the VR model of a construction site. One of them is to use the 3D CAD model, such as Building Information Modeling (BIM) [26]. It is fast and convenient to be loaded as a VR scene but it cannot reflect actual construction site environment since the built structure could deviate from design and there would be obstacles stacking on-site during construction. Another method is to construct point clouds from laser scanners or RGBD cameras [27]. However, it takes significant computational resources to construct the point cloud of a construction site and use it in VR. Therefore, this research uses a combination of the as-design BIM model and as-built point clouds of workspace obtained from the sensors to create the VR digital twin of the construction site (Figure 2).

The general construction site environment is generated from the BIM model. For the non-critical components, the BIM models are directly loaded and used in VR. It creates a realistic construction environment VR experience. The non-critical components indicate components that are outside the robot workspace or components that are inside the robot workspace but their deviations from design do not influence user decision making and robot execution processes. The BIM models of the critical components are set as semi-transparent so that the user can visualize how the structure is designed and supposed to be built.

Meanwhile, the robot workspace is captured by RGBD cameras placed on the construction site. The RGBD images are sent to the computational core for processing and then transferred to Unity3D for visualization in VR in near real-time. The point cloud overlays the semi-transparent as-design BIM model so the differences between the as-design and as-built geometry can be inspected. Point clouds can also capture the dynamic conditions in robot workspace, such as workers and obstacles, and show it in the VR. The human worker can view the as-built workspace conditions for decision making, such as deciding how and where to install the next component.

Figure 1. Collaborative human-robot construction system overview

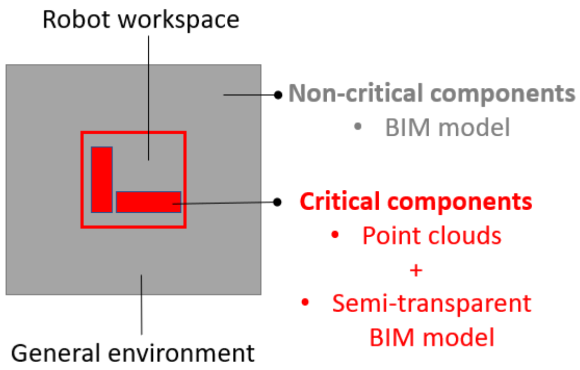

Figure 2. Immersive VR environment construction

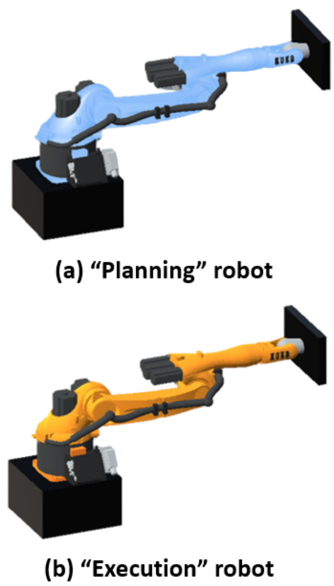

Figure 3. VR robot models (a) "planning" robot (b) "execution" robot

2.1.2 Robot digital twin model

There are two full-scale robot models in the VR scene, which overlap with each other in the original state. One of them shows the planning state of the robot. It is used to visualize the robot motion plan (Figure 3(a)). The other one shows the actual state of the robot for execution status visualization (Figure 3(b)). The two robot models are referred to as the "planning" robot and the "execution" robot respectively in the rest of this paper. The KUKA robot arm model is built in Unified Robotics Description Format (URDF) in the ROS computational core, which has the same size and configuration as the actual robot [28]. The model is then transferred from ROS to be loaded as a game object in VR using the ROS# library [29]. The VR robot models preserve the kinematic and dynamic properties of the robot and can be controlled by subscribing messages from the computational core.

2.1.3 Interactive VR elements and functions

One of the advantages of immersive VR is that the user can have realistic experience while overcoming some restrictions of the real world. For example, users can receive extra information that they cannot directly achieve from the real world, such as the comparison between the as-design and as-built geometry, and overcome some real-world constraints, like gravity.

Our digital twin system includes several interactive VR elements. An interactive billboard with two functions has been developed. First, it shows the user system messages that cannot be directly obtain even from the actual construction environment, such as warning messages from ROS. Second, the billboard can also be used as an input device inside VR where the user can give commands to the system by interacting with the buttons on the screen. Users' sight could easily be occluded in complex construction environments. Therefore, instead of fixing the billboard to one location, our system allows users to manipulate it with the VR controller and adjust its pose and view at their convenience. The billboard can be suspended in the air as how it has been placed. As the environment changes, users can always put the billboard at a new desirable position. Some interactive construction materials have been created for pick-and-place related tasks, which can also be grabbed and suspended in the air, for the user to perform high-level task planning. It should be noted that although the paper mainly discussed pick-and-place related cases, the system can be generalized to many construction tasks by adding customized interactive elements and functions.

2.2 Computational Core

ROS has been used as the system computational core. ROS is an open-source system that combines a variety of tools and software libraries for robot operation [30]. It can communicate with Unity3D, Gazebo, and the actual robot. In our system, ROS is also responsible for sensor data processing, motion planning, and robot control besides communication.

2.2.1 Communication

The system communication framework is shown in Figure 4. ROS# is used for communication between ROS and Unity3D [29], and *gazebo_ros_pkgs* is used to interface Gazebo with ROS [31]. As the program starts, Gazebo starts to constantly publish sensor data and robot states to ROS. In the meantime, ROS processes the sensor data and publish the processed data and robot states to Unity3D, which is then visualized as the point cloud and the state of the “execution” robot in VR. Based on the point cloud, the user develops the task plan and sends it to ROS after confirmation. ROS then generates a collision-free motion plan accordingly.

The motion plan is sent back to Unity3D and is visualized by the user on the “planning” robot. If the user is not satisfied with the motion plan, they can either adjust their task plan or request another motion plan from ROS which in turn generates a new motion plan in response. Upon user approval, a message is sent to ROS which converts the motion plan into execution commands to control the actual robot. As the actual robot executes the work, updated robot state messages are received by ROS and Unity3D. The “execution” robot in VR moves accordingly.

2.2.2 Sensor data processing

Several Microsoft Kinect cameras are placed on the virtual construction site in Gazebo to capture robot workspace. The RGBD images captured are converted into point clouds. Point clouds from different cameras are transformed into the world frame based on respective camera positions and rotations and then concatenated into one single point cloud. The point cloud is then downsampled with the voxel grid filter. Finally, it goes through the self-filtering process. Self-filter removes visible parts of the robot from the point cloud based on the current robot state.

2.2.3 Motion planning

After receiving the user-specified task plan, the corresponding end-effector pose is calculated. The robot then plans a trajectory to that pose so that both the robot itself and the object carried by the robot do not collide with the environment. The motion planning is conducted by MoveIt, a robotics manipulation platform in ROS [32]. The point cloud after processing discussed earlier is further processed by OctoMap into a 3D occupancy grid map of the environment [33]. The Open Motion Planning Library is used as the motion planner and the Flexible Collision Library is used for collision detection [34, 35]. The inverse kinematics is calculated by the Kinematics and Dynamics Library numerical jacobian-based solver [36]. The joint velocity and acceleration limits are taken into consideration to time-parameterize the generated path. After that, the time-parameterized path is sent to Unity3D as separate states for visualization on the “planning” robot.

2.2.4 Robot Control

When the user approves the trajectory plan in VR, ROS will be notified with an approval message. The *ros_control* package is then used to convert the approved trajectory plan into robot control commands [37]. It obtains joint state data from the encoders of robot actuators and generates output with PID controllers to robot actuators.

2.3 Case Study

A drywall installation case study with a 6DOF KUKA robot arm that is capable of real construction work has been conducted to evaluate the immersive digital twin system. The user guides the robot arm to pick up a drywall panel placed on the ground near the robot and place it on a wall frame that is built with deviations from design. Figure 5 shows the robot operating environment in Gazebo, which represents the actual construction site, and its VR digital twin in Unity3D. Three Microsoft Kinect cameras are used to capture the robot workspace environment in Gazebo.

Figure 6 shows the point cloud before and after processing. Points on the ground panel are also removed with the RANSAC plane segmentation algorithm. In VR, a drywall panel is set to be the interactive construction component, which is in the same shape as the actual drywall. The user will first observe the wall frame geometry from the point cloud and decide how to install the drywall panel onto the frame. The user can then demonstrate the task plan by grabbing the interactive panel and placing it at the desired installation position. The buttons on the interactive billboard provide options for fast and accurate adjustment of the orientation of the interactive panel.

The robot will first pick up the drywall panel on the floor and then wait for the user to specify the task plan. The

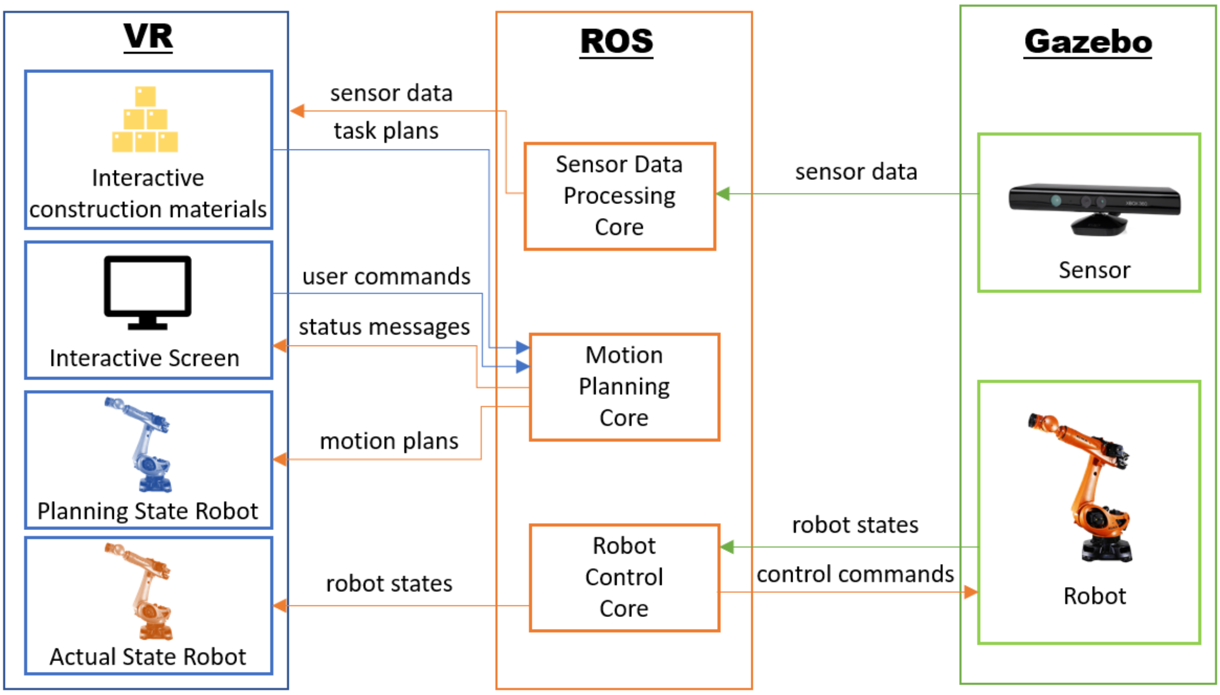

Figure 4. System communication framework

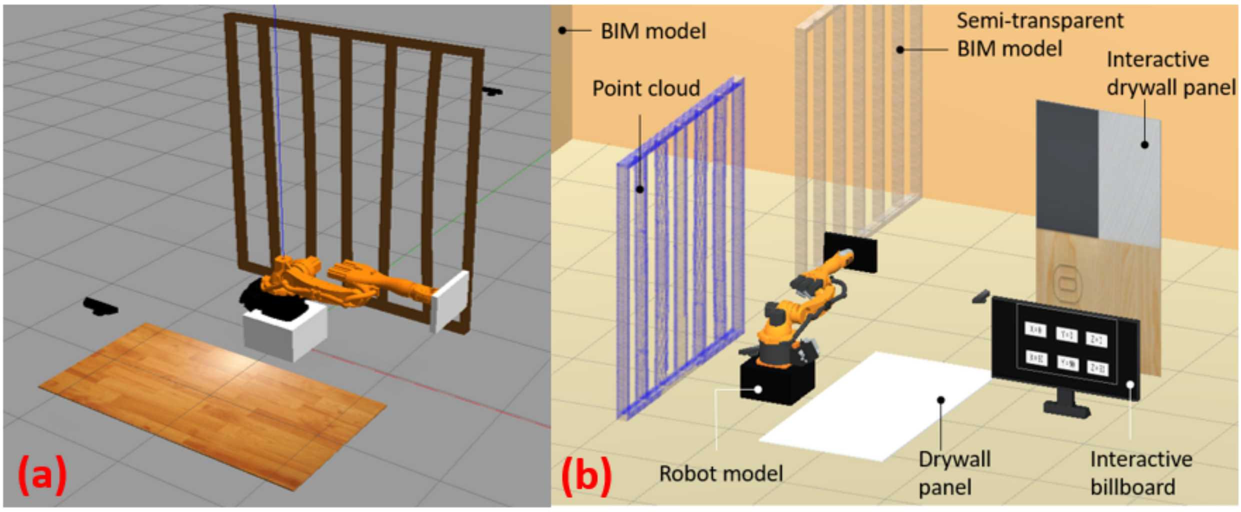

Figure 5. (a) Robot operation environment (b) VR environment

user can know whether the robot has successfully picked up the panel from the billboard and the panel will change color after being picked up. After the user confirms the task plan, ROS starts to develop the detailed motion plan to place the panel to the user-specified position while sending planning status messages (e.g. in progress, success, reasons of failure) to the user via the billboard. After motion planning, the "planning" robot demonstrates the plan to the user while the actual robot stays still (Figure 7).

Upon approval, ROS controls the actual robot to execute the approved motion plan and update the user with execution status messages. At the same time, the "execution" robot is synchronized with the actual robot by subscribing to the actual robot state messages so that the user can perceive actual robot status from VR (Figure 8).

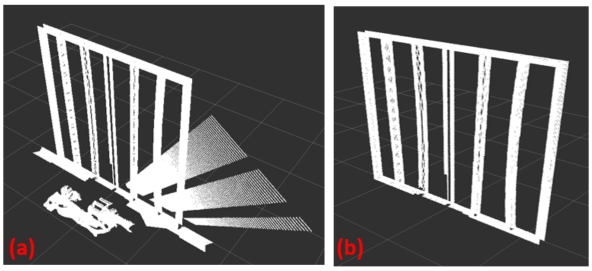

Figure 6. Point cloud (a) before processing (b) after processing

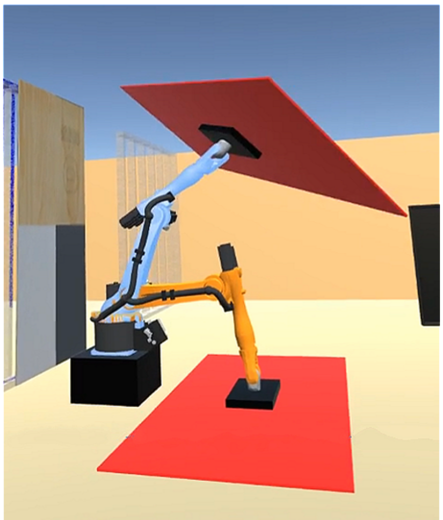

Figure 7. "Planning" robot demonstrating motion plan

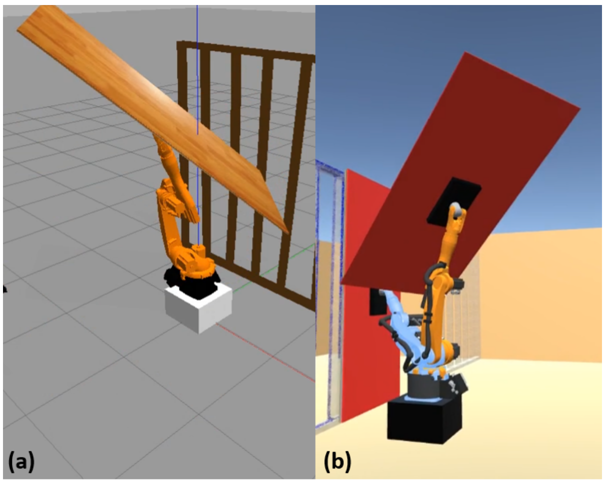

Figure 8. Synchronized movement of (a) actual robot and (b) execution robot

3 Conclusion

In this study, a real-time, process-level, immersive digital twin system for collaborative human-robot construction work is proposed. The system has several advantages. First, human workers can visualize construction site conditions and collaborate with the robot remotely, which protects them from potential dangers on the construction site. Second, the communication network allows the human worker and the robot to exchange task plans and status information in near real-time. Third, it allows the human worker to improvise high-level construction plans based on as-built construction site geometry. Last, the robot develops its motion plan and carries out physical construction work on-site, which significantly reduces human workload. In ongoing work, our research team is experimenting with real robots and implementing the immersive digital twin system with mobile robot arms.

Acknowledgements

The authors would like to acknowledge the financial support received from the U.S. National Science Foundation (NSF) CBET 1638186 and CBET 1804321. Any opinions and findings in this paper are those of the authors and do not necessarily represent those of the NSF.

References

[1] Jan Mischke Maria João Ribeirinho Mukund Sridhar Matthew Parsons Nick Bertram Filipe Barbosa, Jonathan Woetzel and Stephanie Brown. Reinventing construction: A route to higher productivity. On-line: `https://www.mckinsey.com/~/media/McKinsey/Industries/Capital%20Projects%20and%20Infrastructure/Our%20Insights/Reinventing%20construction%20through%20a%20productivity%20revolution/MGI-Reinventing-construction-A-route-to-higher-productivity-Full-report.ashx`, 2017.

[2] Meiyin Liu. *Video-Based Human Motion Capture and Force Estimation for Comprehensive On-Site Ergonomic Risk Assessment*. PhD thesis, 2019.

[3] CJ Liang, VR Kamat, and CC Menassa. Teaching robots to perform construction tasks via learning from demonstration. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 36, pages 1305–1311. IAARC Publications, 2019.

[4] CPWR. The construction chart book. On-line: `https://www.cpwr.com/publications/research-findings-articles/construction-chart-book`, 2018.

[5] CJ Liang, KM Lundeen, W McGee, CC Menassa, S Lee, and VR Kamat. Stacked hourglass networks for markerless pose estimation of articulated construction robots. In *35th International Symposium on Automation and Robotics in Construction*, 2018.

[6] Juan Manuel Davila Delgado, Lukumon Oyedele, Anuoluwapo Ajayi, Lukman Akanbi, Olugbenga Akinade, Muhammad Bilal, and Hakeem Owolabi. Robotics and automated systems in construction: Understanding industry-specific challenges for adoption. *Journal of Building Engineering*, 26:100868, 2019.

[7] ENR. Construction loses 975,000 jobs in april, due to covid-19 impacts. On-line: `https://www.enr.com/articles/49333-construction-loses-975000-jobs-in-april-due-to-covid-19-impacts`, 2020.

[8] Kurt M Lundeen, Vineet R Kamat, Carol C Menassa, and Wes McGee. Scene understanding for adaptive manipulation in robotized construction work. *Automation in Construction*, 82:16–30, 2017.

[9] Toshio Fukuda, Yoshio Fujisawa, Fumihito Arai, H Muro, K Hoshino, Kenji Miyazaki, and K Uehara. A new robotic manipulator in construction based on man-robot cooperation work. In *Proc. of the 8th International Symposium on Automation and Robotics in Construction*, pages 239–245. Citeseer, 1991.

[10] Chen Feng, Yong Xiao, Aaron Willette, Wes McGee, and Vineet R Kamat. Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites. *Automation in Construction*, 59:128–138, 2015.

[11] Xing Su and Hubo Cai. Enabling construction 4d topological analysis for effective construction planning. *Journal of Computing in Civil Engineering*, 30 (1):04014123, 2016.

[12] Lichao Xu, Chen Feng, Vineet R Kamat, and Carol C Menassa. An occupancy grid mapping enhanced visual slam for real-time locating applications in indoor gps-denied environments. *Automation in Construction*, 104:230–245, 2019.

[13] Seungyeol Lee and Jeon Il Moon. Introduction of human-robot cooperation technology at construction sites. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 31, page 1. IAARC Publications, 2014.

[14] Ci-Jyun Liang, Vineet R Kamat, and Carol C Menassa. Teaching robots to perform quasi-repetitive construction tasks through human demonstration. *Automation in Construction*, 120:103370, 2020.

[15] Colin Milberg and Iris Tommelein. Role of tolerances and process capability data in product and process design integration. In *Construction Research Congress: Wind of Change: Integration and Innovation*, pages 1–8, 2003.

[16] Kurt M Lundeen, Vineet R Kamat, Carol C Menassa, and Wes McGee. Autonomous motion planning and task execution in geometrically adaptive robotized construction work. *Automation in Construction*, 100: 24–45, 2019.

[17] Yap Hwa Jen, Zahari Taha, and Lee Jer Vui. Vr-based robot programming and simulation system for an industrial robot. *International Journal of Industrial Engineering*, 15(3):314–322, 2008.

[18] Construction Robotics. Mule. On-line: `https://www.construction-robotics.com/mule/`, Accessed: 06/01/2020.

[19] Kyungmo Jung, Baeksuk Chu, Shinsuk Park, and Daehie Hong. An implementation of a teleoperation system for robotic beam assembly in construction. *International Journal of Precision Engineering and Manufacturing*, 14(3):351–358, 2013.

[20] P Chotiprayanakul, DK Liu, and G Dissanayake. Human–robot–environment interaction interface for robotic grit-blasting of complex steel bridges. *Automation in Construction*, 27:11–23, 2012.

[21] Dongmok Kim, Jongwon Kim, Kyouhee Lee, Cheolgyu Park, Jinsuk Song, and Deuksoo Kang. Excavator tele-operation system using a human arm. *Automation in construction*, 18(2):173–182, 2009.

[22] Ying-Hao Yu, Chun-Hsien Yeh, Tsu-Tian Lee, Pei-Yin Chen, and Yeu-Horng Shiau. Chip-based real-time gesture tracking for construction robot's guidance. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 31, page 1. IAARC Publications, 2014.

[23] Sangseok You, Jeong-Hwan Kim, SangHyun Lee, Vineet Kamat, and Lionel P Robert Jr. Enhancing perceived safety in human–robot collaborative construction using immersive virtual environments. *Automation in Construction*, 96:161–170, 2018.

[24] Pedro Tavares, Carlos M Costa, Luís Rocha, Pedro Malaca, Pedro Costa, António P Moreira, Armando Sousa, and Germano Veiga. Collaborative welding system using bim for robotic reprogramming and spatial augmented reality. *Automation in Construction*, 106:102825, 2019.

[25] Nathan Koenig and Andrew 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. IEEE, 2004.

[26] Jing Du, Zhengbo Zou, Yangming Shi, and Dong Zhao. Zero latency: Real-time synchronization of bim data in virtual reality for collaborative decision-making. *Automation in Construction*, 85:51–64, 2018.

[27] Qian Wang, Jingjing Guo, and Min-Koo Kim. An application oriented scan-to-bim framework. *Remote sensing*, 11(3):365, 2019.

[28] kuka - ros wiki. On-line: `http://wiki.ros.org/kuka`, Accessed: 06/01/2020.

[29] ros-sharp. On-line: `https://github.com/siemens/ros-sharp`, Accessed: 06/01/2020.

[30] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[31] gazebo_ros_pkgs - ros wiki. On-line: `http://wiki.ros.org/gazebo_ros_pkgs`, Accessed: 06/01/2020.

[32] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit![ros topics]. *IEEE Robotics & Automation Magazine*, 19(1):18–19, 2012.

[33] A Hornung, KM Wurm, M Bennewitz, C Stachniss, and W Burgard. An efficient probabilistic 3d mapping framework based on octrees armin hornung. *Autonomous Robots Journal. Springer*, 2013.

[34] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[35] Jia Pan, Sachin Chitta, and Dinesh Manocha. Fcl: A general purpose library for collision and proximity queries. In *2012 IEEE International Conference on Robotics and Automation*, pages 3859–3866. IEEE, 2012.

[36] Ruben Smits, H Bruyninckx, and E Aertbeliën. Kdl: Kinematics and dynamics library, 2011.

[37] Sachin Chitta, Eitan Marder-Eppstein, Wim Meeussen, Vijay Pradeep, Adolfo Rodríguez Tsouroukdissian, Jonathan Bohren, David Coleman, Bence Magyar, Gennaro Raiola, Mathias Lüdtke, and Enrique Fernández Perdomo. ros_control: A generic and simple control framework for ros. *The Journal of Open Source Software*, 2017.