Affecting Fundamental Transformation in Future Construction Work Through Replication of the Master-Apprentice Learning Model in Human-Robot Worker Teams

by

Ci-Jyun Liang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2021

Doctoral Committee:

Professor Vineet R. Kamat, Co-Chair
Associate Professor Carol C. Menassa, Co-Chair
Professor SangHyun Lee
Associate Professor Wes McGee
Assistant Professor Xi Jessie Yang

Ci-Jyun Liang

cjliang@umich.edu

ORCID iD: 0000-0002-0213-8471

Acknowledgments

I would like to express my sincere gratitude to my advisors Professor Vineet Kamat and Professor Carol Menassa, for supporting and guiding me throughout my Ph.D. study. Professor Kamat has encouraged me to pursue cutting-edge research and provided me insightful and valuable opinions that helped me tackle research questions. Professor Menassa has always provided me inspiring and delightful ideas that helped me improve my work. They have offered me various opportunities to gain experience in becoming a future faculty member, as well as mental support during my doctoral study, especially for the difficult time in my last year of study. It has been a great honor working with them.

I would like to sincerely thank Professor SangHyun Lee for his critical and constructive feedback on my project and the collaboration in the NRI project. His feedback in my research proposal defense truly helped me rethink the logical coherence of my research methodology.

I am deeply indebted to Professor Wes McGee for providing the robot system in the Digital Fabrication Laboratory at Taubman College of Architecture and Urban Planning to conduct experiments. Without his generous support, my dissertation will not go through smoothly. His assistance in technical support and valuable discussions helped me resolve technical issues during the system development.

I would also like to thank Professor Xi Jessie Yang for her valuable comments on improving my project. These comments and discussions during the project meetings enlightened me on thinking of human factors and the relations between humans and robots.

I would also like to thank Professor Jia Deng and other colleagues in the NRI project, Dr. Daeho Kim, Ankit Goyal, and Dr. Meiyin Liu. The delightful discussions and constructive feedback from them during the collaboration helped me build my work on pose estimation.

I want to thank my lab mates Drs. Feng Chen, Albert Thomas, Yong Xiao, Bharadwaj Mantha, Kurt Lundeen, Da Li, Lichao Xu, Xi Wang, Sumukha Udupa, Min Deng, Hongrui Yu, Somin Park, and many others from the LIVE, SICIS, and DPM groups. I enjoyed the time we spent together and the memories in the past few years.

I would also like to acknowledge the generous financial support for this dissertation received from the U.S. National Science Foundation (NSF) NRI #1734266 and FW-HTF #2025805.

Last but not least, I would like to thank all my family members and friends who have always supported me throughout this journey.

Table of Contents

List of Tables

List of Figures

Abstract

Construction robots continue to be increasingly deployed on construction sites to assist human workers in various tasks to improve safety, efficiency, and productivity. Due to the recent and ongoing growth in robot capabilities and functionalities, humans and robots are now able to work side-by-side and share workspaces. However, due to inherent safety and trust-related concerns, human-robot collaboration is subject to strict safety standards that require robot motion and forces to be sensitive to proximate human workers. In addition, construction robots are required to perform construction tasks in unstructured and cluttered environments. The tasks are quasi-repetitive, and robots need to handle unexpected circumstances arising from loose tolerances and discrepancies between as-designed and as-built work. It is therefore impractical to pre-program construction robots or apply optimization methods to determine robot motion trajectories for the performance of typical construction work.

This research first proposes a new taxonomy for human-robot collaboration on construction sites, which includes five levels: Pre-Programming, Adaptive Manipulation, Imitation Learning, Improvisatory Control, and Full Autonomy, and identifies the gaps existing in knowledge transfer between humans and assisting robots. In an attempt to address the identified gaps, this research focuses on three key studies: enabling construction robots to estimate their pose ubiquitously within the workspace (Pose Estimation), robots learning to perform construction tasks from human workers (Learning from Demonstration), and robots synchronizing their work plans with human collaborators in real-time (Digital Twin).

First, this dissertation investigates the use of cameras as a novel sensor system for estimating the pose of large-scale robotic manipulators relative to the job sites. A deep convolutional network human pose estimation algorithm was adapted and fused with sensor-based poses to provide real-time uninterrupted 6-DOF pose estimates of the manipulator's components. The network was trained with image datasets collected from a robotic excavator in the laboratory and conventional excavators on construction sites. The proposed system yielded an uninterrupted and centimeter-level accuracy pose estimation system for articulated construction robots.

Second, this dissertation investigated Robot Learning from Demonstration (LfD) methods to teach robots how to perform quasi-repetitive construction tasks, such as the ceiling tile installation process. LfD methods have the potential to be used in teaching robots specific tasks through human demonstration, such that the robots can then perform the same tasks under different conditions. A visual LfD and a trajectory LfD methods are developed to incorporate the context translation model, Reinforcement Learning method, and generalized cylinders with orientation approach to generate the control policy for the robot to perform the subsequent tasks. The evaluated results in the Gazebo robotics simulator confirm the promise and applicability of the LfD method in teaching robot apprentices to perform quasi-repetitive tasks on construction sites.

Third, this dissertation explores a safe working environment for human workers and robots. Robot simulations in online Digital Twins can be used to extend designed construction models, such as BIM (Building Information Models), to the construction phase for real-time monitoring of robot motion planning and control. A bi-directional communication system was developed to bridge robot simulations and physical robots in construction and digital fabrication. Through empirical studies, the high accuracy of the pose synchronization between physical and virtual robots demonstrated the potential for ensuring safety during proximate human-robot co-work.

Chapter 1

Introduction

The construction industry is confronted with chronic problems of stagnant productivity and a shortage of skilled workers [1,2]. This is largely due to an aging and retiring workforce that has not been offset by an influx of younger workers, who are generally reluctant to enter the construction industry [2]. The reasons for this are two-fold. First, construction work involves the use of a variety of hand and power tools to perform repetitive basic tasks such as cutting, connecting, and spreading to complete a project [3]. Such construction tasks are often perceived to be dull, dirty, and dangerous (construction 3D [4]) and the cause of strain and long-term fatigue [5]. Second, the physical demands of construction work often cause serious occupational injuries, such as musculoskeletal disorders that are common among workers leading to early retirement [6], which further contributes to keeping new recruits away from the construction industry. It has thus been essential to explore new methods to relieve human workers from hazardous working conditions and reduce physical-demanding work in construction projects.

Human-robot collaborative teams are broadly envisioned to be deployed on future construction sites to assist or relieve human workers from various quasi-repetitive construction tasks. The objective of applying automation and robotics in construction is to increase efficiency and productivity in construction projects, improve safety, prevent accidents, and reduce health issues from strenuous construction work. To overcome the technical barriers inherent due to the unstructured and cluttered nature of construction work, this research explores the prospect of

robots learning the knowledge of how to perform quasi-repetitive construction tasks from skilled human workers to enable robots as valuable collaborators and assistants on construction sites. Particularly, this research aims to develop construction robots that accurately locate themselves within their work environments (Pose Estimation), learn to perform construction tasks alongside human workers (Learning from Demonstration), and inform their work plans to human supervisors in real-time (Online Digital Twins). Figure 1.1 shows the overview of the dissertation.

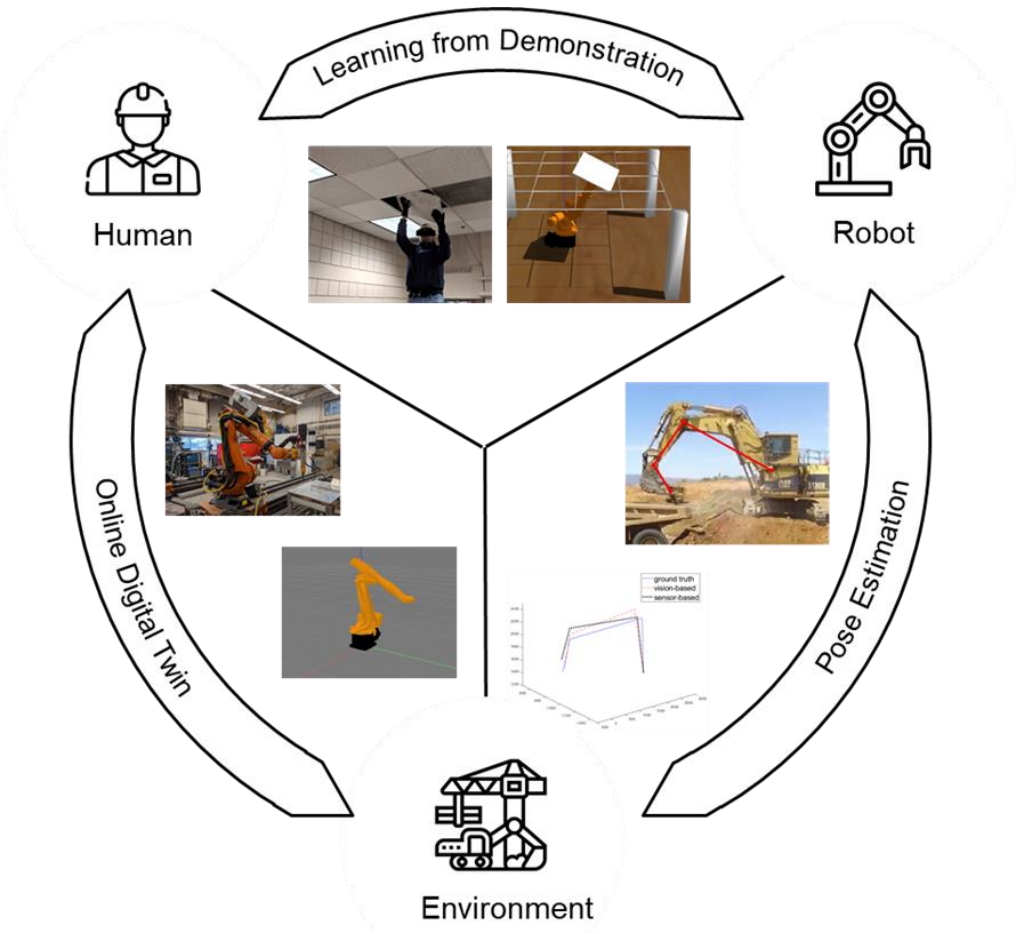


Figure 1.1 Overview of the Dissertation: Robot Localization and Pose Estimation, Robot Learning from Demonstration, and Robot Online Digital Twin

1.1 Importance of Research

Automation and robotics have been introduced to the construction industry decades ago for assisting human workers with a variety of construction tasks [7]. The global market for construction robotics is estimated to grow from $22.7 million in 2018 to $226 million by 2025 with corresponding shipments predicted to grow from 358 to 1,475 units by 2025 [8]. Different types of single-task construction robots have been developed over the years for specific construction tasks and deployed on factory-like construction sites to assist human workers with physically-demanding construction tasks [9]. However, the development of such robots has been limited, and they have not been broadly employed on actual construction sites due to the computational limitations of the hardware, the quality of the robot actuators, and more importantly the nature of the unstructured working environments [10–12].

Recent advances in hardware, software, and machine learning methods have progressed general development in robotics, which has also increased the performance of construction robots [13]. In addition, the new paradigm of collaborative robot teams and human-robot collaboration (cobot teams) has also been introduced and is envisioned to be deployed on future construction sites to assist or relieve human workers from hazardous, dangerous, and repetitive construction work [14]. By introducing human-robot collaborative teams on construction sites, human workers could potentially transition their current duties to the performance of high-level planning and cognitive work as cobot supervisors while benefiting from the assistance of the robots in repetitive physical tasks such as heavy-lifting and precise motion control of tools.

A robot arm is typically used by the construction sector [7] due to the nature of construction assembly tasks. The robot arm can provide higher degrees of freedom (up to six degrees of freedom when stationary) and more flexibility to adapt to the complexity of construction. In addition, the

high precision of the robot arm can help minimize errors in construction tasks caused by traditional human-operated equipment. For example, the holes drilled by an imprecise drilling machine often result in loose tolerances in the performed work. Robot arms typically have higher precision but require calibration using vision-based method [15], laser-based method [16], or mechanism-based method [17] to ensure its performance. Examples of other robot arm applications from the literature include maintenance and cleaning tasks [18–20], component assembly tasks [10,11,21–24], or additive manufacturing and fabrication [25–27].

When applying robots for complicated or multi-step construction tasks, it is challenging to pre-program or automatically plan the trajectory due to the discrepancy between the design model and actual workpieces [11]. Even with the feedback from sensors, a human worker still needs to assist the robot with additional guidance. Stumm et al. [28] developed a new human-robot collaboration strategy for on-site robotic assembly, called haptic programming. The robot performs the assembly task by pre-programmed trajectory, and the human worker adapts the plan based on the environmental and material conditions. The robot utilizes haptic technology to record human performance and applies it to future assembly tasks.

This concept can be further extended to robot learning from human performance or demonstration, which is similar to the apprenticeship learning modality already prevalent in the construction industry for human-to-human training [29]. The imitation learning (IL) method or Learning from Demonstration (LfD) method utilizes the demonstration data from human experts to guide the robot, while the robot tries to mimic the human behavior and explores the environment to find the optimal policy [30]. The robot first extracts and learns the knowledge from the demonstration data and then applies it to the encountered situation. The human workers switch

their role to that of a supervisor of the robot, where they first demonstrate the task several times to the robot and then monitor the robot's performance during the execution.

LfD methods offer a promising opportunity to deploy robots on construction sites. The traditional robot programming methods require an exhaustive specification of robot actions by programmers and are difficult to adapt to unknown geometry in the workplace, whereas the IL methods require task-specific experts for demonstration [31,32]. Thus, in the construction industry, instead of replacing any human workers on-site, the skilled human workers have to continually train construction robots and work with them to supervise the process. LfD research has been one of the current trends in the robotics community. As indicated by Ravichandar et al. [31], the number of publications in the LfD area has been consistently growing in the past decade.

When establishing a human-robot collaboration team on construction sites, three key aspects have to be considered. First, robots have to continuously locate themselves on highly occluded construction sites in real-time without interruption. Second, human workers have to demonstrate construction tasks to robots and supervise the construction process to ensure the quality of work. Additional improvisation from human supervisors must be provided if robots cannot complete a task successfully. Finally, human workers have to be aware of the ongoing and future changes in the environment, including the robot collaborator's work plan, to ensure the safety of human-robot collaboration on construction sites. Therefore, this research explores three key aspects: robot pose estimation, robot Learning from Demonstration, and robot online Digital Twin, which are the foundations and critical steps to enable human-robot collaboration teams in the construction industry in the near future.

1.2 Literature Review and Taxonomy

Construction robots are used on construction sites to assist with heavy-duty tasks [33] or navigate to hazardous or narrow locations to perform construction work [34]. Each robot has its specific designed functionality and typically performs a single-task [9], such as bricklaying [10,35], welding [36], or beam assembly [37–39]. To analyze the existing research on construction robots and explore the challenges and knowledge gaps for future research, a critical review of the relevant state of the art research in the construction discipline and categorizing prior and ongoing work into a logical and encompassing taxonomy are necessary.

Everett and Slocum [3] first proposed a taxonomy of construction field operations specifically for automation and robotics research. This taxonomy categorized the construction operation to the level of the basic task, such as "connect," "cover," "cut," and "dig." Single-task construction robots that existed at the same time or were developed later mapped well to a specific basic task in the taxonomy. For instance, robots developed for screwing/bolting identified best with the "connect" basic task [38,39]. Saidi et al. [12] further grouped these operations into three types of functional operators: materials handling, materials shaping, and structural joining. In addition, they also classified construction robots into three general categories based on the level of onboard intelligence: tele-operated systems, programmable construction machines, and intelligent systems [40].

On the other hand, Tan et al. [41] proposed a framework for formulating the robot-inclusive environments by measuring the inclusiveness of environments to robots, developing a taxonomy of robot-environment interaction, and identifying design criteria of autonomous robots in indoor and outdoor environments. Bock [42] identified three modules of construction robots with different tasks in interior assembly, i.e., transportation, drilling and mounting, and assembly, and then

proposed a procedure of applying these three robot modules and evaluated it in an office building construction simulation.

Although the existing taxonomies of construction robotics have reviewed prior studies and categorized them, they have not considered the effect of the human-in-the-loop collaboration. This dissertation bridges this critical gap and reviews the existing construction automation and robotics studies and applications in the context of a new proposed taxonomy that is based on the level of the human-robot interaction in the performance of work.

1.2.1 Background on Human-Robot Collaboration

Human-robot collaboration (HRC) is defined as human(s) and robot(s) in contact with each other to establish a dynamic system for accomplishing tasks in the environment [43]. The goal of HRC is to ease the workload of humans in performing repetitive and physical-demanding tasks [44]. In the manufacturing industry, humans and robots work in the shared work-space performing manufacturing tasks, such as welding [45], transporting [46], and assembling [44,47,48]. In the domestic or healthcare facility, robots are utilized to assist humans with various daily tasks such as pick up objects [49] or rehabilitation such as walking assistants [50], or arm reinforcement [51]. These applications are typically deployed in structured environments with dynamic objects and uncertainties [43], such as industrial assembly lines with moving workers.

The level of robot autonomy (LoRA) in HRC proposed by Beer et al. [52] categorizes the HRC into ten levels based on the role that the human and robot plays in the robot primitive tasks, i.e., sensing, planning, and acting. In the lower level of the LoRA, the human performs most aspects of the tasks with some assistance from the robot. For example, the robot utilizes sensing feedback to avoid obstacles during the tele-operation. The human determines the plan of a task and programs the robot to execute it. In the middle of the LoRA, the human and the robot contribute

to a task equally. Both humans and robots come up with the task plan, and then the human decides for the robot to proceed with the selected plan.

In the higher level of LoRA, the robot performs most aspects of a task with some human interventions. For example, the robot first plans the task and executes it. If the robot encounters difficulty, the human will intercede with a new plan for the robot. The human can also give an abstract high-level goal to the robot. Finally, in the highest level of the LoRA, the robot performs all aspects of a task without any intervention or assistance from the human. Based on these generalized ten levels of LoRA, the following five categories in taxonomy are proposed to organize collaborative human-robot work in construction: Pre-Programming, Adaptive Manipulation, Imitation Learning, Improvisatory Control, and Full Autonomy.

1.2.2 Taxonomy of Human-Robot Collaboration in Construction Work

Due to the complexity of construction operations and the need for robots to perform quasi-repetitive tasks, the interaction relationship in human-robot construction teams can be defined as multiplex. It is thus difficult and adds little insight to categorize human-robot interaction in construction at the level of detail in LoRA proposed by Beer et al. [52]. Therefore, a condensed taxonomy of five distinct groups is proposed: Pre-Programming, Adaptive Manipulation, Imitation Learning, Improvisatory Control, and Full Autonomy, to adequately classify construction work performed by human-robot teams. Figure 1.2 illustrates the taxonomy of construction human-robot collaboration depicting the levels of human effort and robot autonomy.

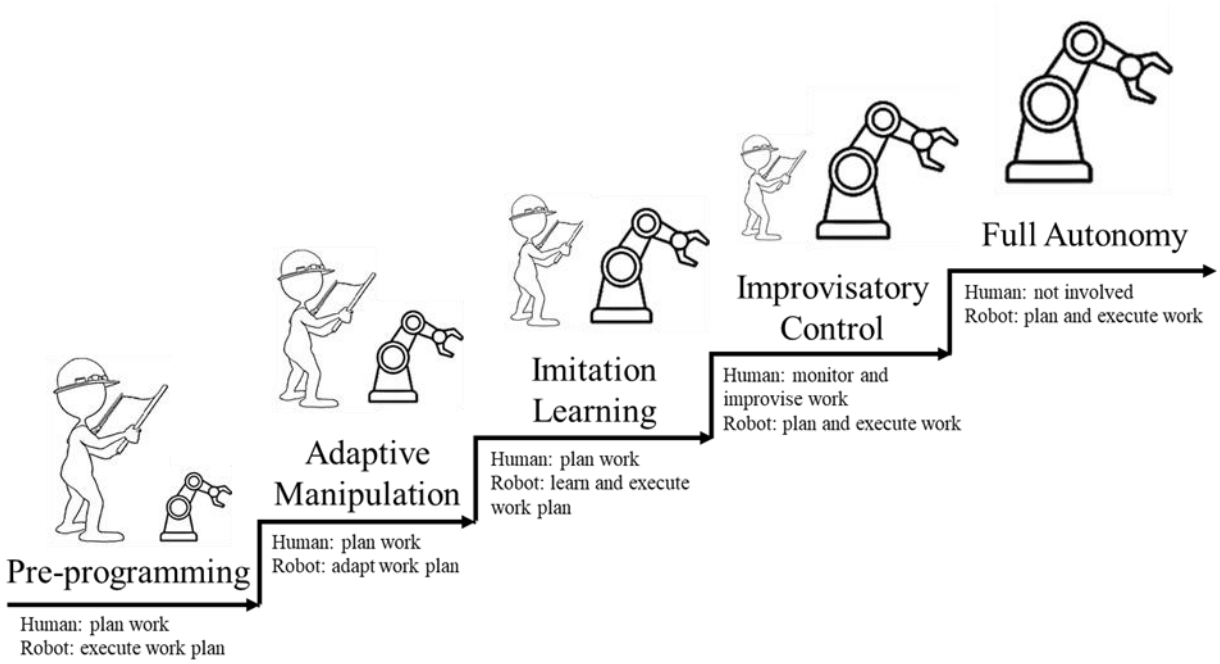


Figure 1.2 Taxonomy of Human-Robot Collaboration in Construction with the Level of Robot Autonomy and Human Effort.

The relative size of the humans and robots in the figure represents the level of effort and autonomy during the process. In the Pre-Programming category, the human undertakes the majority of the effort to plan the work, and the robot only executes the plan, which maps to the Tele-Operation group in the LoRA described in Beer et al. [52]. In the Adaptive Manipulation category, the human plans the work while the robot adapts the plan based on the encountered geometry, representing a combination of the Assisted Tele-Operation group and Batch Processing group in the LoRA. For example, the tile placement robot, bricklaying robot, or 3D printing robot (contour crafting) [53–55] are programmed with the robot code generated by the designed pattern and executed on-site [35,56].

Sensors such as laser profiler [11,21], force sensor [23], or camera [10] can help adjust the pre-programmed work plan or remote control command from the human worker to resolve the issue of the minor design-built discrepancy. However, if the quality of the robot-built component

is unacceptable, the human worker has to demolish the component and reconstruct it manually. The unforeseen situations such as arbitrary obstruction or significant discrepancy due to loose tolerances on construction sites will prevent the robot from accomplishing the task.

In the Imitation Learning category, the human worker plans the work, and the robot learns the knowledge of the work and executes it, which is categorized under the Decision Support group in the LoRA. In the Improvisatory Control category, the robot plans and executes the work while the human monitors the work and improvises if necessary. The Shared Control with Human Initiative group, Shared Control with Robot Initiative group, Executive Control group, and Supervisory Control group in the LoRA are combined to the proposed Improvisatory Control category.

In the Full Autonomy category, the robot performs every aspect of the work without intervention from the human, which corresponds to the Full autonomy group in the LoRA. For example, the drilling robot that is utilized for landslide consolidation can autonomously operate supervised by human workers remotely and switch to tele-operation mode when necessary [57]. Autonomous navigation robots are used in built environments or construction sites for maintenance and construction applications without human control or intervention [58,59], especially for the indoor or GPS-denied environment [60]. However, these robots are unable to perform complex construction tasks, which require human worker guidance such as drywall installation or ceiling tile installation process. In addition, when the robot encounters an unexpected or unforeseen situation, the human worker can intervene in the process and control the robot to complete the task. However, the robot will not absorb the concept of how human workers resolve the situation, and next time the robot still requires assistance from human workers.

A vector can represent the interplay between robot autonomy and the level of human effort to indicate the involved distribution between the human worker and the robot in the construction human-robot collaborative team, as shown in Figure 1.3. For the Pre-Programming method, the human programs the trajectory for the robot or tele-operates it, and the robot is only responsible for the acting job. Therefore, robotic autonomy is the lowest, and human effort is the highest in the taxonomy. For the Adaptive Manipulation method, the human still programs or tele-operates the robot, but the robot adapts the work plan through the use of sensor data. In addition to the acting, the robot is involved in the sensing aspect of the process. Thus, the robot has a higher level of autonomy than the Pre-Programming method.

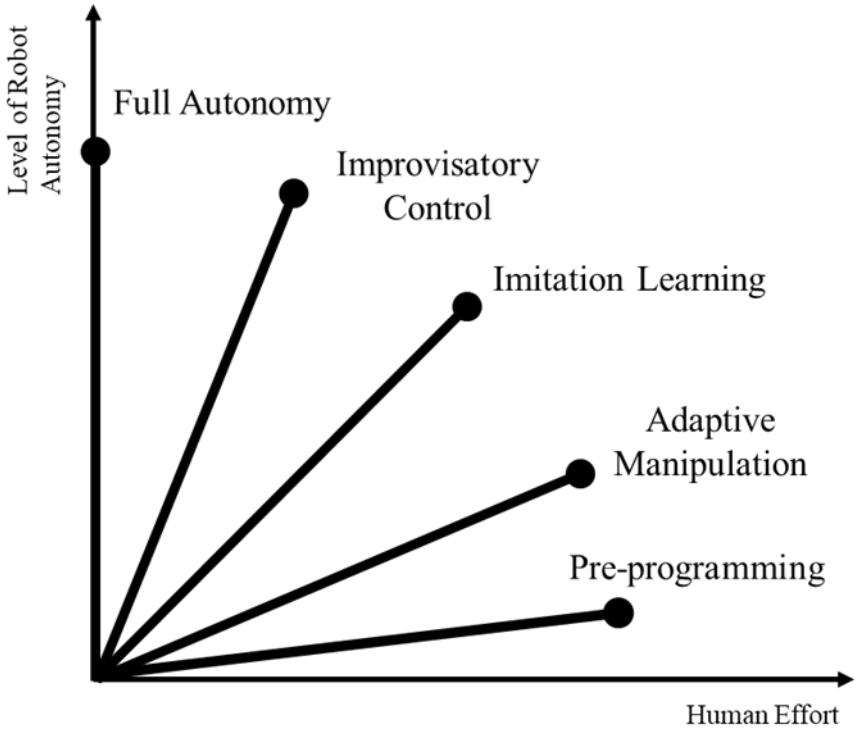


Figure 1.3 Vector Representation of Involvement Distributions Between The Human and the Robot in Collaborative Human-Robot Construction.

For the Imitation Learning method, the robot learns the skill from humans and generates the work plan to complete the task, wherein the human and the robot are equally involved in the process. For the Improvisatory Control method, the robot first explores the possible solution and determines the work plan, then the human supervises the work plan and improvises if necessary. Such collaboration requires a higher level of robot autonomy and lower human effort in the process. For the Full Autonomy method, the robot finds the work plan without support from humans. Thus, the level of robot autonomy is the highest in the taxonomy, and no human effort is involved.

1.2.3 Knowledge Gaps

Based on the categorization and the review of the literature, this research found a critical gap in the transfer of knowledge between human workers and robots in existing construction co-robots. The transfer of knowledge can enable robots to learn tasks from humans directly and resolve problems that manifest in typical quasi-repetitive construction tasks, which cannot be easily represented in or solved by optimization approaches. For example, the ceiling tile installation process requires complex manipulation trajectories to pass through the grid area while avoiding collision above the suspended grids. Such a process can be easily performed by experienced human workers who can transfer the knowledge to robots through Learning from Demonstration (LfD) or imitation learning (IL) methods.

The prospect of robots learning how to perform quasi-repetitive construction tasks from human workers offers significant promise in overcoming the described challenges, and in turn facilitating the deployment of HRC on real construction sites. Such a learning structure parallels the vocational education and training model prevalent in today's construction industry, wherein novice construction human workers (e.g., bricklayers, carpenters) develop their skills and

credentials by completing apprenticeships under the tutelage of human experts who provide technical instruction and on-the-job learning [29]. Skilled workers have thus been training and educating newer recruits to produce qualified and productive employees for future construction jobs. When training a human worker to perform a construction task in an apprenticeship program, the human recruit will learn the task by observing demonstrations from experts, absorbing the practical knowledge and practicing the task. Similarly, a robot can learn a task by observing demonstrations from human experts.

1.3 Research Objectives

The objectives of this research include developing an ubiquitous pose estimation system for interaction between robots and the environment, Learning from Demonstration methods for interaction between humans and robots, and an online Digital Twin for interaction between humans and the environment. The detailed objectives are listed as follows:

- Investigate a vision-based marker-less 2D and 3D pose estimation approach for large-scale articulated construction robots.
    - Design a fast dataset collection approach for 3D vision-based articulated construction robot pose estimation.
    - Develop a DNN-based sensor fusion uninterrupted pose estimation system which combines the vision- and sensor-based pose estimation.
- Explore the application of the Learning from Demonstration (LfD) method for teaching quasi-repetitive construction tasks to robots.
    - Develop a visual demonstration method and a trajectory demonstration method for construction tasks involving manipulation.

    - Evaluate the proposed visual and trajectory-based LfD methods in a virtual environment with a robot arm.
- Develop an online, bi-directional, process-level Digital Twin system to serve as a real-time communication bridge between human workers and robots.
    - Devise a bi-directional communication mechanism and a pose checking algorithm (PCA) to ensure the synchronization between the physical work environment and the robot's virtual representation.

1.4 Dissertation Outline

This dissertation is a compilation of peer-reviewed scientific manuscripts, which describe the research on construction robot pose estimation, learning from human demonstration, and online Digital Twin. The remainder of the dissertation is organized as follows.

Chapter 2 introduces the development of a vision-based marker-less pose estimation system and a fusion-based pose estimation system for articulated construction robots. A fast image dataset collection approach is also described in this chapter. The proposed pose estimation system is evaluated with the image dataset and compares the results with sensor-based pose estimation system and ground truth data.

Chapter 3 presents the development of a visual Learning from Demonstration method and a trajectory Learning from Demonstration method for teaching robots quasi-repetitive construction tasks. The ceiling tile installation process is selected as the target construction task. The two proposed LfD methods are evaluated by having the robot install tiles in the ROS Gazebo simulator environment and comparison of the success rate of both methods.

Chapter 4 describes the development of an online process-level Digital Twin system for construction and digital fabrication robots. The bi-directional communication mechanism is

developed to ensure state synchronization between the virtual and the physical robot. The proposed Digital Twin is evaluated by comparing the joint angles and end-effector coordinates of the virtual and physical robot.

Chapter 5 summarizes the dissertation and discusses the significance and contribution of the research. Finally, future research directions are also articulated.

Chapter 2

Ubiquitous Pose Estimation for Large-Scale Articulated Construction Robots

2.1 Introduction

Due to the hazardous, unstructured, and dynamic working environment and labor-intensive nature, the construction industry has a higher rate of workplace fatalities and injuries compared to other industries [37,61]. According to reports from the U.S. Bureau of Labor Statistics and CPWR, on average 53% of the fatal accidents that happen on construction sites are either struck by vehicle or equipment overturns and collisions between 2003 and 2010 [62], which costs approximately $13 billion per year in the U.S. [63]. On a typical construction site, workers and heavy equipment have to work together closely, which increases the potential safety risks [64].

Blind spots around the equipment are the primary cause of struck-by accidents [65]. When workers need to interact with the equipment on job sites, the equipment operator sometimes cannot locate all workers nearby, and the workers also cannot monitor the equipment components clearly, especially for articulated equipment such as excavators that usually work around trenches or earth mounds that serve as potential occlusions leading to the increased possibility of blind spots. To prevent these types of accidents, manual jobsite safety observations and inspections are required on construction sites [66]. However, safety personnel has to pay attention to entire job sites continuously, which is time-consuming and incurs additional costs [67].

Underground utility strike incidents are another category of accidents related to the operation of articulated construction robots such as excavators [68,69]. According to the Common Ground Alliance (CGA) 2016 Damage Information Reporting Tool (DIRT) report, approximately 379,000 underground utility damage incidents were reported in 2016 in the U.S., which was an increase of 20% from 2015 and cost an additional $1.5 billion [70]. One key reason for the high incident rate is the location uncertainty of the underground utilities [71]. Many of the existing buried utilities are abandoned or undocumented, and locating hidden utilities is the first step to address this issue [72]. The underground utility record could help workers and excavator operators avoid potential utility locations. However, the operators sometimes cannot locate the bucket or utilities directly from the cabin. The indirect guidance from workers near the bucket does little to reduce the risks of utility strikes.

Thus, utilizing sensors to estimate the excavator pose and providing real-time information to workers and operators has emerged as a feasible method and has been studied in developing on-site articulated construction robot pose estimation systems [71,73–75], and enhancing the on-site information with Augmented Reality [76–78]. Furthermore, the pose estimation system also provides the potential application of productivity analysis [79]. The existing productivity analysis methods only tracked the construction equipment or part of the equipment by sensors or computer vision method [80–82]. For example, the part of the excavator and the haul truck were identified and tracked during the dirt-loading cycle and utilized to estimate the productivity [80]. Motion analysis or action recognition methods are required to classify similar excavator activities such as digging and dumping to enhance productivity analysis [83]. This can be achieved by providing the detailed pose of the excavator for identifying the action [84].

The prospect of human-robot collaboration (HRC) on construction sites further heightens these proximity safety concerns [85]. Unlike HRC in typical manufacturing settings, the robot on the construction site has to maneuver around the unstructured environment to their following task location. The workplace of the robot changes dynamically based on their location, which is a challenge for HRC safety. According to standards ISO 10218-1, ISO 10218-2, and ISO/TS 15066, the safety of the HRC must be adhered to either by stopping the robot before human contact or be controlled by regulating force and speed limits [86].

The recently developed dynamic safety system utilized human detection sensors and optical sensors to adjust the robot speed according to the detected human action and the protective distance [87]. However, the protective distance, or safety zone, has to be very large since the optical sensors only identify the difference between the current frame and the previous frame instead of tracking the robot's exact pose, which causes the poor utilization of space [86]. On the other hand, the robot's onboard sensors are often failed due to magnetic disturbance by artifacts (IMU) or signal blockage in an urban canyon (GPS) [88]. In addition, the articulated construction robot has arbitrary and expansive movement around the unstructured construction site and is difficult to make the construction site a structured environment [89]. This highlights the need for developing an effective on-site pose estimation system for articulated construction robots and human workers.

The experimental testbed of the construction articulated robot in this research was an excavator since it is ubiquitous equipment on jobsite and has a large blind spot [65]. The pose of the excavator can be described as the angle between each component (boom, stick, and bucket) and the six-degree-of-freedom (6 DOF) coordinates of each joint (cabin-boom, boom-stick, stick-bucket, bucket end-effector). Figure 2.1 depicts a 2D pose estimation system.

Figure 2.1 Illustration of the 2D Pose Estimation System on a Video Frame for Both Articulated Construction Robot and Human Workers.

The pose of the construction equipment, such as an excavator, can be described as the angle between each component (boom, stick, and bucket) and the six-degree-of-freedom (6 DOF) coordinates of each joint (cabin-boom, boom-stick, stick-bucket, bucket end-effector). In the 2D case, the pose is defined as the pixel-wise coordinate and angle (X, Y, θ), whereas in the 3D case, the pose is defined as the world coordinate and roll-pitch-yaw $(X, Y, Z, \phi, \theta, \psi)$. Figure 2.2 illustrates the excavator side view with the kinematic chain and the corresponding parameters. The pose of each excavator joint P can be calculated using the angle between each component θ and effective lengths of each component L by Forward Kinematics [90] or directly estimated by sensors

or vision [71]. Therefore, determining the location of each joint and the angle between each link is the primary goal of articulated construction robot pose estimation.

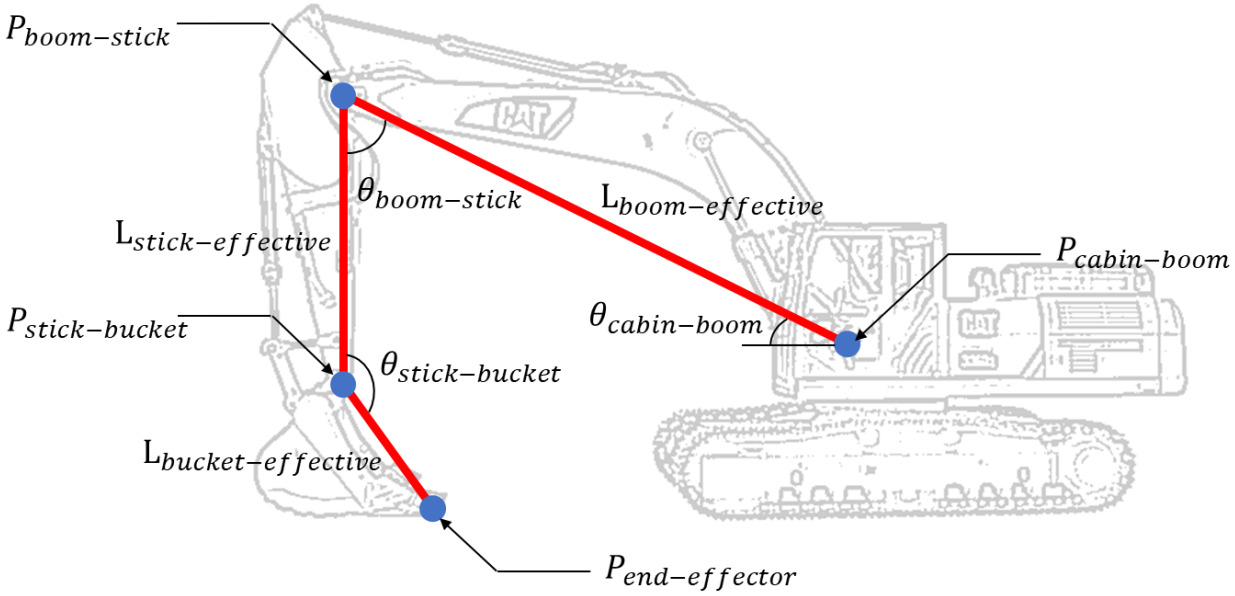


Figure 2.2 Definition of the Excavator Pose.

The remainder of this chapter is organized as follows. First, the existing pose estimation methods are reviewed. Second, a deep neural network pose estimation method for articulated construction robots is introduced. 2D, 3D, and sensor fusion version baselines are established and evaluated. Third, the performance of the proposed pose estimation method is investigated via an experiment and compared with the IMU-based pose estimation method and ground truth data. Lastly, an articulated construction robot pose estimation dataset is collected and evaluated.

2.2 Related Work

This section reviews the existing pose estimation methods, including sensor-based methods, vision-based marker-based methods, and vision-based marker-less methods, and their applications in the construction industry based on the performance.

2.2.1 Existing Pose Estimation Methods

In current practice, two types of pose estimation methods are mainly used on construction equipment or human workers. These are non-visual sensor-based and vision-based pose estimation methods. For non-visual sensor-based pose estimation methods, sensors such as Inertial Measurement Unit (IMU), Global Positioning System (GPS), Wireless Local Area Network (WLAN), Radio Frequency Identification (RFID), and Ultra-Wide Band (UWB) are mainly deployed on construction equipment and construction sites. IMU sensors need to be mounted on excavator links to measure the angle [68,91–94], which suffers from drift issues over time and magnetic interference [95]. GPS is effective for outdoor use only and also suffers from signal blockage in an urban canyon [88], which is not suitable for some indoor or urban construction sites.

WLAN systems require a significant amount of effort for calibration [96]. The accuracy of the WLAN estimation depends on the distribution of the access point [97]. RFID and UWB methods both require sufficient preinstalled tags and readers on equipment and infrastructure [98–101]. They generally suffer from missing data issues [75] and are inadequate for pose estimation [102]. Besides, most of these methods cannot provide orientation information directly, except for IMU sensors, and are thus not suitable for construction scenarios.

On the other hand, vision-based pose estimation methods are capable of analyzing position information as well as orientation information directly from input data, such as videos or point clouds [103]. These methods generally recognize construction equipment on-site [80,104–106], then estimate their six-degrees-of-freedom (6 DOF) pose [74,107,108], and can be categorized into two different groups: marker-based and marker-less pose estimation methods. The marker-based pose estimation method recognizes all the markers mounted on equipment and estimates the pose by their geometric relations or marker network [71,109,110], or projects infrared LEDs and analyzes the pattern to determine the pose [111,112]. In contrast, the marker-less pose estimation method directly extracts image features and estimates the pose from them [74,107,108]. The marker-based method has been extensively applied in indoor localization and facility management [10,113–115]. Similar to the sensor-based pose estimation method, they also require pre-installed markers on equipment and environment.

In comparison to the marker-based method, the marker-less pose estimation method only requires an on-site camera system, which is common on typical construction sites today, or utilizes RGB-D cameras [116–119]. Feature descriptor based is the first type of marker-less pose estimation method, such as Histograms of Oriented Gradient (HOG) [80], 3D principal axes descriptor (PAD) [104], Iterative Closest Point (ICP) [11], or Viewpoint Feature Histogram (VFH) [107]. On the other hand, the recently emerging Convolutional Neural Networks (CNN) is another type of pose estimation method [120], which has improved performance (accuracy and speed) in comparison with all other vision-based methods, especially for human pose estimation. The majority of the human pose estimation methods are 2D-based methods [121,122], which estimate the human pose in 2D pixel-wise coordinates, as shown in Figure 2.1. Existing human pose estimation can be categorized as detection-based and repression-based [123]. The detection-based

methods utilized a heat-map to predict the joint location [124], whereas the regression-based methods utilized a nonlinear function to compute the joint coordinates directly [125].

The stacked hourglass network method proposed by Newell et al. [126] built the foundation of the state-of-the-art 2D human pose estimation method. Generative Adversarial Networks (GAN) [127], Pyramid Residual Module (PRM) [128], Conditional Random Field (CRF) [129] were applied to the stacked hourglass network to improve the performance. Besides, several existing 3D human pose estimation methods adopted the stacked hourglass network with coarse-to-fine volumetric architecture [130] or weakly-supervised approach [131]. The existing pose estimation methods were mainly focused on the 2D pose due to the lack of 3D ground truth posture data [132]. For human pose data collection, the motion capture system is primarily used to obtain the ground truth data of the human skeleton in an indoor environment [133], which is difficult to employ for construction equipment in an outdoor environment.

2.2.2 Application of Pose Estimation Methods

The existing pose estimation methods used in construction have different target applications. The accuracy and the specific shortcomings of any pose estimation method affect the method selected for each specific construction application. Table 2.1 lists the accuracy and the limitations of the existing pose estimation methods. For the 3D markerless vision-based pose estimation method, the accuracy can be achieved at 1 m. However, the largest distance of the target equipment from the camera is 50 m; otherwise, the accuracy drops dramatically [74]. For the 3D marker-based vision-based pose estimation method, the accuracy can be achieved at 2 cm when the distance between camera and bucket teeth is under 6.1 m [71]. The camera occlusion is the main drawback of the marker-based method since the markers have to be visible in the camera view at all times in order to estimate the pose [71].

Table 2.1 Comparison of the Existing Pose Estimation Methods by Accuracy and Limitations.

	3D markerless vision-based [74]	3D marker-based vision-based [71]	Sensor-based [92]	2D vision-based [108]
Accuracy	1 m	2 cm	5 cm	10°
Disadvantage	Distance < 50 m	Camera occlusion	Data missing	No depth data and 3D pose

For the sensor-based pose estimation method, the accuracy can be achieved at 5 cm when testing on a real excavator arm with IMU sensors [92] but could be improved depending on the type of sensor used. In addition, data missing or signal block is the major issue of the sensor-based method [75,92]. Finally, for the 2D vision-based pose estimation method, the angular accuracy can be achieved at 10° between the excavator components, which results in 122 cm vertical displacement when the reaching length of the excavator boom is 7 m [108]. However, this type of method can only provide 2D pixel-wise location or angle in each image and requires extra post-processing to acquire the depth data or 3D pose [108].

Pose estimation methods have been applied on construction sites to address safety and quality-related issues. Table 2.2 compares the different pose estimation-related construction applications comparing their acceptable location uncertainty and the methods currently used. The first application is preventing accidental utility strikes during excavation, which has a 2.5 cm acceptable location uncertainty [71]. The sensor-based method [75,92] and the 3D marker-based vision-based method [71] are two methods used for such applications. The second application is grade control, which also has a 2.5 cm acceptable location uncertainty [71]. Several sensor-based grade control commercial products have claimed that their accuracy can approach 1 mm [134,135]. The above two applications can tolerate relatively low uncertainty in pose estimates due to their precise control features.

Table 2.2 Comparison of Equipment Pose Estimation Applications in the Construction Industry by Location Uncertainty and Corresponding Methods.

	Preventing utility strikes	Grade control	Object detection and tracking	Proximity detection	Autonomous excavation
Location uncertainty	2.5 cm [71]	2.5 cm [71]	< 1 m [136]	< 0.7 m [137]	4 cm [138]
Methods	Sensor [75,92] 3D vision [71]	Sensor [134,135]	Sensor [136] 2D vision [73,139]	Sensor [98,137] 2D vision [140]	Sensor [138,141] 3D vision [74]

The third application is object detection and tracking. The object detection and tracking methods have demonstrated a location uncertainty of less than 1 m [136], and sensor-based methods and 2D vision methods are mainly utilized in this application [73,139]. The fourth application is proximity detection, in which the location uncertainty is shown to be under 0.7 m [137]. Similar to object detection and tracking, the sensor-based method [98,137] and the 2D vision method [140] are used in proximity detection applications. Instead of the high accuracy, data consistency is more important for these two types of applications. Finally, the fifth application is autonomous excavation, and the acceptable location uncertainty is 4 cm [138]. The sensor-based method [138,141] and the 3D vision-based method [74] are applied.

2.3 Research Goal and Contribution

This research first proposes a vision-based marker-less pose estimation system for articulated construction robots, which can distinguish robot joints and estimate their poses in images or video frames. The excavator is used as the experimental testbed. This system is built on a state-of-the-art human pose estimation deep neural network called the stacked hourglass network [126,142] and trained on an excavator image dataset collected from a factory environment with a robotic manipulator. The network is adapted and modified for the excavator skeleton. Both 2D and

3D versions of the system are built and evaluated in order to characterize the location uncertainty requirement illustrated in Table 2.2.The performance of the proposed system is validated based on the dataset annotation and the 3D ground truth data and compared with the sensor-based pose estimation method (IMU sensors).

Furthermore, a DNN-based sensor fusion pose estimation system is proposed, which combines the vision pose and the sensor pose data (IMU sensors) to obtain an uninterrupted and high accurate 3D pose estimation system. The proposed DNN-based sensor fusion system is compared with the Extended Kalman filter methods [143,144], pure 3D vision-based method, sensor-based method, and ground truth data. Finally, a fast dataset collection approach for articulated construction robot pose estimation is also developed and described to overcome the lack of 3D ground data on construction sites.

2.4 Vision-based Marker-less Pose Estimation

This section first discusses the development of the 2D pose estimation system. Then, the system is extended to the 3D pose estimation. Third, the sensor-based pose estimation system is developed. Finally, the sensor fusion pose estimation system is introduced.

2.4.1 2D Pose Estimation System

The proposed vision-based marker-less 2D pose estimation system is developed based on a state-of-the-art human pose estimation algorithm, namely the stacked hourglass network by Newell et al. [126,142]. This network scales the training images into different resolutions and captures features, and then combines the information to predict the pose. Compared to the complicated human pose, the construction equipment pose is relatively simpler and thus requires less information across different image resolutions.

Unlike the complicated human skeleton, the excavator pose only requires identifying three components: the bucket, stick, boom, and corresponding joints. Therefore, the complexity of the network needed is lower than the original network. Two convolutional layers followed by a max-pooling layer are first applied to the training images, which shrinks the images down to the size of 64 pixels. Then three subsequent convolutional layers upscale the images to the size of 256 pixels before the hourglass module. Finally, four hourglass modules, output prediction modules, and residual link modules are used in the network.

According to Newell et al. [126], eight hourglass modules are used for human pose estimation. The reason for using four hourglass modules for the excavator pose estimation is that the excavator pose is relatively simpler than the human and thus requires less information across different image resolutions. All the convolutional layers are followed by the ReLu activation function, with stride one except the first convolutional layer (Conv1 layer) with stride two, and with batch normalization except the convolutional layers in the output prediction module. Figure 2.3 shows the detailed network structure.

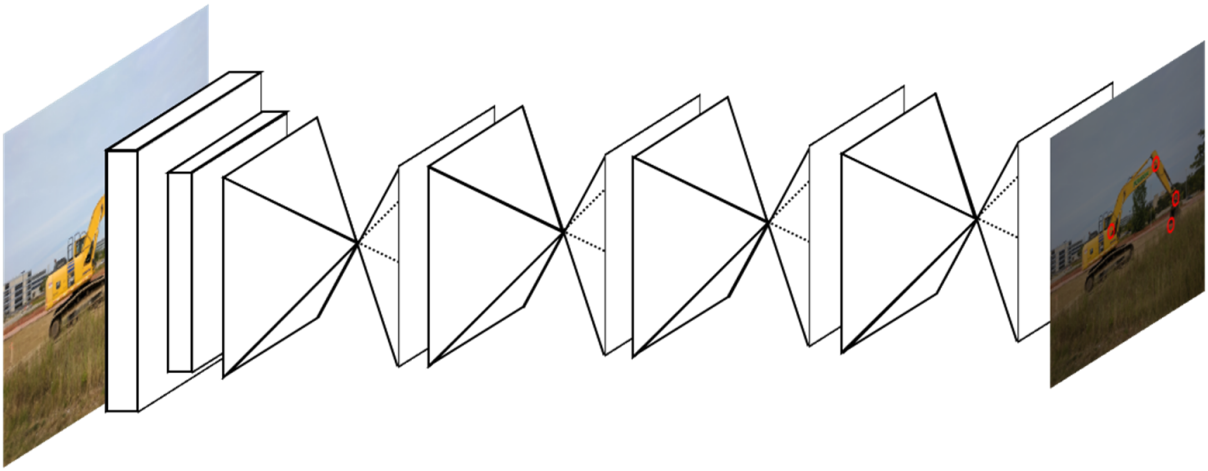


Figure 2.3 Vision-Based Marker-Less 2D Pose Estimation Network Structure [126].

The hourglass modules are the main components that collect features across the different resolutions of the images. Figure 2.4 shows the network structure of the hourglass module. The input passes into two parallel routes. In the first route, only one convolutional layer is applied to upscale the input to the size of 256 pixels. In the second route, one max pooling layer followed by three convolutional layers are applied to downscale the input to the size of 384 pixels, then resized to the size of 256 pixels, as the first route result. Finally, two route results are added together through elementwise summation to generate the output. This can preserve the global features and capture the local features as well.

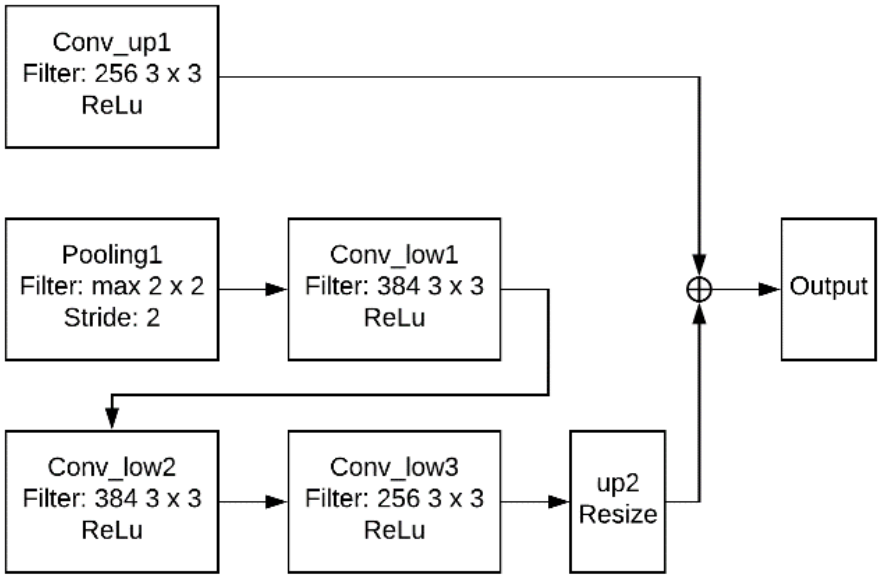


Figure 2.4 Hourglass Module Network Structure.

The output prediction module and residual link module are applied after the hourglass module. Two convolutional layers are used in the output prediction module to generate the heat-map of the possibility distribution of the location of each joint. Figure 2.5 shows the concept of the prediction heat-map. Each circle in the image represents the highest probability of the corresponding joint location. The final layer is a one-by-one convolutional layer, which aims to

calculate the possibility across the depth of the output of the previous layer. On the other hand, the residual link module combines the output of the prior hourglass and the output prediction module to generate the input for the next hourglass. The repeated hourglass and residual link modules can preserve the spatial location and relation of each feature and apply to the final prediction step.

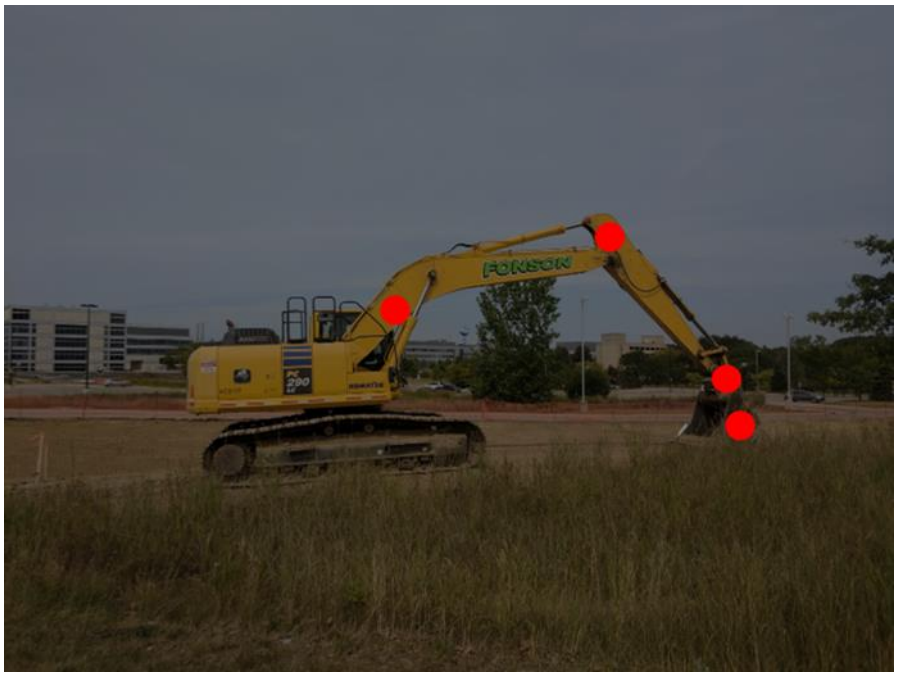


Figure 2.5 The Concept of the Prediction Heat-Map Generated by the Output Prediction Module.

2.4.2 3D Pose Estimation System

The proposed vision-based marker-less 3D pose estimation system is adapted and modified from a 3D human pose estimation baseline network [132]. This network uses the 2D pose estimation result, such as the stacked hourglass network, to predict and reconstruct the 3D pose. This can expedite the estimation process in order to accomplish the real-time pose estimation.

The objective of the baseline network is to predict and reconstruct the 3D pose of the articulated equipment based on the input 2D pose data. The 2D pose data from the previous vision-based marker-less 2D pose estimation result is passed to two subsequent linear layers, which are

followed by the ReLu activation function and the 0.5 dropout. The batch normalization is also applied to the linear layer output, which can increase the performance of the network. Next, the residual link module combines the output of the linear layer and the input 2D pose data to generate the predicted 3D pose, similar to the 2D pose estimation network. The entire process is repeated twice to generate a higher accuracy of the prediction and prevent overfitting. Based on the experiment results from [132], the best performance of the network can be achieved by repeating the process twice, and it will saturate after repeating the process four times due to overfitting the network. Figure 2.6 shows the 3D pose estimation network structure.

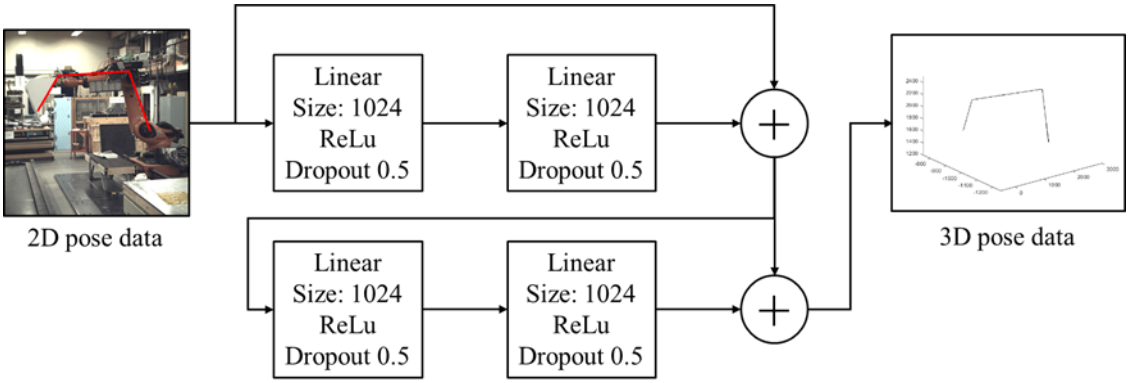


Figure 2.6 Vision-Based Marker-Less 3D Pose Estimation Network Structure.

2.4.3 Sensor Fusion Pose Estimation System

Occlusion is the primary issue of the vision-based method, especially on highly occluded construction sites. The excavators are usually blocked by trucks or soil when digging. Increasing the number and variety of training datasets has the potential to address this issue [145]. In addition, with the help of multiple cameras, the occlusion can be reduced and increases the accuracy of the pose estimation [109]. On the other hand, the sensor-based pose estimation methods can resolve the occlusion issues but suffer from data missing, drift, magnetic interference, or signal blockage issues [75,88,90,95]. By combining the advantage of the vision-based and sensor-based method,

the pose can be tracked uninterruptedly with high accuracy, i.e., sensor fusion pose estimation [146,147].

First, the sensor-based pose estimation is developed. Four IMU sensors are deployed to measure the angular change of the robot joints, as shown in Figure 2.7. These sensors are placed on the axis of each joint so that they can measure the correct angle when the robot changed its pose. The 3D pose of each joint can be calculated by Forward Kinematics. Since the exact location of a joint requires location sensors such as GPS, which are not available in the system, the first joint (A1) is aligned with the ground truth A1 joint location, and then the other joints are calculated relative to the first joint. The Xsens MTw Awinda wireless motion tracker system [148] is used for the sensor-based method. The system contains four motion trackers with IMU embedded and a wireless receiver to transmit the data. The sensor data is also synchronized with the vision-based pose data and the ground truth data so that it can be compared and fused with each other.

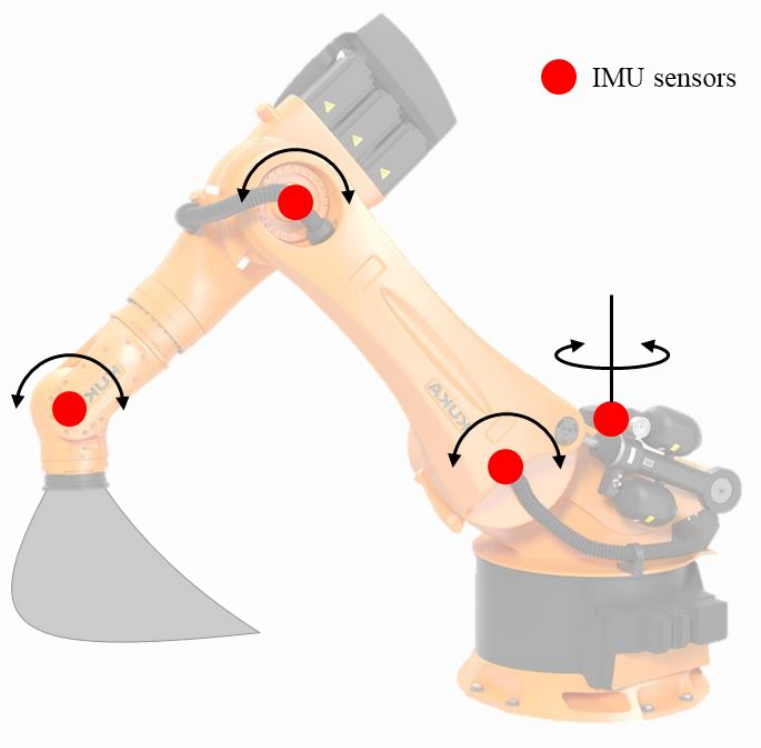


Figure 2.7 IMU Sensors Deployment.

The fusion-based pose estimation method combines the vision-based methods, which can acquire joint position accurately but are sensitive to occlusion and illumination. The IMU-based pose estimation methods can obtain the link orientation and joint position based on the body model but are sensitive to missing data and magnetic interference. In previous work, the IMU orientation data were fused with the multi-viewpoint images in the network to estimate the pose. They were either fusing the volumetric probabilistic visual hull data (PVH) from multi-viewpoint video and the IMU orientation data with Forward Kinematics at the last layer [149] or fusing the IMU orientation data as the link layer and multi-viewpoint images at the early stage [150].

In the proposed system, the IMU pose data is fused with the 3D pose network at the last residual connection layer before the final fully-connected layer. If the IMU pose data are missing, the network will simply skip the IMU pose data in the residual connection layer. Figure 2.8 shows the 3D fusion-based pose estimation network structure.

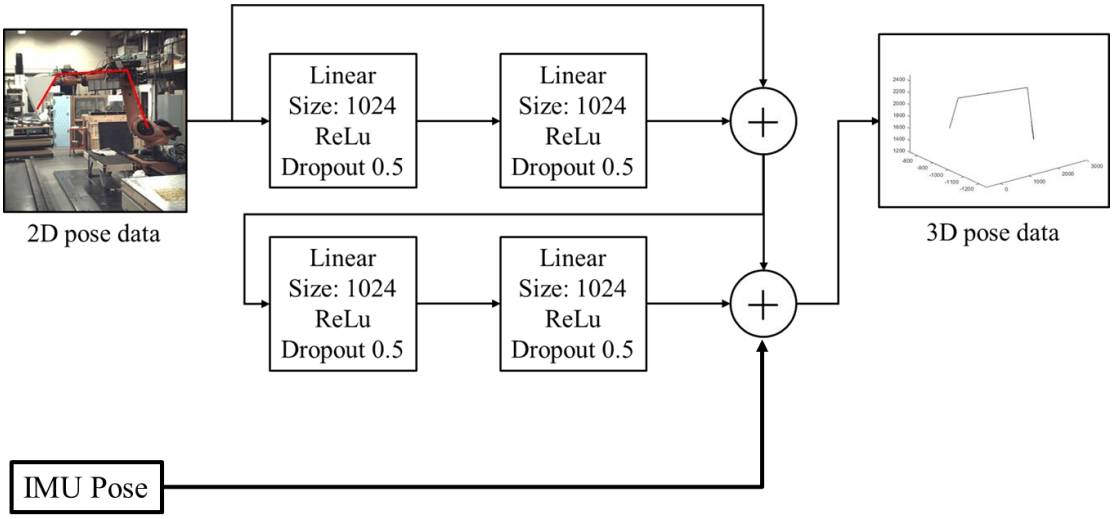


Figure 2.8 Fusion-Based 3D Pose Estimation Network Structure.

On the other hand, Kalman Filter methods are well-known in the sensor fusion domain to provide consistent, real-time, and uninterrupted data in high occlusion environments. Extended Kalman Filter (EKF) [143,146,147], Unscented Kalman Filter [151,152], and Information Filter [153] are usually utilized to fuse multi-sensor data for mobile robots. The EKF method is applied to fuse the vision pose data and the IMU pose data to compare with the DNN method.

The IMU pose data has high accuracy at the beginning of the trajectory, and the data stream latency is low but has drift issues over time. In addition, Vision pose data has high accuracy but requires more computing time, and the data stream latency is high. The occlusion is another issue that affects accuracy. Therefore, in the EKF method, the IMU pose data is used in the predicting step, and the vision pose data is used in the updating step to correct the sensor drifts. Figure 2.9 and Figure 2.10 show the flowchart and the equations of the EKF-based pose estimation. For each iteration, if the sensor data is unavailable such as missing data, the vision data will be used for the prediction step.

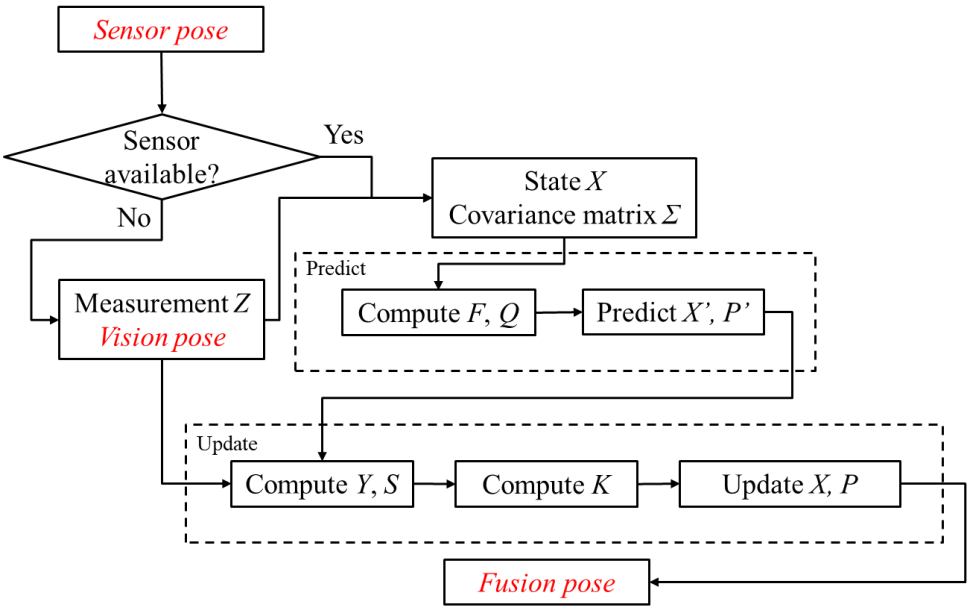


Figure 2.9 EKF-Based Pose Estimation Procedure.

– Predict
  - Predicted state estimate $X' = f(X, u_k)$
  - Predicted covariance $P' = F_k P {F_k}^T + Q$
– Update
  - Innovation $Y = z - h(X')$
  - Innovation covariance $S = H_k P' {H_k}^T + R$
  - Kalman gain $K = P' {H_k}^T S^{-1}$
  - Updated state estimate $X_{t+1} = X' + KY$
  - Updated covariance $P_{t+1} = (I - KH_k)P'$

X: mean state
P: state covariance
u: motion
Q: covariance of the noise
F: state transition function
h: observation
z: measurement
H: measurement function
R: covariance of the noise

Figure 2.10 EKF-Based Pose Estimation Equations.

2.4.4 Training Details and Implementation

For the 2D pose estimation, the L_2-norm loss function is used to train the network, as shown in Eq 2.1:

$$L_2(\hat{X}_P, X_L) = \sum \left(\hat{X}_P - G(X_L)\right)^2 \qquad \text{Eq 2.1}$$

where $\hat{X}_P$ represents the predicted pose and X_L represents the labeled ground truth training data, and $G(\cdot)$ represents the Gaussian kernel function with 1-pixel standard deviation. The loss function directly calculates the error between the training ground truth heat-map and the predicting heat-map and minimizes it.

The 2D network system is implemented by modifying the original network using PyTorch and the loss function Eq 2.1. The RMSprop method with learning rate 2e-4 is used for optimization. Batch normalization is used for the training process [154]. The network is trained with NVIDIA GeForce GTX 1060 graphic card on an excavator image dataset, which is collected from a factory setup laboratory environment with a simulated robotic excavator. The excavator dataset contains 2,500 training images and 500 testing images aligned with their 2D pose annotation. The detailed laboratory environment setup and data annotation are discussed in sections 2.5.

For the 3D pose estimation and the sensor fusion pose estimation, the L_2-norm loss function is also used to train the network, as shown in Eq 2.2:

$$L_2(X_{2D}, X_{sensor}, X_{3D}) = \sum (f(X_{2d}, X_{sensor}) - X_{3D})^2 \qquad \text{Eq 2.2}$$

where X_{2d} represents the input 2D pose data, X_{sensor} represents the input IMU pose data, and X_{3D} represents the labeled ground truth 3D training data, and $f(\cdot)$ represents the function that maps the 2D input data to the 3D prediction and fuses with the sensor pose. The loss function minimizes the prediction error between 3D prediction and 3D ground truth data. The L_2-norm loss function is derived from the loss function of the 3D human pose estimation baseline network [132]. If the sensor pose data is unavailable, the loss function will ignore X_{sensor} and not update the corresponding parameters.

The 3D network is implemented using TensorFlow and the loss function described in Eq 2.2. The Adam method with starting learning rate 2e-3 and exponential decay is used for optimization instead of starting learning rate 1e-3 [132]. Batch normalization is also used for the training process. The network is trained with NVIDIA GeForce GTX 1060 graphic card on the same image dataset collected from the laboratory with a robotic excavator. The 3D ground truth data is measured directly from the robot's embedded joint sensors.

The DNN model is implemented using PyTorch and trained with NVIDIA GeForce GTX 1060 graphic card on the lab dataset with robot ground truth and the site dataset with IMU pose data as ground truth. On the other hand, the EKF method is implemented using MATLAB and tested with the lab dataset. For evaluation, we compare the performance of the fusion-based pose estimation (both DNN model and EKF method) with vision-based, sensor-based, and ground truth data by the bucket 3D location.

2.5 Dataset Collection Approach

The image dataset is collected with an articulated robotic manipulator outfitted with a simulated excavator bucket. The dataset is separated into training and testing groups. The proposed networks are trained by the training group and then evaluated by the testing group.

2.5.1 Dataset Collection Setup

For the dataset collection setup, a KUKA 7 DOF robot arm (KUKA KR120) [155] in the Digital Fabrication Laboratory at the Taubman College of Architecture and Urban Planning was used to simulate the excavator, and the images of the robot arm with different poses were captured. Figure 2.11 illustrates the simulated excavator in the laboratory. The upper arm represents the excavator stick and the lower arm represents the excavator boom. A bucket is mounted on the robot arm end-effector for a more realistic simulation. In order to control the robot as an excavator, the profile of the mounted bucket must remain perpendicular to the ground level. Thus, only four of the robot joints were moved during the dataset collection process, and the others were fixed at all times.

The robot arm was controlled to follow trajectories to perform several excavator-like tasks such as digging, swinging, or unloading. The ground truth of the excavator pose data was acquired from the robot arm's embedded encoders, including 6 DOF pose of the robot's end-effector (X, Y, Z, A, B, C) and angles of all joints $(A_1, A_2, A_3, A_4, A_5, A_6)$. The joint angles were used to calculate the 6 DOF pose of the robot's joint.

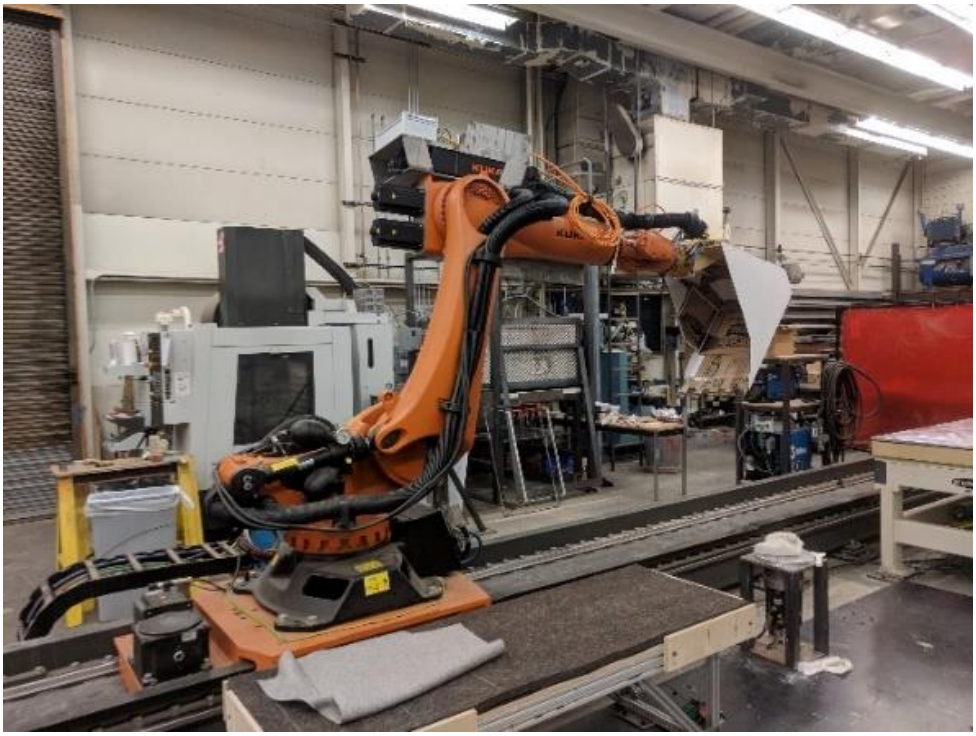


Figure 2.11 Simulated Robotic Excavator.

In order to collect the images of the simulated excavator, a Point Grey camera [156] was used in the process. The camera was mounted on a second KUKA robot arm in the laboratory, as shown in Figure 2.12. This could not only provide several different locations and orientations of the camera to increase the variety of the dataset, but also helped obtain the 6 DOF pose of the camera itself, which is the end-effector of the camera robot, for further processing. The mounted camera on the second robot arm was triggered by the same controller (Programmable Logic Controller, PLC) to control the first robot arm. Thus, the captured image and the recorded ground truth pose data were synchronized with each other. In the data collection process, a total of 2,500 images were collected; 2,000 of them were used as training images and 500 of them were used as testing images.

The data augmentation method was applied to increase the verity of the dataset to 3,000 training images [157]. The human pose benchmark dataset FLIC [158] is composed of 3,987

training images. In addition, the human pose is much more complicated than the excavator pose and constraint-free. The excavator has 1 DOF joints which are finite in number, and that reduces very dramatically the number of images needed for training. Figure 2.13 shows a set of the collected images from the dataset. The size of each image is 2048x2048 pixels.

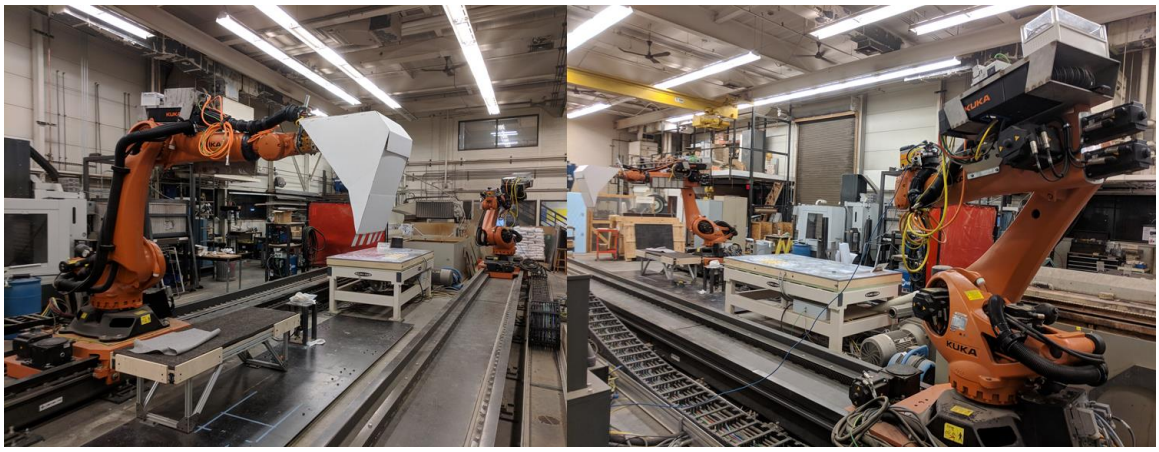


Figure 2.12 The Camera is Mounted on the Second Robot Arm to Capture Images.

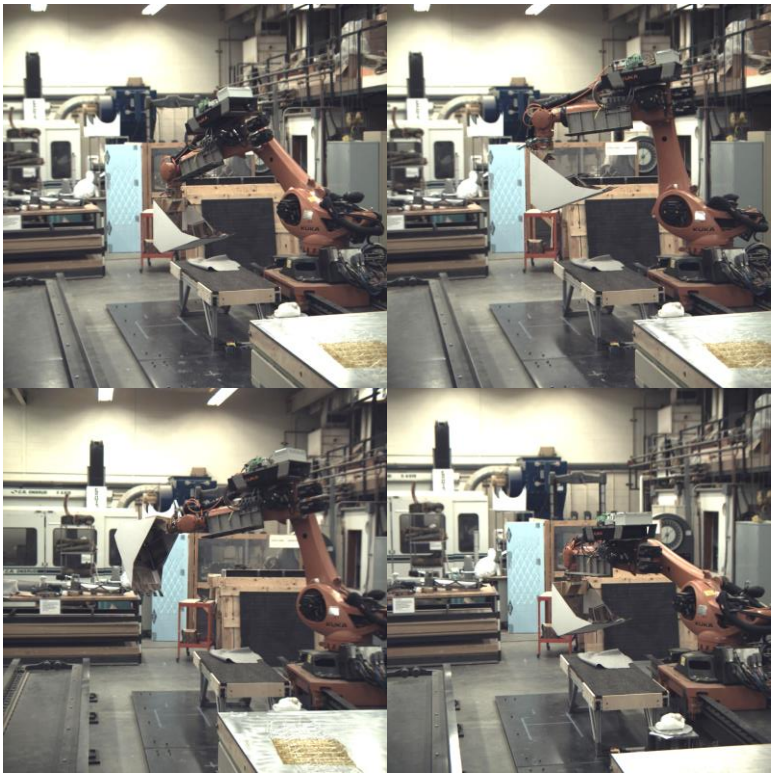


Figure 2.13 A Set of the Captured Images for the Excavator Dataset with Different Camera Locations, Orientation, and Excavator Pose.

In addition, to increase the variety of the dataset and vary the background of the dataset images, several images from outdoor construction sites with working excavators were also collected, as shown in Figure 2.14. The images contain a variety of excavator operations on different construction sites with single or multiple machines. These images were only used for evaluating the 2D pose estimation network since the 3D ground truth data could not be obtained from these images. To overcome the 3D ground truth issue, we mounted three IMU sensors on real excavators to collect the cabin, boom, and stick pose data as ground truth along with the video captures on real sites by stationary cameras and drone cameras. A total of 6,508 images were collected; 4,234 of them were used as training images and 2,274 of them were used as testing images. The size of the images was different and thus needed re-scaling and cropping to 1024x1024 pixels before inputting into the network.

Figure 2.14 A Set of Working Excavator Images from the Dataset.

2.5.2 Data Annotation

Data annotation is required in order to indicate the location of the excavator's joints in the images as the ground truth. The structure of the excavator data annotation follows a similar structure to the human pose dataset annotation, MPII for 2D pose [120] and Human3.6M for 3D pose [133]. In the 2D pose annotation, excavator joint locations were annotated in the pixel-wise coordinate. The visibility of each joint was also marked in the annotation data. The scale of the image was measured with respect to a height of 200 pixels.

On the other hand, in the 3D pose annotation, the locations of the excavator's joints were labeled as (X, Y) in pixel-wise coordinates and Z was considered as the depth value from the camera to each joint, which was calculated from the robot arm end-effector and joints' ground truth data. The bounding box was also labeled to show the area of the excavator in the image. The annotations were performed via MATLAB and saved as two separated annotation files, one for the 2D pose and the other for the 3D pose. Figure 2.15 shows an example of an annotated image.

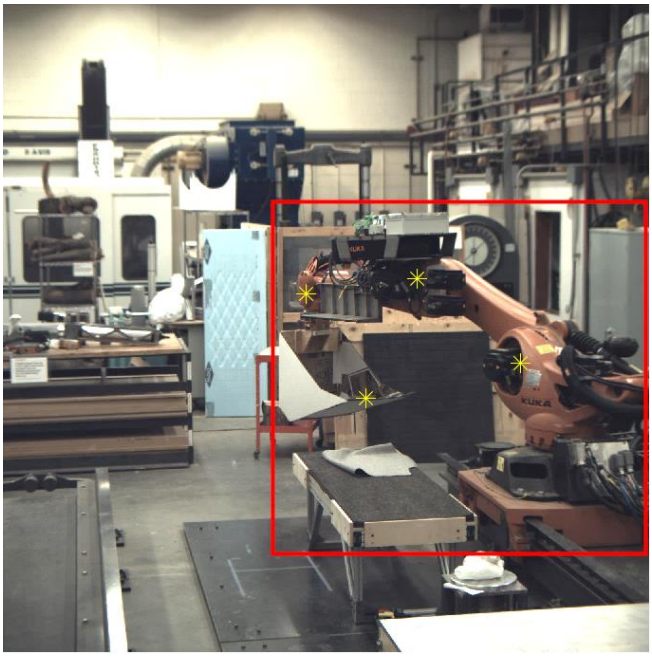


Figure 2.15 Example of the Annotated Image.

The 3D ground truth data was acquired by the robot arm's built-in encoders and the Programmable Logic Controller (PLC). Figure 2.16 illustrates the framework of the pose data and image acquisition. The PLC sent the control command to both robot arms (North and South). The South robot would perform the predefined trajectory, such as digging or unloading, whereas the North robot would stay as it is to capture the images. Several trigger points were set to trigger the camera on the North robot to capture the image and acquire the pose of both robots, and then transfer them to a computer. After the South robot finished the entire trajectory, the North robot would move to a different pose and re-run the process. This could increase the variety of the dataset by having different orientations in the images. The 3D pose of the end-effector was directly read from the robot arm, and the 3D pose of the rest of the robot joints was obtained using inverse kinematics.

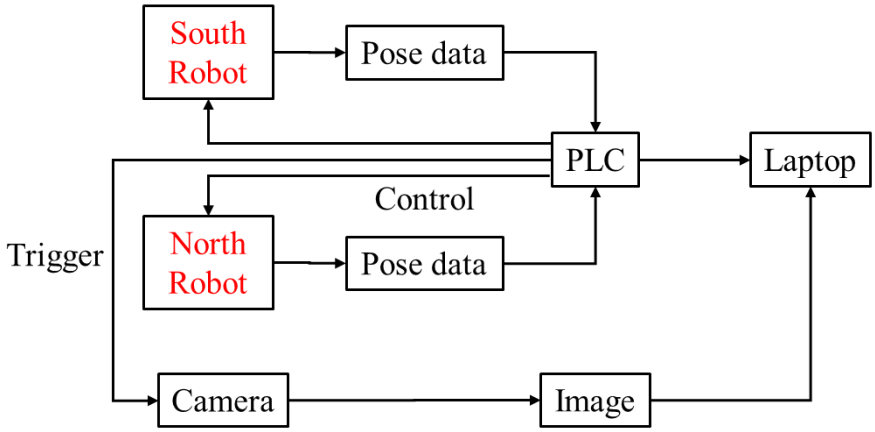


Figure 2.16 Framework of the Pose Data and Image Acquisition.

2.6 Experimental Results

The results of the pose estimation experiments are explained in the following subsections. The 2D vision-based method is first compared with the ground truth. Then, the 3D vision-based method, sensor-based method, and ground truth are compared with each other. Finally, the DNN-

based fusion method, EKF fusion method, 3D vision-based method, sensor-based method, and ground truth are compared with each other.

2.6.1 Results of the 2D Pose Estimation System

The proposed 2D network was evaluated by comparing the prediction results of the testing images and the ground truth. Figure 2.17 demonstrates the results of the excavator pose estimation. The lines represent the bucket, stick, and boom prediction. These images are estimated in the testing dataset. The proposed model is first evaluated by average PCKb@0.5, which reports the percentage of the distance between ground truth heat map and predicted heat map that is below 50% of the bucket segment length. The PCK (Percentage of Correct Key-points) is a well-known evaluation metric for measuring the performance of human pose estimation models [120]. Then, the Euclidean distance between the estimated joint location and the ground truth joint location in pixel-wise units and converted to mm units are used to evaluate the performance, and the error percentage of the predicted component length and the ground truth, which can be found in Table 2.3 and Table 2.4.

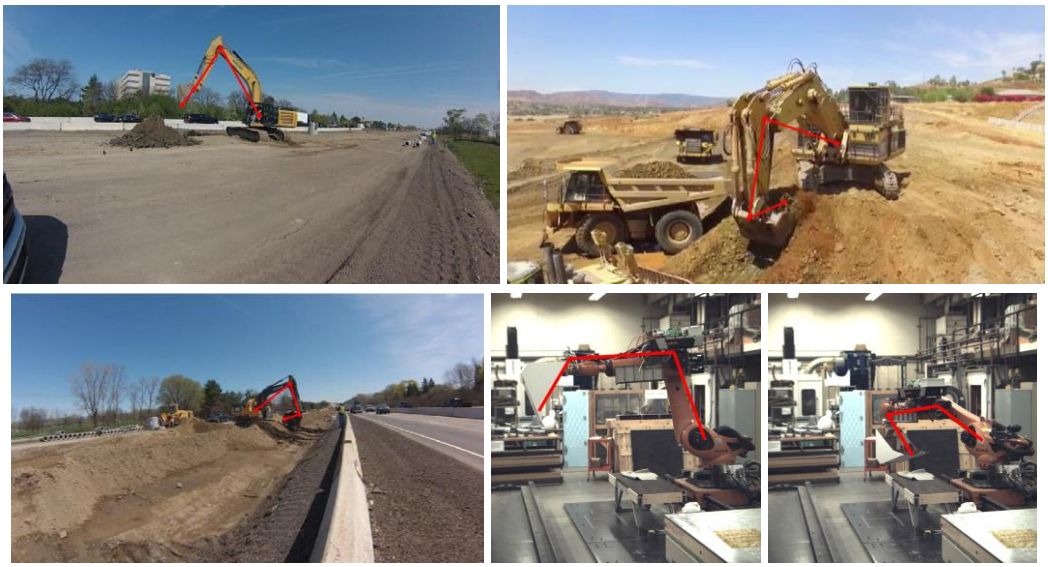


Figure 2.17 Results of the Excavator 2D Pose Estimation.

Table 2.3 Results of the Average Euclidean Distance Between the Predicted and the Ground Truth Joint Location.

(mm)	Laboratory Dataset	Real Site Dataset
Boom	31.58	777.12
Boom Stick	39.47	701.40
Stick Bucket	35.65	753.36
Bucket	55.84	1216.44

During the training phase of the entire dataset (lab and site dataset), the average PCKb@0.5 is 87.1% for the training dataset and 79.5% for the validation dataset. The average PCKb@0.5 is 71.3% for the entire testing dataset. Moreover, the average Euclidean distance between the laboratory testing dataset and ground truth is 40.64 pixels (image size is 2048x2048), and between the real site testing dataset and the ground truth is 71.84 pixels (image size is 1024x1024). In the laboratory dataset, the pixel size is measured by averaging the length of the robot arm across the entire dataset, which resulted in 1 pixel approximated to 1 mm. Therefore, the average Euclidean distance in the laboratory dataset can be converted to 40.64 mm. The distance between the camera and the robotic excavator is 10 m. The distance between the camera and the robotic excavator is 10 m.

In the real site dataset, some of the image data are collected by our team, and the distance between the camera and the excavator is measured by GPS. However, some of the distances of the image data are unknown since they are collected randomly online. Thus, it is difficult to convert the result from pixel to mm. The result is roughly converted to mm by measuring the length of the excavator stick in the testing image and calculating the ratio with the actual excavator stick length. The size of the excavator must be similar throughout the entire testing dataset. The stick size in the testing image is 40 pixels and the actual stick size is 2,500 mm, which resulted in 1 pixel

approximated to 12 mm. Therefore, the average Euclidean distance in the real site dataset can be converted to 862.08 mm.

The results show that the bucket location has the highest error because the bucket is blocked (occluded) or out of range in some of the images. The network still tries to find the bucket location in these cases, which increases the error distance. The error in the real site dataset is higher than the laboratory dataset. This is because the real site dataset has a greater variety of excavators and backgrounds. Only some of these variations were included in the testing dataset, which caused a decrease in accuracy. The number of images in the real site dataset is also insufficient for training purposes.

For the error percentage of the predicted component length and the ground truth, only the laboratory dataset is evaluated because the length of each robot arm skeleton is known, but some of the component sizes in the real site dataset are unknown due to occlusion. The results are shown in Table 2.4. The error percentage of the boom and stick is approximately 40% and 31%, and the bucket is 59%. The reason for the high error percentage in the bucket case is the occlusion issue. When the bucket is blocked or out of range in the image, the predicted bucket location will be far from its actual location.

Table 2.4 Results of the Error Percentage of the Predicted Component Length and the Ground Truth in the Laboratory Dataset.

(%)	Error Percentage of the Component Length
Boom	39.1
Stick	30.7
Bucket	58.8

In addition, the ground truth length of the bucket is short, which increases the differences between the ground truth and the false predicted result. Figure 2.18 shows the result of a false prediction of the bucket caused by occlusion. The excavator is partially blocked by another equipment, and the network mispredicts the bucket pose. The occlusion issue can be resolved by deploying multi-cameras system on construction sites to collect several video streams with different viewpoints or deploying sensor systems on the excavator to fuse the pose data.

Figure 2.18 False Prediction Result of the Bucket Due to Occlusion.

2.6.2 Results of the 3D Pose Estimation

The proposed 3D pose estimation method is first evaluated by comparing the prediction results and the ground truth of the laboratory dataset. Figure 2.19 shows the result of the 3D pose estimation. The left image is the result of the 2D pose estimation, which is the input to the 3D network. The right image was the 3D predicted result. The dashed line is the vision-based result, the dotted line is the sensor-based result, and the solid line is the ground truth. The average

Euclidean distance, i.e., the average mean square error, between the estimated joint location and the ground truth joint location in 3D coordinate is used to evaluate the performance, as shown in Table 2.5. Since the boom location is aligned together, it is not considered in the comparison. On the other hand, we also compare the average difference in the 3D coordinate between the IMU pose data and the vision estimated bucket location with different drone view angles on real sites.

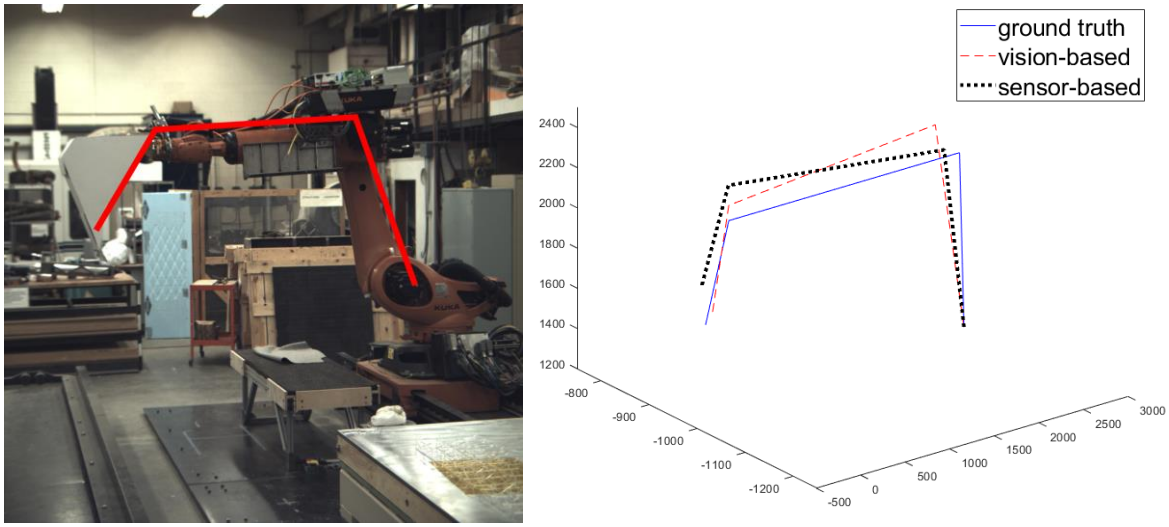


Figure 2.19 Results of the Excavator 3D Pose Estimation, Including the 2D (Left) and the 3D Result (Right).

Table 2.5 Results of the Average Euclidean Distance Between the Predicted and the Ground Truth Joint Location.

(mm)	3D Vision-based	Sensor-based
Boom	--	--
Boom-Stick	148.16	84.35
Stick-Bucket	134.22	97.21
Bucket	151.58	99.42

The overall average Euclidean distance between the 3D vision-based method and ground truth is 144.65 mm (distance between the camera and the robotic excavator is 10 m), and between the sensor-based method and the ground truth is 93.66 mm. The results show that the error of the 3D vision-based is higher than the sensor-based method. One of the reasons is that the 3D vision-

based method predicted the pose based on the 2D pose estimation result, wherein the error would accumulate from 2D prediction and decrease the accuracy in the 3D prediction. The other reason is that the camera coordinates preprocessing mentioned in [132] was not applied to the ground truth data because the camera matrix is not determined in the laboratory dataset. In addition, the occlusion issue also affects the prediction result similar to the 2D results. The error caused by the occlusion also accumulates from the 2D pose estimation results, especially for the bucket.

Second, the bucket pose estimation accuracy is evaluated by comparing the estimated bucket location with the sensor-based result and the ground truth. In the laboratory dataset, a sequence of the excavator trajectory is repeated ten times and is captured with different camera orientations, as demonstrated in Figure 2.20. A total of 16 images are captured in the trajectory yielding a total of 160 images that are used in the evaluation. The average pose of each of the 16 data points is calculated and compared between pose estimation methods.

Figure 2.20 A Sequence of the Excavator Trajectory.

Figure 2.21 shows the results of the pose estimation. The star-line is the 3D vision-based result, the circle-line is the sensor-based result, and the cross-line is the ground truth. The error of the 3D vision-based method is larger than the sensor-based method at the beginning of the trajectory. The sensor-based pose is closer to the ground truth pose than the vision-based pose before data 5 in X and Y location. After data 6, the sensor-based pose has a higher error than the

vision-based pose. The error of the X and Y location in the sensor-based result increased over time. The difference in the Z location in sensor-based and vision-based pose does not change significantly. This is because the drift occurred in the heading direction (Yaw). The sensor system has a stabilizing mechanism to calibrate the sensors. The earth's magnetic field is used to stabilize the heading but is susceptible to disturbance by artifacts such as nearby metal objects.

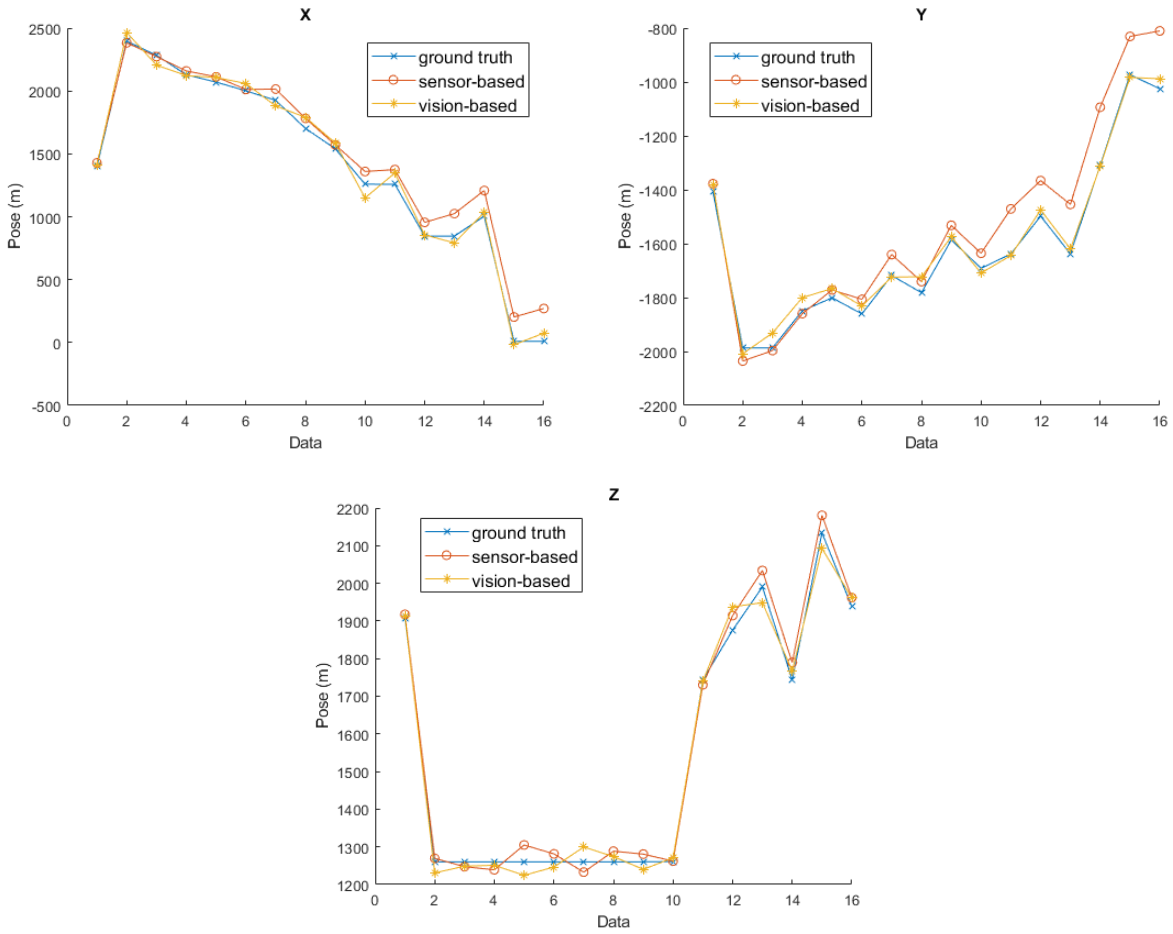


Figure 2.21 Results of the Bucket 3D Pose Estimation.

In addition, the cumulative error of the bucket 3D vision-based pose estimation is illustrated in Figure 2.22. The straight line is the error in X-axis, the cross-line is the error in Y-axis, and the circle-line is the error in Z-axis. The cumulative error along the X- and Y-axes is higher than the cumulative error along the Z-axis since the movement of the excavator bucket in

the data points is larger in the X and Y direction. In addition, the cumulative error along the X-axis is much higher than along the Y- and Z-axes. This is because the X direction has a higher projection in the camera viewing direction and the movement in such direction is difficult to identify by a single camera. Moreover, the Z direction is tangent to the viewing direction of the camera (pointing up), which has a larger displacement in the image and results in better performance.

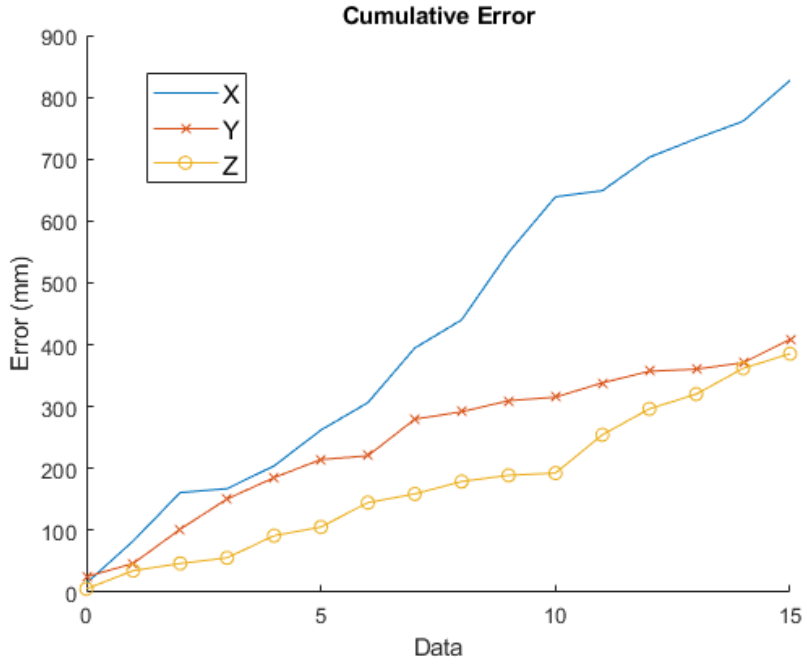


Figure 2.22 Cumulative Error of the Bucket 3D Vision-Based Pose Estimation.

Finally, different drone view angles for 3D pose estimation are compared with each other. The drone is controlled to collect videos of excavation works with three different view angles, i.e., 45°, 70°, and 90°. Figure 2.23 three different drone views with different angles (45°, 70°, and 90°). The IMU pose is used as ground truth for training and evaluating. Table 2.6 shows the results of the average bucket 3D pose estimation difference with different drone view angles. The average difference of the 45° angle results in X: 198.7 mm, Y: 184.5 mm, and Z: 199.6 mm. The average

difference of the 70° angle results in X: 181.1 mm, Y: 202.8 mm, and Z: 178.8 mm. The average difference of the 90° angle (top view) results in X: 170.4 mm, Y: 183.2 mm, and Z: 334.5 mm. The Z direction of the 90° view angle has the highest difference between the estimated pose and the ground truth data since the axis is parallel to the camera view direction.

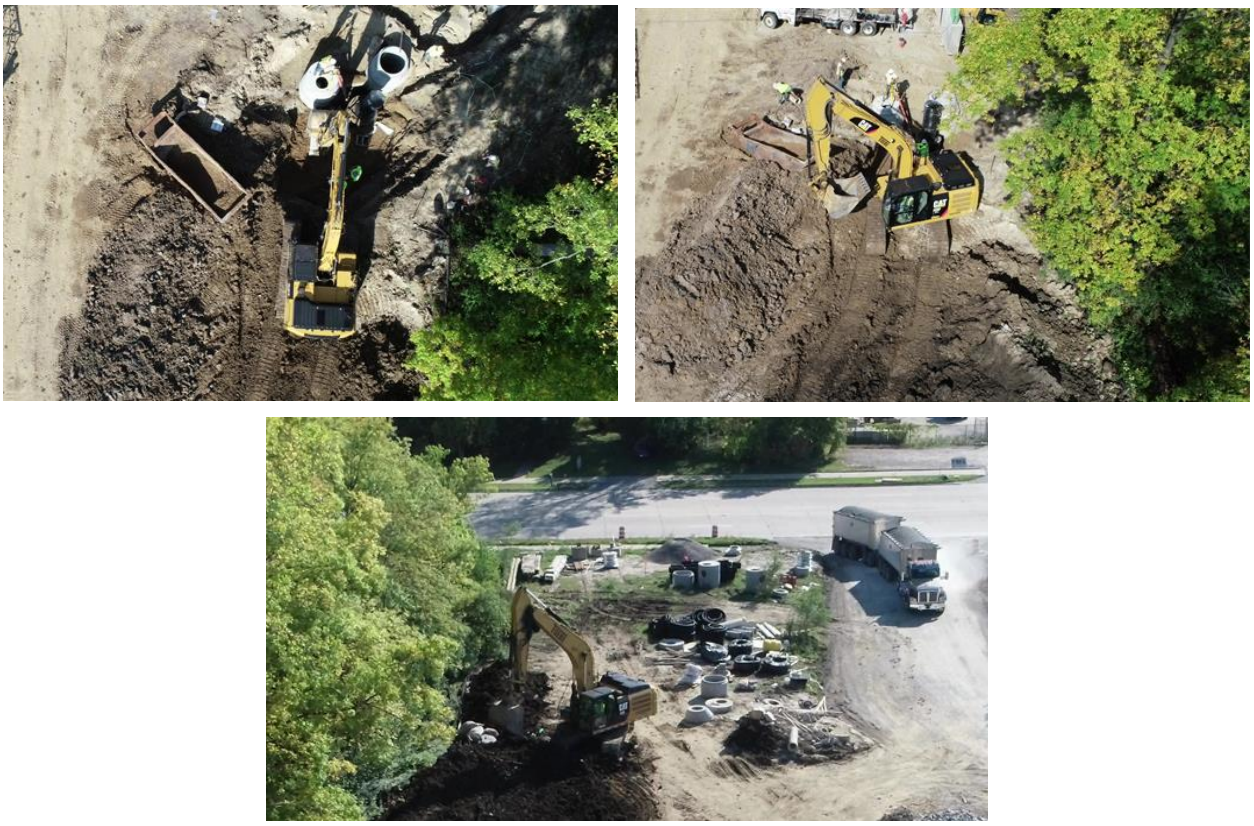


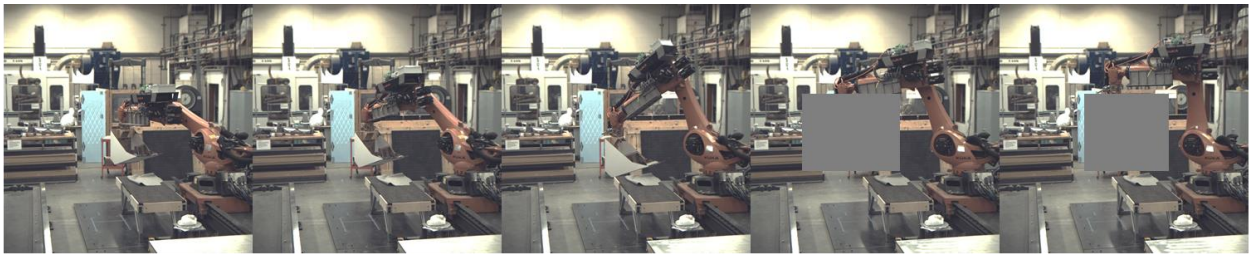
Figure 2.23 Different Drone Views with Different Angles.

Table 2.6 Results of the Average Bucket Pose Error with Different Drone View Angles.

View Angle	X (mm)	Y (mm)	Z (mm)
45°	198.7	184.5	199.6
70°	181.1	202.8	178.8
90°	170.4	183.2	334.5

2.6.3 Results of the Sensor Fusion Pose Estimation

In the lab dataset, a sequence of the robotic excavator trajectory is repeated ten times and is captured with different camera orientations. A total of 16 images are captured in the trajectory yielding a total of 160 images that were used in the evaluation. Figure 2.13 shows one set of images. Parts of the images and sensor data are manually blocked to evaluate sensor-fusion performance. In each 16 data points, 1-3 and 7-9 are all clear. For 4-6 data points, the sensor data are missed. For 10-12 data points, the bucket area in the image is blocked by a mean dataset color rectangle. For 13-16 data points, both image and sensor data are blocked. Figure 2.24 shows a set of testing data with occluded images and missed sensor data.



Data Point	1-3	4-6	7-9	10-12	13-16
Vision	Clear	Clear	Clear	Occluded	Occluded
Sensor	Received	Missed	Received	Received	Missed

Figure 2.24 A Set of Testing Data with Occluded Images and Missed Sensor Data.

The average difference in the 3D coordinate of the bucket location is used to evaluate the fused location with vision-based, sensor-based, and ground truth data. Figure 2.25 shows the results of the bucket pose estimation by DNN fusion-, EKF-, vision-, and sensor-based methods. The average difference of the DNN fusion method is X:54.3 mm, Y: 57.0 mm, and Z: 19.5 mm, and the EKF-based method is X: 65.3 mm, Y: 88.1 mm, and Z: 40.3 mm. For the vision-based

method, the average difference is X: 113.1 mm, Y: 117.6 mm, and Z: 75.3 mm. For the sensor-based method, the average difference is X: 84.0 mm, Y: 83.1 mm, and Z: 38.5 mm.

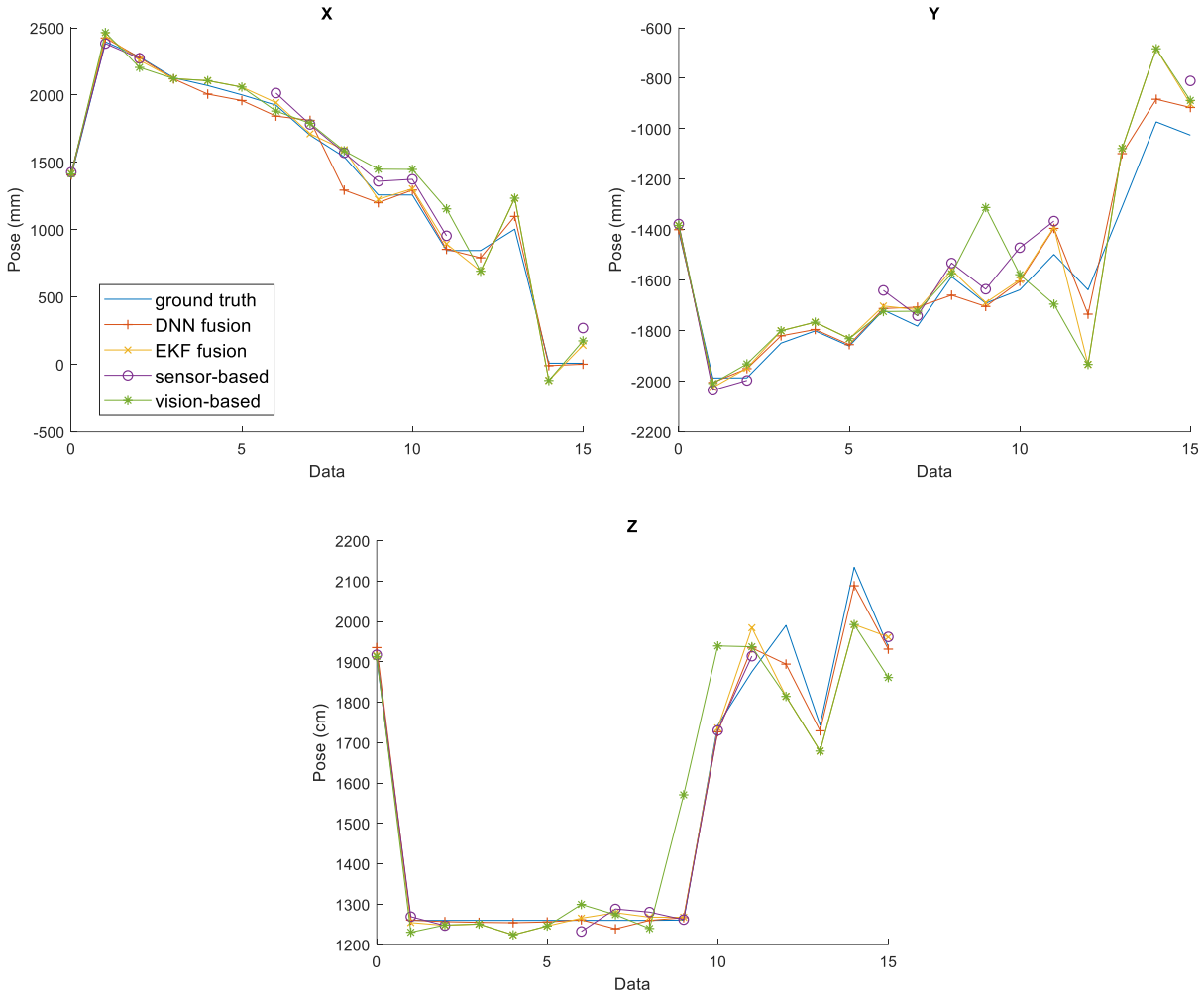


Figure 2.25 Results of the Bucket Pose Estimation by DNN Fusion-, EKF-, Vision-, and Sensor-Based Method.

The occluded results are compared with the results of the original data. The errors of both the sensor-based method and the vision-based method are increased due to the blocked view and data missing. The sensor-based method accumulates the error over time due to the drift issue. Both DNN fusion and EKF-based could improve the pose estimation performance even when some of the data are missed or occluded. The DNN fusion method has a lower error than the EKF-based

method since the EKF-based method utilized only the vision pose data to estimate the pose when the IMU data is unavailable.

2.7 Discussion

Based on the evaluation results, occlusion is the primary issue of the proposed vision-based method, which can potentially be addressed by increasing the number and variety of the training dataset. Dataset augmentation and expansion techniques can also help address this issue. Another problem is the multiple-machine situation. The proposed network can only identify one machine's pose. If there are two or more articulated machines in the image, the result is likely to fail. The other issue is the accumulated error from the 2D pose estimation result. The proposed 3D pose estimation network utilizes the 2D pose estimation results as input to predict the 3D pose, which results in accumulated errors. Therefore, a new network or method for the multiple-machine situation and the 3D pose direct training can be designed in future work.

The accuracy of the proposed 2D pose estimation method is 40.64 mm in the laboratory dataset, which is acceptable for the object detection and tracking and the proximity detection application discussed in Table 2.2. On the other hand, the accuracy of the proposed 3D pose estimation method is 144.65 mm and the DNN fused pose estimation method is 43.6 mm, which are not adequate for preventing utility strikes, grade control, and autonomous excavation applications, even though it may be acceptable in proximity detection applications.

The camera distance is important for pose estimation on construction sites. The scope of some existing human pose estimation methods is not suitable for the articulated construction robot due to short camera distance. The typical excavator working range is within 6.1 m, according to Lundeen et al. [71]. Thus, the performance of the articulated construction robot pose estimation

should be evaluated over 6.1 m for the camera distance. The camera distance in the evaluation of the proposed method is 10 m.

The proposed pose estimation method has three limitations. First, the network trained on the laboratory dataset is unable to achieve high performance when applying to an excavator operating in the field. The background and the light conditions of the laboratory dataset do not have a wide variety since they were collected in the same indoor environment, compared to actual construction sites where such conditions may vary. Second, the latency of the proposed method is affected by the hardware specifications and the complexity of the network structure, which would need extra cost for the advanced hardware in order to achieve outstanding performance.

Third, the system assumes the consistency and quality of the source video stream, which is not always available in real practice, especially on hazardous and unstructured construction sites. Further research of the data consistency on construction sites needs to be conducted to explore this issue. Fourth, the proposed 3D pose estimation method is trained and evaluated on the laboratory dataset and some real site images due to the lack of the 3D ground truth data for the real site dataset. Future work on augmenting the real site dataset with ground truth data from onboard sensors of the excavator or exploring a new network to train without ground truth data needs to be conducted.

2.8 Conclusions and Future Work

In this research, vision-based marker-less 2D and 3D pose estimation methods and DNN-based fusion pose estimation methods for articulated construction robots were proposed, in which an excavator was used as the experimental machine test-bed. The excavator boom, stick, and bucket joint positions are estimated with both 2D and 3D coordinates. State-of-the-arts human pose estimation deep convolutional networks, i.e., the stacked hourglass network and baseline network, were adapted and modified for the application. The network model was trained on an

excavator dataset, which was collected and annotated with a KUKA robot arm representing an excavator and from real construction sites with working excavators. The IMU sensor-based pose estimation method was also implemented to evaluate the performance of the proposed network. The results showed that the proposed network could estimate the boom and stick joints but had higher estimation errors for the bucket location due to typically encountered occlusion issues.

To overcome the occlusion issues, the DNN-based fusion method was proposed to combine the vision-based and the sensor-based pose data to estimate the uninterrupted and accurate pose. The pose data were integrated into the last residual module. The network was trained on the excavator dataset and tested on some occluded image data and signal blocked data. An EKF-based fusion pose estimation method was also implemented to compare the performance. The results showed that the proposed fusion network could achieve a higher accuracy than the vision-only and sensor-only methods in the occluded scenario.

Moreover, the accumulated error in the 3D pose estimation resulting from the 2D predicted pose input needs to be resolved as well. Therefore, in proposed future work, additional training image data with greater variety will be collected. A further modification of the proposed network will also be explored to adapt to the multiple-machine situation and address the accumulated error issues. Finally, the data consistency on construction sites will also be considered and surveyed.

Chapter 3

Robot Learning from Human Demonstration for Quasi-Repetitive Construction Tasks

3.1 Introduction

Applying robots on construction sites is one of the current trends to resolve construction safety, productivity, and quality [40]. In the United States, the fatal injury rate continuously grew in the past decade and reached the highest fatalities in 2019 [159]. Such safety issues cause high costs and delay to the construction project. Statistics show that the construction industry accounts for 5% of all industry annual employment but 15% of all industry annual injury costs [160]. Accidents on construction sites are one of the significant reasons for construction project delays since entire sites have to be shut down and suspended for recovery or investigation [161]. On the other hand, productivity and quality are highly dependent on the skill and experience of human workers. Manual errors happen occasionally and decrease the productivity and quality of the construction project [37].

The adoption of construction robots on job sites has demonstrated improvement in the safety, the productivity of projects, and the quality of work [162]. Similar to applications in disparate fields such as manufacturing and surgery, where a robot can assist with repetitive or precise work in a narrow workspace, robots on the construction site can assist with physically demanding and repetitive construction tasks. However, unlike manufacturing or surgery robots, where the robots are typically placed at stationary locations to perform work by pre-programming

or tele-operating, on-site construction robots have to navigate to different locations in an unstructured environment to perform work that is often susceptible to loose tolerances and discrepancies between the designed and built versions [11]. It is therefore impractical to pre-program a robotic construction work plan or to define it as an optimization problem. In addition, tele-operating robots to complete construction tasks requires significant training of operators that must include both construction experts as well as robot technicians [163].

3.1.1 Challenges in On-Site Construction Robot Deployment

Even though a series of construction tasks are often similar, there are distinct reasons why they cannot be considered purely repetitive, particularly when considered from a robot's perspective. For example, in the ceiling tile installation process, several tiles are sequentially installed into grid openings in a ceiling frame. Even though the dimensions of the tiles and the frame openings may be nominally identical, there are subtle differences in the installation process of each subsequent tile. For instance, each tile is installed in a different, albeit adjacent, opening in the ceiling frame and no two tiles are placed at the same location.

In addition, even though the motions involved in installing tiles in a frame may be generally similar, the geometry of the obstructions above such a frame (e.g., ducts or other utilities) may be dramatically different in each frame cell. This requires the basic motions involved in maneuvering a tile above the frame and dropping it onto the supports to be uniquely dependent on the cell geometry. Such construction tasks that are conceptually similar but differ in location and required elemental motions are defined as quasi-repetitive. The distinct quasi-repetitive characteristic of construction tasks makes the deployment of on-site robots particularly challenging. For instance, the subtleties of ceiling tile installation tasks described above are intuitively overcome by human

workers but are extremely challenging, if not impossible, to pre-program as a set of installation instructions for robots.

3.1.2 Robot Apprentices Learning from Human Experts

Imitation learning or Learning from Demonstration (LfD) methods eliminate the requirement of pre-programming or tele-operation to control a robot to accomplish a task. Instead, these methods enable the robot to imitate the behavior of human experts directly [30]. The human worker and the robot coexist in the workspace to teach and perform the construction task respectively. Human experts demonstrate the task to the robot during the teaching process, and the robot generates models to reproduce the task under similar yet non-identical circumstances.

In the performance phase, the robot first observes the scene to determine the start and target locations through scene understanding methods or human instructions [11,21]. Then, the robot uses the model to reproduce the task based on the encountered circumstance under the human worker's supervision. For typical installation tasks, an experienced worker can pick up and install construction components in desired locations while the robot observes the procedure and learns the model. Then, the robot reproduces such installation tasks at different locations with similar components.

This procedure of teaching construction robots is, in some aspects, comparable to a construction apprenticeship program involved in training new construction workers [29]. The novice construction workers follow instructions from veteran experts and develop their skills by observing and practicing the craft. They complete the necessary training and and are evaluated through examinations before being qualified as independent construction craftworkers. The robot imitation learning or LfD methods have a learning structure that parallels such an apprenticeship program. The robot develops a model for performing work by observing demonstrations from

expert workers and practicing the skills through supervision, thereby developing the capability to adapt the skills to other similar work contexts.

Based on the success of robots learning industrial assembly tasks such as pick-and-place and bolt-screwing from demonstration [164], the application of the LfD method in construction provides opportunities to train robots to perform work semi-autonomously (i.e., not pre-programmed or tele-operated) in partnership with human workers, specifically on quasi-repetitive tasks on unstructured construction sites. Such collaboration between humans and robots provides a sustainable and scalable model for robot deployment in construction in the upcoming future where robots' training on quasi-repetitive tasks can be easily transferable to other projects in the future.

The remainder of this chapter is organized as follows. First, existing robot LfD methods are identified and discussed in the context of construction applications. Second, a visual LfD-based context translation method for construction task learning is introduced. A Reinforcement Learning (RL)-based robot control method for construction task performance is developed to utilize the translated context to generate the control policy and perform the construction task. Third, a trajectory LfD method, i.e., generalized cylinder with orientations approach (GCO), is developed. A trajectory adaptation approach and a human-in-the-loop refinement approach are developed in the GCO. Lastly, the ceiling tile installation experiments are conducted and used to evaluate the proposed robot LfD methods.

3.2 Related Work

In this section, the existing works on robot Learning from Demonstration (LfD) are reviewed and discussed based on the demonstration methods and the learning methods. The

imitation from observation (IfO) methods and the trajectory-based LfD methods are discussed for construction applications.

3.2.1 Demonstration Methods

Robot learning from demonstration (LfD) or imitation learning methods enable a robot to acquire new skills by imitating observed demonstrations from human experts [30], and is an advantageous approach when the involved skills can neither be pre-programmed nor expressed as optimization problems [31]. LfD methods are typically applied to manipulation tasks or assembly tasks [164,165] that can be easily demonstrated by human experts. Existing LfD methods can be categorized based on the demonstration methods or the learning methods [31,32]. The demonstration methods are concerned with how the skills are demonstrated to the robot, including trajectory demonstration and passive observation. On the other hand, the learning methods are categorized based on how the skills are being learned by the robot, which includes probabilistic approach, dynamic system approach, and reward-based approach.

The demonstration methods are concerned with how the skills are demonstrated to the robot, including trajectory demonstration and passive observation. On the other hand, the learning methods are categorized based on how the skills are being learned by the robot, which includes probabilistic approach, dynamic system approach, and reward-based approach. Figure 3.1 shows the categories of the demonstration methods. In the kinesthetic demonstration, the human expert demonstrates the task by manually moving the robot to the desired waypoints [166]. The onboard sensors or external sensors are used to record the trajectory data, including joint angles, end-effector poses, and motor torques [167–170]. The kinesthetic demonstration provides an intuitive way to interact with the robot and eliminates the correspondence problem [30] but is limited to lightweight and small object manipulation [164]. Thus, it is challenging to apply kinesthetic

demonstration to construction tasks where the objects being manipulated are typically heavy and oversized.

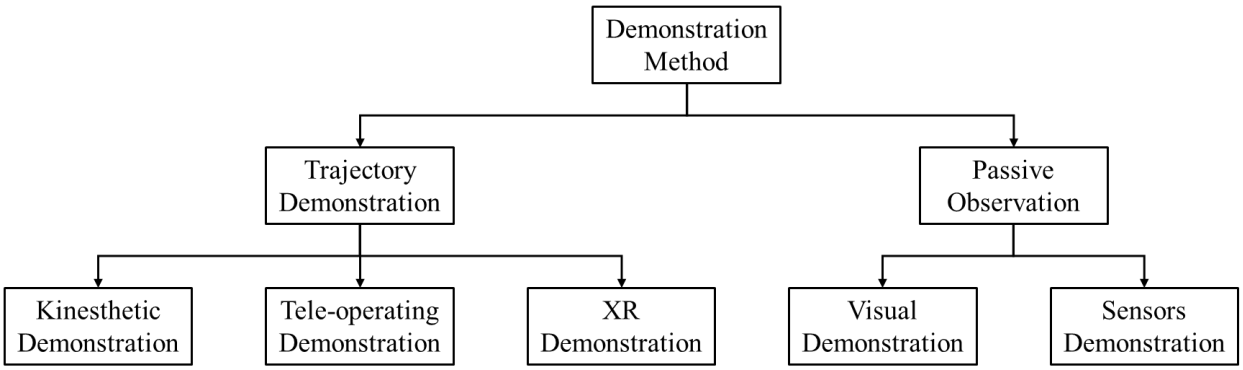


Figure 3.1 Categorization of the Demonstration Method

In the tele-operating demonstration, the human expert controls the robot with a remote controller to demonstrate the task [171,172]. Similar to the kinesthetic demonstration, the onboard sensors or external sensors such as haptic sensors on the controller are used to record the trajectory data [170]. The tele-operating demonstration supports broader applications such as autonomous helicopter learning [173], mobile robot positioning [174], and hierarchical tasks learning [175] due to straightforward and efficient communication of control from the human. However, in some complicated construction tasks such as drywall installation, it is not intuitive to control a robot to complete the task using a joystick and requires additional human effort.

Lastly, in the extended reality demonstration, the use of virtual reality (VR) or augmented reality (AR) provides various scenarios for human experts to demonstrate tasks to the robot [176,177]. The human expert can either control the virtual robot to demonstrate the task [178] or directly demonstrate the task in the virtual environment and record the video [179,180]. The trajectory of the robot or manipulated object can be recorded easily inside the controlled virtual environment to construct the demonstration dataset. With the assistance of extended reality, the

human expert can demonstrate different construction tasks inside the virtual environment with different components and backgrounds.

On the other hand, passive observation is the second group of demonstration methods, and allows human experts to demonstrate the task directly and utilizes sensors or cameras to collect the demonstration data. The passive observation is a particularly intuitive way for the human expert to demonstrate the task, and the robot is not involved during the demonstration phase. This type of method includes visual demonstration and sensor demonstration. In the visual demonstration, the videos of the demonstration are collected for the robot using camera or motion sensors, and then used to extract features from video frames or track the motion of humans or objects in the scene [181,182]. Visual demonstration is susceptible to typical computer vision-related challenges such as occlusion.

Motion capture systems provide accurate human whole-body motion data and can record various human motions such as lifting objects for demonstration [183]. However, such systems have limited applicability for deployment in dynamic and constantly-changing construction environments. In the sensors demonstration, multiple sensors are used to collect demonstration data, such as tactile sensors or motion sensors [172,184]. The trajectory of the human expert's movement or the contact force between the human and the object is collected. Furthermore, the sensor demonstration can also be combined with the visual demonstration to provide abundant and informative demonstration data [169,184,185]. This type of demonstration method is usually applicable to tasks requiring contact forces, e.g., fastening bolts, but needs additional data mapping approach to ensure the correspondence.

3.2.2 Learning Methods

The probabilistic approach, dynamic system approach, and reward-based approach are three subgroups of the learning method, as shown in Figure 3.2. The probabilistic approach is the first group of learning methods that encode the feature using probabilistic representations and learn the policy. Hidden Markov Model (HMM) is the common probabilistic method applied in LfD to learn the skill by regression [164,166,186]. In addition, HMM method can be combined with other probabilistic methods to obtain more reliable learning outcomes, such as the combination of Gaussian Mixture Regression (GMR) and Gaussian Mixture Model (GMM) to obtain smooth trajectory [166,168,187]. However, these methods usually require extensive parameter tuning to get robust manipulation [188]. In order to minimize the parameter tuning process, Ahmadzadeh and Chernova [188] developed the Generalized Cylinders-based LfD method to generate trajectories within the geometry.

The dynamic system approach is the second group of learning methods, which utilizes nonlinear dynamic systems to represent demonstrations and generate trajectories. The Dynamic Movement Primitives (DMP) method uses the spring-damper model to represent the demonstration and GMM to learn the movement [189,190] but also requires a notable parameter tuning process. Stable Estimator of Dynamical Systems (SEDS) optimizes the parameters of the dynamic system to imitate the demonstration as a function of the velocity data [191].

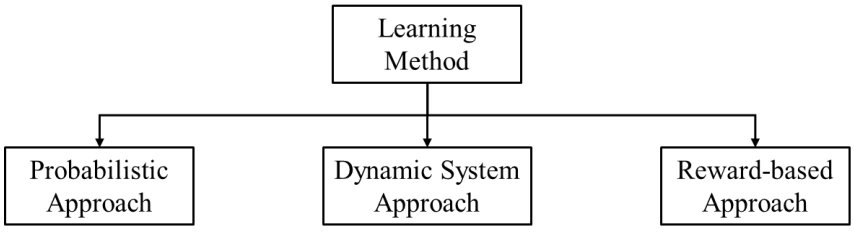


Figure 3.2 Categorization of the Learning Method.

Finally, the reward-based approach is the third group of learning methods, which defines a reward or cost function and optimizes the policy with maximum reward or minimum cost. However, it is difficult to define a reward or cost function for the LfD method since it requires assumptions about the task and the workspace [31]. Behavior cloning or trajectory optimization approaches directly use expert demonstration data to learn the policy with the assumption that the expert always provides optimal solutions to the task, and thus the hidden cost function in the demonstration data is minimum [192,193].

Inverse reinforcement learning (IRL) methods first infer a hidden reward function using demonstration data, then apply reinforcement learning (RL) methods to determine the policy based on the inferred reward function [173,194], which requires significant computational time. Recent advances in IRL methods combine the IRL structure with other methods to reduce the computational effort. Generative adversarial imitation learning (GAIL) [195,196] is one of the examples of combining IRL with generative adversarial networks (GAN) [197].

Imitation learning from observation (IfO) is the special form of LfD where the robot only has access to state demonstrations, i.e., visual observation, instead of the state-action demonstrations, i.e., visual observation with expert's action [32,181]. The challenge of the IfO is how to extract actions from demonstration states. The dynamics model is the first type of the IfO method [198,199] that learns the action from state demonstration using forward or inverse dynamics models. For example, one of the inverse dynamics method extracts actions a_t from state transitions (s_t, s_{t+1}) in the demonstration video to train the model, then used it to determine the actions for robots to follow. Reinforcement learning (RL) is the second type of the IfO method that applies GAN to learn the policy [200] or manually defines the reward function [181,201,202].

In this research, the visual demonstration and reward-based IfO method is first adapted and applied to teach robots the quasi-repetitive construction task, i.e., ceiling tile installation process. Second, the trajectory demonstration and probabilistic approach, i.e., Generalized Cylinders with Orientation approach, trajectory adaptation approach, and human-in-the-loop refinement approach, for teaching robots the same quasi-repetitive construction task is developed and compared with the first IfO method.

3.3 Research Goal and Contribution

The objective of this research is to investigate the robot LfD method [181,188] and evaluate the feasibility of applying visual LfD and trajectory-based LfD for teaching quasi-repetitive construction tasks to robot apprentices. The visual demonstrations and the trajectory demonstrations of the construction task by human experts are provided to the robots to first learn the knowledge by extracting state-action pairs and then perform the task to evaluate its performance. Trained robots can then collaborate with human workers to perform physical tasks while human workers focus on the planning and cognitive aspects of the work.

Four specific research questions are pursued in this work: (i) To what extent can a robot learn a quasi-repetitive construction task through visual demonstration or trajectory demonstration by a human expert; (ii) What is the effect of the viewpoint of observed demonstrations on the subsequent robot learning and performance; and (iii) What is the general relationship between the number of visual demonstrations provided and the corresponding learning and performance of a quasi-repetitive construction task by a robot. (iv) How to adapt to the new start and target locations in different scenes.

The installation of suspended ceiling tiles, which fits the definition of quasi-repetitive construction tasks where the encountered geometry is different in each instance, is used as the

target construction process to describe and validate the developed methods. Figure 3.3 illustrates the procedure of installing ceiling tiles by a human worker or a robot. First, the worker or robot navigates to the desired assembly location based on their interpretation of digital maps (e.g., Building Information Models) and specifications [21]. Second, the worker measures the ceiling grid layout by tape measure, whereas the robot measures the layout and the geometry of the workspace using its sensors. Third, the worker or robot maneuvers, positions, and places a tile at the target grid location. Finally, the worker or robot inspects the alignment of the tile.

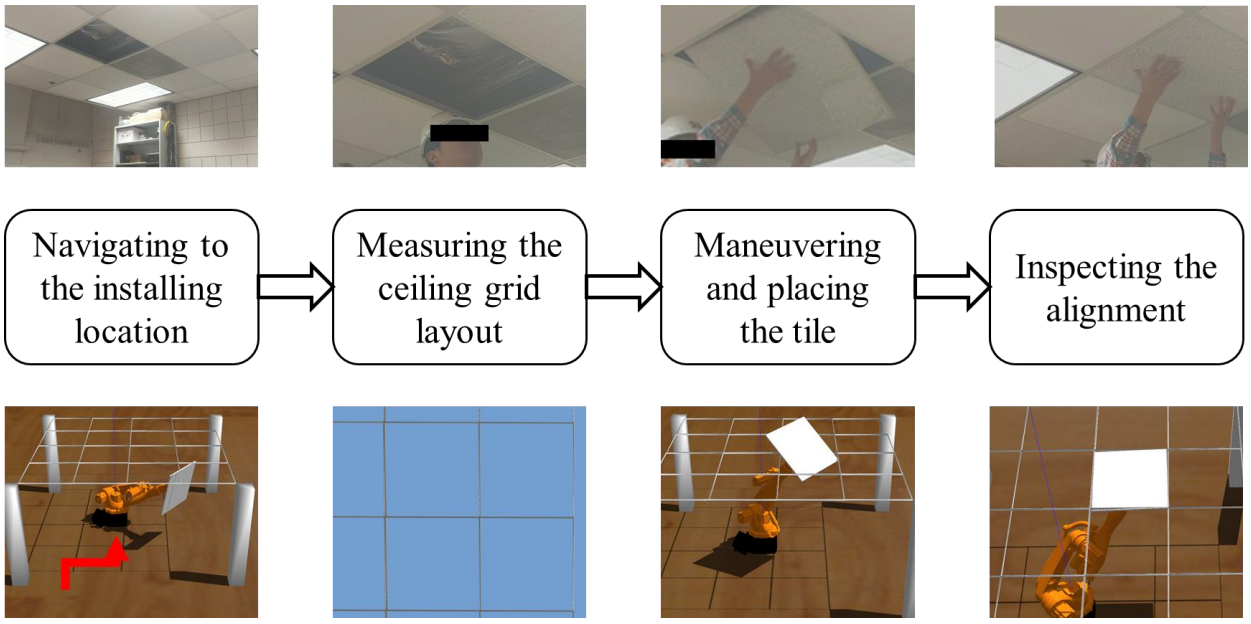


Figure 3.3 Procedure of Installing a Ceiling Tile by a Human and a Robot.

This research specifically focuses on the third step in the procedure and assumes that the robot is able to navigate to the correct location and gathers the correct geometric information using mapping and adaptive manipulation methods such as those developed in our prior work [11,203]. The research is conducted in the Gazebo robotics simulator using Robot Operating System (ROS) tools [204,205] and a KUKA mobile industrial robot arm emulator [155]. Virtual simulators allow

rapid prototyping of new robotic algorithms and methods such as the ones developed, and allow the testing of their feasibility and iterative adjustments based on the achieved results [206].

This research has three main contributions. First, this research discusses the challenges of the on-site construction robot deployment and identifies the need for robot learning and adapting construction tasks. Second, this research explores the existing robot Learning from Demonstration (LfD) method and rigorously extends and adapts them to a new application in construction, where the visual demonstration method, i.e., the context translation model [181] and Trust Region Policy Optimization (TRPO) [207] method, and the trajectory demonstration method, i.e., generalized cylinders with orientation (GCO) approach, are applied to learn quasi-repetitive construction tasks such as ceiling tile installation. Third, this research conducts the simulation experiment to investigate the capabilities and limitations of the context translation model, TRPO, and GCO for construction tasks.

3.4 Robot Learning from Visual Demonstration

The visual LfD method utilized for teaching a robot the experimental construction tasks, i.e., ceiling tile installation, is an extension of the context translation and imitation method [181]. This method only uses the visual demonstration to train the robot and has been applied before for tasks such as robot arm reaching, object pushing, sweeping, and pouring in controlled and uncluttered environments [181,202,208]. The context translation model is constructed by several encoders, decoders, and autoencoder networks. In this research, the ceiling tile installation can be considered as an advanced and intricate version of object pushing.

The challenge of applying the context translation model to construction tasks is the unstructured and cluttered environment, which can be overcome by modifying the network structure and fine-tuning on the construction tasks demonstration videos. The knowledge of the

construction task, such as the pose of the ceiling tile, is extracted from the demonstration videos as the work context features and translated to the target scene, i.e., the scene that the robot observed, to place the ceiling tile. The robot can further learn the results of the translation and generate the control policy through the RL method [209].

Figure 3.4 shows the work process of the developed construction task learning and performance method, which takes demonstration videos and the target scene as input and output the robot commands for execution. The context translation model and Reinforcement Learning are two primary algorithms in the proposed method, which are introduced in Section 3.4.2 and Section 3.4.4.

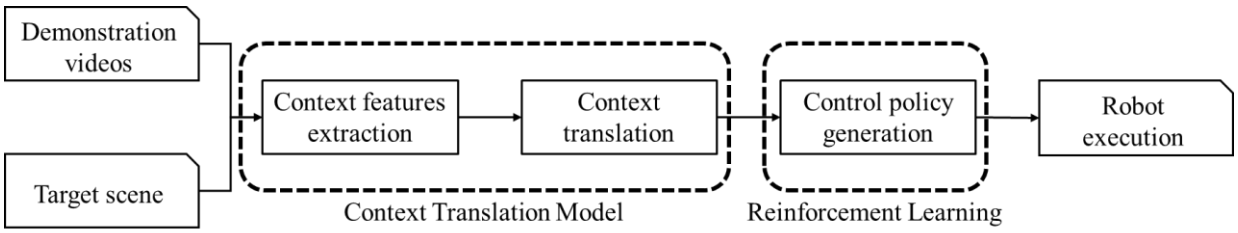


Figure 3.4 Process of the Proposed Visual LfD Method.

3.4.1 Problem Definition and Assumptions

When teaching a robot to perform a specific task, the knowledge, or work context, from the expert's demonstration must be defined in order to let the robot know what information needs to be tracked and assimilated, as well as how to determine the action to take for achieving the task. In the ceiling tile installation process as an example, the work context is to pick up a tile from a staged location, maneuver it through a target ceiling grid, and place it at the correct grid location. The context tracks the intention of how the ceiling tile is being manipulated to the target location. When the robot subsequently faces additional tile installation tasks with different staging and target grid locations, it can still use the context learned to manipulate those tiles to the target grid cells.

The context variable ω can be defined as the pose of the object and human expert, the viewpoint of the camera, the condition of the environment, and the target location. Variations in the camera viewpoint increase the complexity of the robot learning problem since the camera viewpoint is one of the context variables [181]. In this research, the camera is assumed to be fixed in two different viewpoints, i.e., bottom view and iso view, and the task is to be performed in the same environment to reduce the complexity.

The demonstration of the task is defined as [181] Eq 3.1:

$$\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \end{bmatrix} = \begin{bmatrix} O_0^1 & O_1^1 & \cdots & O_T^1 \\ O_0^2 & O_1^2 & \cdots & O_T^2 \\ \vdots & \ddots & & \vdots \\ O_0^n & O_1^n & \cdots & O_T^n \end{bmatrix} \quad \text{Eq 3.1}$$

where D_n is the n-th demonstration and O_t is the observation at time t, which is generated from the Partially Observable Markov Decision Process (POMDP) [210]. For example, the observation O_5 is the demonstration video frame containing the condition of the ceiling tile at time 5. The probability observation distribution $p(O_t|s_t, \omega_i)$, dynamics $p(s_{t+1}|s_t, a_t, \omega_i)$, and the policy of the expert $p(a_t|s_t, \omega_i)$ are utilized to define the POMDP, where s_t and s_{t+1} are the current and next state (e.g., unknown Markovian state), a_t is the action of the agent at time t (e.g., maneuvering direction), and ω_i is the i-th context (e.g., pose of the ceiling tile, the viewpoint of the camera).

The demonstration domain and the learner domain are two domains in the environment, as shown in Figure 3.5. Each demonstration D_n is extracted with different context ω_1, ω_2, and ω_3 in the demonstration domain. In the learner domain, D_o is the scene that the robot learner observes and is also extracted with different contexts ω'_1, ω'_2, and ω'_3. Since this is a partially observable environment, the context from the demonstration domain is unknown to the robot learner in the

probability distribution, i.e., ω_i and ω'_i are unrelated and noncorresponding. The robot learner might try to track the mismatch context from the demonstration domain. For example, the robot learner might follow the context of the ceiling tile pose (e.g., ω_1) but consider it as the context of the expert pose (e.g., ω'_2).

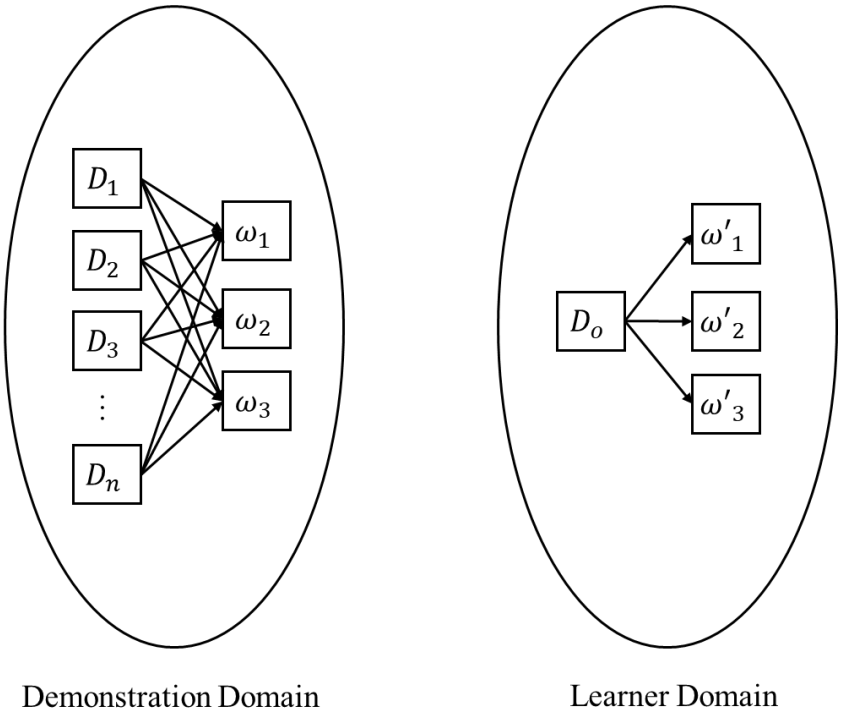


Figure 3.5 The Demonstration Domain and the Learner Domain.

This can be overcome by applying the context translation model to translate the context from the source (demonstration domain) to the target (learner domain) and aligning the context variable in the learner domain. This is followed by defining a reward function based on the translated context feature for the RL method, such as Trust Region Policy Optimization (TRPO) [207], Proximal Policy Optimization (PPO) [211], or Deep Deterministic Policy Gradient (DDPG) [212], to learn the robot control policy. The details of the RL method and the reward function definition are described ahead in the section 3.4.4 and section 3.4.5.

The source and the target demonstrations are defined as follows Eq 3.2 and Eq 3.3:

$$D_s = [O_0^s \quad O_1^s \quad \cdots \quad O_T^s] \qquad \text{Eq 3.2}$$

$$D_t = [O_0^t \quad O_1^t \quad \cdots \quad O_T^t] \qquad \text{Eq 3.3}$$

where D_s is the demonstration from the unknown context extracted from the source video frames with observations O_s, and D_t is from the unknown context extracted from target video frames with observations O_t. For example, the context of the ceiling tile pose from the source demonstration is translated to the target scene, which is the first frame of different demonstration videos, and the model is trained with the loss function calculated by the translated context and the different demonstration video frames. After training with a sufficient number of demonstration examples, the context translation model is capable of translating the source demonstration D_s into a new target scene D_n with the robot learner's context ω_l so that the robot can track and learn the context feature. Figure 3.6 illustrates the steps of the context translation from the source demonstration to the target scene. The translation function takes the source demonstration and translates it to the target scene to obtain translated context. The details of the context translation model are discussed in the following section.

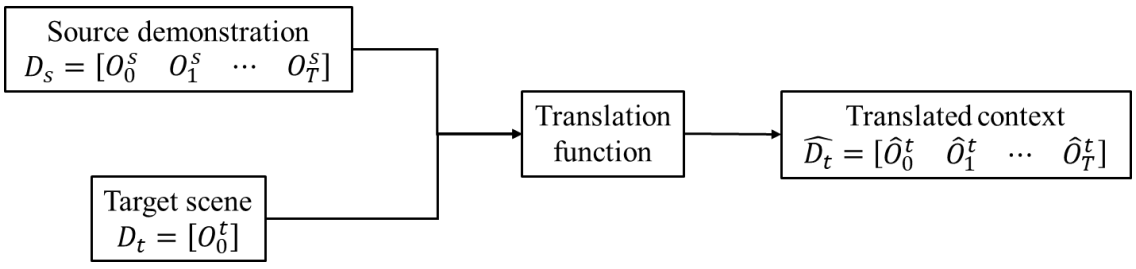


Figure 3.6 Flow Chart of the Context Translation from the Source Demonstration D_s to the Target Scene D_t Resulting in the Translated Context $\widehat{D_t}$.

3.4.2 Context Translation Model

The objective of the context translation model is to learn the translation function that can translate the source demonstration $D_s = [O_t^s],\ t = 0,1, \dots T$ to the target scene $D_t = [O_0^t]$, that is, the scene of the ceiling tile at the starting location for the robot to install, with the context ω_t, e.g., the pose of the ceiling tile or pose of the demonstrator. The full translation function is defined as Eq 3.4:

$$M(O_t^s, O_0^t) = \left(\hat{O}_t^t\right)_{trans} \quad \text{Eq 3.4}$$

where $\left(\hat{O}_t^t\right)_{trans}$ represents the translated observations in the robot learner's context.

The context translation model is constructed by several encoders, decoders, and autoencoders [213], which includes a source encoder $En_s(O_t^s)$, a target first observation encoder $En_t(O_0^t)$, and a target context decoder $De_t(z_{trans})$, as shown in Figure 3.7. The encoder extracts the features from observations (video frames), and the decoder recovers the features back to observations. The features extracted by the encoder are pixel values representing movement in the frames. On the left side is the framework of translating the source observations to the target observation through the translation function $T(z_s, z_t) = z_{trans}$, where z_s, z_t and z_{trans} represent the features of the encoded source, target, and translation. The loss function for training the translation function is defined as L_2-norm (Eq 3.5):

$$L_{trans} = \left\| \left(\hat{O}_t^t\right)_{trans} - O_t^t \right\|_2^2 \quad \text{Eq 3.5}$$

The loss function calculates the differences between translated observations and the target observations, and the translation function tries to minimize the difference between these two observations.

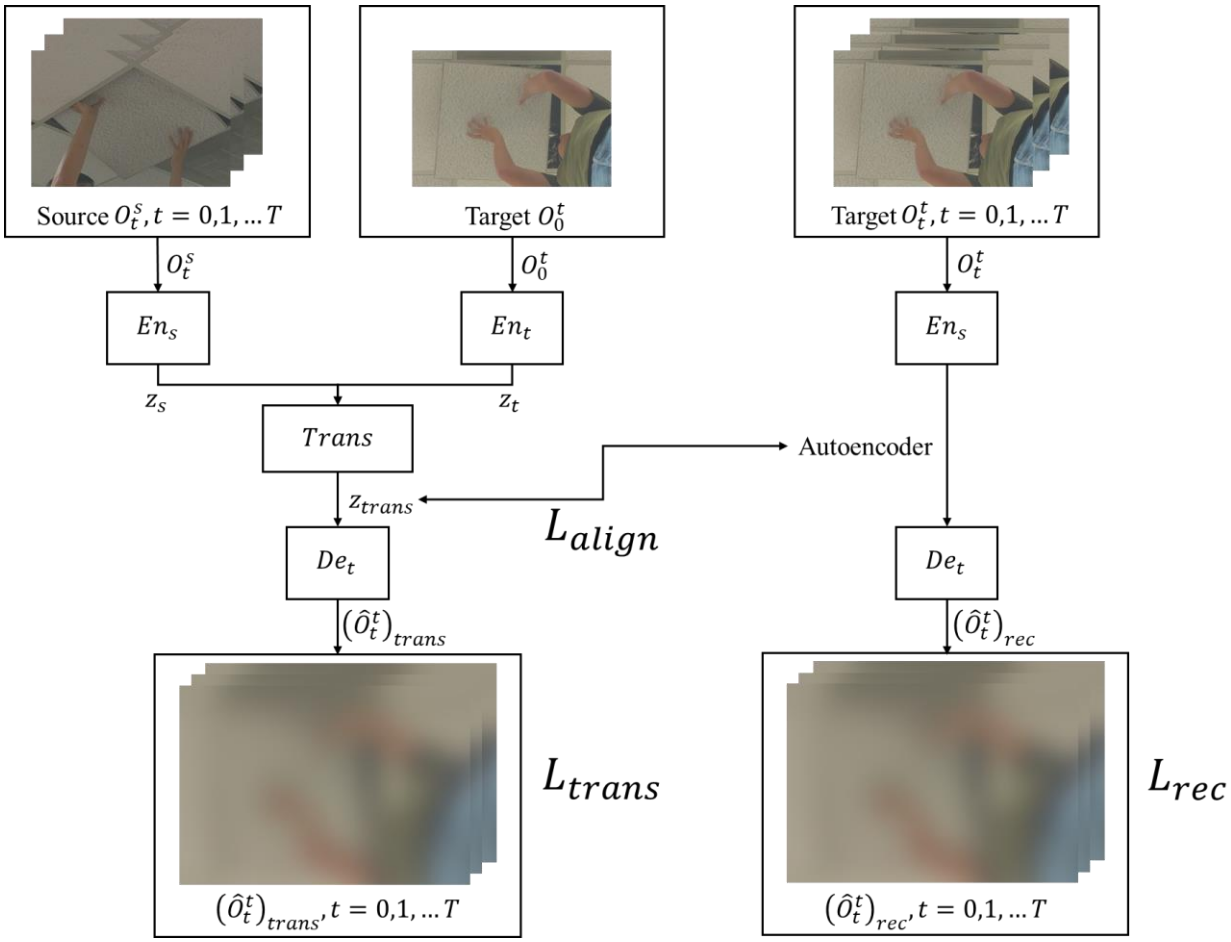


Figure 3.7 The Structure of the Context Translation Model.

Since the unknown context feature is translated from source to target, it must be ensured that the feature in the target domain is the same as the feature in the source. The features need to be trained on the target demonstration video in order to ensure the consistency of the feature representation between the source encoder En_s and the target decoder De_t. The autoencoder is applied for aligning the feature extraction from En_s and De_t. The right side of Figure 3.7 is the

framework of the encoder for training the En_s and De_t with a reconstruction loss, which is defined as Eq 3.6:

$$L_{rec} = \left\| De_t\left(En_s(O_t^t)\right) - O_t^t \right\|_2^2 \quad \text{Eq 3.6}$$

The encoder En_s extracts the features from the target, and then the decoder De_t reconstructs the observation from these features. The reconstruction loss calculates the difference between the reconstructed target observations and the actual target observations, and the autoencoder tries to minimize this loss function to ensure the En_s and De_t are extracting the proper features.

The final step is to align the feature representation of the autoencoder with the translation function features z_{trans} so that the encoder and the decoder in the source and target domains can track the same feature representation. The loss function for the alignment is defined as Eq 3.7:

$$L_{align} = \| z_{trans} - En_s(O_t^t) \|_2^2 \quad \text{Eq 3.7}$$

The alignment loss calculates the difference between the translation features z_{trans} and the features in the autoencoder $En_s(O_t^t)$, and the model tries to align these two features. After training with sufficient demonstrations, the context translation model is able to translate the context from the source to a target observation that is never shown in the training steps for determining the procedure of completing the construction task, in this case, the ceiling tile installation process.

3.4.3 Network Structure of the Context Translation Model

The source encoder, the target encoder, and the target decoder are constructed by deep neural networks to process the demonstration video frames and extract features. The demonstration video frames are cropped and resized to the size of 180x120 and then fed into the networks. The

network of the encoder and decoder is illustrated in Table 3.1. All the network parameters are adapted from [181] and fine-tuned with the ceiling tile installation demonstration videos. In the encoder network, four convolutional layers with different filter size (32, 16, 16, and 8) and stride (1, 2, 1, and 2) followed by two linear layers with a size of 100 and 0.5 dropout are applied to the training video frames to extract the context features, which are unknown to the learner (partially observed).

In the translation function, one linear layer with a size of 100 and 0.5 dropout is used, which translates the source features to the target domain. In the decoder network, one linear layer with 0.5 dropout followed by four deconvolutional layers with different filter size (16, 16, 32, and 3) and stride (½, 1, ½, and 1) are applied to the translated feature, which reconstructs the translated feature to the observations. The translated feature is the trajectory of the ceiling tile. All the convolutional and deconvolutional layers are followed by LeakyReLu activation function [214] with 0.2 leak except the last deconvolutional layer in the decoder. The filter size of the first linear layer in the decoder is dependent on the size of the input image. Batch normalization [154] is applied to the network for the training. The input image is cropped to the size of 48 by 48 pixels for training and testing in order to reduce the computation complexity.

The target objects do not need to be specified in the video frames. The encoder extracts the features from video frames, and then the decoder reconstructs the frames from translated features. The reconstructed frames are used to calculate the loss by Eq 3.5 for tuning the weights of the network. Therefore, when the networks receive new observation data, it can reconstruct a series of manipulation frames, and the robot can determine the trajectory based on the manipulation frames.

Table 3.1 The Network Structure of the Encoder and Decoder.

Type	Layer	Filter size	Stride	Other
Encoder	Conv	32	1	LeakyReLu leak = 0.2
	Conv	16	2	
	Conv	16	1	
	Conv	8	2	
	Linear	100	n/a	Dropout = 0.5
	Linear	100	n/a	
Translation function	Linear	100	n/a	Dropout = 0.5
Decoder	Linear	*	n/a	Dropout = 0.5
	Deconv	16	½	LeakyReLu leak = 0.2
	Deconv	16	1	
	Deconv	32	½	
	Deconv	3	1	n/a

*Depends on the size of the input image

3.4.4 Reinforcement Learning Method

After extracting the context from the demonstration videos and translating it to the target initial observation, the observing robot apprentice must determine the action to take for accomplishing tasks. The RL method [209] is utilized for the robot to learn the sequence of the actions, i.e., the policy, to reach the goal state, i.e., the tile installation location. Deep RL methods have been proven successful in robot continuous control [212] and visual LfD [202,208]. A reward function needs to be defined for the robot to find the policy with the highest rewards.

The RL method is agent-oriented [209], where agents learn by interacting with an environment to reach a goal. The agent (e.g., robot) will receive a reward (e.g., reaches the goal) or penalty (e.g., hits obstacles) after taking a sequence of actions and interacting with the environment. The agent learns a policy (e.g., a sequence of actions) mapping states to actions and seeks to maximize its cumulative reward in the long run, which is the optimal policy to accomplish the task. Figure 3.8 shows the RL concept. One of the main challenges for the Reinforcement Learning application is defining the reward function [194]. For the developed method, the reward

function is defined based on the translated context, and the agent will receive a higher reward if it follows the translated context, which is discussed in section 3.4.5.

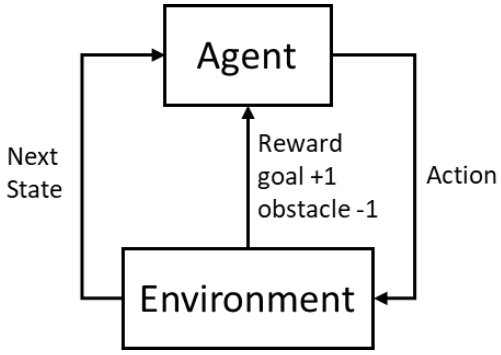


Figure 3.8 Reinforcement Learning (RL) Concept.

When applying the RL method to control a robot, the robot action needs to be formulated in continuous form, where the action in the RL setting is discrete since the agent will choose the action at each iteration [215]. The Policy Gradient (PG) based methods, such as Trust Region Policy Optimization (TRPO) [207], Proximal Policy Optimization (PPO) [211], and Deep Deterministic Policy Gradients (DDPG) [212], are the state-of-the-art methods in the robot RL continuous control domain [215] for solving the linearity and the non-linearity feature of the control.

The PG methods utilize gradient ascent to find the optimal policy with the most precipitous increase in rewards, which have issues of choosing a proper learning rate (step size). If the learning rate is too large, one step could result in overshoot and never converge to the maximum. If the learning rate is too small, the model learns too slow and might take an inordinate amount of time to find the optimal policy. Some PG methods use the learning rate adjustment method [209] to change the learning rate over-time based on the situation, or optimization methods such as Adam optimizer [216]. For instance, in the flat region of the gradient, a large learning rate is used, and a small learning rate is used in the steep region of the gradient. However, there are still some issues

in the cliff situation or unexpected significant changes in the policy even with a small learning rate.

The Trust Region Policy Optimization (TRPO) method [207] is applied in this work for robots to generate the control policy for installing the ceiling tiles, which utilizes a trust region to address the learning rate issue in the policy gradient. TRPO defines a trust-region by the Minorize-Maximization algorithm (MM algorithm) [217] and KL divergence- which is the maximum step size. The PG method will then find the optimal point within the trust region and use that point to define a new trust region. This process will be repeated until the global maximum location is found, which is the control policy with the most precipitous increase in rewards.

The challenge of the RL method in construction application is the partial observability and non-stationarity, similar to the challenge of deploying the RL method to physical systems [218]. This challenge can be overcome by constructing an RL environment in the simulator for the robot to interact with and find the trajectory by the TRPO method. This learning environment includes the non-stationarity of the construction site, such as loose tolerances and unexpected obstacles.

3.4.5 Reward Function for Robot Learning

After the source context is translated to the target initial observation, a reward function must be defined for the robot to learn the control policy through the TRPO RL method. Since the context translation model extracts the features from the image of the source demonstration and translates it to the target scene, the reward function is defined based on the extracted features, i.e., the feature tracking reward function Eq 3.8 [181]:

$$\hat{R}_f\left(O_t^l\right) = -\left\| En_s\left(O_t^l\right) - \frac{1}{n}\sum_{i}^{n} T(z_s, z_t) \right\|_2^2 \qquad \text{Eq 3.8}$$

The feature tracking reward function calculates the difference between the features in the encoder $En_s(O_t^t)$, which encodes the learner's observation O_t^l to z_l, and the average translated feature $\frac{1}{n}\sum_i^n T(z_s, z_t)$. If these two features have large differences, the robot should receive a large penalty to avoid following this translated context. If these two features match perfectly, the robot should receive the highest reward (in this case, zero) to follow this translated context. For example, if the context features in the translated context frame are similar to the context features in one of the ceiling tile demonstration video frames, which means the translated context at that time is close to what the demonstration did, the robot should receive a higher reward and use this translated context for performing the task.

On the other hand, in order to avoid the robot overfit to the extracted features, a second reward function is defined to provide supplemental information, i.e., an image tracking reward function Eq 3.9 [181]:

$$\hat{R}_i(O_t^l) = -\left\| O_t^l - \frac{1}{n}\sum_i^n M(O_t^s, O_0^t) \right\|_2^2 \qquad \text{Eq 3.9}$$

The image tracking reward function calculates the difference between the images of the target observation O_t^l and the average translated observation $\frac{1}{n}\sum_i^n M(O_t^s, O_0^t)$. The robot will receive the highest reward if these two observations match with each other.

Finally, the total reward function combines the feature tracking reward function and the image tracking reward function, which is defined as Eq 3.10 [181]:

$$\hat{R}(O_t^l) = \hat{R}_f(O_t^l) + \alpha \hat{R}_i(O_t^l) \qquad \text{Eq 3.10}$$

where α represents the weight of the image tracking reward function.

3.5 Robot Learning from Trajectory Demonstration

The generalized cylinder (GC) is a generic representation of an arbitrary cylinder. The center axis of the cylinder is defined as arbitrary spline curve $\Gamma(s) = \big(x(s), y(s), z(s)\big)$ and the cross-section boundary of the cylinder is a closed curve $\gamma(r,s) = \big(x(r,s), y(r,s)\big)$ with different shapes. Each cross-section along the center axis is perpendicular to each other. Figure 3.9 illustrates an example of GC with the center axis and three cross-sections. The GC can be represented as:

$$G(r,s) = \Gamma(s) + x(r,s)\nu(s) + y(r,s)\xi(s) \qquad \text{Eq 3.11}$$

where ν represents the unit norm vector of the center axis and ξ represents the unit binormal vector of the center axis. The GC has been applied for robotics applications, including collision detection [219], mapping and state estimation [220], and learning from demonstration [188]. The GC for LfD is modified to suit basic construction manipulating tasks, and a new generalized cylinders with orientation approach is further proposed to perform complex construction manipulating tasks. The strategies of handling unforeseen situations and obstacle avoidance are also developed for human-robot collaboration. Details of each element are discussed in the following subsections.

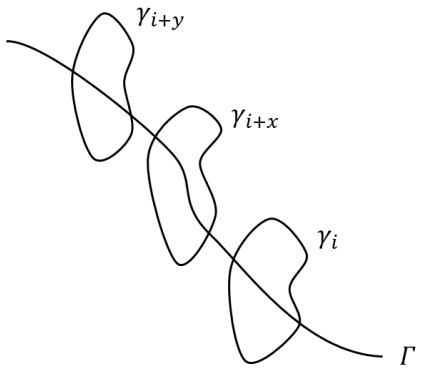


Figure 3.9 Example of the Generalized Cylinder. Each Cross-Section γ is Perpendicular to Each Other Along the Center Axis Γ.

3.5.1 Generalized Cylinders for Robot Learning from Demonstration

For the Learning from Demonstration approach, the GC is constructed from demonstration data and determines the robot trajectory within the GC space. Figure 3.10 shows the detailed procedure of the GC for LfD. First, the demonstration data is pre-processed to obtain aligned data. Second, the center axis curve and the cross-section boundary of the GC are calculated using the aligned data. Third, the GC is constructed using the center axis curve and the set of cross-section curves. Finally, the new robot trajectory is sampled within the GC space starting from the new initial pose.

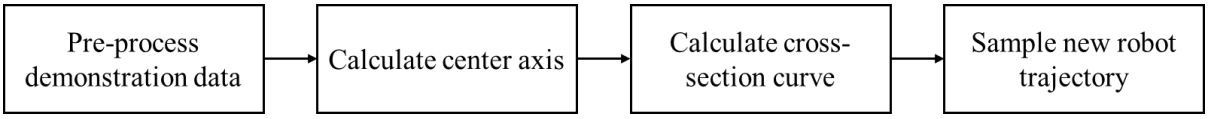


Figure 3.10 Procedure of the Generalized Cylinder Method for Trajectory LfD Method.

In the first step, the demonstration data is processed and aligned. The demonstration data is captured using the virtual robot simulator, where the human expert controls the robot to complete the construction task manually, and the robot's end-effector poses are recorded. Since the demonstrations are recorded separately and manually, they are not aligned with each other and have some redundant waypoints, e.g., the robot is idle during the demonstration since the human expert has to ensure a collision-free manipulation. The group of m demonstration data sets is defined as $D^i = \left(D_x^i, D_y^i, D_z^i\right)$ where $i = 1, \dots, m$ represents ith demonstration data set, i.e., robot's end-effector pose in 3D Cartesian coordinates.

Each demonstration data set has different numbers of data points and requires the alignment process. The Ramer–Douglas–Peucker (RDP) algorithm is first applied to simplify and remove the redundant points in the demonstration data. Only key points remain in the simplified trajectory.

Then, we resample the trajectory with n new data points, including key points from the simplified trajectory. Finally, the Dynamic Time Warping (DTW) algorithm is applied to align each resampled demonstration data. The resulting demonstration data becomes $\widehat{D^i} = \left(\widehat{D_x^i}, \widehat{D_y^i}, \widehat{D_z^i}\right)$, where $\widehat{D} \in \mathbb{R}^{3\times n\times m}$ represents the set of demonstration having m different demonstration trajectories where each trajectory has n data points in 3D Cartesian coordinates. Figure 3.11 shows the original and the processed demonstration data. On the left side is the original demonstration data with three trajectories and different numbers of data points in each trajectory. On the right side is the processed demonstration data, where all three trajectories have the same number of data points and are aligned with each other.

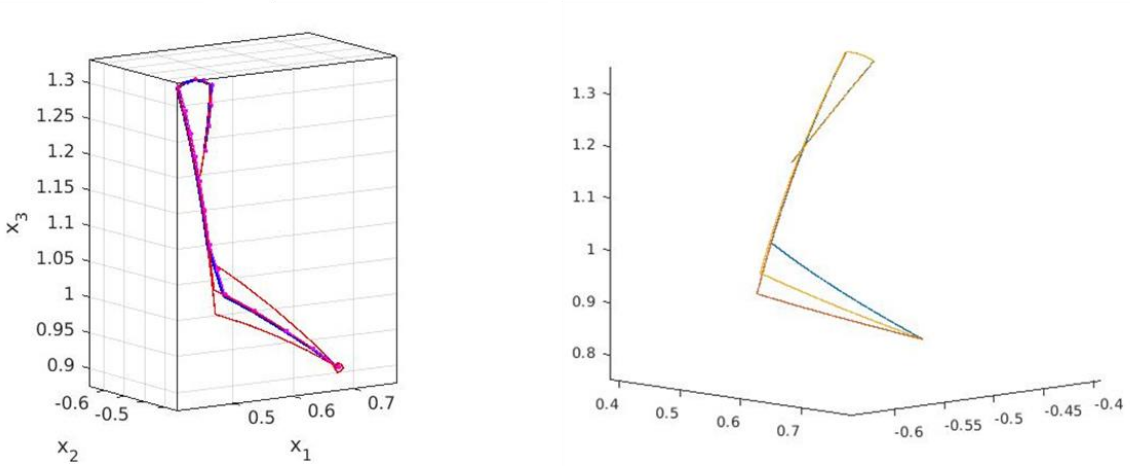


Figure 3.11 Example of the Original (left) and the Processed Demonstration Data (right).

In the second step, the center axis Γ of the GC is calculated using processed demonstration data. The average location of the demonstration data is simply computed at s^{th} data point and assigns to $\Gamma(s)$:

$$\Gamma(s) = \big(x(s), y(s), z(s)\big) = average\left(\widehat{D}(s)\right) \qquad \text{Eq 3.12}$$

This ensures that the center axis is aligned with the demonstration data at each timestep. In the third step, the cross-section curve γ of the GC is calculated using the processed demonstration data and the center axis Γ. In order to construct the cross-section curve at s^{th} data point, all corresponding points are taken from processed demonstration data $\widehat{D}(s)$, and cubic spline interpolation is applied to fit the data with the closed curve [188]. Figure 3.12 illustrates one of the cross-section curves $\gamma(r, s)$ defined by three demonstration data and the center axis data point. After calculating all cross-section curves, the GC can be constructed by the center axis Γ, cross-section curve γ, and Eq 3.12. Figure 3.13 shows an example of the GC constructed by three demonstration data.

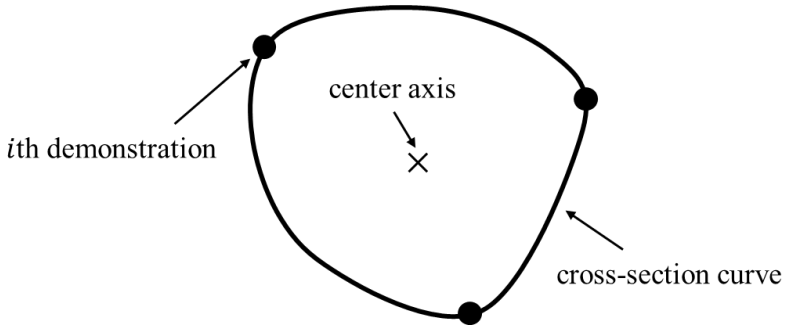


Figure 3.12 A Cross-Section Curve is Defined by Three Demonstration Data and a Center Axis.

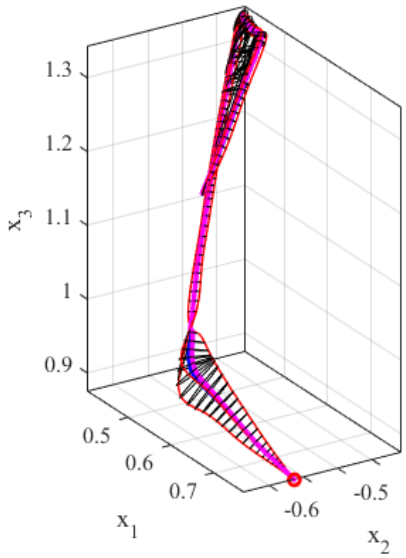


Figure 3.13 Example of the GC Constructed by Three Demonstration Data.

In the final step, a new robot trajectory has to be sampled within the GC. The skill reproduction process in [188] is followed to sample the robot trajectory. The initial pose of the trajectory p_0 is randomly sampled from the first cross-section plane S_0, i.e., the cross-section defined by $\gamma(0)$ and $\Gamma(0)$. To determine the next point on the second cross-section plane S_1, the initial pose p_0 is first projected onto S_1 and get the new pose p'_1. The variable p_t is used to represent the current pose and p_{t+1} is used to represent the new pose in Eq 3.13, Eq 3.14, and Eq 3.15 to keep consistency ($t = 0,1, \dots n-1$):

$$p'_{t+1} = T_t^{t+1} p_t \qquad \text{Eq 3.13}$$

where T_t^{t+1} represents the projection matrix between two coordinates. In order to preserve the feature of the previous pose p_t, a similarity ratio η is defined to shift the new pose p'_{t+1} to a different pose p_{t+1}. On the previous cross-section plane S_t, the center axis point $\Gamma(t)$ is projected to the cross-section curve $\gamma(t)$ through p_t and find the projection point g_t. The similarity ratio is calculated using the following equation:

$$\eta = \frac{\left|\overline{p_t \Gamma(t)}\right|}{\left|\overline{g_t \Gamma(t)}\right|} \qquad \text{Eq 3.14}$$

After obtaining the new pose p'_{t+1}, the center axis point $\Gamma(t+1)$ is again projected to the cross-section curve $\gamma(t+1)$ through p'_{t+1} and find the projection point g_{t+1}. Finally, the shifted new pose p_{t+1} is calculated by:

$$p_{t+1} = \left(\eta \left|\overline{g_{t+1} \Gamma(t+1)}\right|\right) p'_{t+1} \qquad \text{Eq 3.15}$$

The entire process is repeated through every cross-section to generate the new robot trajectory. Figure 3.14 illustrates the new pose sampling process from the cross-section S_t to S_{t+1} along the center axis Γ. The pseudo-code of the GC for LfD can be found in Algorithm 1.

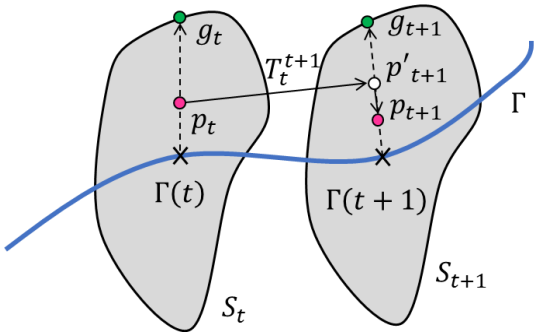


Figure 3.14 Process of Sampling New Pose from the Cross-Section S_t to the Next Plane S_{t+1}.

Algorithm 1. Generalized Cylinder for Robot Learning from Demonstration

procedure ENCODING DEMONSTRATIONS($\widehat{\boldsymbol{D}}$)
    $\Gamma(s) \leftarrow average(\widehat{D})$
    $\gamma(r,s) \leftarrow constructBoundary(\widehat{D})$
    $G(r,s), S \leftarrow constructGeneralizedCylinder(\Gamma(s), \gamma(r,s))$
    return $G(r,s), S$
end procedure
procedure CONSTRUCTGENERALIZEDCYLINDER($\boldsymbol{\Gamma(s), \gamma(r,s)}$)
    for each s **do**
        $v(s), \xi(\mathrm{s}) \leftarrow calculateUnitVector(\Gamma(s))$
        $S_s \leftarrow getTransformation(v(s), \xi(\mathrm{s}))$
        $G(r,s) \leftarrow \Gamma(s) + \gamma^x(r,s)v(s) + \gamma^y(r,s)\xi(\mathrm{s})$
    end for
    return $G(r,s), S$
end procedure
procedure GENERATE TRAJECTORY($\boldsymbol{G(r,s), S}$)
    $p_0 \leftarrow randomSample(G(r,0))$
    $\eta \leftarrow \dfrac{|p_0 - \Gamma(0)|}{|g_0 - \Gamma(0)|}$
    $p_t \leftarrow p_0$
    for each *cross section* S_t **do**
        $p_{t+1} \leftarrow project(p_t, \eta, S_{t+1}, S_t)$
        $t \leftarrow t+1$
    end for
    return p
end procedure

3.5.2 Orientation Constraint

After obtaining the robot learned trajectory from the GC, the robot control policy could be determined using Inverse Kinematics. The robot trajectory is in 3D Cartesian coordinate as (x, y, z) triplets without end-effector's orientation information. However, it is necessary for some complex construction tasks to strictly follow the manipulating orientation. For example, in the ceiling tile installation process, the tile has to be manipulated to some specific orientations in order to pass through the grid area. In the end, the tile also has to be placed with the same orientation to fit the grid.

When the tile is approaching the grid, the orientations are similar across every demonstration data. Figure 3.15 illustrates the orientation information of the tile manipulation. The demonstration trajectories are close to each other when nearing the grid area to insert a tile into the grid. The demonstration data points with minimum distance to each other are defined as the insertion points since all demonstration trajectories have to go through that region. The average orientation at the insertion point is defined as the critical orientation, i.e., the robot must use it at the insertion point to pass the tile into the grid area. Figure 3.16 shows an example of the insertion point and the critical orientation.

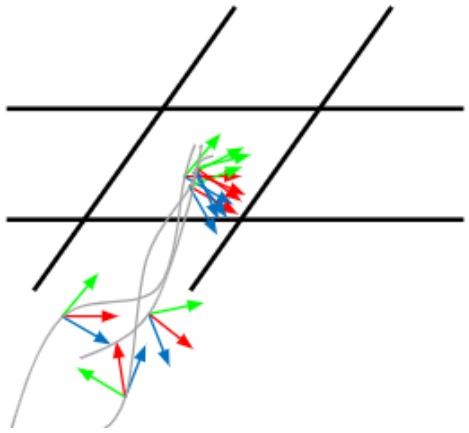


Figure 3.15 Orientation Information of the Ceiling Tile Installation Manipulation.

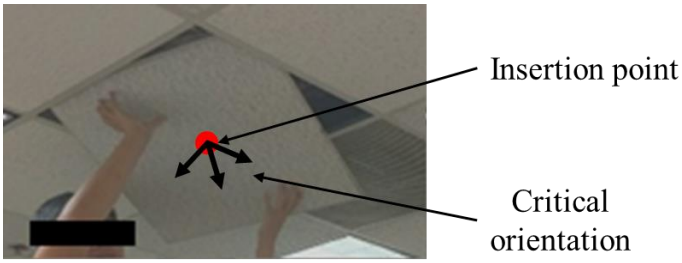


Figure 3.16 Example of the Insertion Point and the Critical Orientation.

The new algorithm called Generalized Cylinder with Orientations approach (GCO) is proposed using the GC method and orientation constraint method. Figure 3.17 shows the procedure of the orientation constraint for the GC method. First, the cross-section with the minimum area and the insertion point is determined. The demonstration data points on this cross-section are closest to each other. Second, the critical orientation is calculated by averaging all orientation data at the insertion point. Finally, the robot pose is constrained by the critical orientation at the insertion point and determines the new control policy.

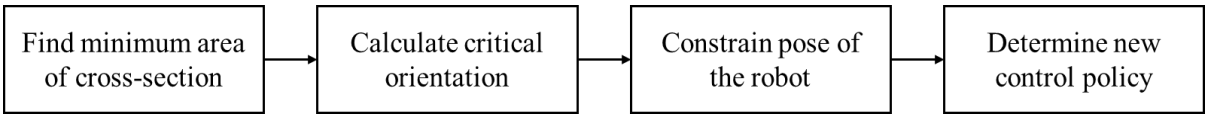


Figure 3.17 Procedure of the Orientation Constraint for the GC Method.

In the first step, the area of every cross-section is calculated and the one with the minimum area is found. Using the GC representation from Eq 3.11, the cross-section curve at the s^{th} data point is $\gamma(r,s) = (x(r,s), y(r,s))$. The area of the cross-section curve can be calculated by:

$$area(s) = \int_0^p \left(x\frac{dy}{dr} - y\frac{dx}{dr} \right) dr \qquad \text{Eq 3.16}$$

where p represents the perimeter of the cross-section curve. Then, the data point with the minimum cross-section can be determined, i.e., the insertion point $\hat{s}$.

In the second step, the critical orientation is calculated at the insertion point $\hat{s}$. The orientation of the demonstration data is represented using quaternions. The critical orientation is defined as the average of the quaternions. Based on the definition, the average quaternion is the argument of the minima of the following equation [221]:

$$\bar{q} = \underset{q \in \mathbb{S}^3}{\operatorname{argmin}} \sum_{i}^{n} \omega_i \|A(q) - A(q_i)\|_F^2 \qquad \text{Eq 3.17}$$

Using Eq 3.17, the average quaternion can further derived by:

$$\bar{q} = \underset{q \in \mathbb{S}^3}{\operatorname{argmax}} q^T M q \qquad \text{Eq 3.18}$$

$$M = \sum_{i}^{n} \omega_i q_i q_i^T \qquad \text{Eq 3.19}$$

Therefore, the average quaternion is the normalized eigenvector corresponding to the maximum eigenvalue of M. The average quaternion is calculated using Eq 3.19 and the eigendecomposition process. In the final step, Inverse Kinematics is applied with the critical orientation constraint and the last orientation data (the orientation for fitting the grid) to find the robot control policy. The pseudo-code of the GCO can be found in Algorithm 2.

Algorithm 2. Orientation Constraint for the Generalized Cylinder Approach

procedure ORIENTATION CONSTRAINT($\widehat{\boldsymbol{D}}, \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{s})$)
    for each s **do**
        $area(s) \leftarrow \int_0^p \left(x \frac{dy}{dr} - y \frac{dx}{dr} \right) dr$
    end for
    $\hat{s} \leftarrow findMinimumArea\big(area(s)\big)$
    $M \leftarrow constructQuaternionMatrix\left(\widehat{D}(\hat{\boldsymbol{s}})\right)$
    $\bar{q} \leftarrow findMaximumEigenvector(M)$
    return $\bar{q}$
end procedure

3.5.3 New Locations and Obstacle Avoidance

The GCO approach provides the geometric space constructed by demonstration data to sample the new robot trajectory. However, the start and the target pose have to lie in the GC space, and the process is unable to overcome unforeseen situations such as arbitrary obstruction. One way to overcome such unforeseen situations is to apply a nonrigid registration technique, e.g., Thin-Plate Splines or Laplacian Trajectory Editing that takes a set of points in the unforeseen geometry to deform the GC [188]. Since construction tasks are quasi-repetitive and subject to various start and target locations, a trajectory adaptation approach is proposed to refine the robot trajectory based on the new start and target locations.

Figure 3.18 shows the procedure of the trajectory adaptation approach. First, each demonstration data is translated to the new scene and matches the target data point with the new target location. Then, the GC is constructed using the translated demonstration data. Second, the GC is updated by the new start location. Third, the collision of the new GC is checked by the collision detection algorithm. If the collision exists, the GC will be updated to avoid the obstacle. Finally, a new robot trajectory is sampled within the new GC with the orientation constraint and applies Inverse Kinematics to determine the robot control policy.

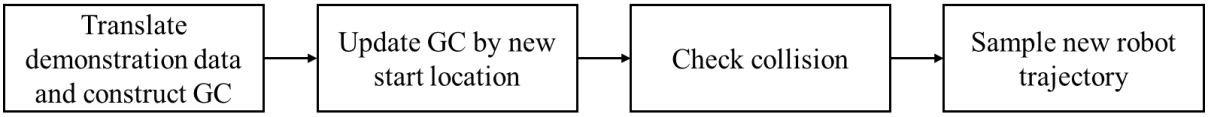


Figure 3.18 Procedure of the Trajectory Adaptation Approach to Refine the Robot Trajectory.

In the first step, the human worker indicates the new target location p_t in the new scene to the robot. The robot translates the demonstration data $\widehat{D}$ to the new scene and matches the new target location p_t. Next, the GC is constructed using the translated demonstration data $\widehat{D'}$ and

Algorithm 1 before sampling a new trajectory. In the second step, the GC is updated with the new start location p_0, i.e., the current pose of the robot's end-effector.

If the new start location p_0 is within the GC space, the new trajectory can be simply sampled starting from the cross-section of the new start location p_0 to the new target location p_t, as shown in Figure 3.19(a). If the new start location p_0 is outside the GC space and coplanar with the first cross-section plane of the GC, the new start location is directly connected to the previous start location, as shown in Figure 3.19(b). By following this process, the robot can maneuver on the first cross-section plane S_0 and follow the same trajectory afterward.

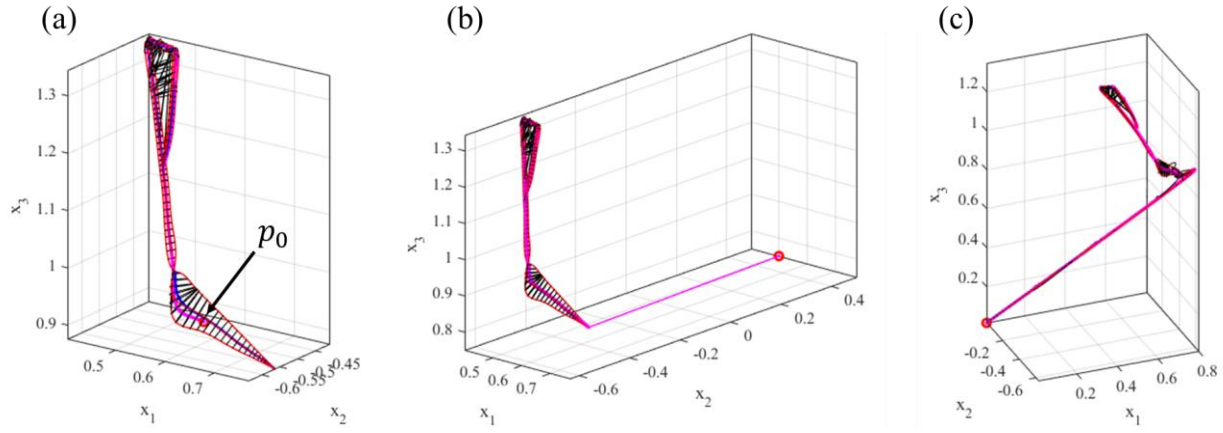


Figure 3.19 Updated GC with Different Start and Target Locations.

If the new start location p_0 is outside the GC space and not coplanar with the first cross-section of the GC, the new start location p_0 is connected to every start waypoint of the demonstration data $\widehat{D'}(0)$ with straight lines. Then, a new GC $G'(r,s)$ is constructed using these updated straight-line demonstration data $\widehat{D''}(0)$ and resample the robot trajectory. Because all updated demonstration data $\widehat{D''}$ are started from the same initial waypoint p_0, the vertex of the GC $G'(r,s)$ is the new start location p_0 and the first similarity ratio η cannot be determined (S_0 is the vertex of the GC).

To simplify the trajectory sampling process, instead of using the projection and similarity ratio to determine the second robot waypoint p_1, a waypoint is randomly sampled on the cross-section plane S_1 and utilize it as the second robot waypoint p_1. Then, the generating trajectory procedure in Algorithm 1 can be repeated to find the new robot trajectory. Figure 3.19(c) shows an example of the updated GC with a new start location outside the GC space and not coplanar with the first cross-section plane of the GC. Algorithm 3 shows the pseudo-code of the trajectory adaptation approach.

Algorithm 3. Trajectory Adaptation Approach

procedure TRAJECTORY ADAPTATION($\widehat{\boldsymbol{D}}, \boldsymbol{p_0}, \boldsymbol{p_t}$)
    $\widehat{D'} \leftarrow translate(\widehat{D}, p_t)$
    $G'(r,s), S' \leftarrow Encode\ Demonstration(\widehat{D'})$
    if $p_0 \in G'(r,s)$ **do**
        $S'_0 \leftarrow getCrossSection(S', p_0)$
        $p \leftarrow Generate\ Trajectory(G'(r, S'_0), S')$
    else if $p_0 \notin G'(r,s)$ **and** $isCoplanar(p_0, S'_0)$ **do**
        $p \leftarrow Generate\ Trajectory(G'(r,s), S')$
        $p \leftarrow connectTrajectory(p_0, p)$
    else do
        $\widehat{D''} \leftarrow connect(p_0, \widehat{D'})$
        $G''(r,s), S'' \leftarrow Encode\ Demonstration(\widehat{D''})$
        $p \leftarrow Generate\ Trajectory(G''(r,s), S'')$
        $G'(r,s) \leftarrow G''(r,s)$
    end if
    return $p, G'(r,s)$
end procedure

In the third step, the collision detection algorithm is applied to validate the GC. The bounding box algorithm is used to create bounding boxes around each geometry in the environment. the robot is assumed to have all information of the surrounding environment using an approach described in previous work to collect and synchronize the geometry data [11] and construct bounding boxes around each geometry in the environment. If the GC or the handled

component is intersecting with any of the bounding boxes, the GC must be reconstructed to avoid collision. Existing methods used the adaptive ratio and deformation function to avoid the obstacle intersecting with the GC [188]. A human-in-the-loop refinement approach is proposed to resolve the situation. When a collision occurs, the human worker will demonstrate one solution to the robot and record the trajectory. The new demonstration data is combined with all other demonstration data to construct a new GC.

Instead of randomly sampling a waypoint on the first cross-section plane S_1, the center axis $\Gamma(1)$ is connected to the new demonstration data $\widehat{D^m}(1)$ with a straight line and define a shift ratio ρ to select the waypoint:

$$\rho = \frac{n}{\left|\overline{\Gamma(1)\widehat{D^m}(1)}\right|} \qquad \text{Eq 3.20}$$

where n represents the total number of the data points in one demonstration. Then, the first waypoint is determined by the demonstration data $\widehat{D^m}(1)$ and the shift ratio ρ. By using the shift ratio, the new robot trajectory will stay close to the new demonstration data in order to avoid obstacles. Figure 3.20 illustrates the process of determining the robot waypoint on the first cross-section plane. The dashed curve is the original cross-section, and the solid curve is the new cross-section extended by the new demonstration data $\widehat{D^m}(1)$. The new waypoint p_1 is selected by the line $\overline{\Gamma(1)\widehat{D^m}(1)}$ and the shift ratio ρ. Next, the rest of the robot trajectory can be sampled using the updated GC and Algorithm 1.

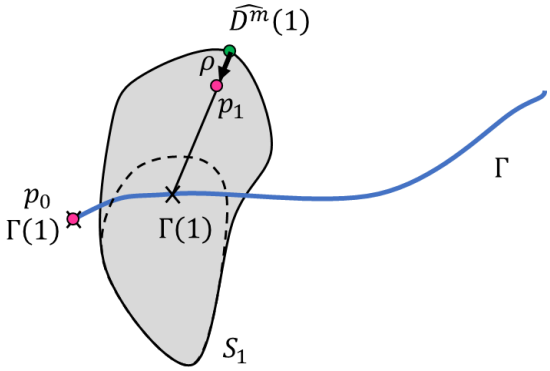


Figure 3.20 Process of Determining the Waypoint p_1 on the First Cross-Section Plane S_1.

On the other hand, a collision usually occurs when the manipulated object is approaching the installation location, e.g., a tile collides with the suspended grids. To overcome such collisions, more critical orientation data points near the installation location are included. The number of critical orientation data points depends on the demonstration trajectory. If the demonstration trajectory is close to the obstacle, it requires more critical orientations, and the computational time also increases significantly to calculate the average quaternions.

Two steps are proposed to determine if the critical orientation is required. First, if the distance between the current robot waypoint and the installation plane is smaller or equal to the half-length of the manipulated object bounding box's largest diagonal, the critical orientation is required at this waypoint. Second, if the collision still occurs, the human-in-the-loop refinement approach will be applied to avoid the obstacle and repeat the entire process to determine the robot trajectory. Finally, the robot control policy is determined based on the robot trajectory and the critical orientations. Algorithm 4 shows the pseudo-code of the human-in-the-loop refinement approach.

Algorithm 4. Human-in-the-Loop Refinement Approach

procedure HUMAN-IN-THE-LOOP REFINEMENT($G'(r,s)$)

$B \leftarrow getBoundingBox()$

if $isFoundCollision(G'(r,s), B)$ **do**

$\widehat{D^m} \leftarrow getHumanDemonstration()$

$G''(r,s), S \leftarrow Encode\ Demonstration(\widehat{D})$

$\rho \leftarrow \frac{n}{\left|\overline{\Gamma(1)\widehat{D^m}(1)}\right|}$

$p_1 \leftarrow shift\left(\rho, \widehat{D^m}(1), \Gamma(1)\right)$

$insert\ p_1\ into\ p$

$p \leftarrow Generate\ Trajectory(G''(r,s), S)$

return p

end if

end procedure

3.6 Experiments and Results

To evaluate the feasibility of applying the visual and trajectory LfD method for teaching quasi-repetitive construction tasks, the ceiling tile installation task was chosen as the experimental construction process. The visual LfD method is first evaluated. The ceiling tile installation demonstration videos were collected in the laboratory with a camera, as shown in Figure 3.21, and utilized to train the context translation model and the RL method (TRPO). The performance of the model and the TRPO method was evaluated by the success rate of the installation task in the Gazebo robotics simulator using ROS (Robot Operating System) [204] and rviz [222] tools and a KUKA industrial robot arm emulator. Second, the trajectory LfD method is evaluated. The robot simulator ROS Gazebo [205] is used to build the robot's work environment, collect demonstration data, and evaluate the robot's performance. The success rate of the installation is used as the evaluation metric.

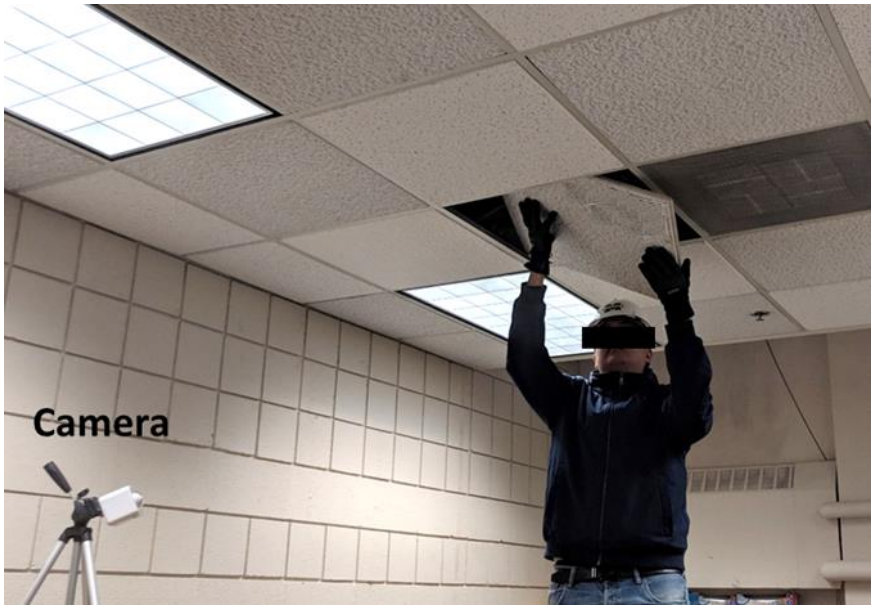


Figure 3.21 Ceiling Tile Installation Demonstration Video Collected in the Laboratory with an Iso View Camera.

3.6.1 Implementation and Training Details

The context translation model was implemented by modifying the original network using TensorFlow. A total of 85 videos were collected in the real-world for training the networks. The imitation learning by observation methods typically require thousands of demonstration videos to train accurate learning models [223]. This can be coupled with pre-programming a robot with primitive actions and providing an action sequence to complete tasks. However, collecting such data at this scale in real construction work settings is not practical due to the human effort required. Therefore, we propose the use of robot simulation in the virtual simulator or humans demonstrating in Virtual Reality to help collect a rich training dataset by interacting with different ceiling tiles or placing the camera at different locations[178]. Additional 1,500 simulation videos were collected in the virtual simulator and augmented to 3,000 training data by dataset augmentation method [157].

The network was trained by the Adam optimizer [216] with learning rate 10^{-5} and the loss function described in the subsection Context Translation Model. The pre-trained model of the pushing task from [181] is used and fine-tuned the entire network's weight with our training dataset. The network was trained for 10,000 iterations, and the batch size was 50. Since the total number of the training data is 3,085, each epoch has 3,085 / 50 = 62 iterations and a total of 10,000 / 62 = 162 epochs. The network weights were updated in every iteration. The training loss was found to converge after 1,000 iterations. Thus, considering time constraints and for ensuring the quality of the training, the model that trained for 10,000 iterations was selected. Table 3.2 listed the training parameters of the context translation model network. For validating the trained network, 60 different initial scenes were used as the testing data. In the demonstration video collection, the camera was set up at two fixed viewpoints for reducing the complexity, i.e., iso view and bottom view, as shown in Figure 3.22.

Table 3.2 Context Translation Model Network Training Parameters

	Parameters
Optimizer	Adam
Learning rate	10^{-5}
Batch size	50
Training iterations	10,000

Figure 3.22 Example of the Ceiling Tile Installation Demonstration Video with Two Different Camera Viewpoints (Iso View and Bottom View).

The robot construction task performance was implemented in the virtual simulator environment for demonstrating the learned skill. ROS, Gazebo, rviz, OpenAI Gym [224], and gym-gazebo [225] were utilized to create the simulation environment and the TRPO algorithm. The use of a virtual simulator such as ROS Gazebo is the first step for evaluating the feasibility of a new method [226]. Different types of construction conditions could also be demonstrated and evaluated in virtual simulations, e.g., different sizes of the ceiling tile. In addition, the ROS and Gazebo systems are capable of communicating with real robots for testing in real environmental settings in subsequent work.

A 6 DOF KUKA robot arm with a gripper, a ceiling tile, and a suspended grid were built and included in the ROS Gazebo simulator. Figure 3.23 illustrates the robot construction task performance in the simulation environment. The target grid location is on top of the robot arm. The real demonstration videos collected from the previous experiment were used as the training source, and 60 different initial observations with the same viewpoint (iso view) but different target suspended grid location and initial tile location were used as the testing data.

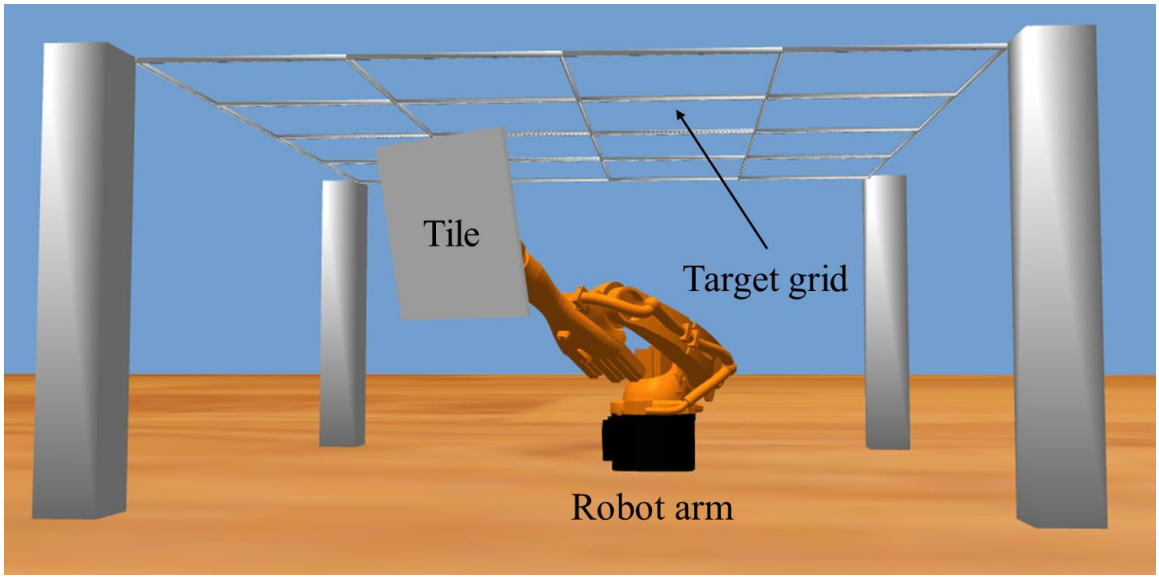


Figure 3.23 A 6-DOF Robot Arm with a Gripper, a Ceiling Tile, and a Suspended Grid are Included in the Simulation Environment.

For the trajectory demonstration, the data were also collected in the robot simulation. The human expert controlled the robot to complete the ceiling tile installation process while the robot's end-effector 6-DOF poses (x, y, z, q) were recorded as the trajectory demonstration. For the human demonstration, three different target locations, i.e., three different grids, were defined and demonstrated four different trajectories for each target location from similar start locations (total 12 sets of demonstration trajectories). The robot was assumed to have picked up the tile, and thus the tile was secured on the robot's end-effector. The demonstration data were pre-processed using the method discussed in Section 3.5.1 to smoothen and align the trajectories. Each demonstration trajectory was resampled to 1,500 waypoints, as suggested in [188]. The number of the waypoints affects the computational time, the trajectory smoothness, and the collision checking ability. Figure 3.24 shows one of the processed demonstration trajectories.

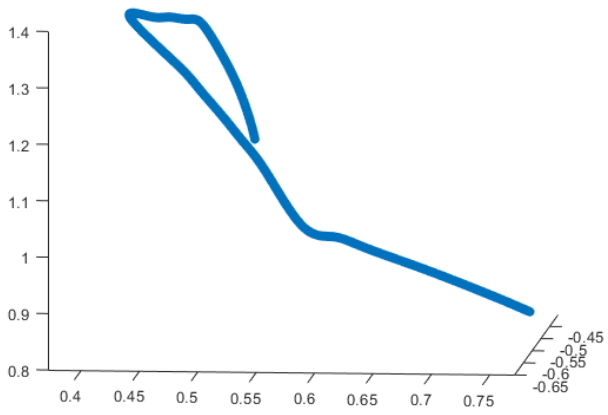


Figure 3.24 One of the Processed Demonstration Trajectories with 1,500 Waypoints.

The generalized cylinders with orientation approach was implemented in MATLAB and sent the control policy to the Gazebo robot. The experiment was designed in two phases to evaluate the GCO approach and the adaptation approach. In the first phase, the start and the target locations were both inside the GC space. Fifty start locations were defined on the first cross-section of each GC and, therefore, 150 cases to test the GCO approach.

In the second phase, the start and the target locations were both outside the GC space, i.e., unforeseen situations. Ten different start locations and ten different target grids were defined to test the trajectory adaptation approach. The image of the start locations was captured for the Imitation from Observation (IfO) approach to compare with the GCO approach. Figure 3.25 shows two examples of different start locations and target grids.

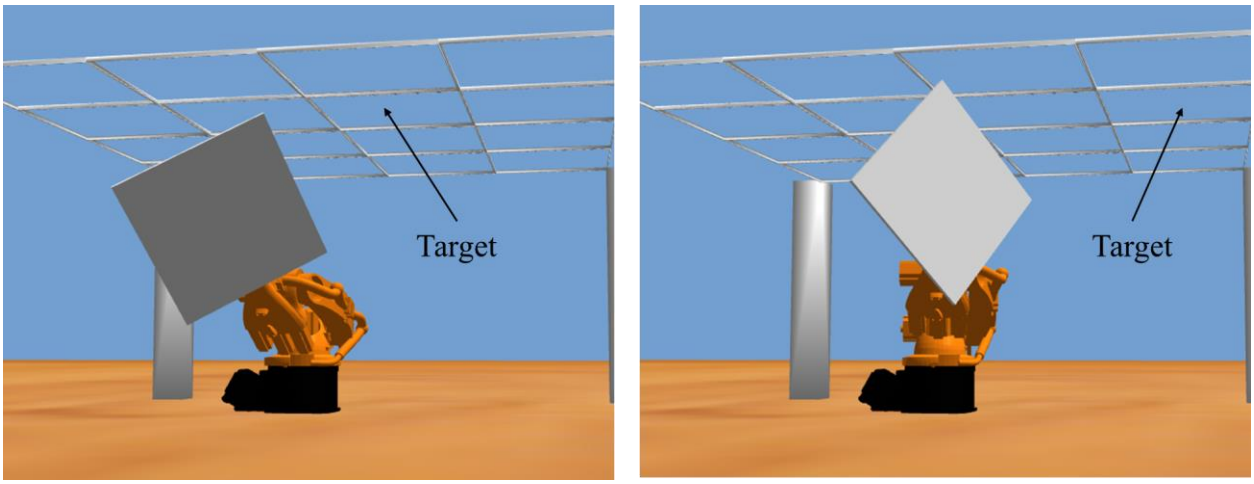


Figure 3.25 Two Examples of Different Start Locations and Target Grids

For evaluation and validation, the existing validation method in Liu et al. [181] was followed to devise the benchmark for success based on the understanding of the ceiling tile installation process and its success by consultation with experts and viewing tutorial videos online [227]. The volumetric success metric was defined, i.e., how close to the desired ceiling tile volume the robot places the tile, based on the ceiling tile tolerance identified in the product manual [228] for validation.

3.6.2 Results of Context Translation Model

The success metric is defined as whether the final non-overlapping area between the ceiling tile and the target grid cells is within a predefined threshold. Based on the ceiling tile tolerance

manual [228], the size of 595 by 595 mm (23.4" by 23.4") ceiling tile allows for a 5 mm (0.2") difference of tolerance between the tile and the grid; thus, the threshold is defined as 30 cm^2 ($0.5cm \times 60cm = 30cm^2$) in the experiment. Figure 3.26 is an example of the source video of ceiling tile installation. The human worker demonstrates how to install the ceiling tile, and the frame is sampled every 1/3 second in the video, which is approximately 15 frames in every training data point.

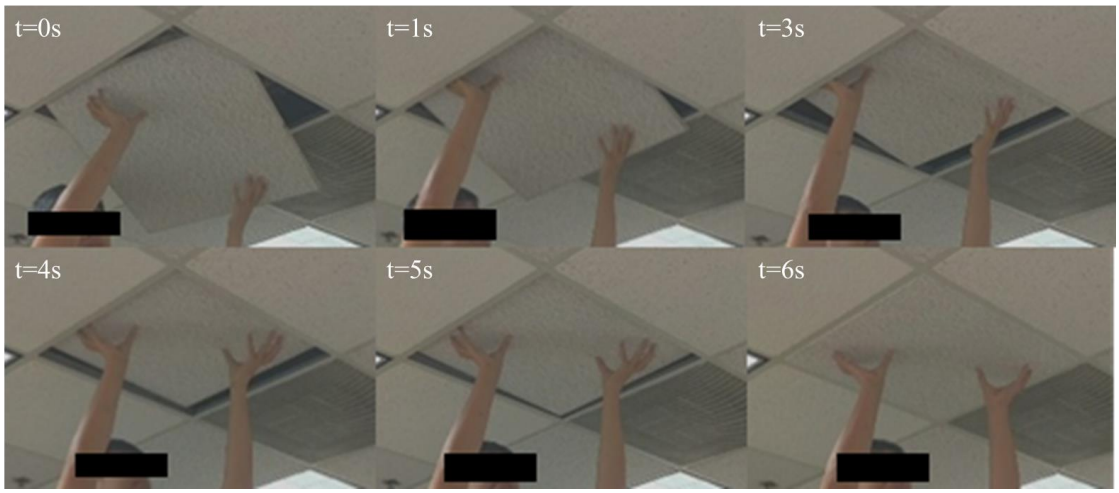


Figure 3.26 Example of a Source Video of Ceiling Tile Installation.

Figure 3.27 shows examples of the target initial observations with two different viewpoints (iso view and bottom view). The target initial observation is the first frame of the demonstration video, which is utilized for translating the context from the source video. Figure 3.28 shows the results of the translated scene. The top row is the successful result, and the bottom row is the unsuccessful result. The red rectangle represents the ceiling tile, and the green rectangle represents the target grid. The distance between the ceiling tile and the grid is over the predefined threshold of 5 mm; thus, it is determined as an unsuccessful result.

Figure 3.27 Example of Target Initial Observations, i.e., First Frame of the Video.

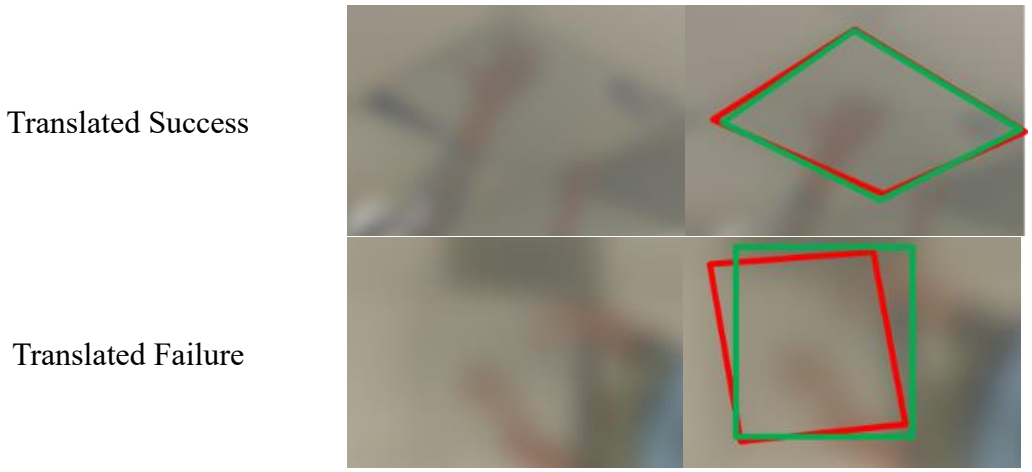


Figure 3.28 Results of the Translated Scene by Reconstructing Images with the Successful Result (Top) and Unsuccessful Result (Bottom).

The success rate of the context translation model is compared with the different types of viewpoints (bottom view and iso view), as shown in Table 3.3. The success rate of the bottom view is 40% with 15 initial scenes, the iso view is 53% with 15 initial scenes, and the overall success rate is 47% with 30 initial scenes. In comparison with the result from Liu et al. [181], where the lowest success rate is 66% for pouring almonds with 3,060 demonstration videos, our context translation success rate is acceptable given the relatively modest number of demonstration videos (85) used for training. In addition, the results show that the bottom view has a lower success rate since most of the trajectories of ceiling tile installation are vertical types of movement, and the bottom view cannot provide sufficient information. This can be addressed by avoiding the vertical type viewpoint, and thus for evaluating the construction task performance, only the iso view types of the initial observation are utilized.

Table 3.3 Success Rate of the Context Translation Result.

Viewpoint	Success	Failure	Success rate
Bottom view	6	9	40%
Iso view	8	7	53%
Overall	14	16	47%

3.6.3 Results of Reinforcement Learning Method

The success metric is defined as whether the final non-overlapping volume between the ceiling tile and the target grid cells is within a predefined threshold. Figure 3.29 shows a series of sequential frames for one of the results of the ceiling tile installation task performed by the robot arm in the virtual simulation environment. For the 2ft-by-2ft ceiling tile ($60cm \times 60cm \times 1cm$), 30cm^3 ($0.5cm \times 60cm \times 1cm = 30cm^3$) is selected as the threshold for determining the successful and failed cases. Figure 3.30 shows the cases of successful and failed ceiling tile installation. On the left side, the ceiling tile is placed at the incorrect location and exceeded the 30 cm^3 threshold. The success rate of the ceiling tile installation is 78% (47/60) in the simulation experiment. Among the thirteen failed cases, five of them exceeded the threshold (> 30cm^3), and eight of them failed to pass the grid area.

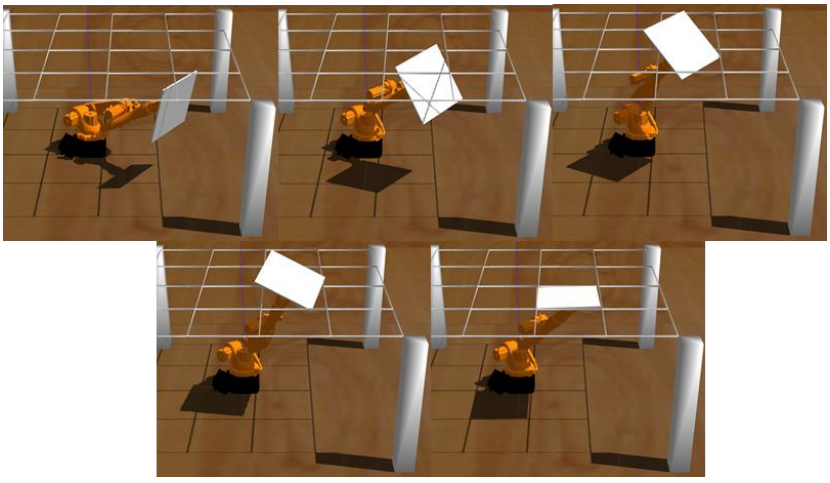


Figure 3.29 Result of the Ceiling Tile Installation Task Performed by the Robot Arm in the Virtual Simulation Environment.

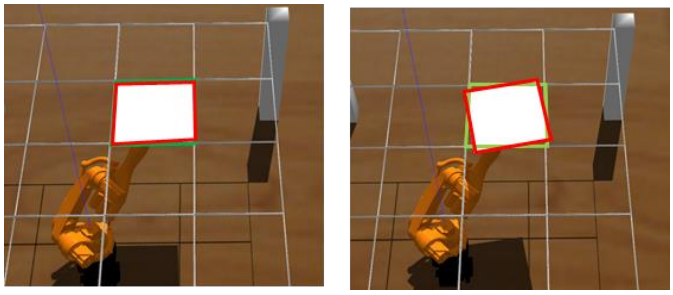


Figure 3.30 Example of the Successful (Left) and Failed (Right) Cases.

For comparing the number of real-world training data and the success rate, the network was trained with different numbers of real demonstration videos (30, 50, 70, 75, 80, and 85 videos) by 10,000 iterations. The success rate and the number of training data were approximated by a log function to predict the success rate with thousands of real training data, as shown in Figure 3.31. The success rate converged to 80% after 3,000 training data. The result of the 3,085 virtual and real training data was 78%. In the results of Liu et al. [181], a total of 3,000 training videos in virtual environments were used for the reaching task, which resulted in 81% success rate, and 4,500 training videos in virtual environments for the pushing task, which resulted in 78% success rate. Therefore, the proposed method can achieve similar performance in the ceiling tile installation task, which is a more complex task than the simple reaching tasks, when provided with sufficient demonstration videos in both virtual and real world.

Figure 3.31 The Prediction of the Success Rate by Approximating the Number of Training Data and Success Rate with Log Function.

3.6.4 Results of Generalized Cylinder with Orientations Approach

In the first phase of the experiment, the success rate of the GCO approach is compared with the GC approach and the Context Translation Reinforcement Learning method (CTRL). Table 3.4 shows the results of the GCO approach, GC approach, and the CRTL method for the robot installing ceiling tiles. First, the success rate of the GCO approach is 75.3%, with 113 success cases and 37 failed cases. In the 37 failed cases, the tiles were found to have collided with the grids before reaching the critical orientations. Second, the success rate of the GC approach is 16.0%, with 24 success cases and 126 failed cases. Among the 126 failed cases, 103 were unable to pass the grid, and 23 exceeded the threshold. Finally, the success rate of the CTRL method is 71.3%, with 107 success cases and 43 failed cases. Among the 43 failed cases, 23 were unable to pass the grid, and 8 exceeded the threshold.

Table 3.4 Results of the GCO, GC, and CTRL Method for the Ceiling Tile Installation.

Method	Success	Failure	Success Rate
GCO	113	37	75.3%
GC	24	126	16.0%
CTRL	107	43	71.3%

Figure 3.32 shows the results of the GCO approach and the generated robot trajectory. The GC is constructed by the four sets of demonstration data, which are the thin lines inside the GC. The generated trajectory is shown as the thick line inside the GC, and the insertion point (critical orientation) is shown as the red dot. The robot will manipulate from the start location p_0 to the target location p_t. For the failed cases in this experiment, they were unable to complete the task due to the demonstration trajectories being close to the suspended grid, and the tile colliding before reaching the critical orientation. The multiple critical orientation approach was applied and resolved 34 failed cases. Only 3 cases still collided with the grid due to the inaccurate

demonstration orientation recording. Figure 3.33 shows one of the sequences of the robot executing the ceiling tile installation process.

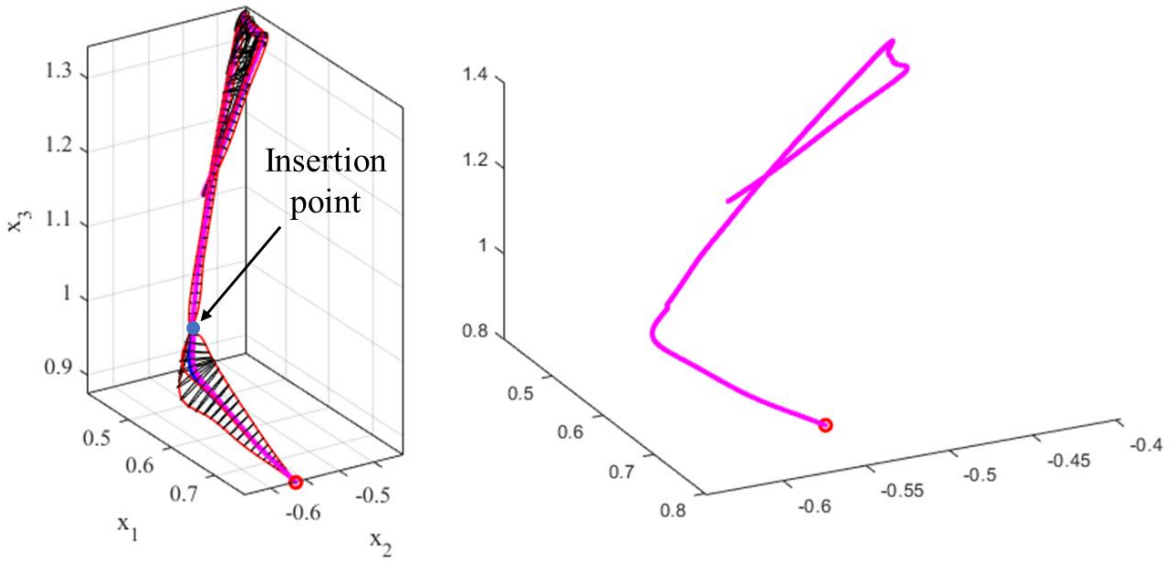


Figure 3.32 Results of the GCO Approach and the Generated Robot Trajectory.

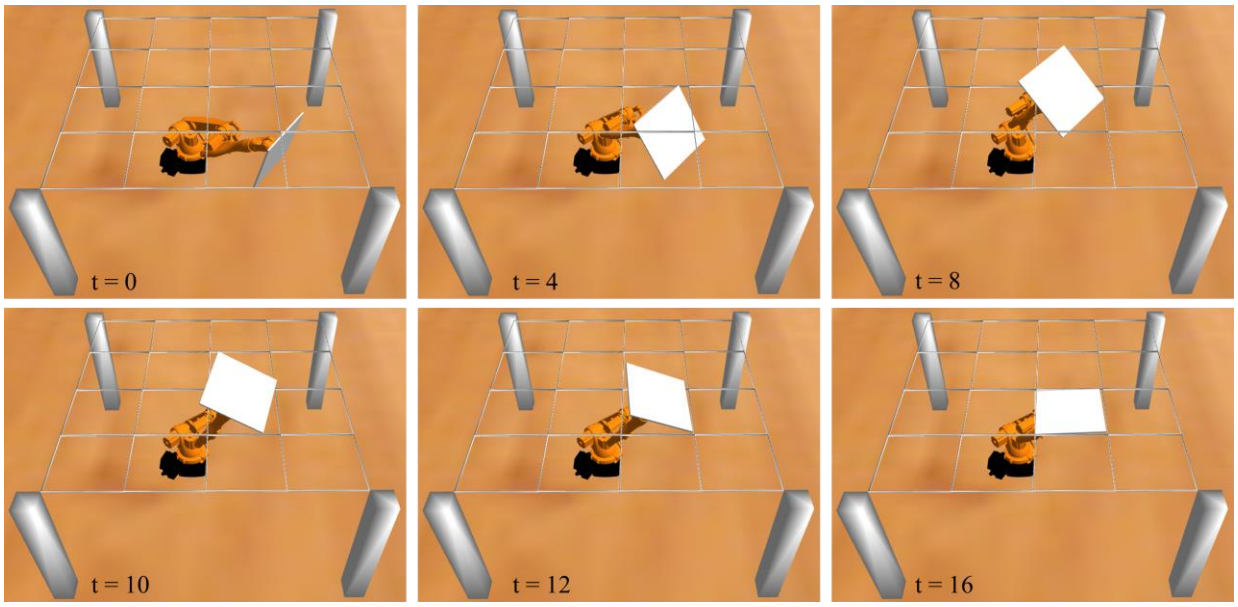


Figure 3.33 A Sequence of the Robot Executing the Ceiling Tile Installation Process Using the GCO Approach.

In the second phase of the experiment, the success rate of the GCO and trajectory adaptation approach is compared with the GC approach and the CTRL method. Table 3.5 shows the results of the GCO and trajectory adaptation approach (GCOT), GC and trajectory adaptation approach (GCT), and the CTRL method for the new start and target locations. First, the success rate of the GCOT is 82.0%, with 82 success cases and 18 failed cases. The 18 failed cases collided with the grid during the inserting process. Even with the multiple critical orientations, the tile still could collide with the grid after passing the last critical orientation. The other reason is the noisy demonstration orientation during the data collection phase. Second, the success rate of the GCT is 3.0%, with 3 success cases and 97 failed cases, which were unable to pass the grid. The low success rate of the GCT is due to incorrect orientation near the grid. Finally, the success rate of the CTRL method is 66.0%, with 66 success cases and 34 failed cases. Among the 34 failed cases, 24 were unable to pass the grid, and 10 exceeded the threshold.

Table 3.5 Results of the GCO and Trajectory Adaptation Approach, GC and Trajectory Adaptation Approach, and CTRL Method for the New Start and Target Locations.

Method	Success	Failure	Success Rate
GCOT	82	18	82.0%
GCT	3	97	3.0%
CTRL	66	34	66.0%

Figure 3.34 shows the results of the GCOT approach and the generated robot trajectory. The human worker first determines the new target grid location p_t to the robot. Then, the robot constructs the GC using four sets of translated demonstration data (thin lines) and connects to the new start location p_0. Lastly, the robot can generate the adapted trajectory (thick line) using the new GC. Figure 3.35 shows one of the sequences of the robot executing the ceiling tile installation process using the GCOT approach with the new start and target locations.

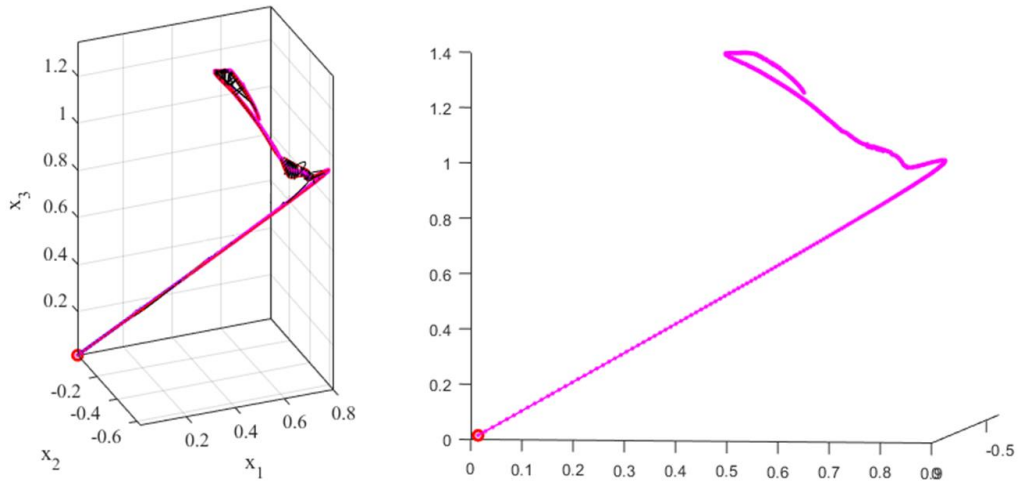


Figure 3.34 Results of the GCOT Approach and the Generated Robot Trajectory.

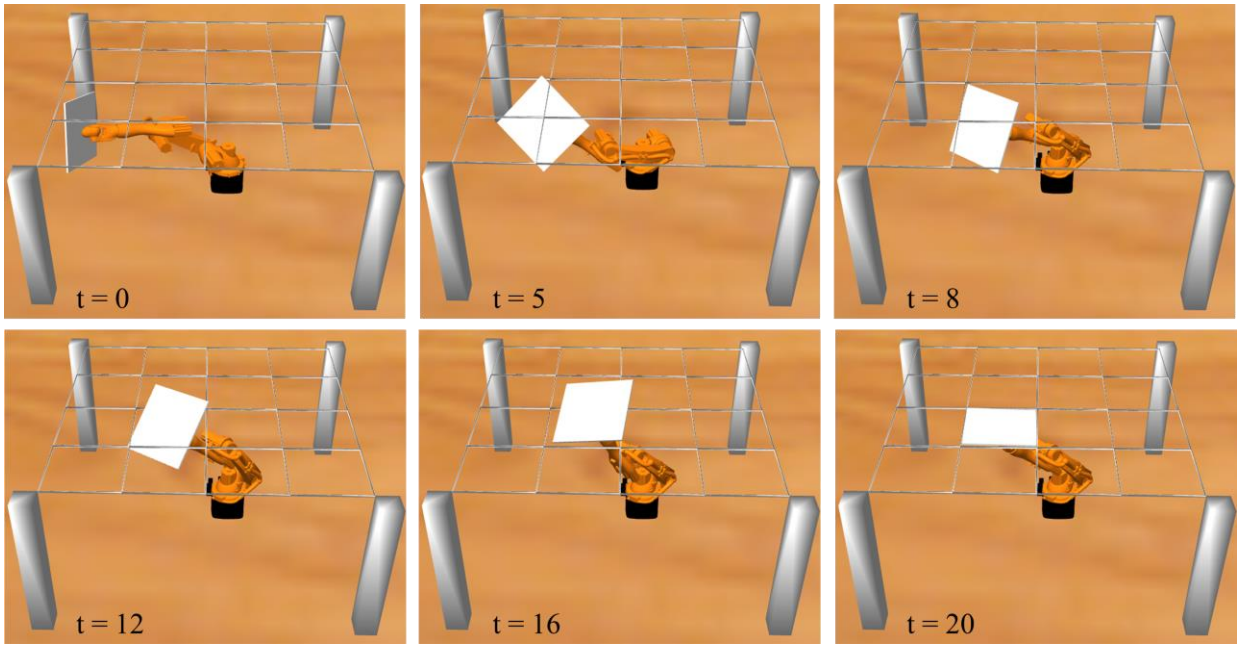


Figure 3.35 A Sequence of the Robot Executing the Ceiling Tile Installation Process Using the GCOT Approach with the New Start and Target Locations.

3.7 Discussion

In comparison with the results of Liu et al. [181], where they used the context translation model to teach a robot to push an object onto the target area, ladle almonds into a frying pan, and sweep the almonds into a dustpan, the success rate is 80% with 4,635 training videos (135 in real-

world and 4,500 in the virtual environment) of the object pushing, 66% with 3,060 training videos (60 in real-world and 3,000 in the virtual environment) of the almonds pouring, and 75% with 994 training videos (100 in real-world and 894 in the virtual environment) of the almonds sweeping.

In this research, the result of the ceiling tile installation is 78% with 3,085 training videos (85 in real and 3,000 in virtual), which achieves a similar performance. The primary issue encountered in the experiment is the inability to pass the suspended grid due to incorrect manipulating angles, which can be improved by providing more demonstration videos from several tile installation projects with a variety of geometric conditions. Furthermore, the use of the augmented, virtual, mixed reality (AVMR) can also help collect more demonstration videos easily [176,178] since the human can demonstrate different tasks by interacting with different construction conditions in virtual environments directly.

On the other hand, to evaluate the proposed trajectory-based Learning from Demonstration approach, the demonstration data and the experiment results were compared with the imitation learning from observation methods (IfO). For the demonstration data, the IfO methods typically require thousands of visual demonstration data (3,000 demonstration data in our experiment) [223], whereas the proposed trajectory-based approach only needs a few-shot demonstration (four sets of demonstrations in our implementation). The GCO approach requires detailed demonstration with the robot's end-effector 6-DOF trajectory, which can be collected using the robot simulator or the parametric VR system. Furthermore, demonstration data can also be in the physical environment using the object pose estimation method [229] or markers such as AprilTag and KEG algorithm [113,230] to track the 6-DOF pose of the manipulated object.

For the experiment results, the GCO approach achieves a higher success rate (75.3% for single critical orientation and 98.0% for multiple critical orientations) than the pure GC approach

(16%) and the CTRL method (71.3%) due to its close following of the human demonstration with detailed 6-DOF end-effector pose information. In the new start and target locations experiment, the GCOT approach also achieves a higher success rate (82.0%) than the GCT approach (3.0%) and the CTRL method (66.0%). The GC-based approach requires humans to indicate the target grid location, whereas the CTRL method requires the camera to point at the target grid. The CTRL methods can also achieve high accuracy by providing sufficient and variety of visual demonstration data.

There are some limitations to the proposed method of teaching robots construction tasks from visual or trajectory demonstrations. First, the camera viewpoint in the visual demonstration is fixed at the iso view in order to reduce the complexity. On a real construction site, however, it is difficult to set up the camera with the same viewpoint during the robot performing process. In addition, the bottom view cannot provide sufficient information since the trajectory of the ceiling tile is parallel to the viewpoint of the camera. A new set of training data with a variety of viewpoints in both real-world and virtual world needs to be collected in future work, which can also tackle the variation in the demonstration video and the target scene. Humans can also demonstrate tasks in Virtual Reality by interacting with different objects and placing the virtual camera at different locations [178].

Second, the experiment is conducted in a virtual prototyping simulator, which will have some differences when the method is applied in the real-world with a real robot. For example, the size of the ceiling tile and grids might not be perfectly matched, and additional scene understanding and adaptive manipulation techniques such as those described in [11,21] may be necessary for implementation. The human-in-the-loop adaptation can also help the robot adjust the component and provide additional instructions.

Third, the proposed method only used visual demonstration or trajectory demonstration as input for learning the construction task skill. However, some of the construction tasks require different types of demonstrations to learn the skill [23]. For example, if the ceiling tile and the grid are perfectly snug without any workable gap between them, the human worker has to push the tile up and down to overcome friction and place it at the correction location. This process will need multiple types of demonstrations, such as tactile observations to measure the contact force corresponding to the visuals, so that the movements can be recorded and fused with such additional observation streams to teach the robot. Thus, an ongoing study of combining visual demonstration and force feedback, or haptic feedback, for teaching construction robots is currently being investigated.

Fourth, the tile alignment must be checked in real practice to ensure quality, as illustrated in Figure 3.3 step 4. For this study, the ceiling tile installation is only concerned with ensuring that it is placed in its cell within the threshold volume, and the alignment is assumed to be already addressed in the grid construction phase (Figure 3.3 step 2). Thus, the overall grid alignment is assumed to be sufficient for this study. In real practice, an alignment checking mechanism such as a robot with a laser profiler must be included in the system to measure the alignment of the tile in the future, especially for other types of tiles such as those requiring plaster. If the alignment is incorrect or below the acceptable quality, the robot has to repeat the installation process until the tile alignment is correct.

Fifth, the environment feedback is assumed to be collected by additional sensors and registered to the virtual simulator [11,231]. However, when dealing with the dynamic changing environment on construction sites, synchronization between the virtual and the physical environment is required to provide real-time information. The online process-level Digital Twins

can ensure state synchronization between the physical and the virtual environment, which is introduced in the next chapter.

Sixth, the context translation model can handle a similar task under different scenarios. For example, placing the tile from a different starting position. However, when dealing with an unforeseen object such as the different shapes of the tile, additional algorithms must be applied to address the issue. For example, one-shot imitation learning [202] can be adapted to provide the robot the solution to the situation with limited demonstrations. The human can show the robot how to manipulate the different shape tiles with only one demonstration video, and the robot can improve their skill to handle such circumstances in the future.

3.8 Conclusions and Future Work

In this research, two robot LfD methods for training a construction robot to perform quasi-repetitive construction tasks were proposed and evaluated, in which the ceiling tile installation was utilized as the experimental construction process. First, a visual LfD method, i.e., the context translation model, was adapted and extended for the construction application. The context translation model only uses the visual demonstration as input to teach the robot how to perform a specific task.

There were two stages in the context translation model: construction task learning and construction task performing. In the construction task learning, the context from the training data was translated to the target initial observation, and then the robot could learn the translated context via the RL method, i.e., Trust Region Policy Gradient (TRPO) method. The model was trained on a set of ceiling tile installation demonstration videos, which was collected from the laboratory, and evaluated in the virtual ROS and Gazebo systems with a virtual KUKA robot arm.

Second, a trajectory-based Learning from Demonstration method to train robots to perform quasi-repetitive construction tasks is developed. The generalized cylinders approach was adapted and combined with orientation constraints to construct a geometric representation using demonstration data and generate the robot trajectory within the space with critical orientations (GCO approach). The trajectory adaptation approach and human-in-the-loop refinement approach were proposed to overcome unforeseen situations and avoid collisions (GCOT approach). The proposed GCO and GCOT approaches were evaluated in the robot simulator ROS Gazebo with ceiling tile installation demonstration trajectories collected from the human-controlled robot simulator. Only four sets of demonstration trajectories were required to construct the GC space and generate the robot trajectory.

For evaluating the visual LfD method, a total of 3,085 demonstration videos (85 in real-world and 3,000 in the virtual environment) were used for training, and 60 different scenes were used as test cases. The results showed that the model could translate the work context from the source video to the target observation with a success rate of 78% when using the iso camera view and used to find the robot control policy of the construction task with the TRPO method. For evaluating the trajectory LfD method, 150 test cases were selected for the GCO, and 100 test cases were selected for the GCOT and compared with the visual LfD method. The results showed that the GCO and GCOT could achieve 98.0% and 82.0% success rates with different start and target locations.

For future work, additional demonstration videos with a variety of viewpoints and environmental conditions will first be collected for enhancing the training data in both the real-world and virtual environments. Second, a human subjects study aimed at understanding how human workers interact with the robot using the proposed system to indicate the target location

and supervise the process will also be conducted to evaluate the human-in-the-loop refinement approach. Third, the deployment of real physical learning robots on actual construction jobsites will be developed and investigated in subsequent work. Fourth, the extension of the proposed approach to other quasi-repetitive construction tasks, such as drywall installation, will be investigated. Finally, the combination of multiple types of demonstration and sensor fusion, including trajectory, visual, and tactile demonstration, will be developed to tackle more complex construction tasks in cluttered work environments.

Chapter 4

Online State Synchronization Between Physical Robots and Process-Level Digital Twins

4.1 Introduction

Construction work is generally characterized by 3D (dull, dirty, and dangerous) [4]. Among all U.S. industries, statistics show that the construction sector ranks the highest in occupational injuries and fatalities [232]. Falls, struck by objects, electrocution, and caught-in/between are four leading construction worker death causes (Fatal Four) [233] due to active and close proximity interaction between human workers and heavy equipment [109,234]. Accidents happen when the machine operator and human workers are not aware of each other or misunderstand the intention. For example, human workers present in the blind spot of the excavator and are struck by the bucket [235]. On the other hand, human workers also have to repetitively handle large and heavy materials with machines to complete construction tasks, where the struck-by and caught-in/between hazards usually occur in such processes [236].

Research has explored with various solutions to help prevent construction accidents. The tracking system is one of the methods to continuously locate equipment and human workers on construction sites. This can be achieved by sensors such as RFID [237], UWB [238], IMU [68], or GPS [239], or by cameras and computer vision algorithms [139]. Furthermore, different types of sensors can be combined and fused to improve tracking accuracy [90,95]. Recent advances in computer vision algorithms have extended the construction equipment and workers tracking

algorithm to trajectory prediction [234]. Blindspot analysis and prediction can also identify the possible collision zone near the heavy equipment [65,240]. Lastly, tele-operated or autonomous equipment can help reduce the requirement of humans working close to the equipment [241,242]. This can be extended to using robots on construction sites.

4.1.1 Human-Robot Collaboration Safety

Robots deployment on construction sites can help relieve safety issues [21]. For instance, the construction robot can group with human workers on job-sites to assist with physically demanding tasks, while human workers focus on the work process plan and decision-making tasks. This is similar to the assembly line in the manufacturing industry, where the robots focus on repetitive and precise tasks, and humans focus on planning and checking tasks. Human-robot interaction is defined as humans and robots working in a shared environment with all types of interactions [243]. Wang et al. classified the relationships between humans and robots into four categories: coexistence, interaction, cooperation, and collaboration [244]. Human-robot collaboration (HRC) has among the most active interaction between humans and robots, where humans and robots are sharing the workspace and coordinating on the same task synchronously [245].

Symbiotic human-robot collaboration is one of HRC that applies to solve complex tasks [246,247] by combining their expertise and complementing proficiencies, which typically requires significant computational effort and training data. For example, the human has cognitive skills, decision-making ability, and ability to react reasonably to unexpected situations that might arise on a construction site, where as the robot has the advantage of high precision, strength, and repeatability [248]. However, such human-robot collaborative work suffers from safety and trust-related concerns [14,249] and is subject to strict safety standards [86]. For example, the robot must

be restricted for speed and force while collaborating with nearby human workers even though physical contact is allowed. A real-time human and robot tracking system can ensure safety by providing the information of the robot state to human workers [250].

On the other hand, since the symbiotic HRC consistently engages the human and robot with each other during the process, bi-directional communication is required to minimize the interruption and ensure safety [244]. In the human-to-robot direction, In the human-to-robot direction, communication can be achieved by directly commanding through the user interface to determine the robot goal or by using sensors to observe human movement, such as hand gestures, and extract the command [251]. The robot can easily understand the situation and execute the work plan. In the robot-to-human direction, the human has to be informed of the robot's work plan before execution. This can be achieved by providing a virtual representation, i.e., Digital Twin, of the robot and the environment. The robot's work plan can be demonstrated in the Digital Twin to the human in real-time and high-precision [244], allowing the human can make decisions based on the information.

4.1.2 Process-Level Digital Twin

The Digital Twin (DT) offers opportunities to virtually mimic the conditions of the physical (real) environment. This allows for a cyber-physical system (CPS) [252] where information of the current and forecasted future states of the robot can be displayed for decision making and evaluation prior to task execution [249]. Figure 4.1 shows the physical robot arm and its Digital Twin in the Gazebo simulator. Madni et al. [253] defined four levels of Digital Twin (Pre-Digital Twin, Digital Twin, Adaptive Digital Twin, and Intelligent Digital Twin) based on the level of intelligence. Of these four levels, the Adaptive Digital Twin combines user interface and machine

learning with regular DT, whereas the Intelligent Digital Twin further utilizes reinforcement learning to process the state in a partially observed and uncertain environment.

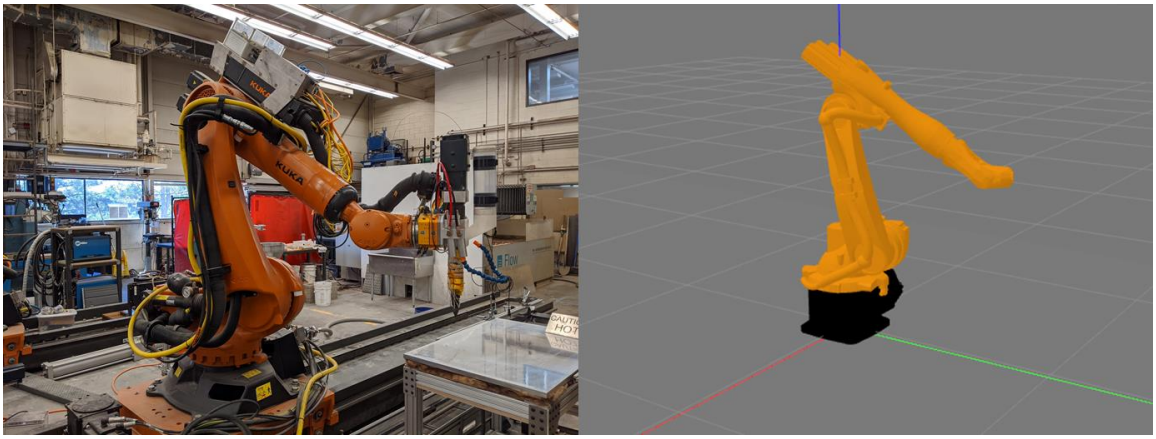


Figure 4.1 The Physical Robot Arm (Left) and its Digital Twin in Gazebo simulator (Right).

Adaptive Digital Twin or process-level Digital Twin replicates the entire physical process in real-time [254], such as the manufacturing assembly line process. Real-time is defined as whether the digital twin is able to complete the process correctly within pre-defined timestamps, i.e., deadlines [255]. The real-time system can be categorized as a hard real-time system, firm real-time system, and soft real-time system [256]. The hard real-time system has to accomplish each subtask before deadlines and will cause failure if missing any of the deadlines. For example, a 3D printer is considered a hard real-time system since the filament must be extruded at the right time as the extruder crosses the print bed. The firm real-time system can tolerate infrequent missing deadlines and count those as low-quality results. The soft real-time system can accommodate missing deadlines by reducing the quality of the result, such as live broadcasting video. The real-time process-level DT has to meet all deadlines in order to represent the physical environment and thus is defined as a hard real-time system [257].

One of the major aspects of the process-level DT is the synchronized model [258]. The DT first constructs the virtual model based on the physical environment, then records and tracks the

changes in the physical environment and reflects them in the virtual model. The virtual model can be extracted from the designed construction model such as BIM or scanned 3D point cloud of the as-built environment [259–261]. On the other hand, a communication mechanism is required to synchronize the data between the physical environment and the virtual model [244,252]. The communication needs to be bi-directional so that the virtual model can reflect the changes of the physical environment, and the user can determine the next steps in the virtual model and send the command to the physical environment. This level of data communication and connectivity is one of the challenges to applying DT in architecture, engineering, and construction discipline [262].

The remainder of this chapter is organized as follows. First, existing digital modeling methods and Digital Twin robotic systems are identified and reviewed to define the research gap. Second, the real-time process-level Digital Twin of the robotic construction process is developed. Third, the communication system and an algorithm for checking synchronization are introduced. Finally, experiments of robot motion planning and executing are conducted and used to evaluate the synchronization of the proposed real-time process-level Digital Twin.

4.2 Related Work

To enable the virtual simulator to mimic the physical robot and its workspace, two aspects need to be considered. First, the virtual simulator has to reconstruct the physical environment and dynamically reflect the changes, which is the digital modeling method. Second, the virtual simulator has to plan the robot's work plan and send commands back to the physical robot, which is the Digital Twin for the robotic system. The existing digital modeling methods and Digital Twin for construction robots are discussed in the following subsections.

4.2.1 Digital Modeling Methods

Digital modeling methods, such as 3D visualization or BIM, are used in the construction industry for design, management, and operation throughout the building life cycle [263,264]. These modeling methods document the project information and provide a platform for stakeholders to record changes, collaborate, and resolve conflicts [265,266]. In order to achieve a high-quality collaboration, the model must be fully synchronized with the physical environment. It is time and cost-prohibitive to manually update the model [267]. Thus, existing research focuses on automatically generating and updating the 3D model [268].

Collecting the 3D point cloud is one of the methods for generating the 3D model of the indoor environment [231]. This type of method requires a registration method for obtaining 3D points from cameras or laser scanners [10,203,269] and then applies segmentation methods to separate objects and reconstructs the semantic model [270–272]. Object recognition algorithms are also applied to identify different objects in the point cloud [273,274]. Finally, algorithms are required to automatically update the digital model based on the identified objects [275,276]. In the Digital Twin system, geometry assurance is developed to ensure the quality of the model [277,278]. The data transmission in these types of methods is from the physical environment to the virtual environment.

4.2.2 Digital Twin for Construction and Assembly Robots

Digital Twin has been envisioned to be the next generation of construction cyber-physical systems that can benefit the construction industry in decision-making and monitoring [279]. A similar approach can be used to integrate a construction robot with digital modeling methods for visualization and task planning [280]. For example, Yang et al. [281] utilized BIM and robot path

planners to find and visualize the construction process of modular construction. Shahmiri and Ficca [282] developed a parametric model that can directly control industrial robots to assemble the structure. Bruckmann et al. [283] used BIM as the data source to program a cable-driven parallel robot to construct masonry buildings. Similarly, Usmanov et al. [284] used BIM to program an industrial robot arm to lay bricks.

However, these types of systems are typically not synchronized between the virtual model and physical robot and require further adaptation to the design-build discrepancy [11]. One way to resolve the discrepancy is to use sensors to adapt the robot control [21,285]. On the other hand, the robot Digital Twin system developed in this work fulfills the demand for real-time data exchange, which is widely utilized in the manufacturing industry, digital fabrication, and human-robot collaboration assembly [286–288]. For example, Naboni and Kunic [289] used DT for complex wood structure manufacturing and assembly. Furthermore, by combining with other techniques such as Augmented Reality, the synchronization and communication mechanism of the robot DT system can be improved [290]. The data transmission in these types of methods is from the virtual environment to the physical environment.

4.3 Research Objective and Contributions

To address the issue of human-robot collaboration in construction work, an online process-level Digital Twin system is developed to bridge the virtual robot and physical robot in construction and digital fabrication. Robot Operating System (ROS) [204] is utilized to construct the framework of the system and create a robot arm model representing the physical robot arm in the Gazebo simulation environment [205]. ROS and Gazebo simulator have been utilized as modeling and operating tools for robotic buildings and environments [291] or multi-robot collaboration across different robot platforms [292].

In terms of bi-directional communication, MQTT [293] or TwinCAT ADS [294] are used to connect the virtual robot arm with the physical robot arm. The algorithm of checking the synchronization between the physical robot arm and the virtual twin is also developed. Various robot motion planning methods are implemented in the Digital Twin system to control the physical robot arm, including joint angle control and Cartesian path planning.

The proposed framework can be adapted to any robot arm models reflecting physical robots. The system is implemented in a fabrication laboratory with a full-scale KUKA KR120 6 DOF robot arm and evaluates the system by comparing the pose of the physical robot arm with the virtual robot arm. Several complex trajectories and sets of joint angles are collected to test the proposed system. Finally, to validate the hard real-time feature of our process-level Digital Twin system, the data transmission time between the virtual robot and the physical robot is measured.

4.4 Digital Twins of Robotic Construction Process

The proposed real-time process-level robot Digital Twin system consists of three modules: the physical robot module, the virtual robot module, and the communication module, as shown inFigure 4.2. First, the virtual robot module includes the Digital Twin for visualizing the robot and the motion planner for planning the trajectory and solving the inverse kinematics (IK). Second, the physical robot module includes the real robot arm and the embedded sensors for measuring joint angles. Finally, the communication module includes two different communication protocols (MQTT and ADS) for data exchange and synchronization.

The system is developed in Robot Operating System (ROS) since it is a meta-operating system that provides a message exchange mechanism between platforms across a network. For instance, the motion planner in the virtual robot module plans a trajectory and then sends the

control commands to the DT robot for execution and visualization. Each platform can be operated under different operating systems or programming languages.

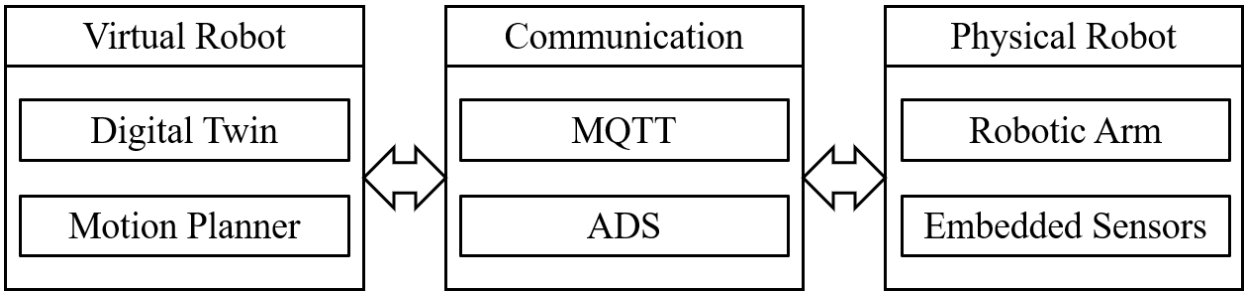


Figure 4.2 The Framework of the Online Process-Level Robot Digital Twin System.

Figure 4.3 and Figure 4.4 show the flowchart of the data exchange between each platform in the MQTT version and ADS version. The system requires at least one PC to run the Digital Twin system and one PC embedded on the robot to process the control commands. The two PCs are connected with ethernet for data exchange and communication using the MQTT or ADS protocol. A detailed description of each module is provided in the following subsections.

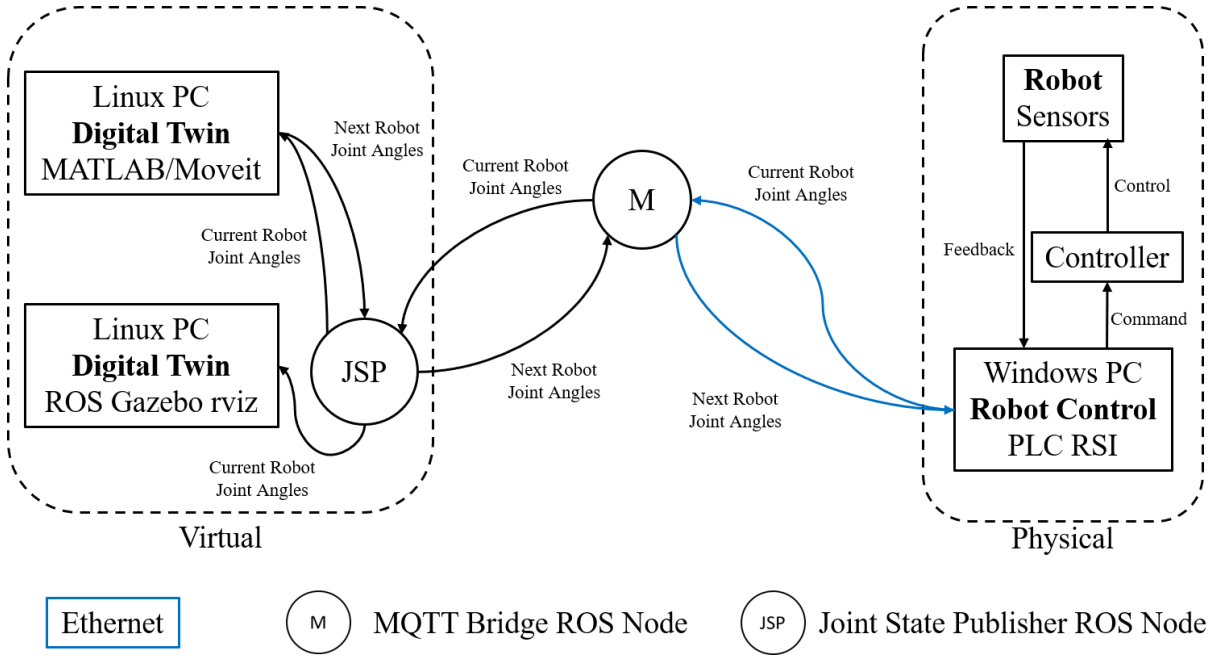


Figure 4.3 Flowchart of the Data Exchange Between Each Platform (MQTT Version).

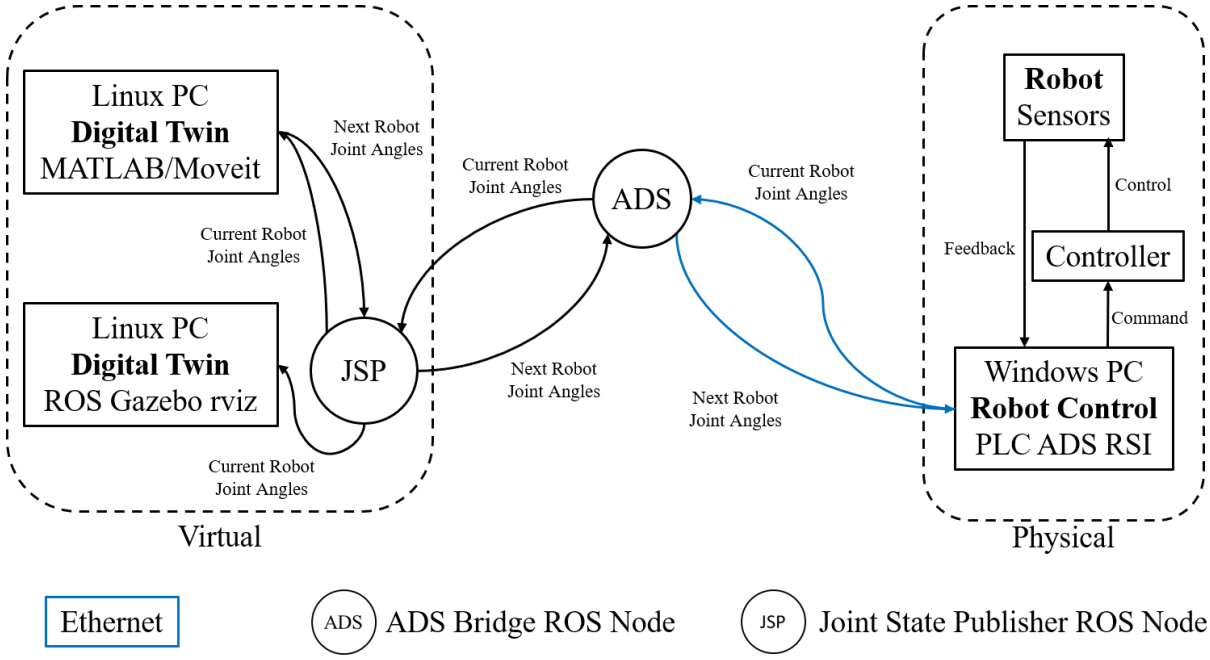


Figure 4.4 Flowchart of the Data Exchange Between Each Platform (ADS Version).

4.4.1 Virtual Robot Module

The ROS Gazebo and rviz are used to develop the DT in the virtual robot module on a Linux PC [205,222]. The Gazebo is a real-world physics simulator that creates a world and simulates the robot, whereas the rviz is visualization software that can read and display the data from Gazebo or real-world sensors. The robot arm model is imported to the Gazebo and rviz environment using the urdf format, as shown in Figure 4.5 and Figure 4.6. Two different robots are used as examples in the DT, i.e., KUKA KR5 and KUKA KR120 robot arms. The KR5 robot is a stationary robot arm whereas the KR120 is mounted on a track system with additional DOF. The joint angles of the robot arm are exchanged between the two programs to ensure synchronization.

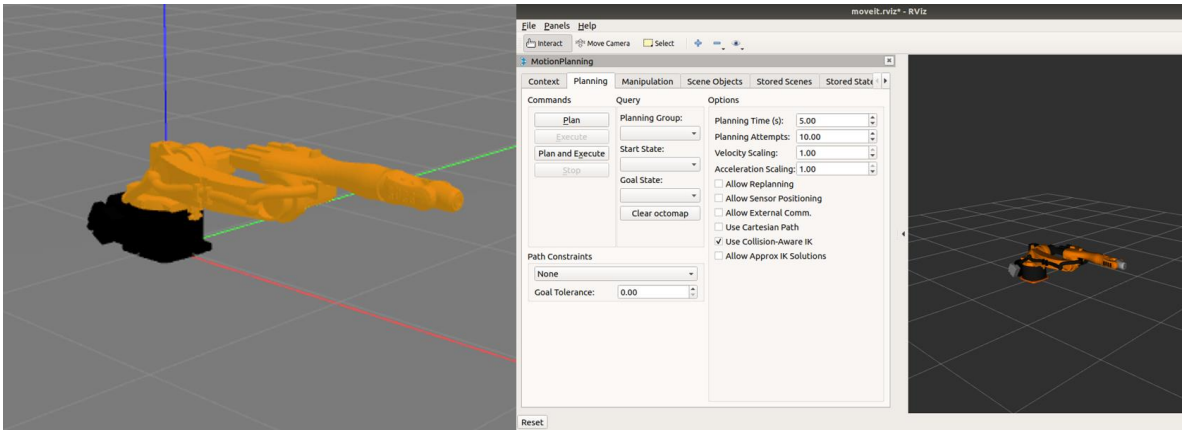


Figure 4.5 The KUKA KR5 Robot Arm in Gazebo (Left) and rviz with MoveIt! Package (Right).

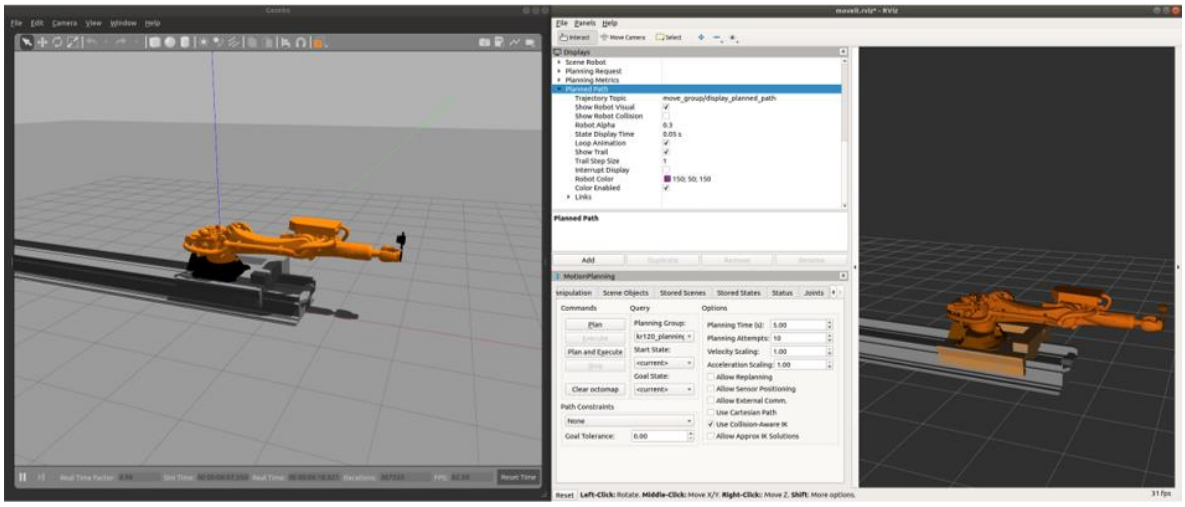


Figure 4.6 The KUKA KR120 Robot Arm in Gazebo (Left) and rviz with MoveIt! Package (Right).

In order to plan the specific construction task or motion, a motion planner is required in the module. Either MATLAB or MoveIt! can be used as the motion planner to achieve the task [295]. The Robotic System Toolbox in the MATLAB can plan the trajectory and solve the inverse kinematics of the robot. The built-in functions in the MATLAB provide faster programming ability to control the robot in various ways. However, it suffers from the latency issue and is not fast enough for real-time planning purposes.

On the other hand, the MoveIt! is a motion planning package for ROS, which plans the motion inside the rviz and sends it to the Gazebo. Figure 4.5 and Figure 4.6 right side shows the interface of the MoveIt! motion planning in rviz. The goal state, velocity, and time parameters can be customized and determined by the user as input to the motion planner. The result of the motion planning will then be demonstrated in rviz and sent back to Gazebo for execution. Both MATLAB and MoveIt! can be run on the same Linux PC as the DT or run on a different PC and connected through the network.

To allow the robot to perform different construction tasks, two control modes are included in the DT: joint state control mode and Cartesian path control mode. In the joint state control mode, the user can determine the target joint angles of the robot and let the MoveIt! package plan the trajectory starting from the current robot joint states. This is an intuitive way for the user to control the robot to the desired pose. In the Cartesian path control mode, the user can specify a list of waypoints and let the robot end-effector follow the trajectory. The MoveIt! package will calculate the robot joint angles using inverse kinematics and control the robot to execute the plan. For example, the user extracts the waypoints from a BIM model geometry for a 3D printing robot arm to determine the work plan. Note that in the path-driven programming, such as 3D printing, the joint control mode is rarely being used. The algorithm will extract waypoints and geometric instructions, typically linear and circular movements, to generate the robot control commands. It will require significantly more waypoints to achieve the same path fidelity with joint movements due to the complexity of the 7 DOF robot kinematics.

For the data exchange, only the current robot joint angles and the next robot joint angles are displayed within the virtual robot module. Both Gazebo and rviz read the current robot joint angles to visualize the robot state. The MATLAB or MoveIt! package read the robot joint angles,

determine the next robot joint angles, and send back to the Gazebo and rviz for execution. The joint state publisher (JSP) is the ROS node for publishing the current robot state to different ROS nodes, including the current robot joint angles from the physical robot module.

4.4.2 Physical Robot Module

In the proposed process-level DT system, the KUKA KR120 robot arm on the track system is the physical robot, as shown in Figure 4.7. The KUKA KR120 robot arm is a six-degrees-of-freedom robot with an additional external degree-of-freedom for the track system. The programmable logic controller (PLC), TwinCAT software system, and robot sensor interface (RSI) are running on an embedded Windows PC to control the robot arm and retrieve the sensor data. The embedded encoders on the robot arm are used to measure the joint angles and read by the RSI. In the ADS communication version, the TwinCAT ADS is also running on the embedded Windows PC to publish and receive the messages.

Figure 4.7 The KUKA KR120 Robot Arm for the Physical Robot Module.

After activating the robot arm, the system first records the current robot joint angles as the origin of the robot for robot controlling purposes. The sensor readings are directly connected to

the TwinCAT system and correlate with the robot position data. Once the physical robot receives the next joint angles from the virtual robot, it will calculate the differences of the joint angles and then use the recorded origin to control the robot arm in the relative mode. The robot control command and the sensor measurement are two data exchanges inside the physical robot module, as shown on the right side of Figure 4.3 and Figure 4.4.

Due to the limitation of the hardware data transmission speed and the missing data issue, some jitter effects might happen on the physical robot. To resolve this issue, different methods are used in the MQTT communication and the ADS communication. Currently, the first-order delay filter is applied in the TwinCAT program to smooth the robot trajectory for both MQTT and ADS communication. If a situation where missing data might arise, the delay filter can still interpolate and smooth the robot trajectory and avoid the jitter effects or sudden movements. However, the delay filter might produce slightly different trajectory compared to the planned trajectory. This can be resolved by applying the TwinCAT CNC package to generate the physical robot motion in the future. The CNC package can plan and interpolate the received waypoints to control the robot while respecting all dynamic limits and singularities of the robot. The density of the robot waypoints can be very high and synchronize all axes at the same time.

4.4.3 Communication Module

Finally, the communication module links the virtual robot module and the physical robot module. Two different communication protocols, i.e., MQTT communication protocol and TwinCAT ADS communication protocol, are developed for data exchange between the ROS system in the virtual robot module and the PLC in the physical robot module. Both MQTT communication protocol and TwinCAT communication protocol are capable of real-time communication and thus are suitable for smooth robotic control. First, an MQTT Bridge ROS node

(M) is developed to connect the MQTT to the ROS system, as shown in the middle of Figure 4.3. The MQTT Bridge node is run on the same Linux PC as the DT system to exchange the joint angles with the JSP node and connect with PLC in the physical robot module through the ethernet cable. The data exchange frequency in the MQTT Bridge is set to be 250 Hz to ensure the transmission speed on the robot arm.

The joint angles of the robot arm and the location of the track system are the primary data streams exchanged in the MQTT bridge ROS node. Figure 4.8 illustrates the data structure and exchange process in the MQTT bridge ROS node. The data stream concatenates the robot joint angles from A1 to A6 and the track location E1 joint with a plus-minus sign and comma. Each joint angle is rounded to three decimal places (E1 joint is rounded to four decimal places) and pads zeros to the left. Thus, the length of the data is consistent and quickly retrieved by PLC.

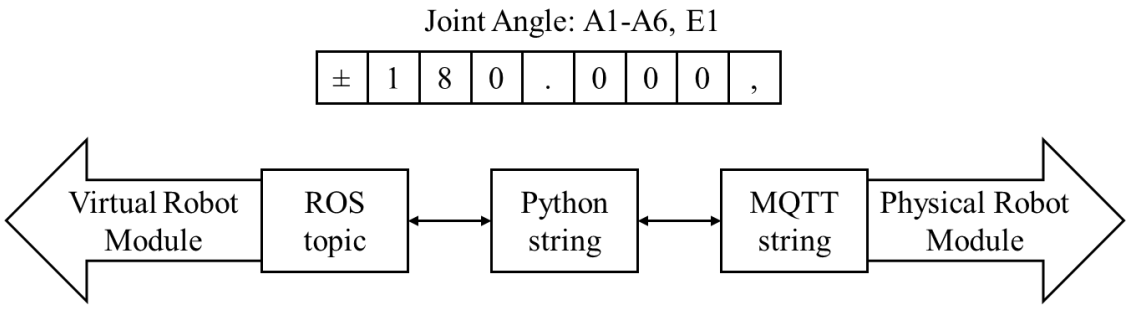


Figure 4.8 Data Structure and Exchange in the MQTT Bridge ROS Node.

After receiving the joint angles data from the virtual robot module through the ROS topic, the system first converts the data to python string for easy storage and access. Next, the data is converted to the MQTT string type and sent to the physical robot module. This process can also avoid the garbled text issue when directly converting from the ROS topic to the MQTT string type. The data stream from the physical robot module is also processed with the same procedure and data structure and sent to the virtual robot module.

Second, a TwinCAT ADS Bridge ROS node (ADS) is developed to connect the TwinCAT ADS to the ROS system, as shown in the middle of Figure 4.4. Similar to the MQTT Bridge node, the TwinCAT ADS Bridge node is also run on the same Linux PC as the DT system to exchange the joint angles with the JSP node and connect with the TwinCAT ADS and PLC in the physical robot module through the ethernet cable. The data exchange frequency in the TwinCAT ADS Bridge is set to be 1,000 Hz to ensure the transmission speed on the robot arm. The joint angles of the robot arm and the location of the track system are stored in an array and directly sent between the ADS and the ROS system. Both ADS and the ROS system can read and change the array data to reflect the work plan and the robot condition.

When exchanging the data between the virtual robot module and the physical robot module, the system must ensure the control commands are executed completely, and the pose of the physical and virtual robot is synchronized. A robot pose checking algorithm is designed to confirm the synchronization between the two robot arms. Algorithm 5 shows the pseudo-code of the pose checking algorithm (PCA). The algorithm takes the current virtual robot pose $\theta_{virtual}$, current physical robot pose $\theta_{physical}$, and the next robot pose θ_{next} as input.

Algorithm 5: Pose Checking Algorithm

procedure $Next\ Pose(\theta_{virtual}, \theta_{physical}, \theta_{next})$
  $diff(\theta) \leftarrow |\theta_{virtual} - \theta_{physical}|$
  if $diff(\theta) > threshold$ **then**
    $\theta_{next} \leftarrow \theta_{virtual}$
    Re-plan the trajectory based on θ_{next}
  else
    $\theta_{next} \leftarrow \theta_{next}$
  end if
  return θ_{next}
end procedure

First, the PCA calculates the difference of $\theta_{virtual}$ and $\theta_{physical}$. If the difference exceeds the pre-defined threshold, the next joint angle θ_{next} will be assigned with the current joint angles $\theta_{virtual}$ to ensure the physical robot can reach the desired joint angles. The trajectory also needs to be re-planned to reflect the current joint angles. On the other hand, if the difference does not exceed the threshold, the robot will simply execute the following joint angles.

4.5 Experiment and Results

The online process-level robot Digital Twin system is implemented and deployed in the Digital Fabrication Laboratory at the Taubman College of Architecture and Urban Planning, and the Structural Laboratory at the Department of Civil and Environmental Engineering at the University of Michigan. Three KUKA KR120 robot arms are the target physical robots, as shown in Figure 4.1 and Figure 4.7.

4.5.1 Experimental Setup

To evaluate the proposed system, experiments are conducted to verify the transmission time between the ROS system and PLC, and the pose between the physical robot and its DT is synchronized during trajectory execution. In the first experiment, a local network between two computers is set up to build the MQTT communication protocol and the TwinCAT ADS communication protocol to test the ROS Bridge node. Figure 4.9 shows the first experimental setup and the data exchange. The robot motion is planned on the first computer, and the trajectory is sent to the second computer through the MQTT or ADS for execution. The Cartesian path control mode is applied to plan four different motions, i.e., X-axis motion, Y-axis motion, Z-axis motion, and triangle motion. During the experiment, the timer is triggered when sending the pose from the 1st

computer and stopped when receiving the corresponding pose from the second computer to record the data transmission time.

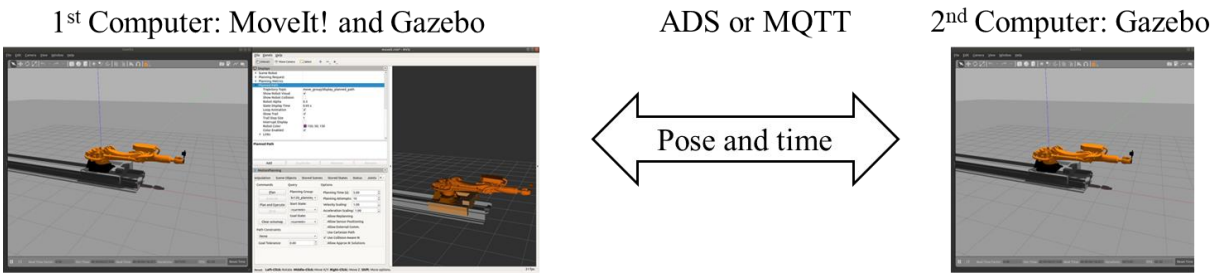


Figure 4.9 Two Virtual Robots are Used to Evaluate the MQTT and ADS Data Transmission Time.

In the second experiment, the MATLAB package is used to plan the robot trajectory and send the work plan through the MQTT or TwinCAT ADS Bridge node. One reaching task trajectory is prepared and executed in the MATLAB and Gazebo DT, then the joint angles are sent to the physical robot using MQTT or ADS communication. The robot poses are generated by the Inverse Kinematic package in the MATLAB. Figure 4.10 shows the planned reaching task trajectory (pink line) in the MATLAB. The embedded encoders on the KUKA robot arm are used to measure and record the joint angles of the physical robot. Figure 4.11 shows the procedure of the second process-level robot Digital Twin system experiment.

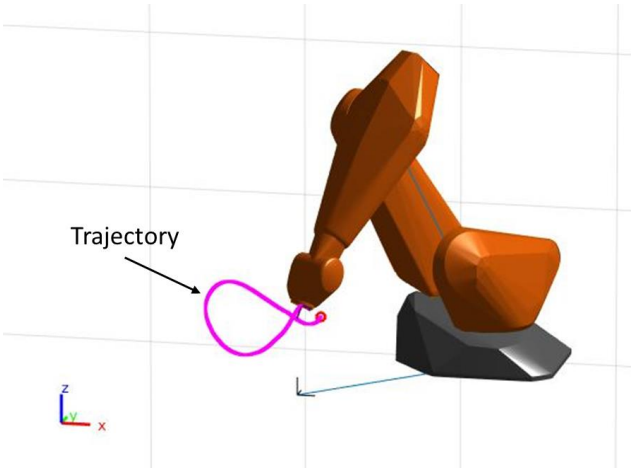


Figure 4.10 The Planned Reaching Task Trajectory (Pink Line) in MATLAB.

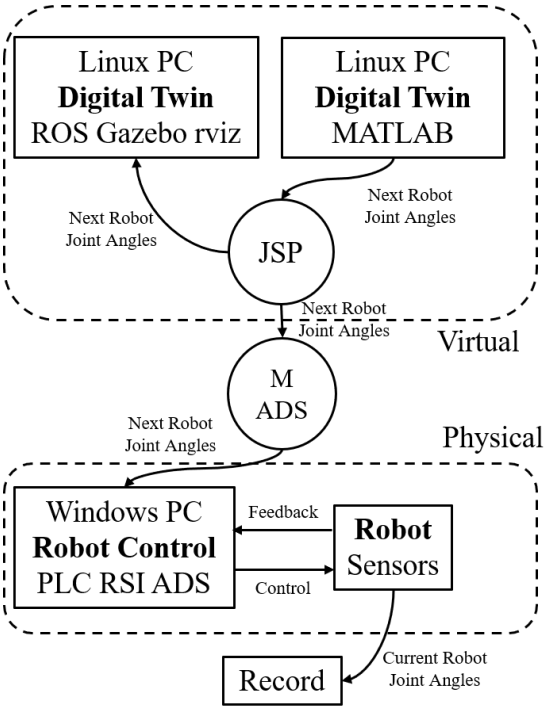


Figure 4.11 Procedure of the MATLAB Planned Digital Twin Experiment Using the MQTT or ADS Communication Protocol.

In the third experiment, the MoveIt! package is used to plan the robot trajectory and compare the accuracy of the trajectory execution between the two communication methods. Figure 4.12 shows the procedure of the MoveIt! planned process-level robot Digital Twin system experiment using the MQTT or ADS communication. The joint angles control mode is evaluated in this experiment. Ten different goal joint angles are randomly generated and planned the trajectories using the MoveIt! package. This information is then executed in the Gazebo DT and sent to the physical robot. Finally, the joint angles of the physical robot arm are measured and recorded to compare with the joint angles of the virtual robot arm.

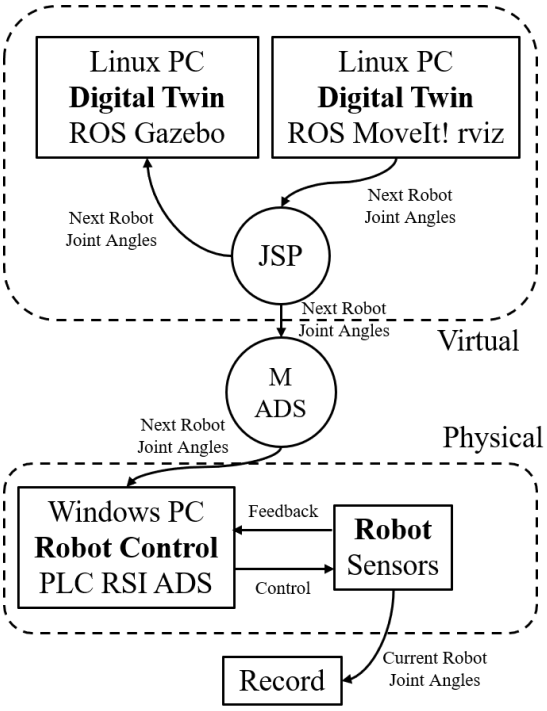


Figure 4.12 Procedure of the MoveIt! Planned Digital Twin Experiment Using the MQTT or ADS Communication Protocol.

In the final experiment, the MoveIt! package is used to plan the trajectory using Cartesian path control mode. Four different sets of end-effector waypoints are prepared (x-axis motion, y-axis motion, z-axis motion, and triangle motion) and executed in the Gazebo DT and sent to the physical robot. The pose of the physical robot arm end-effector is measured and recorded to compare with the planned waypoints and the end-effector of the virtual robot arm.

4.5.2 Experiment Results

In the first experiment, the data transmission time between two virtual robots is recorded and compared with the virtual to physical robot transmission time using both MQTT and ADS communication. Table 4.1 shows the result of the data transmission time experiment. MQTT (VtoV), ADS (VtoV), MQTT (VtoP), and ADS (VtoP) are four different settings, where VtoV represents Virtual robot to Virtual robot and VtoP represents Virtual robot to Physical robot. We

executed four trajectories (x-axis, y-axis, z-axis, and triangle motion) 100 times and collected 400 data points for two VtoV settings. For two VtoP settings, we executed four trajectories four times and collected 16 data points. The average data transmission time for MQTT (VtoV) is 8.786ms, and for ADS (VtoV) is 5.173ms due to the transmission frequency limitation of the MQTT communication protocol (250hz). For MQTT (VtoP), the average data transmission time is 12.547ms. Finally, ADS (VtoP) has an average of 9.483ms of data transmission time to exchange data between the DT and the physical robot to execute the work plan.

Table 4.1 Data Transmission Time Between Two Robots Using MQTT and ADS Communication

	Average Time	Maximum Time	Minimum Time
MQTT (VtoV)	8.786 ms	9.024 ms	8.141 ms
ADS (VtoV)	3.173 ms	3.905 ms	3.237 ms
MQTT (VtoP)	12.547 ms	15.771 ms	11.754 ms
ADS (VtoP)	7.483 ms	9.688 ms	6.595 ms

VtoV: Virtual to Virtual robot; VtoP: Virtual to Physical robot

In the second experiment, the joint angles of the physical robot and the MATLAB planned virtual robot are recorded and compared with each other. The stationary robot arm is used in this experiment, that is, excluding the track system (E1) joint. Figure 4.13 shows the results of the MQTT communicated virtual and physical robot joint angles using the MATLAB planned reaching trajectory. Each line represents the angle of each joint (A1, A2, A3, A4, A5, and A6) in radians. The trajectory from the virtual robot consists of 1,500 waypoints, and the measurement from the physical robot includes 18,802 data points. The results showed that the line of each joint angle had the same trend in the two robots, which demonstrated the consistency of the synchronization between the two robots using the MQTT communication protocol.

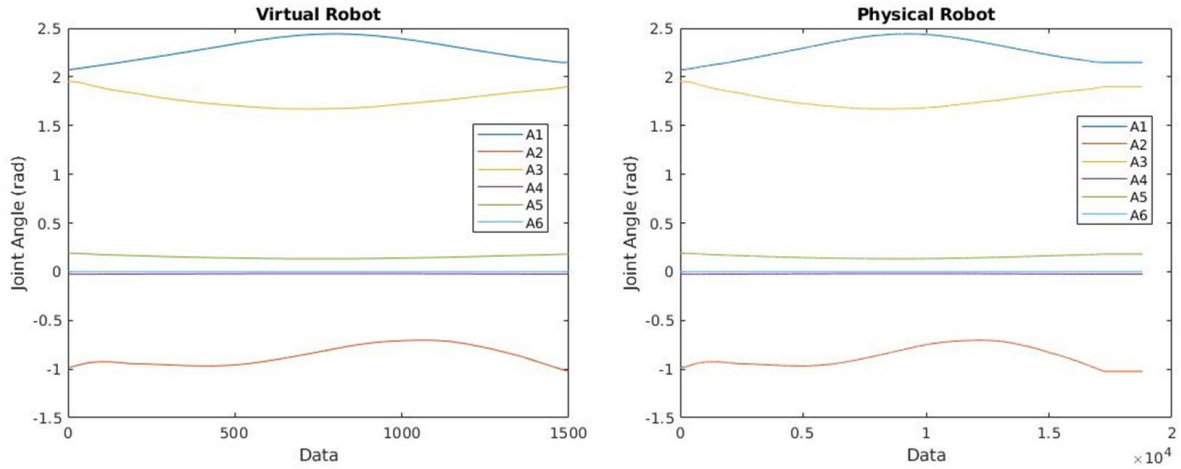


Figure 4.13 Results of the MQTT Communicated Virtual and Physical Robot Joint Angles Using the MATLAB Planned Reaching Trajectory.

On the other hand, the ADS version is also evaluated using the same procedure. Figure 4.14 shows the results of the ADS communicated virtual and physical robot joint angles using the same MATLAB planned reaching trajectory. The trajectory from the virtual robot consists of 1,500 waypoints, and the measurement from the physical robot includes 17,145 data points. The results showed that the ADS communication could also synchronize the joint angles between two robots.

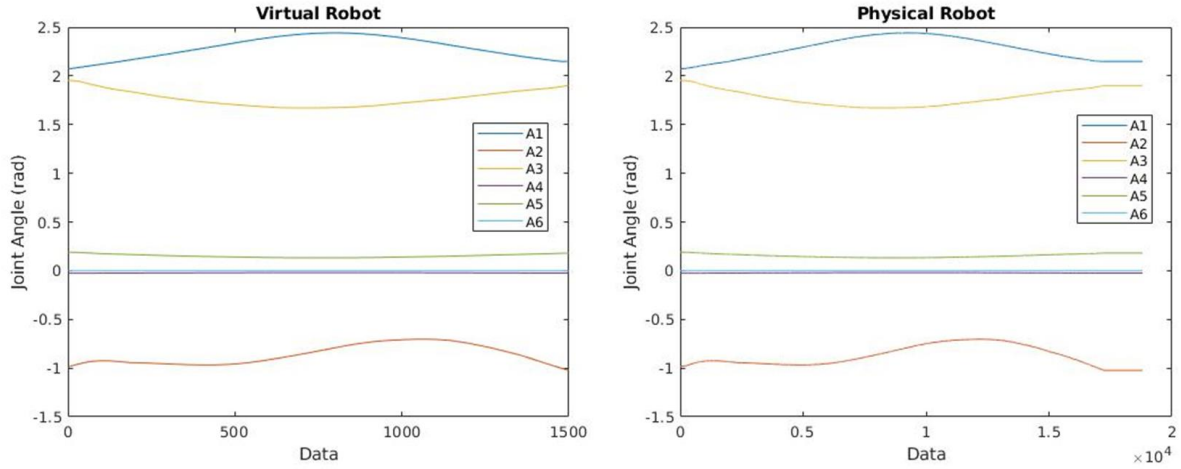


Figure 4.14 Results of the ADS Communicated Virtual and Physical Robot Joint Angles Using the MATLAB Planned Reaching Trajectory.

To further evaluate the accuracy of the synchronization, the average error and the maximum error of each joint angle between the two robots are calculated. Virtual robot and physical robot results are first aligned to obtain the same numbers of data from two robots in order to calculate the mean square error. Table 4.2 lists the results of the average and maximum joint angle error using the MATLAB planned trajectory in the MQTT and ADS communication. In the MQTT communication, the average errors of each joint angle are less than 0.0013 in radians, and the maximum errors of each joint angle are less than 0.0025 in radians. In the ADS communication, the average errors of each joint angle are less than 0.0012 in radians, and the maximum errors of each joint angle are less than 0.003 in radians. Note that there are numerous translations of the datatype between ROS and the physical robot. The joint angles were scaled up by 1,000 in the RSI layer for data processing, and then scaled down in the ROS system to convert to angles for the virtual robot. Therefore, the number under 0.001 radians is insignificant. These results indicated that the synchronization of the virtual and the physical robot demonstrated high accuracy. The proposed pose checking algorithm (PCA) also helps minimize the discrepancy between two robots during the data transmission.

Table 4.2 Average and Maximum Joint Angle Errors Between the Virtual and Physical Robot Using the MATLAB Planned Trajectory.

Joint	MQTT		ADS	
(rad)	Average Error	Maximum Error	Average Error	Maximum Error
A1	0.00024	0.00056	0.00034	0.00136
A2	0.00040	0.00076	0.00032	0.00073
A3	0.00077	0.00149	0.00077	0.00148
A4	0.00127	0.00241	0.00120	0.00293
A5	0.00030	0.00068	0.00037	0.00068
A6	3.535e-05	7.623e-05	4.345e-05	0.00017

In the third experiment, the joint angles of the physical robot and the MoveIt! joint angle control mode planned virtual robot are recorded and compared with each other. The track system

is included in this experiment (E1). Figure 4.15 and Figure 4.16 show the results of the MQTT and ADS communicated virtual and physical robot joint angles using the MoveIt! joint angle control mode planned trajectory. The trajectory from the virtual robot consists of 10 random goal poses, which includes 2,507 waypoints, and the measurement from the physical robot includes 2,513 data points. The trajectory patterns of the virtual and the physical robot are similar and only have minor errors.

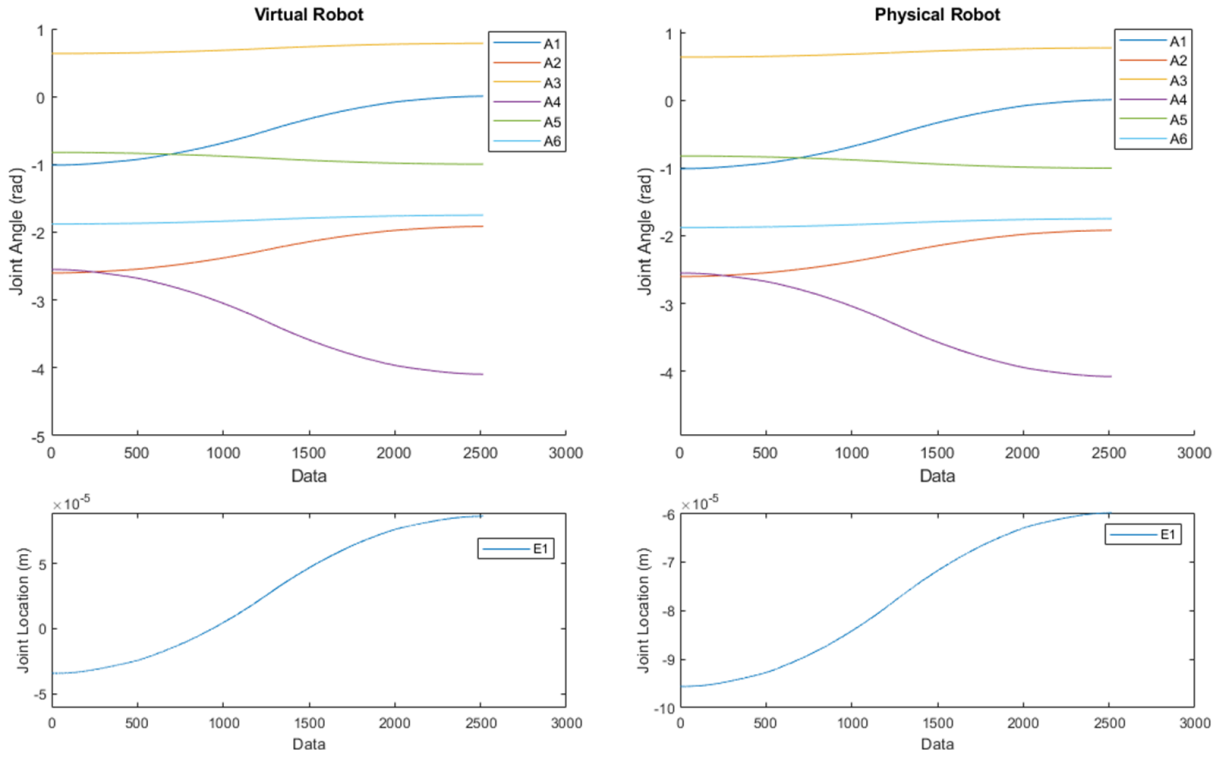


Figure 4.15 Results of the MQTT Communicated Virtual and Physical Robot Joint Angles Using the MoveIt! Joint Angle Control Mode.

Table 4.3 shows the results of the average and maximum joint angle error using the MoveIt! joint control mode in the MQTT and ADS communication. In the MQTT communication, the average errors of each joint angle are less than 0.099 in radians and less than 0.0042 in m for the E1 joint. The maximum errors of each joint angle are less than 0.195 in radians and less than 0.0011 in m for the E1 joint. The A5 joint has the highest error in the MQTT experiment.

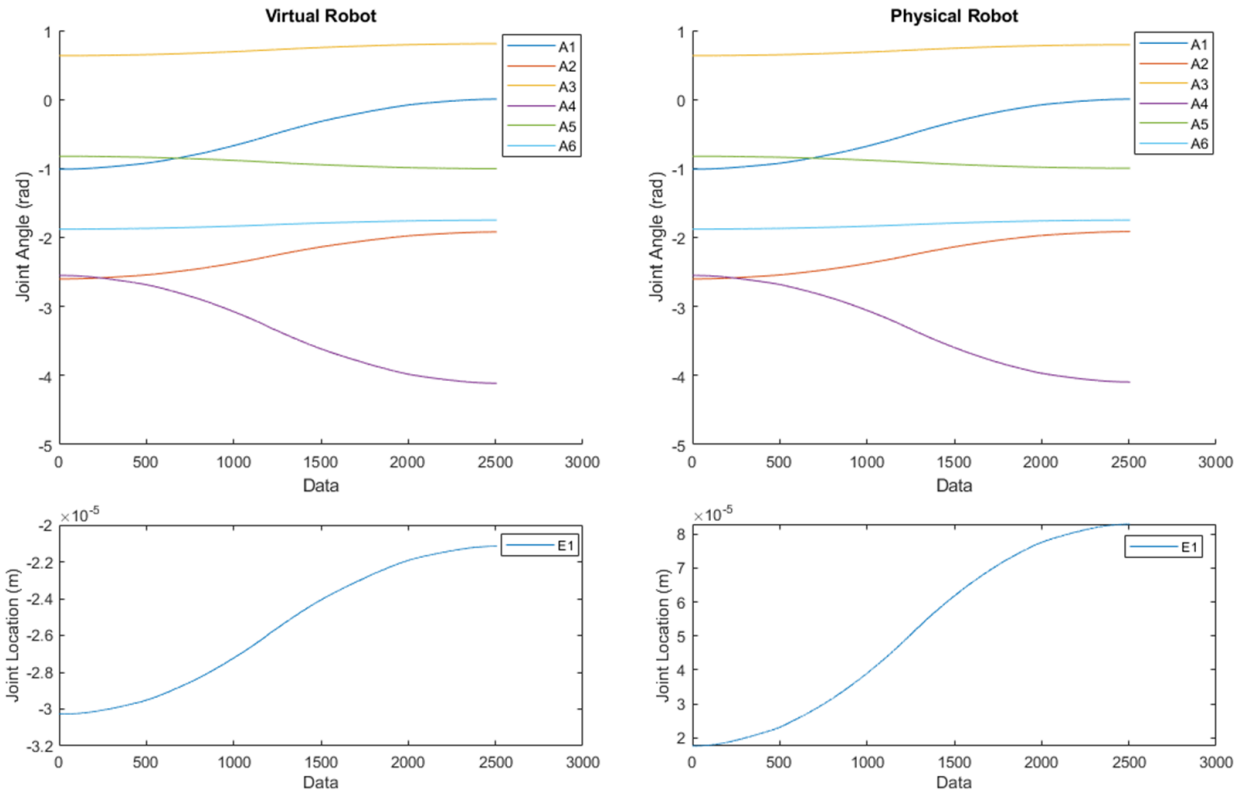


Figure 4.16 Results of the ADS Communicated Virtual and Physical Robot Joint Angles Using the MoveIt! Joint Angle Control Mode.

In the ADS communication, the average errors of each joint angle are less than 0.099 in radians and 0.0005 in m for the E1 joint. The maximum errors of each joint angle are less than 0.196 in radians and 0.0015 in m for the E1 joint. The A5 joint also has the highest error in the ADS experiment, and the average error of the E1 joint in ADS is smaller than the MQTT E1 joint. The results showed that the joint angle control mode in the MoveIt! package can precisely control the robot to the goal pose with some minor errors due to the first-order delay filter and synchronize with the physical robot by using the proposed PCA algorithm to ensure the accuracy of the joint angles.

Table 4.3 Average and Maximum Joint Errors Between the Virtual and Physical Robot Using the MoveIt! Joint Control Mode.

Joint	MQTT		ADS	
(rad)	Average Error	Maximum Error	Average Error	Maximum Error
A1	0.00239	0.00549	0.00272	0.00711
A2	9.117e-05	0.00024	0.00061	0.00099
A3	0.00072	0.00193	0.00088	0.00269
A4	0.00434	0.01143	0.00601	0.01600
A5	0.09804	0.19458	0.09887	0.19594
A6	0.00162	0.00427	0.00190	0.00565
E1 (m)	0.00410	0.00108	0.00048	0.00144

In the final experiment, the pose of the end-effector of the physical robot and the MoveIt! Cartesian path control mode planned virtual robot are recorded and compared with each other. Figure 4.17 shows the results of the MQTT and ADS communicated virtual and physical robot end-effector pose using the MoveIt! Cartesian path control mode planned trajectory. The solid blue line represents the planned trajectory in the virtual robot module, the red dash line represents the MQTT executed trajectory, and the yellow dot line represents the ADS executed trajectory. Each line represents the position of the robot end-effector in world coordinate (X, Y, Z).

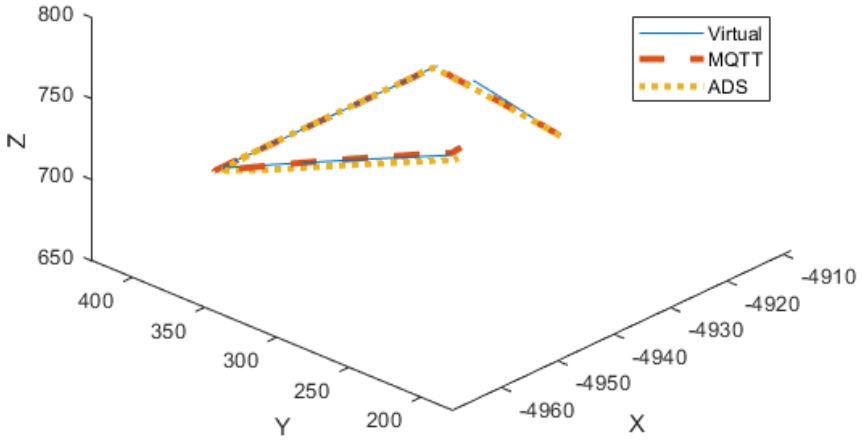


Figure 4.17 Results of the MQTT and ADS Communicated Virtual and Physical Robot's End-Effector Pose Using the MoveIt! Cartesian Path Control Mode.

In addition, the average and maximum errors of the end-effector pose are also calculated, as listed in Table 4.4. The average errors of the robot end-effector are 1.422 mm on the X-axis, 5.015 mm on the Y-axis, 1.967 mm on the Z-axis, and overall 6.487 mm for the MQTT communication protocol, where the maximum errors are 7.787 mm on the X-axis, 14.345 mm on the Y-axis, 7.204 mm on the Z-axis, and overall 16.052 mm. For the ADS communication protocol, the average errors are 1.543 mm on the X-axis, 3.667 mm on the Y-axis, 1.842 mm on the Z-axis, and overall 5.284 mm. The maximum errors are 8.027 mm on the X-axis, 7.695 mm on the Y-axis, 6.854 mm on the Z-axis, and overall 10.314 mm.

Table 4.4 Average and Maximum End-Effector Pose Errors Using the MoveIt! Cartesian Path Control Mode.

(mm)	MQTT		ADS	
	Average Error	Maximum Error	Average Error	Maximum Error
X	1.422	7.787	1.543	8.027
Y	5.015	14.345	3.667	7.695
Z	1.967	7.204	1.842	6.854
Overall	6.487	16.052	5.285	10.314

4.6 Discussion

The difference in the transmission time between the MQTT communication and the ADS communication is less than 5 ms. The TwinCAT system is capable of running in real-time as the code is execute within 1 ms. The ADS communication ROS node can directly modify the joint angle variable in the TwinCAT system, and the MQTT communication ROS node has to convert joint angle data to several different formats (Figure 4.8), thus requiring extra time to process the data. In addition, both the ADS and MQTT communications are capable of connecting to the CNC package and precisely controlling the physical robot in the future. For the accuracy of the physical robot execution, both the MQTT and ADS communication reach similar performance, and the

errors of the joint angle are within 0.1 rad. The first-order delay filter causes some errors since the joint angle data received by TwinCAT are slightly different each time, and the smoothed trajectories are different. The errors are reduced by the proposed pose checking algorithm (PCA). Both MATLAB and the MoveIt! package achieve similar performance on planning the robot motion by joint control mode and Cartesian path mode and communicating with the communication module. However, the MATLAB package has limited the data communication frequency (up to 100 Hz) and requires a code generation function to improve the execution speed.

The advantages of using the proposed bi-directional communication protocol to synchronize the virtual and real robot arms are to ensure that the collaborating human worker is informed of the work plan of the robot, to supervise and review the robot's work plan, and to control the robot accurately. The use of the ROS framework also has the advantage of adapting to various robot systems or controlling software. For example, instead of using MATLAB or MoveIt! package, the modeling software such as Rhino can be used to directly extract the trajectory from components and send it to the communication module to exchange with the physical robot module.

However, there are several limitations of the proposed system that need to be addressed in future work. First, the pose of the physical robot is measured by the onboard sensors. If any of the sensors fail, the virtual robot will not be able to synchronize with the physical robot. In the industrial robots, a failed encoder will cause an error state stopping all motion, and there is failsafe component to ensure the data correctness. This can be resolved by applying additional sensors, such as camera [71], to supplement the pose estimation and fusing with the onboard sensor data.

Second, some of the limitations of the physical robot are not reflected in the virtual robot. For example, the velocity and the acceleration limits of the robot joints are not incorporated into the path planning correctly and the physical robot will stop due to the sudden high acceleration.

The dynamic limits can be reflected in the TwinCAT CNC package to control the physical robot. Third, the detailed information of the surrounding environment, e.g., obstacles in the workspace, is not included in the proposed Digital Twin system. When planning the robot trajectory, those obstacles need to be considered to avoid the collision.

4.7 Conclusions and Future Work

This chapter presents the development of the online process-level robot Digital Twin system for human-robot collaboration in the construction and digital fabrication. The system includes the virtual robot module, the physical robot module, and the communication module. The ROS Gazebo and rviz were leveraged to develop the virtual robot module, i.e., Digital Twin of the physical robot, and connected to the physical robot module through the MQTT Bridge or TwinCAT ADS Bridge in the communication module. The joint angles of the robot arm were exchanged and synchronized between two robots. The MATLAB or MoveIt! package was also utilized to plan and control the robot arm in the virtual robot module, then sent the command to the physical robot module for execution. In addition, two different control modes, i.e., joint angle control mode and Cartesian path control mode, were implemented in the MoveIt! program to control the virtual robot by joint angles or end-effector pose. Finally, a pose checking algorithm (PCA) was developed to ensure the pose of the two robots were synchronized.

The system was implemented and deployed on a KUKA KR120 robot arm in the Digital Fabrication Laboratory and the Structural Laboratory to evaluate the synchronization and the data transmission time. Although the system was developed for the specific KUKA robot arm, it can be easily adapted to other robot models. The system was evaluated by comparing the data transmission time, joint angles, and end-effector pose between the virtual and physical robot using several planned trajectories and calculated the average and maximum mean square errors. The

results showed that the proposed online process-level robot Digital Twin system can plan the robot trajectory inside the virtual environment and execute it in the physical environment with high accuracy and real-time performance.

In future work, the user interface will be designed for displaying the information of the physical robot in the Digital Twin. The robot planning mechanism will also be developed such that the robot can first demonstrate the planned trajectory inside the Digital Twin before executing by the physical robot. The human can thus expect the movement of the robot in advance and approve the task.

Chapter 5

Conclusion

5.1 Significance of the Research

Robots have the potential to benefit construction workers by assisting with dirty, dull, and dangerous work, such as lifting and installing heavy components, while letting human workers focus on high-level sequential planning and supervising jobs. This research focuses on three key aspects to enable construction robots to work with human workers as a human-robot collaboration on construction sites performing quasi-repetitive tasks while ensuring the safety between human workers and their robot apprentices. In addition, the use of robot pose estimation, robot Learning from Demonstration, and robot Digital Twin can also benefit the construction work in other applications. For example, the robot pose estimation method can be applied to productivity analysis to locate multiple machines on-site and identify their working cycles.

The proposed robot Learning from Demonstration method can also be applied to teach new construction worker recruits different tasks. For example, when a skilled worker is unavailable, the robots can demonstrate the construction task they learned before to a group of novice workers. Then, the robots can observe how recruits perform and practice the task. The observation by the robots can be used for updating the knowledge for future demonstration. The proposed robot Digital Twin method can be extended to BIM model synchronization. For instance, the virtual robot module gathers the geometric information from the BIM model and plans the construction

task. The physical robot module will execute the work plan, gather the geometric outcome from the environment, and send it back to the virtual robot module to update the BIM model.

Finally, the proposed human-robot-environment collaboration can be applied to field robots in a built environment, such as resilient infrastructure robots. These robots have to navigate in an open and environmentally hostile workspace to different locations or in an underground hazardous environment to perform tasks for disaster recovery. The ability to learn from human workers and accomplish the work plan in a dangerous working environment can benefit community resilience and provide sustainable construction.

5.2 Research Contributions

This research investigates the collaboration and interaction between humans, robots, and the environment on construction sites by leveraging pose estimation, Learning from Demonstration (LfD), and Digital Twins. The contributions of each research topic are listed as follows.

1. Vision-based and fusion-based pose estimation methods for large-scale articulated construction robots
    - A DNN-based 2D and 3D vision pose estimation system was modified and applied to articulated construction robots.
    - A fast dataset collection approach was proposed to rapidly collect image data and 3D ground truth data of the robot.
    - A sensor-based (IMU) pose estimation system was implemented to evaluate the performance of the proposed vision pose estimation system.

- A DNN-based sensor fusion pose estimation system was proposed to combine vision pose and sensor pose and improve the accuracy and consistency in highly occluded construction environments.
- The proposed 2D, 3D, and fusion pose estimation systems were tested using the excavator dataset collected by the fast dataset collection approach and demonstrated the applicable for proximity-related applications.

2. Teaching robots quasi-repetitive construction tasks using Learning from human Demonstration
    - A robot Learning from Demonstration method was adapted for quasi-repetitive construction tasks using visual demonstration.
    - A trajectory-based Learning from Demonstration method was proposed to teach robots construction processes involving manipulation.
    - A trajectory adaptation approach and a human-in-the-loop refinement approach were designed to refine the robot trajectory in unforeseen scenarios and avoid obstacles.
    - The proposed visual LfD and trajectory LfD methods were tested in a virtual simulator with a KUKA KR120 robot arm performing the ceiling tile installation process.
3. Online process-level Digital Twin and bi-directional state synchronization
    - A process-level Digital Twin and a bi-directional state synchronization method were developed to bridge the virtual and physical robot for construction and digital fabrication processes.

- Two different communication protocols were implemented in the process-level Digital Twin.
- A pose checking algorithm was developed to ensure the state synchronization between two robots.
- The proposed Digital Twin was evaluated with several robot trajectories and shown to be feasible for construction robot precise motion control.

5.3 Future Directions

This section discusses future research directions in the human-robot-environment interplay for the performance of construction work.

5.3.1 Pose Estimation and Localization

In order to improve the performance of the pose estimation system, additional research can be conducted to analyze the camera coverage and determine the optimal camera deployment network. Furthermore, the advanced robot localization methods can be investigated that adapt to the circumstances on unstructured and cluttered construction sites. Specifically, the reinforcement learning, occupancy grid mapping, and simultaneous localization and mapping (SLAM) [60,203] algorithm could be incorporated to enable construction robots to navigate and localize in unstructured environments.

5.3.2 Learning from Demonstration

The robot LfD, IL, or programming by demonstration methods [30,296,297] open avenues to new research areas of teaching robots complicated construction tasks. The human workers can transition their work profiles to that of demonstrators and supervisors without the need to harbor

any concerns about being displaced by robots. Since the human demonstrates the task to the robot, additional interaction methods are mandatory, such as voice or gesture [298], to control or indicate intentions to the robot. Moreover, Michalos et al. [299] proposed the enhancement of LfD by using voice and natural language to command robots and using visual recognition methods and force sensors to demonstrate the tasks. The sensor fusion methods are also required to obtain a reliable LfD result by combining different types of demonstration data [300].

When the human supervisor and the robot apprentice are collaborating on an actual construction site, the way they interact with each other is critical. There is a need for seamless communication regardless of whether the human is directly observing the robot or doing general work planning in the vicinity. The use of extended reality (XR) can be explored by developing mechanisms to encode the robot's perception of their environment and display their planned trajectory to the human worker for approval. Mixed Reality (MR) can be employed to develop the human-robot supervision system and combine with Digital Twin. The construction robot will first use the learned knowledge to determine the control policy and trajectory and let its Digital Twin perform the task in MR. The human supervisor will then confirm whether the steps demonstrated by the virtual robot are acceptable and permit the real robot to perform the construction task.

In addition, the improvisation method of the robot work plan can be further explored. If the work plan of the robot displayed in the MR is unacceptable or prone to failure, improvisation intervention from the human worker is required to resolve the issue. One option is to ask the supervisor to wear haptic gloves and interact with the construction component in MR to demonstrate the improvised actions to the robot apprentice. The robot will then imbibe the improvisation steps, determine a new control policy and trajectory, and demonstrate to the supervisor again for approval. Once approved, the robot can perform the task accordingly.

5.3.3 Process-Level Digital Twin

This research assumes the geometric information of the environment is available and thus only focuses on the robot. The nature of the unstructured and dynamic-changing construction sites needs to be considered to develop the fully functional Digital Twin. Future study can investigate the object recognition methods by computer vision approaches to identify real-world objects, point cloud generation methods by SLAM algorithm to create the 3D model, and model registration methods to register the object in the 3D model, which can then be used to update the Digital Twin and the underlying BIM model representing the project's design.

Appendix A

Links to Datasets

Excavator 3D Dataset

The excavator 3D dataset can be found at this link. The annotation of the dataset is documented as follows:

- bbox represents the bounding box $(X_{top\_left}, Y_{top\_left}, X_{bottom\_right}, Y_{bottom\_right})$.
- camPose represents the camera pose $(X, Y, Z, Yaw, Pitch, Roll)$.
- image_train indicates the training images.
- Y2d represents the 2D pose ground truth data $(X_c, Y_c, X_{bo}, Y_{bo}, X_s, Y_s, X_{bu}, Y_{bu})$.
- Y3d represents the 3D pose ground truth data $(X_c, Y_c, Z_c, X_{bo}, Y_{bo}, Z_{bo}, X_s, Y_s, Z_s, X_{bu}, Y_{bu}, Z_{bu})$.

Excavator 2D Dataset

The excavator 2D dataset can be found at this link. The annotation of the dataset is documented as follows:

- objpos represents the center of the bounding box (X, Y).
- joint_self represents the 2D pose ground truth data $(X_c, Y_c, V_c, X_{bo}, Y_{bo}, V_{bo}, X_s, Y_s, V_s, X_{bu}, Y_{bu}, V_{bu})$, where V indicates the visibility.
- scale_provided represents the excavator scale w.r.t. 200 px height.

- isValidation indicates the validating images.

Ceiling Tile Demonstration Video

The ceiling tile demonstration videos can be found at this link.

Bibliography

[1] P.M. Goodrum, M.F. Yasin, Relationship between changes in material technology and construction productivity, Journal of Construction Engineering and Management. 135 (2009) 278–287. https://doi.org/10.1061/(ASCE)0733-9364(2009)135:4(278).

[2] A.H. Behzadan, A. Iqbal, V.R. Kamat, A collaborative augmented reality based modeling environment for construction engineering and management education, in: Proceedings of the Winter Simulation Conference (WSC), IEEE, Phoenix, AZ, USA, 2011: pp. 3568–3576. https://doi.org/10.1109/WSC.2011.6148051.

[3] J.G. Everett, A.H. Slocum, Automation and robotics opportunities: construction versus manufacturing, Journal of Construction Engineering and Management. 120 (1994) 443–452. https://doi.org/10.1061/(ASCE)0733-9364(1994)120:2(443).

[4] K.M. Lundeen, Autonomous scene understanding, motion planning, and task execution for geometrically adaptive robotized construction work, Dissertation, University of Michigan, 2019. https://deepblue.lib.umich.edu/handle/2027.42/149785 (accessed April 30, 2020).

[5] W. Lee, K.-Y. Lin, E. Seto, G.C. Migliaccio, Wearable sensors for monitoring on-duty and off-duty worker physiological status and activities in construction, Automation in Construction. 83 (2017) 341–353. https://doi.org/10.1016/j.autcon.2017.06.012.

[6] V. Arndt, D. Rothenbacher, U. Daniel, B. Zschenderlein, S. Schuberth, H. Brenner, Construction work and risk of occupational disability: a ten year follow up of 14 474 male workers, Occupational and Environmental Medicine. 62 (2005) 559–566. https://doi.org/10.1136/oem.2004.018135.

[7] T. Bock, Construction robotics, Autonomous Robots. 22 (2007) 201–209. https://doi.org/10.1007/s10514-006-9008-5.

[8] Tractica, Construction & demolition robots, 2019. https://tractica.omdia.com/research/construction-demolition-robots/ (accessed April 23, 2020).

[9] T. Bock, T. Linner, Construction robots: elementary technologies and single-task construction robots, 1st ed., Cambridge University Press, 2016. https://doi.org/10.1017/CBO9781139872041.

[10] C. Feng, Y. Xiao, A. Willette, W. McGee, V.R. Kamat, Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites, Automation in Construction. 59 (2015) 128–138. https://doi.org/10.1016/j.autcon.2015.06.002.

[11] K.M. Lundeen, V.R. Kamat, C.C. Menassa, W. McGee, Scene understanding for adaptive manipulation in robotized construction work, Automation in Construction. 82 (2017) 16–30. https://doi.org/10.1016/j.autcon.2017.06.022.

[12] K.S. Saidi, J.B. O'Brien, A.M. Lytle, Robotics in construction, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer, Berlin, Heidelberg, 2008: pp. 1079–1099. https://doi.org/10.1007/978-3-540-30301-5_48.

[13] C. Balaguer, M. Abderrahim, Trends in robotics and automation in construction, in: Robotics and Automation in Construction, IntechOpen, 2008. https://doi.org/10.5772/5865.

[14] S. You, J.-H. Kim, S. Lee, V. Kamat, L.P. Robert, Enhancing perceived safety in human–robot collaborative construction using immersive virtual environments, Automation in Construction. 96 (2018) 161–170. https://doi.org/10.1016/j.autcon.2018.09.008.

[15] G. Puskorius, L. Feldkamp, Global calibration of a robot/vision system, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 1987: pp. 190–195. https://doi.org/10.1109/ROBOT.1987.1087836.

[16] A. Nubiola, I.A. Bonev, Absolute calibration of an ABB IRB 1600 robot using a laser tracker, Robotics and Computer-Integrated Manufacturing. 29 (2013) 236–245. https://doi.org/10.1016/j.rcim.2012.06.004.

[17] A. Nubiola, I.A. Bonev, Absolute robot calibration with a single telescoping ballbar, Precision Engineering. 38 (2014) 472–480. https://doi.org/10.1016/j.precisioneng.2014.01.001.

[18] J.G. Victores, S. Martínez, A. Jardón, C. Balaguer, Robot-aided tunnel inspection and maintenance system by vision and proximity sensor integration, Automation in Construction. 20 (2011) 629–636. https://doi.org/10.1016/j.autcon.2010.12.005.

[19] P. Chotiprayanakul, D.K. Liu, G. Dissanayake, Human–robot–environment interaction interface for robotic grit-blasting of complex steel bridges, Automation in Construction. 27 (2012) 11–23. https://doi.org/10.1016/j.autcon.2012.04.014.

[20] E. Menendez, J.G. Victores, R. Montero, S. Martínez, C. Balaguer, Tunnel structural inspection and assessment using an autonomous robotic system, Automation in Construction. 87 (2018) 117–126. https://doi.org/10.1016/j.autcon.2017.12.001.

[21] K.M. Lundeen, V.R. Kamat, C.C. Menassa, W. McGee, Autonomous motion planning and task execution in geometrically adaptive robotized construction work, Automation in Construction. 100 (2019) 24–45. https://doi.org/10.1016/j.autcon.2018.12.020.

[22] M.-S. Gil, M.-S. Kang, S. Lee, H.-D. Lee, K. Shin, J.-Y. Lee, C.-S. Han, Installation of heavy duty glass using an intuitive manipulation device, Automation in Construction. 35 (2013) 579–586. https://doi.org/10.1016/j.autcon.2013.01.008.

[23] S. Yousefizadeh, J. de D. Flores Mendez, T. Bak, Trajectory adaptation for an impedance controlled cooperative robot according to an operator's force, Automation in Construction. 103 (2019) 213–220. https://doi.org/10.1016/j.autcon.2019.01.006.

[24] P. Tavares, C.M. Costa, L. Rocha, P. Malaca, P. Costa, A.P. Moreira, A. Sousa, G. Veiga, Collaborative welding system using BIM for robotic reprogramming and spatial augmented reality, Automation in Construction. 106 (2019) 102825. https://doi.org/10.1016/j.autcon.2019.04.020.

[25] M. Jovanović, M. Raković, B. Tepavčević, B. Borovac, M. Nikolić, Robotic fabrication of freeform foam structures with quadrilateral and puzzle shaped panels, Automation in Construction. 74 (2017) 28–38. https://doi.org/10.1016/j.autcon.2016.11.003.

[26] E. Lublasser, T. Adams, A. Vollpracht, S. Brell-Cokcan, Robotic application of foam concrete onto bare wall elements - Analysis, concept and robotic experiments, Automation in Construction. 89 (2018) 299–306. https://doi.org/10.1016/j.autcon.2018.02.005.

[27] X. Zhang, M. Li, J.H. Lim, Y. Weng, Y.W.D. Tay, H. Pham, Q.-C. Pham, Large-scale 3D printing by a team of mobile robots, Automation in Construction. 95 (2018) 98–106. https://doi.org/10.1016/j.autcon.2018.08.004.

[28] S. Stumm, P. Devadass, S. Brell-Cokcan, Haptic programming in construction, Constr Robot. 2 (2018) 3–13. https://doi.org/10.1007/s41693-018-0015-9.

[29] R. Grytnes, M. Grill, A. Pousette, M. Törner, K.J. Nielsen, Apprentice or student? The structures of construction industry vocational education and training in denmark and sweden and their possible consequences for safety learning, Vocations and Learning. 11 (2018) 65–87. https://doi.org/10.1007/s12186-017-9180-0.

[30] B.D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robotics and Autonomous Systems. 57 (2009) 469–483. https://doi.org/10.1016/j.robot.2008.10.024.

[31] H. Ravichandar, A.S. Polydoros, S. Chernova, A. Billard, Recent advances in robot learning from demonstration, Annual Review of Control, Robotics, and Autonomous Systems. 3 (2020) 297–330. https://doi.org/10.1146/annurev-control-100819-063206.

[32] F. Torabi, G. Warnell, P. Stone, Recent advances in imitation learning from observation, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Macau, China, 2019: pp. 6325–6331. http://arxiv.org/abs/1905.13566 (accessed October 20, 2020).

[33] J. Chung, S.H. Lee, B.-J. Yi, W.K. Kim, Implementation of a foldable 3-DOF master device to a glass window panel fitting task, Automation in Construction. 19 (2010) 855–866. https://doi.org/10.1016/j.autcon.2010.05.004.

[34] A. Beckett, R. Ross, PyroShield - A HVAC fire curtain testing robot, Automation in Construction. 81 (2017) 234–239. https://doi.org/10.1016/j.autcon.2017.06.009.

[35] S.-N. Yu, B.-G. Ryu, S.-J. Lim, C.-J. Kim, M.-K. Kang, C.-S. Han, Feasibility verification of brick-laying robot using manipulation trajectory and the laying pattern optimization, Automation in Construction. 18 (2009) 644–655. https://doi.org/10.1016/j.autcon. 2008.12.008.

[36] M. Nagata, N. Baba, H. Tachikawa, I. Shimizu, T. Aoki, Steel frame welding robot systems and their application at the construction site, Computer-Aided Civil and Infrastructure Engineering. 12 (1997) 15–30. https://doi.org/10.1111/0885-9507.00043.

[37] C.-J. Liang, S.-C. Kang, M.-H. Lee, RAS: a robotic assembly system for steel structure erection and assembly, International Journal of Intelligent Robotics and Applications. 1 (2017) 459–476. https://doi.org/10.1007/s41315-017-0030-x.

[38] B. Chu, K. Jung, M.-T. Lim, D. Hong, Robot-based construction automation: An application to steel beam assembly (Part I), Automation in Construction. 32 (2013) 46–61. https://doi.org/10.1016/j.autcon.2012.12.016.

[39] K. Jung, B. Chu, D. Hong, Robot-based construction automation: An application to steel beam assembly (Part II), Automation in Construction. 32 (2013) 62–79. https://doi.org/10.1016/j.autcon.2012.12.011.

[40] K.S. Saidi, T. Bock, C. Georgoulas, Robotics in construction, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer International Publishing, Cham, 2016: pp. 1493–1520. https://doi.org/10.1007/978-3-319-32552-1_57.

[41] N. Tan, R.E. Mohan, A. Watanabe, Toward a framework for robot-inclusive environments, Automation in Construction. 69 (2016) 68–78. https://doi.org/10.1016/j.autcon.2016.06. 001.

[42] T. Bock, Procedure for the implementation of autonomous mobile robots on the construction site, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Jeju, Korea, 2004: pp. 1–6. https://doi.org/10.22260/ISARC2004/0053.

[43] A. Ajoudani, A.M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, O. Khatib, Progress and prospects of the human–robot collaboration, Autonomous Robots. 42 (2018) 957–975. https://doi.org/10.1007/s10514-017-9677-2.

[44] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, P. Fraisse, Collaborative manufacturing with physical human–robot interaction, Robotics and Computer-Integrated Manufacturing. 40 (2016) 1–13. https://doi.org/10.1016/j.rcim.2015.12.007.

[45] M.S. Erden, A. Billard, End-point impedance measurements at human hand during interactive manual welding with robot, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014: pp. 126–133. https://doi.org/10.1109/ICRA.2014.6906599.

[46] A. Levratti, A. De Vuono, C. Fantuzzi, C. Secchi, TIREBOT: A novel tire workshop assistant robot, in: 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2016: pp. 733–738. https://doi.org/10.1109/AIM.2016.7576855.

[47] E. Matsas, G.-C. Vosniakos, Design of a virtual reality training system for human–robot collaboration in manufacturing tasks, International Journal on Interactive Design and Manufacturing (IJIDeM). 11 (2017) 139–153. https://doi.org/10.1007/s12008-015-0259-2.

[48] P. Tsarouchi, A.-S. Matthaiakis, S. Makris, G. Chryssolouris, On a human-robot collaboration in an assembly cell, International Journal of Computer Integrated Manufacturing. 30 (2017) 580–589. https://doi.org/10.1080/0951192X.2016.1187297.

[49] M. van Osch, D. Bera, K. van Hee, Y. Koks, H. Zeegers, Tele-operated service robots: ROSE, Automation in Construction. 39 (2014) 152–160. https://doi.org/10.1016/j.autcon.2013.06.009.

[50] K. Wakita, J. Huang, P. Di, K. Sekiyama, T. Fukuda, Human-walking-intention-based motion control of an omnidirectional-type cane robot, IEEE/ASME Transactions on Mechatronics. 18 (2013) 285–296. https://doi.org/10.1109/TMECH.2011.2169980.

[51] J. Huang, W. Huo, W. Xu, S. Mohammed, Y. Amirat, Control of upper-limb power-assist exoskeleton using a human-robot interface based on motion intention recognition, IEEE Transactions on Automation Science and Engineering. 12 (2015) 1257–1270. https://doi.org/10.1109/TASE.2015.2466634.

[52] J.M. Beer, A.D. Fisk, W.A. Rogers, Toward a framework for levels of robot autonomy in human-robot interaction, Journal of Human-Robot Interaction. 3 (2014) 74–99. https://doi.org/10.5898/JHRI.3.2.Beer.

[53] B. Khoshnevis, Automated construction by contour crafting—related robotics and information technologies, Automation in Construction. 13 (2004) 5–19. https://doi.org/10.1016/j.autcon.2003.08.012.

[54] P. Carneau, R. Mesnil, N. Roussel, O. Baverel, Additive manufacturing of cantilever - From masonry to concrete 3D printing, Automation in Construction. 116 (2020) 103184. https://doi.org/10.1016/j.autcon.2020.103184.

[55] G. Vantyghem, W. De Corte, E. Shakour, O. Amir, 3D printing of a post-tensioned concrete girder designed by topology optimization, Automation in Construction. 112 (2020) 103084. https://doi.org/10.1016/j.autcon.2020.103084.

[56] N. King, M. Bechthold, A. Kane, P. Michalatos, Robotic tile placement: Tools, techniques and feasibility, Automation in Construction. 39 (2014) 161–166. https://doi.org/10.1016/j.autcon.2013.08.014.

[57] R.M. Molfino, R.P. Razzoli, M. Zoppi, Autonomous drilling robot for landslide monitoring and consolidation, Automation in Construction. 17 (2008) 111–121. https://doi.org/10.1016/j.autcon.2006.12.004.

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

[59] T. Tsuruta, K. Miura, M. Miyaguchi, Mobile robot for marking free access floors at construction sites, Automation in Construction. 107 (2019) 102912. https://doi.org/10.1016/j.autcon.2019.102912.

[60] L. Xu, C. Feng, V.R. Kamat, C.C. Menassa, A scene-adaptive descriptor for visual SLAM-based locating applications in built environments, Automation in Construction. 112 (2020) 103067. https://doi.org/10.1016/j.autcon.2019.103067.

[61] Z. Zhou, Y.M. Goh, Q. Li, Overview and analysis of safety management studies in the construction industry, Safety Science. 72 (2015) 337–350. https://doi.org/10.1016/j.ssci.2014.10.006.

[62] BLS, An analysis of fatal occupational injuries at road construction sites, 2003–2010, Monthly Labor Review. (2013). https://stats.bls.gov/opub/mlr/2013/article/pdf/an-analysis-of-fatal-occupational-injuries-at-road-construction-sites-2003-2010.pdf (accessed February 27, 2019).

[63] CPWR, The construction chart book: the U.S. construction industry and its workers, 4th ed, CPWR - The Center for Construction Research and Training, Silver Spring, MD, 2008.

[64] W.-H. Hung, C.-W. Liu, C.-J. Liang, S.-C. Kang, Strategies to accelerate the computation of erection paths for construction cranes, Automation in Construction. 62 (2016) 1–13. https://doi.org/10.1016/j.autcon.2015.10.008.

[65] J. Teizer, B.S. Allread, U. Mantripragada, Automating the blind spot measurement of construction equipment, Automation in Construction. 19 (2010) 491–501. https://doi.org/10.1016/j.autcon.2009.12.012.

[66] J. Hinze, R. Godfrey, An evaluation of safety performance measures for construction projects, Journal of Construction Research. 04 (2003) 5–15. https://doi.org/10.1142/S160994510300025X.

[67] R.E. Levitt, N.M. Samelson, Construction safety management, John Wiley & Sons, 1993.

[68] S. Talmaki, V.R. Kamat, Real-time hybrid virtuality for prevention of excavation related utility strikes, Journal of Computing in Civil Engineering. 28 (2014) 04014001. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000269.

[69] A.H. Behzadan, V.R. Kamat, Interactive augmented reality visualization for improved damage prevention and maintenance of underground infrastructure, in: Proceedings of the Construction Research Congress (CRC), ASCE, Seattle, WA, USA, 2009: pp. 1214–1222. https://doi.org/10.1061/41020(339)123.

[70] Common Ground Alliance, New common ground alliance dirt report estimates that damage to buried utilities cost society at least $1.5 billion last year, (2017). http://commongroundalliance.com/media-reports/press-releases/new-common-ground-alliance-dirt-report-estimates-damage-buried (accessed August 15, 2018).

[71] K.M. Lundeen, S. Dong, N. Fredricks, M. Akula, J. Seo, V.R. Kamat, Optical marker-based end effector pose estimation for articulated excavators, Automation in Construction. 65 (2016) 51–64. https://doi.org/10.1016/j.autcon.2016.02.003.

[72] S. Li, H. Cai, V.R. Kamat, Uncertainty-aware geospatial system for mapping and visualizing underground utilities, Automation in Construction. 53 (2015) 105–119. https://doi.org/10.1016/j.autcon.2015.03.011.

[73] E. Rezazadeh Azar, B. McCabe, Part based model and spatial–temporal reasoning to recognize hydraulic excavators in construction images and videos, Automation in Construction. 24 (2012) 194–202. https://doi.org/10.1016/j.autcon.2012.03.003.

[74] M.M. Soltani, Z. Zhu, A. Hammad, Framework for location data fusion and pose estimation of excavators using stereo vision, Journal of Computing in Civil Engineering. 32 (2018) 04018045. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000783.

[75] F. Vahdatikhaki, A. Hammad, H. Siddiqui, Optimization-based excavator pose estimation using real-time location systems, Automation in Construction. 56 (2015) 76–92. https://doi.org/10.1016/j.autcon.2015.03.006.

[76] V.R. Kamat, A.H. Behzadan, GPS and 3DOF tracking for georeferenced registration of construction graphics in outdoor augmented reality, in: Proceedings of the EG-ICE International Workshop on Intelligent Computing in Engineering and Architecture, Springer, Berlin, Heidelberg, Ascona, Switzerland, 2006: pp. 368–375. https://doi.org/10.1007/11888598_34.

[77] A.H. Behzadan, V.R. Kamat, Animation of construction activities in outdoor augmented reality, in: Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering (ICCCBE), Montréal, Canada, 2006: pp. 1135–1143.
http://pathfinder.engin.umich.edu/documents/Behzadan&Kamat.ICCCBEXI.2006.pdf (accessed April 2, 2019).

[78] A.H. Behzadan, V.R. Kamat, Integrated information modeling and visual simulation of engineering operations using dynamic augmented reality scene graphs, Journal of Information Technology in Construction (ITcon). 16 (2011) 259–278. http://www.itcon.org/paper/2011/17 (accessed March 28, 2019).

[79] C. Feng, S. Dong, K.M. Lundeen, Y. Xiao, V.R. Kamat, Vision-based articulated machine pose estimation for excavation monitoring and guidance, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Oulu, Finland, 2015. https://doi.org/10.22260/ISARC2015/0029.

[80] E. Rezazadeh Azar, S. Dickinson, B. McCabe, Server-customer interaction tracker: computer vision–based system to estimate dirt-loading cycles, Journal of Construction Engineering and Management. 139 (2013) 785–794. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000652.

[81] A. Montaser, O. Moselhi, RFID+ for tracking earthmoving operations, in: Proceedings of the Construction Research Congress (CRC), ASCE, West Lafayette, IN, USA, 2012: pp. 1011–1020. https://doi.org/10.1061/9780784412329.102.

[82] M. Ibrahim, O. Moselhi, Automated productivity assessment of earthmoving operations, Journal of Information Technology in Construction (ITcon). 19 (2014) 169–184. http://www.itcon.org/paper/2014/9 (accessed February 26, 2019).

[83] J. Gong, C.H. Caldas, An object recognition, tracking, and contextual reasoning-based video interpretation method for rapid productivity analysis of construction operations, Automation in Construction. 20 (2011) 1211–1226. https://doi.org/10.1016/j.autcon.2011.05.005.

[84] C. Yuan, S. Li, H. Cai, Vision-based excavator detection and tracking using hybrid kinematic shapes and key nodes, Journal of Computing in Civil Engineering. 31 (2017) 04016038. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000602.

[85] J. Kim, S. You, S. Lee, V.R. Kamat, L.P. Robert, Evaluation of human robot collaboration in masonry work using immersive virtual environments, in: Proceedings of the International Conference on Construction Applications of Virtual Reality (CONVR), Banff, Canada, 2015: pp. 132–141. http://pathfinder.engin.umich.edu/documents/KimEtAl.CONVR.2015.pdf (accessed February 27, 2019).

[86] T. Salmi, J.M. Ahola, T. Heikkilä, P. Kilpeläinen, T. Malm, Human-robot collaboration and sensor-based robots in industrial applications and construction, in: H. Bier (Ed.), Robotic Building, Springer, Cham, 2018: pp. 25–52. https://doi.org/10.1007/978-3-319-70866-9_2.

[87] T. Salmi, I. Marstio, T. Malm, J. Montonen, Advanced safety solutions for human-robot-cooperation, in: Proceedings of the International Symposium on Robotics (ISR), Munich, Germany, 2016: pp. 610–615. https://cris.vtt.fi/en/publications/advanced-safety-solutions-for-human-robot-cooperation (accessed September 9, 2018).

[88] P.D. Groves, Shadow matching: a new GNSS positioning technique for urban canyons, The Journal of Navigation. 64 (2011) 417–430. https://doi.org/10.1017/S0373463311000087.

[89] C. Feng, Y. Xiao, A. Willette, W. McGee, V.R. Kamat, Towards autonomous robotic in-situ assembly on unstructured construction sites using monocular vision, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Sydney, Australia, 2014: pp. 163–170. https://doi.org/10.22260/ISARC2014/0022.

[90] S. Talmaki, V.R. Kamat, Multi-sensor monitoring for real-time 3D visualization of construction equipment, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Montréal, Canada, 2013: pp. 27–43. https://doi.org/10.22260/ISARC2013/0004.

[91] F.A. Bender, S. Göltz, T. Bräunl, O. Sawodny, Modeling and offset-free model predictive control of a hydraulic mini excavator, IEEE Transactions on Automation Science and Engineering. 14 (2017) 1682–1694. https://doi.org/10.1109/TASE.2017.2700407.

[92] Z. Péntek, T. Hiller, T. Liewald, B. Kuhlmann, A. Czmerk, IMU-based mounting parameter estimation on construction vehicles, in: Proceedings of the DGON Inertial Sensors and Systems (ISS), IEEE, Karlsruhe, Germany, 2017: pp. 1–14. https://doi.org/10.1109/InertialSensors.2017.8171504.

[93] H. Kim, C.R. Ahn, D. Engelhaupt, S. Lee, Application of dynamic time warping to the recognition of mixed equipment activities in cycle time measurement, Automation in Construction. 87 (2018) 225–234. https://doi.org/10.1016/j.autcon.2017.12.014.

[94] C.R. Ahn, S. Lee, F. Peña-Mora, Application of low-cost accelerometers for measuring the operational efficiency of a construction equipment fleet, Journal of Computing in Civil Engineering. 29 (2015) 04014042. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000337.

[95] J. Park, J. Chen, Y.K. Cho, Self-corrective knowledge-based hybrid tracking system using BIM and multimodal sensors, Advanced Engineering Informatics. 32 (2017) 126–138. https://doi.org/10.1016/j.aei.2017.02.001.

[96] Z. Aziz, C.J. Anumba, D. Ruikar, P.M. Carrillo, N.M. Bouchlaghem, Context aware information delivery for on-site construction operations, in: Proceedings of the CIB-W78 International Conference on Information Technology in Construction, CIB Publication, Dresden, Germany, 2005: pp. 321–332. https://itc.scix.net/pdfs/w78-2005-D6-2-Aziz.pdf (accessed March 31, 2021).

[97] C. Rohrig, F. Kiinemund, Mobile robot localization using WLAN signal strengths, in: Proceedings of the IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IEEE, Dortmund, Germany, 2007: pp. 704–709. https://doi.org/10.1109/IDAACS.2007.4488514.

[98] B.-W. Jo, Y.-S. Lee, J.-H. Kim, D.-K. Kim, P.-H. Choi, Proximity warning and excavator control system for prevention of collision accidents, Sustainability. 9 (2017) 1488. https://doi.org/10.3390/su9081488.

[99] H.M. Khoury, V.R. Kamat, Evaluation of position tracking technologies for user localization in indoor construction environments, Automation in Construction. 18 (2009) 444–457. https://doi.org/10.1016/j.autcon.2008.10.011.

[100] J. Teizer, M. Venugopal, A. Walia, Ultrawideband for automated real-time three-dimensional location sensing for workforce, equipment, and material positioning and tracking, Transportation Research Record: Journal of the Transportation Research Board. 2081 (2008) 56–64. https://doi.org/10.3141/2081-06.

[101] C. Zhang, A. Hammad, S. Rodriguez, Crane pose estimation using UWB real-time location system, Journal of Computing in Civil Engineering. 26 (2012) 625–637. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000172.

[102] J. Chai, C. Wu, C. Zhao, H.-L. Chi, X. Wang, B.W.-K. Ling, K.L. Teo, Reference tag supported RFID tracking using robust support vector regression and Kalman filter, Advanced Engineering Informatics. 32 (2017) 1–10. https://doi.org/10.1016/j.aei.2016.11.002.

[103] J. Seo, S. Han, S. Lee, H. Kim, Computer vision techniques for construction safety and health monitoring, Advanced Engineering Informatics. 29 (2015) 239–251. https://doi.org/10.1016/j.aei.2015.02.001.

[104] J. Chen, Y. Fang, Y.K. Cho, C. Kim, Principal axes descriptor for automated construction-equipment classification from point clouds, Journal of Computing in Civil Engineering. 31 (2017) 04016058. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000628.

[105] E. Rezazadeh Azar, B. McCabe, Automated visual recognition of dump trucks in construction videos, Journal of Computing in Civil Engineering. 26 (2012) 769–781. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000179.

[106] M.M. Soltani, Z. Zhu, A. Hammad, Automated annotation for visual recognition of construction resources using synthetic images, Automation in Construction. 62 (2016) 14–23. https://doi.org/10.1016/j.autcon.2015.10.002.

[107] C.-J. Liang, V.R. Kamat, C.C. Menassa, Real-time construction site layout and equipment monitoring, in: Proceedings of the Construction Research Congress, New Orleans, LA, 2018: pp. 64–74. https://doi.org/10.1061/9780784481264.007.

[108] M.M. Soltani, Z. Zhu, A. Hammad, Skeleton estimation of excavator by detecting its parts, Automation in Construction. 82 (2017) 1–15. https://doi.org/10.1016/j.autcon.2017.06.023.

[109] C. Feng, V.R. Kamat, H. Cai, Camera marker networks for articulated machine pose estimation, Automation in Construction. 96 (2018) 148–160. https://doi.org/10.1016/j.autcon.2018.09.004.

[110] E. Rezazadeh Azar, C. Feng, V.R. Kamat, Feasibility of in-plane articulation monitoring of excavator arm using planar marker tracking, Journal of Information Technology in Construction (ITcon). 20 (2015) 213–229. http://itcon.org/paper/2015/15 (accessed February 13, 2017).

[111] W. Yang, X. Zhang, H. Ma, G.-M. Zhang, Infrared LEDs-Based Pose Estimation With Underground Camera Model for Boom-Type Roadheader in Coal Mining, IEEE Access. 7 (2019) 33698–33712. https://doi.org/10.1109/ACCESS.2019.2904097.

[112] C.-J. Liang, Y.-Y. Yang, Y.-S. Lin, S.-C. Kang, P.-C. Lin, Y.-C. Chen, Botbeep - an affordable warning device for wheelchair rearward safety, in: Proceedings of the IEEE International Conference on Orange Technologies (ICOT), IEEE, Tainan, Taiwan, 2013: pp. 159–163. https://doi.org/10.1109/ICOT.2013.6521182.

[113] C. Feng, V.R. Kamat, Plane registration leveraged by global constraints for context-aware AEC applications, Computer-Aided Civil and Infrastructure Engineering. 28 (2013) 325–343. https://doi.org/10.1111/j.1467-8667.2012.00795.x.

[114] L. Xu, V.R. Kamat, C.C. Menassa, Automatic extraction of 1D barcodes from video scans for drone-assisted inventory management in warehousing applications, International Journal of Logistics Research and Applications. (2017) 1–16. https://doi.org/10.1080/13675567.2017.1393505.

[115] B.R.K. Mantha, C.C. Menassa, V.R. Kamat, Robotic data collection and simulation for evaluation of building retrofit performance, Automation in Construction. 92 (2018) 88–102. https://doi.org/10.1016/j.autcon.2018.03.026.

[116] J. Seo, R. Starbuck, S. Han, S. Lee, T.J. Armstrong, Motion data-driven biomechanical analysis during construction tasks on sites, Journal of Computing in Civil Engineering. 29 (2015) B4014005. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000400.

[117] S. Han, S. Lee, F. Peña-Mora, Comparative study of motion features for similarity-based modeling and classification of unsafe actions in construction, Journal of Computing in Civil Engineering. 28 (2014) A4014005. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000339.

[118] S. Han, S. Lee, A vision-based motion capture and recognition framework for behavior-based safety management, Automation in Construction. 35 (2013) 131–141. https://doi.org/10.1016/j.autcon.2013.05.001.

[119] S. Han, S. Lee, F. Peña-Mora, Vision-based detection of unsafe actions of a construction worker: case study of ladder climbing, Journal of Computing in Civil Engineering. 27 (2013) 635–644. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000279.

[120] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2D human pose estimation: new benchmark and state of the art analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Columbus, OH, USA, 2014: pp. 3686–3693. https://doi.org/10.1109/CVPR.2014.471.

[121] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, B. Schiele, DeepCut: joint subset partition and labeling for multi person pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016: pp. 4929–4937. https://doi.org/10.1109/CVPR.2016.533.

[122] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, B. Schiele, DeeperCut: a deeper, stronger, and faster multi-person pose estimation model, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, Cham, Amsterdam, Netherlands, 2016: pp. 34–50. https://doi.org/10.1007/978-3-319-46466-4_3.

[123] D.C. Luvizon, D. Picard, H. Tabia, 2D/3D pose estimation and action recognition using multitask deep learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Salt Lake City, UT, USA, 2018: pp. 5137–5146. https://doi.org/10.1109/CVPR.2018.00539.

[124] A. Bulat, G. Tzimiropoulos, Human pose estimation via convolutional part heatmap regression, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer International Publishing, Amsterdam, Netherlands, 2016: pp. 717–732. https://doi.org/10.1007/978-3-319-46478-7_44.

[125] A. Toshev, C. Szegedy, DeepPose: human pose estimation via deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Columbus, OH, USA, 2014: pp. 1653–1660. https://doi.org/10.1109/CVPR.2014.214.

[126] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, Cham, Amsterdam, Netherlands, 2016: pp. 483–499. https://doi.org/10.1007/978-3-319-46484-8_29.

[127] Y. Chen, C. Shen, X.-S. Wei, L. Liu, J. Yang, Adversarial PoseNet: a structure-aware convolutional network for human pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, Italy, 2017: pp. 1212–1221. https://doi.org/10.1109/ICCV.2017.137.

[128] W. Yang, S. Li, W. Ouyang, H. Li, X. Wang, Learning feature pyramids for human pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, Italy, 2017: pp. 1281–1290. https://doi.org/10.1109/ICCV.2017.144.

[129] X. Chu, W. Yang, W. Ouyang, C. Ma, A.L. Yuille, X. Wang, Multi-context attention for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and

Pattern Recognition (CVPR), IEEE, Honolulu, HI, USA, 2017: pp. 1831–1840. https://doi.org/10.1109/CVPR.2017.601.

[130] G. Pavlakos, X. Zhou, K.G. Derpanis, K. Daniilidis, Coarse-to-fine volumetric prediction for single-image 3D human pose, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, USA, 2017: pp. 7025–7034. https://doi.org/10.1109/CVPR.2017.139.

[131] X. Zhou, Q. Huang, X. Sun, X. Xue, Y. Wei, Towards 3D human pose estimation in the wild: a weakly-supervised approach, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, Italy, 2017: pp. 398–407. https://doi.org/10.1109/ICCV.2017.51.

[132] J. Martinez, R. Hossain, J. Romero, J.J. Little, A simple yet effective baseline for 3d human pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, Italy, 2017: pp. 2659–2668. https://doi.org/10.1109/ICCV.2017.288.

[133] C. Ionescu, D. Papava, V. Olaru, C. Sminchisescu, Human3.6M: large scale datasets and predictive methods for 3D human sensing in natural environments, IEEE Transactions on Pattern Analysis and Machine Intelligence. 36 (2014) 1325–1339. https://doi.org/10.1109/TPAMI.2013.248.

[134] John Deere US, Grade control: construction technology solutions, (2018). https://www.deere.com/en/construction/construction-technology/grade-control/ (accessed September 27, 2018).

[135] Trimble Heavy Industry, Grade control for compact machines, (2021). https://heavyindustry.trimble.com/en/products/grade-control-compact-machines (accessed March 20, 2021).

[136] R. Maalek, F. Sadeghpour, Accuracy assessment of Ultra-Wide Band technology in tracking static resources in indoor construction scenarios, Automation in Construction. 30 (2013) 170–183. https://doi.org/10.1016/j.autcon.2012.10.005.

[137] J. Wang, S.N. Razavi, Low false alarm rate model for unsafe-proximity detection in construction, Journal of Computing in Civil Engineering. 30 (2016) 04015005. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000470.

[138] G.J. Maeda, D.C. Rye, S.P.N. Singh, Iterative autonomous excavation, in: Proceedings of the International Conference on Field and Service Robotics (FSR), Matsushima, Japan, 2012: pp. 369–382. https://doi.org/10.1007/978-3-642-40686-7_25.

[139] M. Memarzadeh, M. Golparvar-Fard, J.C. Niebles, Automated 2D detection of construction equipment and workers from site video streams using histograms of oriented gradients and colors, Automation in Construction. 32 (2013) 24–37. https://doi.org/10.1016/j.autcon.2012.12.002.

[140] D. Kim, K. Yin, M. Liu, S. Lee, V.R. Kamat, Feasibility of a drone-based on-site proximity detection in an outdoor construction site, in: Proceedings of the ASCE International Workshop on Computing in Civil Engineering (IWCCE), ASCE, Seattle, WA, USA, 2017: pp. 392–400. https://doi.org/10.1061/9780784480847.049.

[141] H. Shao, H. Yamamoto, Y. Sakaida, T. Yamaguchi, Y. Yanagisawa, A. Nozue, Automatic excavation planning of hydraulic excavator, in: Proceedings of the International Conference on Intelligent Robotics and Applications, Springer, Berlin, Heidelberg, Wuhan, China, 2008: pp. 1201–1211. https://doi.org/10.1007/978-3-540-88518-4_128.

[142] A. Newell, Z. Huang, J. Deng, Associative embedding: end-to-end learning for joint detection and grouping, in: Advances in Neural Information Processing Systems, NIPS, Long Beach, CA, USA, 2017: pp. 2274–2284. http://papers.nips.cc/paper/6822-associative-embedding-end-to-end-learning-for-joint-detection-and-grouping (accessed February 27, 2019).

[143] A. Assa, F. Janabi-Sharifi, A robust vision-based sensor fusion approach for real-time pose estimation, IEEE Transactions on Cybernetics. 44 (2014) 217–227. https://doi.org/10.1109/TCYB.2013.2252339.

[144] G. Ligorio, A.M. Sabatini, Extended kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: comparative analysis and performance evaluation, Sensors (Basel, Switzerland). 13 (2013) 1919–1941. https://doi.org/10.3390/s130201919.

[145] U. Rafi, J. Gall, B. Leibe, A semantic occlusion model for human pose estimation from a single depth image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, Boston, MA, USA, 2015: pp. 67–74. https://doi.org/10.1109/CVPRW.2015.7301338.

[146] M. Alatise, G. Hancke, M.B. Alatise, G.P. Hancke, Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter, Sensors. 17 (2017) 2164. https://doi.org/10.3390/s17102164.

[147] Y. Dobrev, S. Flores, M. Vossiek, Multi-modal sensor fusion for indoor mobile robot pose estimation, in: Proceedings of the 2016 IEEE/ION Position, Location and Navigation Symposium (PLANS), IEEE, Savannah, GA, USA, 2016: pp. 553–556. https://doi.org/10.1109/PLANS.2016.7479745.

[148] Xsens, MTw Awinda, Xsens 3D Motion Tracking. (2018). https://www.xsens.com/products/mtw-awinda/ (accessed October 7, 2018).

[149] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, J. Collomosse, Total capture: 3D human pose estimation fusing video and inertial sensors, in: Procedings of the British Machine Vision Conference 2017, British Machine Vision Association, London, UK, 2017: p. 14. https://doi.org/10.5244/C.31.14.

[150] F. Huang, A. Zeng, M. Liu, Q. Lai, Q. Xu, Deepfuse: an IMU-aware network for real-time 3D human pose estimation from multi-view image, ArXiv:1912.04071 [Cs]. (2019). http://arxiv.org/abs/1912.04071 (accessed August 11, 2020).

[151] M.L. Anjum, J. Park, W. Hwang, H. Kwon, J. Kim, C. Lee, K. Kim, D. "Dan" Cho, Sensor data fusion using Unscented Kalman Filter for accurate localization of mobile robots, in: Proceedings of the International Conference on Control, Automation and Systems (ICCAS), IEEE, Gyeonggi-do, South Korea, 2010: pp. 947–952. https://doi.org/10.1109/ICCAS.2010.5669779.

[152] D. Jeon, H. Choi, J. Kim, UKF data fusion of odometry and magnetic sensor for a precise indoor localization system of an autonomous vehicle, in: Proceedings of the International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), IEEE, Xi'an, China, 2016: pp. 47–52. https://doi.org/10.1109/URAI.2016.7734018.

[153] A. Chilian, H. Hirschmüller, M. Görner, Multisensor data fusion for robust pose estimation of a six-legged walking robot, in: Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, San Francisco, CA, USA, 2011: pp. 2497–2504. https://doi.org/10.1109/IROS.2011.6094484.

[154] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 2015: pp. 448–456. http://proceedings.mlr.press/v37/ioffe15.html (accessed September 29, 2018).

[155] KUKA Robotics Corporation, KR QUANTEC pro, (2018). https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/kr-quantec-pro (accessed October 7, 2018).

[156] FLIR, FLIR USB 3.1, Gigabit Ethernet and FireWire Machine Vision Cameras, (2018). https://www.ptgrey.com/ (accessed October 7, 2018).

[157] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, ArXiv:1712.04621 [Cs]. (2017). http://arxiv.org/abs/1712.04621 (accessed February 26, 2019).

[158] B. Sapp, B. Taskar, MODEC: multimodal decomposable models for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Portland, OR, USA, 2013: pp. 3674–3681. https://doi.org/10.1109/CVPR.2013.471.

[159] BLS, National census of fatal occupational injuries in 2019, 2020. https://www.bls.gov/news.release/pdf/cfoi.pdf (accessed February 4, 2021).

[160] G.M. Waehrer, X.S. Dong, T. Miller, E. Haile, Y. Men, Costs of occupational injuries in construction in the United States, Accident Analysis & Prevention. 39 (2007) 1258–1266. https://doi.org/10.1016/j.aap.2007.03.012.

[161] K.C. Iyer, N.B. Chaphalkar, G.A. Joshi, Understanding time delay disputes in construction contracts, International Journal of Project Management. 26 (2008) 174–184. https://doi.org/10.1016/j.ijproman.2007.05.002.

[162] M. Pan, T. Linner, W. Pan, H. Cheng, T. Bock, A framework of indicators for assessing construction automation and robotics in the sustainability context, Journal of Cleaner Production. 182 (2018) 82–95. https://doi.org/10.1016/j.jclepro.2018.02.053.

[163] H.-L. Chi, Y.-C. Chen, S.-C. Kang, S.-H. Hsieh, Development of user interface for tele-operated cranes, Advanced Engineering Informatics. 26 (2012) 641–652. https://doi.org/10.1016/j.aei.2012.05.001.

[164] Z. Zhu, H. Hu, Robot learning from demonstration in robotic assembly: a survey, Robotics. 7 (2018) 17. https://doi.org/10.3390/robotics7020017.

[165] L. Schwenkel, M. Guo, Optimizing sequences of probabilistic manipulation skills learned from demonstration, in: Proceedings of the Conference on Robot Learning (CoRL), Osaka, Japan, 2019: p. 10. http://proceedings.mlr.press/v100/schwenkel20a.html (accessed March 31, 2021).

[166] S. Calinon, F. Guenter, A. Billard, On learning the statistical representation of a task and generalizing it to various contexts, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Orlando, FL, USA, 2006: pp. 2978–2983. https://doi.org/10.1109/ROBOT.2006.1642154.

[167] G.J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, J. Peters, Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks, Autonomous Robots. 41 (2017) 593–612. https://doi.org/10.1007/s10514-016-9556-2.

[168] J. Song, Q. Chen, Z. Li, A peg-in-hole robot assembly system based on Gauss mixture model, Robotics and Computer-Integrated Manufacturing. 67 (2021) 101996. https://doi.org/10.1016/j.rcim.2020.101996.

[169] E. Zahedi, F. Khosravian, W. Wang, M. Armand, J. Dargahi, M. Zadeh, Towards skill transfer via learning-based guidance in human-robot interaction: an application to orthopaedic surgical drilling skill, Journal of Intelligent & Robotic Systems. 98 (2020) 667–678. https://doi.org/10.1007/s10846-019-01082-2.

[170] P. Kormushev, S. Calinon, D.G. Caldwell, Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input, Adv Robotics. 25 (2011) 581–603. https://doi.org/10.1163/016918611X558261.

[171] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, S. Savarese, Human-in-the-loop imitation learning using remote teleoperation, ArXiv:2012.06733 [Cs]. (2020). http://arxiv.org/abs/2012.06733 (accessed December 22, 2020).

[172] K. Kukliński, K. Fischer, I. Marhenke, F. Kirstein, M.V. aus der Wieschen, D. Sølvason, N. Krüger, T.R. Savarimuthu, Teleoperation for learning by demonstration: data glove

versus object manipulation for intuitive robot control, in: Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), IEEE, St. Petersburg, Russia, 2014: pp. 346–351. https://doi.org/10.1109/ICUMT.2014.7002126.

[173] P. Abbeel, A. Coates, A.Y. Ng, Autonomous helicopter aerobatics through apprenticeship learning, The International Journal of Robotics Research. 29 (2010) 1608–1639. https://doi.org/10.1177/0278364910371999.

[174] B.D. Argall, B. Browning, M. Veloso, Learning robot motion control with demonstration and advice-operators, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Nice, France, 2008: pp. 399–404. https://doi.org/10.1109/IROS.2008.4651020.

[175] A. Mohseni-Kabir, C. Rich, S. Chernova, C.L. Sidner, D. Miller, Interactive hierarchical task learning from a single demonstration, in: Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI), Association for Computing Machinery, New York, NY, USA, 2015: pp. 205–212. https://doi.org/10.1145/2696454.2696474.

[176] X. Wang, C.-J. Liang, C. Menassa, V. Kamat, Real-time process-level digital twin for collaborative human-robot construction work, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Kitakyushu, Japan (Online), 2020: pp. 1528–1535. https://doi.org/10.22260/ISARC2020/0212.

[177] M.B. Luebbers, C. Brooks, M.J. Kim, D. Szafir, B. Hayes, Augmented reality interface for constrained learning from demonstration, in: Proceedings of the International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI), Daegu, Korea, 2019: pp. 11–14. http://www.bradhayes.info/papers/vamhri19.pdf (accessed March 31, 2021).

[178] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, P. Abbeel, Deep imitation learning for complex manipulation tasks from virtual reality teleoperation, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Brisbane, Australia, 2018: pp. 5628–5635. https://doi.org/10.1109/ICRA.2018.8461249.

[179] J.S. Dyrstad, J.R. Mathiassen, Grasping virtual fish: A step towards robotic deep learning from demonstration in virtual reality, in: Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, Macau, China, 2017: pp. 1181–1187. https://doi.org/10.1109/ROBIO.2017.8324578.

[180] N. Koganti, A. Rahman H. A. G., Y. Iwasawa, K. Nakayama, Y. Matsuo, Virtual reality as a user-friendly interface for learning from demonstrations, in: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA, 2018: pp. 1–4. https://doi.org/10.1145/3170427.3186500.

[181] Y. Liu, A. Gupta, P. Abbeel, S. Levine, Imitation from observation: learning to imitate behaviors from raw video via context translation, in: Proceedings of the IEEE International

Conference on Robotics and Automation (ICRA), IEEE, Brisbane, Australia, 2018: pp. 1118–1125. https://doi.org/10.1109/ICRA.2018.8462901.

[182] T. Fitzgerald, K. McGreggor, B. Akgun, A. Thomaz, A. Goel, Visual case retrieval for interpreting skill demonstrations, in: E. Hüllermeier, M. Minor (Eds.), Proceedings of the International Conference on Case-Based Reasoning (ICCBR), Springer International Publishing, Frankfurt, Germany, 2015: pp. 119–133. https://doi.org/10.1007/978-3-319-24586-7_9.

[183] A. Skoglund, B. Iliev, R. Palm, Programming-by-demonstration of reaching motions—a next-state-planner approach, Robotics and Autonomous Systems. 58 (2010) 607–621. https://doi.org/10.1016/j.robot.2009.12.003.

[184] M. Edmonds, F. Gao, X. Xie, H. Liu, S. Qi, Y. Zhu, B. Rothrock, S.-C. Zhu, Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Vancouver, BC, Canada, 2017: pp. 3530–3537. https://doi.org/10.1109/IROS.2017.8206196.

[185] R. Song, F. Li, W. Quan, X. Yang, J. Zhao, Skill learning for robotic assembly based on visual perspectives and force sensing, Robotics and Autonomous Systems. 135 (2021) 103651. https://doi.org/10.1016/j.robot.2020.103651.

[186] S. Calinon, E.L. Sauser, A.G. Billard, D.G. Caldwell, Evaluation of a probabilistic approach to learn and reproduce gestures by imitation, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Anchorage, AK, USA, 2010: pp. 2671–2676. https://doi.org/10.1109/ROBOT.2010.5509988.

[187] N. Jaquier, D. Ginsbourger, S. Calinon, Learning from demonstration with model-based Gaussian process, in: Proceedings of the Conference on Robot Learning (CoRL), Osaka, Japan, 2019: p. 11. http://proceedings.mlr.press/v100/jaquier20b.html (accessed March 31, 2021).

[188] S.R. Ahmadzadeh, S. Chernova, Trajectory-based skill learning using generalized cylinders, Frontiers in Robotics and AI. 5 (2018). https://doi.org/10.3389/frobt.2018.00132.

[189] A.J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: learning attractor models for motor behaviors, Neural Computation. 25 (2012) 328–373. https://doi.org/10.1162/NECO_a_00393.

[190] P. Pastor, H. Hoffmann, T. Asfour, S. Schaal, Learning and generalization of motor skills by learning from demonstration, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Kobe, Japan, 2009: pp. 763–768. https://doi.org/10.1109/ROBOT.2009.5152385.

[191] S.M. Khansari-Zadeh, A. Billard, Learning stable nonlinear dynamical systems with gaussian mixture models, IEEE Transactions on Robotics. 27 (2011) 943–957. https://doi.org/10.1109/TRO.2011.2159412.

[192] M. Bain, C. Sammut, A framework for behavioural cloning, Machine Intelligence. 15 (2001) 1–37. http://www.cse.unsw.edu.au/~claude/papers/MI15.pdf (accessed March 31, 2021).

[193] H. Ravichandar, S.R. Ahmadzadeh, M.A. Rana, S. Chernova, Skill acquisition via automated multi-coordinate cost balancing, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Montreal, Canada, 2019: pp. 7776–7782. https://doi.org/10.1109/ICRA.2019.8793762.

[194] A.Y. Ng, S.J. Russell, Algorithms for inverse reinforcement learning, in: Proceedings of the International Conference on Machine Learning (ICML), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000: pp. 663–670. http://dl.acm.org/citation.cfm ?id=645529.657801 (accessed May 17, 2020).

[195] J. Ho, S. Ermon, Generative adversarial imitation learning, in: Proceedings of the International Conference on Neural Information Processing Systems (NIPS), Curran Associates Inc., Red Hook, NY, USA, 2016: pp. 4572–4580.

[196] A. Kinose, T. Taniguchi, Integration of imitation learning using GAIL and reinforcement learning using task-achievement rewards via probabilistic graphical model, Advanced Robotics. 34 (2020) 1055–1067. https://doi.org/10.1080/01691864.2020.1778521.

[197] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, in: Proceedings of the International Conference on Neural Information Processing Systems (NIPS), MIT Press, Cambridge, MA, USA, 2014: pp. 2672–2680. https://arxiv.org/abs/1406.2661 (accessed March 4, 2021).

[198] B.S. Pavse, F. Torabi, J. Hanna, G. Warnell, P. Stone, RIDM: reinforced inverse dynamics modeling for learning from a single observed demonstration, IEEE Robotics and Automation Letters. 5 (2020) 6262–6269. https://doi.org/10.1109/LRA.2020.3010750.

[199] A.D. Edwards, H. Sahni, Y. Schroecker, C.L. Isbell, Imitating latent policies from observation, in: Proceedings of the International Conference on Machine Learning (ICML), PMLR, Long Beach, CA, USA, 2019: pp. 1755–1763. http://proceedings.mlr.press/v97/edwards19a.html (accessed March 31, 2021).

[200] J. Merel, Y. Tassa, D. TB, S. Srinivasan, J. Lemmon, Z. Wang, G. Wayne, N. Heess, Learning human behaviors from motion capture by adversarial imitation, ArXiv:1707.02201 [Cs]. (2017). http://arxiv.org/abs/1707.02201 (accessed March 5, 2021).

[201] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, G. Brain, Time-contrastive networks: self-supervised learning from video, in: Proceedings of the IEEE

International Conference on Robotics and Automation (ICRA), IEEE, Brisbane, Australia, 2018: pp. 1134–1141. https://doi.org/10.1109/ICRA.2018.8462891.

[202] C. Finn, T. Yu, T. Zhang, P. Abbeel, S. Levine, One-shot visual imitation learning via meta-learning, in: Proceedings of the Conference on Robot Learning (CoRL), Mountain View, CA, USA, 2017: pp. 357–368. http://proceedings.mlr.press/v78/finn17a.html (accessed November 11, 2018).

[203] L. Xu, C. Feng, V.R. Kamat, C.C. Menassa, An occupancy grid mapping enhanced visual slam for real-time locating applications in indoor gps-denied environments, Automation in Construction. 104 (2019) 230–245. https://doi.org/10.1016/j.autcon.2019.04.011.

[204] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A.Y. Ng, ROS: an open-source Robot Operating System, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Kobe, Japan, 2009: p. 5. https://www.semanticscholar.org/paper/ROS%3A-an-open-source-Robot-Operating-System-Quigley/d45eaee8b2e047306329e5dbfc954e6dd318ca1e#citing-papers (accessed January 5, 2021).

[205] N. Koenig, A. Howard, Design and use paradigms for Gazebo, an open-source multi-robot simulator, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Sendai, Japan, 2004: pp. 2149–2154. https://doi.org/10.1109/IROS.2004.1389727.

[206] F. Sanfilippo, Ø. Stavdahl, P. Liljebäck, SnakeSIM: A ROS-based rapid-prototyping framework for perception-driven obstacle-aided locomotion of snake robots, in: Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, Macau, China, 2017: pp. 1226–1231. https://doi.org/10.1109/ROBIO.2017.8324585.

[207] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in: Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 2015: pp. 1889–1897. http://proceedings.mlr.press/v37/schulman15.html (accessed February 3, 2019).

[208] Y. Duan, M. Andrychowicz, B.C. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, W. Zaremba, One-shot imitation learning, in: Advances in Neural Information Processing Systems, Long Beach, CA, USA, 2017: pp. 1087–1098. http://arxiv.org/abs/1703.07326 (accessed November 2, 2018).

[209] R.S. Sutton, A.G. Barto, Reinforcement learning: an introduction, 2nd ed., MIT Press, Cambridge, MA, USA, 2018.

[210] L.P. Kaelbling, M.L. Littman, A.R. Cassandra, Planning and acting in partially observable stochastic domains, Artificial Intelligence. 101 (1998) 99–134. https://doi.org/10.1016/S0004-3702(98)00023-X.

[211] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, ArXiv:1707.06347 [Cs]. (2017). http://arxiv.org/abs/1707.06347 (accessed March 28, 2019).

[212] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, in: Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2016: pp. 1–14. http://arxiv.org/abs/1509.02971 (accessed February 3, 2019).

[213] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the International Conference on Machine Learning (ICML), ACM, Helsinki, Finland, 2008: pp. 1096–1103. https://doi.org/10.1145/1390156.1390294.

[214] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proceedings of the International Conference on Machine Learning (ICML), Atlanta, GA, USA, 2013: pp. 1–6. http://robotics.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf (accessed March 28, 2019).

[215] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, D. Meger, Deep reinforcement learning that matters, in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI, New Orleans, LA, USA, 2018: pp. 3207–3214. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16669 (accessed March 29, 2019).

[216] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 2015: pp. 1–15. http://arxiv.org/abs/1412.6980 (accessed April 1, 2019).

[217] D.R. Hunter, K. Lange, A tutorial on MM algorithms, The American Statistician. 58 (2004) 30–37. https://doi.org/10.1198/0003130042836.

[218] G. Dulac-Arnold, D. Mankowitz, T. Hester, Challenges of real-world reinforcement learning, ArXiv:1904.12901 [Cs, Stat]. (2019). http://arxiv.org/abs/1904.12901 (accessed May 10, 2020).

[219] B. Martínez-Salvador, M. Pérez-Francisco, A.P. Del Pobil, Collision detection between robot arms and people, Journal of Intelligent and Robotic Systems. 38 (2003) 105–119. https://doi.org/10.1023/A:1026252228930.

[220] T. Özaslan, G. Loianno, J. Keller, C.J. Taylor, V. Kumar, Spatio-temporally smooth local mapping and state estimation inside generalized cylinders with micro aerial vehicles, IEEE Robotics and Automation Letters. 3 (2018) 4209–4216. https://doi.org/10.1109/LRA.2018.2861888.

[221] F.L. Markley, Y. Cheng, J.L. Crassidis, Y. Oshman, Averaging quaternions, Journal of Guidance, Control, and Dynamics. 30 (2007) 1193–1197. https://doi.org/10.2514/1.28949.

[222] H.R. Kam, S.-H. Lee, T. Park, C.-H. Kim, RViz: a toolkit for real domain data visualization, Telecommunications Systems. 60 (2015) 337–345. https://doi.org/10.1007/s11235-015-0034-5.

[223] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, S. Levine, Meta-world: a benchmark and evaluation for multi-task and meta reinforcement learning, in: Proceedings of the Conference on Robot Learning (CoRL), Osaka, Japan, 2019: pp. 1–17. https://arxiv.org/abs/1910.10897 (accessed May 17, 2020).

[224] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, OpenAI Gym, ArXiv:1606.01540 [Cs]. (2016). http://arxiv.org/abs/1606.01540 (accessed February 10, 2019).

[225] N.G. Lopez, Y.L.E. Nuin, E.B. Moral, L.U.S. Juan, A.S. Rueda, V.M. Vilches, R. Kojcev, gym-gazebo2, a toolkit for reinforcement learning using ROS 2 and Gazebo, ArXiv:1903.06278 [Cs]. (2019). http://arxiv.org/abs/1903.06278 (accessed November 3, 2019).

[226] Y.-C. Lin, D. Berenson, Using previous experience for humanoid navigation planning, in: Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids), IEEE, Cancun, Mexico, 2016: pp. 794–801. https://doi.org/10.1109/HUMANOIDS.2016.7803364.

[227] Armstrong Ceilings, Drop ceiling installation, Armstrong Ceilings Residential. (2020). https://www.armstrongceilings.com/residential/en-us/project-ideas-and-installation/drop-ceiling-installation.html (accessed May 15, 2020).

[228] Ceiling Tile UK, Suspended ceiling tiles, Suspended Ceiling Tiles. (2018). https://www.ceilingtilesuk.co.uk/sizes-of-ceiling-tile/ (accessed May 5, 2019).

[229] G. Billings, M. Johnson-Roberson, Silhonet: an rgb method for 6d object pose estimation, ArXiv:1809.06893 [Cs]. (2018). http://arxiv.org/abs/1809.06893 (accessed September 10, 2019).

[230] E. Olson, AprilTag: a robust and flexible visual fiducial system, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Shanghai, China, 2011: pp. 3400–3407. https://doi.org/10.1109/ICRA.2011.5979561.

[231] Y. Xiao, Y. Taguchi, V.R. Kamat, Coupling point cloud completion and surface connectivity relation inference for 3D modeling of indoor building environments, Journal of Computing in Civil Engineering. 32 (2018) 04018033. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000776.

[232] U.S. BLS, Census of Fatal Occupational Injuries (CFOI) - current and revised data, (2019). https://www.bls.gov/iif/oshcfoi1.htm (accessed November 4, 2020).

[233] OSHA, Commonly used statistics | occupational safety and health administration, (2018). https://www.osha.gov/data/commonstats (accessed November 4, 2020).

[234] D. Kim, S. Lee, V.R. Kamat, Proximity prediction of mobile objects to prevent contact-driven accidents in co-robotic construction, Journal of Computing in Civil Engineering. 34 (2020) 04020022. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000899.

[235] J.W. Hinze, J. Teizer, Visibility-related fatalities related to construction equipment, Safety Science. 49 (2011) 709–718. https://doi.org/10.1016/j.ssci.2011.01.007.

[236] OSHA, Materials handling and storage, 2002. https://www.osha.gov/Publications/osha2236.pdf (accessed November 11, 2020).

[237] C. Wu, X. Wang, M. Chen, M.J. Kim, Differential received signal strength based RFID positioning for construction equipment tracking, Advanced Engineering Informatics. 42 (2019) 100960. https://doi.org/10.1016/j.aei.2019.100960.

[238] T. Cheng, M. Venugopal, J. Teizer, P.A. Vela, Performance evaluation of ultra wideband technology for construction resource location tracking in harsh environments, Automation in Construction. 20 (2011) 1173–1184. https://doi.org/10.1016/j.autcon.2011.05.001.

[239] A.R. Andoh, X. Su, H. Cai, A framework of RFID and GPS for tracking construction site dynamics, in: Proceedings of the Construction Research Congress (CRC), ASCE, West Lafayette, IN, USA, 2012: pp. 818–827. https://doi.org/10.1061/9780784412329.083.

[240] K. Oh, S. Park, J. Seo, J.-G. Kim, J. Park, G. Lee, K. Yi, Development of a predictive safety control algorithm using laser scanners for excavators on construction sites, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. (2018). https://doi.org/10.1177/0954407018764046.

[241] D. Kim, J. Kim, K. Lee, C. Park, J. Song, D. Kang, Excavator tele-operation system using a human arm, Automation in Construction. 18 (2009) 173–182. https://doi.org/10.1016/j.autcon.2008.07.002.

[242] W. Wang, H. Chi, S. Zhao, Z. Du, A control method for hydraulic manipulators in automatic emulsion filling, Automation in Construction. 91 (2018) 92–99. https://doi.org/10.1016/j.autcon.2018.03.001.

[243] J. Schmidtler, V. Knott, C. Hölzel, K. Bengler, Human centered assistance applications for the working environment of the future, Occupational Ergonomics. 12 (2015) 83–95. https://doi.org/10.3233/OER-150226.

[244] L. Wang, R. Gao, J. Váncza, J. Krüger, X.V. Wang, S. Makris, G. Chryssolouris, Symbiotic human-robot collaborative assembly, CIRP Annals. 68 (2019) 701–726. https://doi.org/10.1016/j.cirp.2019.05.002.

[245] S. Musić, S. Hirche, Control sharing in human-robot team interaction, Annual Reviews in Control. 44 (2017) 342–354. https://doi.org/10.1016/j.arcontrol.2017.09.017.

[246] R.G. Farrell, J. Lenchner, J.O. Kephjart, A.M. Webb, Mi.J. Muller, T.D. Erikson, D.O. Melville, R.K.E. Bellamy, D.M. Gruen, J.H. Connell, D. Soroker, A. Aaron, S.M. Trewin,

M. Ashoori, J.B. Ellis, B.P. Gaucher, D. Gil, Symbiotic cognitive computing, AI Magazine. 37 (2016) 81–93. https://doi.org/10.1609/aimag.v37i3.2628.

[247] N. Nikolakis, V. Maratos, S. Makris, A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace, Robotics and Computer-Integrated Manufacturing. 56 (2019) 233–243. https://doi.org/10.1016/j.rcim.2018.10.003.

[248] A. Hentout, M. Aouache, A. Maoudj, I. Akli, Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017, Advanced Robotics. 33 (2019) 764–799. https://doi.org/10.1080/01691864.2019.1636714.

[249] A. Freedy, E. DeVisser, G. Weltman, N. Coeyman, Measurement of trust in human-robot collaboration, in: Proceedings of the International Symposium on Collaborative Technologies and Systems, IEEE, Orlando, FL, USA, 2007: pp. 106–114. https://doi.org/10.1109/CTS.2007.4621745.

[250] C. Morato, K.N. Kaipa, B. Zhao, S.K. Gupta, Toward safe human robot collaboration by using multiple kinects based real-time human tracking, Journal of Computing and Information Science in Engineering. 14 (2014) 011006. https://doi.org/10.1115/1.4025810.

[251] A. Mohammed, B. Schmidt, L. Wang, Active collision avoidance for human–robot collaboration driven by vision sensors, International Journal of Computer Integrated Manufacturing. 30 (2017) 970–980. https://doi.org/10.1080/0951192X.2016.1268269.

[252] S. Aheleroff, J. Polzer, H. Huang, Z. Zhu, D. Tomzik, Y. Lu, Y. Lin, X. Xu, Smart manufacturing based on digital twin technologies, in: C. Machado, J.P. Davim (Eds.), Industry 4.0: Challenges, Trends, and Solutions in Management and Engineering, CRC Press, 2020: p. 77. https://doi.org/10.1201/9781351132992-3.

[253] A.M. Madni, C.C. Madni, S.D. Lucero, Leveraging digital twin technology in model-based systems engineering, Systems. 7 (2019) 7. https://doi.org/10.3390/systems7010007.

[254] M. Colledani, W. Terkaj, T. Tolio, Product-process-system information formalization, in: Design of Flexible Production Systems: Methodologies and Tools, Springer, Berlin, Heidelberg, 2009: pp. 63–86. https://doi.org/10.1007/978-3-540-85414-2_4.

[255] K.G. Shin, P. Ramanathan, Real-time computing: a new discipline of computer science and engineering, Proceedings of the IEEE. 82 (1994) 6–24. https://doi.org/10.1109/5.259423.

[256] H. Kopetz, Real-time systems: design principles for distributed embedded applications, Springer Science & Business Media, New York, NY, USA, 2011.

[257] J. Mertens, M. Challenger, K. Vanherpen, J. Denil, Towards real-time cyber-physical systems instrumentation for creating digital twins, in: Proceedings of the Spring Simulation Conference (SpringSim), IEEE, Fairfax, VA, USA, 2020: pp. 1–12. https://doi.org/10.22360/SpringSim.2020.CPS.001.

[258] Y. Lu, X. Xu, Resource virtualization: A core technology for developing cyber-physical production systems, Journal of Manufacturing Systems. 47 (2018) 128–140. https://doi.org/10.1016/j.jmsy.2018.05.003.

[259] T. Delbrügger, L.T. Lenz, D. Losch, J. Roßmann, A navigation framework for digital twins of factories based on building information modeling, in: Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Limassol, Cyprus, 2017: pp. 1–4. https://doi.org/10.1109/ETFA.2017.8247712.

[260] M.Q. Marshall, C. Redovian, An application of a digital twin to robotic system design for an unstructured environment, in: Proceedings of the ASME International Mechanical Engineering Congress and Exposition, ASME, Salt Lake City, UT, USA, 2019: p. V02BT02A010. https://doi.org/10.1115/IMECE2019-11337.

[261] Q. Lu, X. Xie, A.K. Parlikad, J.M. Schooling, E. Konstantinou, Moving from building information models to digital twins for operation and maintenance, Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction. (2020) 1–11. https://doi.org/10.1680/jsmic.19.00011.

[262] R. Al-Sehrawy, B. Kumar, Digital twins in architecture, engineering, construction and operations. a brief review and analysis, in: E. Toledo Santos, S. Scheer (Eds.), Proceedings of the International Conference on Computing in Civil and Building Engineering (ICCCBE), Springer International Publishing, São Paulo, Brazil (Online), 2020: pp. 924–939. https://doi.org/10.1007/978-3-030-51295-8_64.

[263] V.R. Kamat, J.C. Martinez, Large-scale dynamic terrain in three-dimensional construction process visualizations, Journal of Computing in Civil Engineering. 19 (2005) 160–171. https://doi.org/10.1061/(ASCE)0887-3801(2005)19:2(160).

[264] R. Eadie, M. Browne, H. Odeyinka, C. McKeown, S. McNiff, BIM implementation throughout the UK construction project lifecycle: An analysis, Automation in Construction. 36 (2013) 145–151. https://doi.org/10.1016/j.autcon.2013.09.001.

[265] A.Z. Sampaio, E. Berdeja, Collaborative BIM environment as a support to conflict analysis in building design, in: Proceedings of the Experiment@International Conference (Exp.at'17), IEEE, Faro, Portugal, 2017: pp. 77–82. https://doi.org/10.1109/EXPAT.2017.7984348.

[266] T.-H. Wu, F. Wu, C.-J. Liang, Y.-F. Li, C.-M. Tseng, S.-C. Kang, A virtual reality tool for training in global engineering collaboration, Universal Access in the Information Society. 18 (2017) 243–255. https://doi.org/10.1007/s10209-017-0594-0.

[267] S. Ochmann, R. Vock, R. Wessel, R. Klein, Automatic reconstruction of parametric building models from indoor point clouds, Computers & Graphics. 54 (2016) 94–103. https://doi.org/10.1016/j.cag.2015.07.008.

[268] H. Hamledari, B. McCabe, S. Davari, A. Shahi, Automated schedule and progress updating of IFC-based 4D BIMs, Journal of Computing in Civil Engineering. 31 (2017) 04017012. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000660.

[269] F. Bosché, M. Ahmed, Y. Turkan, C.T. Haas, R. Haas, The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components, Automation in Construction. 49 (2015) 201–213. https://doi.org/10.1016/j.autcon.2014.05.014.

[270] A. Dimitrov, M. Golparvar-Fard, Segmentation of building point cloud models including detailed architectural/structural features and MEP systems, Automation in Construction. 51 (2015) 32–45. https://doi.org/10.1016/j.autcon.2014.12.015.

[271] H. Macher, T. Landes, P. Grussenmeyer, From point clouds to building information models: 3D semi-automatic reconstruction of indoors of existing buildings, Applied Sciences. 7 (2017) 1030. https://doi.org/10.3390/app7101030.

[272] V. Stojanovic, M. Trapp, R. Richter, B. Hagedorn, J. Döllner, Towards the generation of digital twins for facility management based on 3D point clouds, in: Proceedings of the ARCOM 34th Annual Conference, Belfast, UK, 2018: pp. 270–279. https://www.researchgate.net/publication/325737190_Towards_The_Generation_of_Digital_Twins_for_Facility_Management_Based_on_3D_Point_Clouds (accessed March 31, 2021).

[273] C. Wang, Y.K. Cho, Smart scanning and near real-time 3D surface modeling of dynamic construction equipment from a point cloud, Automation in Construction. 49 (2015) 239–249. https://doi.org/10.1016/j.autcon.2014.06.003.

[274] J.J. Lin, J.Y. Lee, M. Golparvar-Fard, Exploring the potential of image-based 3D geometry and appearance reasoning for automated construction progress monitoring, in: Proceedings of the ASCE International Conference on Computing in Civil Engineering (I3CE), ASCE, Atlanta, GA, USA, 2019: pp. 162–170. https://doi.org/10.1061/9780784482438.021.

[275] H. Hamledari, E. Rezazadeh Azar, B. McCabe, IFC-based development of as-built and as-is BIMs using construction and facility inspection data: site-to-BIM data transfer automation, Journal of Computing in Civil Engineering. 32 (2018) 04017075. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000727.

[276] F. Xue, W. Lu, K. Chen, A. Zetkulic, From semantic segmentation to semantic registration: derivative-free optimization–based approach for automatic generation of semantically rich as-built building information models from 3D point clouds, Journal of Computing in Civil Engineering. 33 (2019) 04019024. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000839.

[277] R. Söderberg, K. Wärmefjord, J.S. Carlson, L. Lindkvist, Toward a Digital Twin for real-time geometry assurance in individualized production, CIRP Annals. 66 (2017) 137–140. https://doi.org/10.1016/j.cirp.2017.04.038.

[278] R.S. Tabar, K. Wärmefjord, R. Söderberg, L. Lindkvist, Efficient spot welding sequence optimization in a geometry assurance digital twin, Journal of Mechanical Design. 142 (2020) 1–8. https://doi.org/10.1115/1.4046436.

[279] C. Kan, C.J. Anumba, Digital twins as the next phase of cyber-physical systems in construction, in: Proceedings of the ASCE International Conference on Computing in Civil Engineering (I3CE), ASCE, Atlanta, Georgia, 2019: pp. 256–264. https://doi.org/10.1061/9780784482438.033.

[280] S. Tandur, Towards a new BIM "dimension" - translating BIM data into actual construction using robotics, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Oulu, Finland, 2015: pp. 1–7. https://doi.org/10.22260/ISARC2015/0051.

[281] C.-H. Yang, T.-H. Wu, B. Xiao, S.-C. Kang, Design of a robotic software package for modular home builder, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Banff, AB, Canada, 2019: pp. 1217–1222. https://doi.org/10.22260/ISARC2019/0163.

[282] F. Shahmiri, J. Ficca, A model for real-time control of industrial robots, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Auburn, AL, USA, 2016: pp. 1065–1072. https://doi.org/10.22260/ISARC2016/0128.

[283] T. Bruckmann, H. Mattern, A. Spenglerc, C. Reichert, A. Malkwitz, M. König, Automated construction of masonry buildings using cable-driven parallel robots, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Auburn, AL, USA, 2016: pp. 332–340. https://doi.org/10.22260/ISARC2016/0041.

[284] V. Usmanov, M. Bruzl, P. Svoboda, R. Šulc, Modelling of industrial robotic brick system, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Taipei, Taiwan, 2017: pp. 1013–1020. https://doi.org/10.22260/ISARC2017/0140.

[285] S. Sharif, T.R. Gentry, L.M. Sweet, Human-robot collaboration for creative and integrated design and fabrication processes, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Auburn, AL, USA, 2016: pp. 596–604. https://doi.org/10.22260/ISARC2016/0072.

[286] C. Zhuang, J. Liu, H. Xiong, Digital twin-based smart production management and control framework for the complex product assembly shop-floor, The International Journal of Advanced Manufacturing Technology. 96 (2018) 1149–1163. https://doi.org/10.1007/s00170-018-1617-6.

[287] A. Bilberg, A.A. Malik, Digital twin driven human–robot collaborative assembly, CIRP Annals. 68 (2019) 499–502. https://doi.org/10.1016/j.cirp.2019.04.011.

[288] A.A. Malik, A. Brem, Digital twins for collaborative robots: A case study in human-robot interaction, Robotics and Computer-Integrated Manufacturing. 68 (2021) 102092. https://doi.org/10.1016/j.rcim.2020.102092.

[289] R. Naboni, A. Kunic, A computational framework for the design and robotic manufacturing of complex wood structures, in: Proceedings of the Education and Research in Computer Aided Architectural Design in Europe and Iberoamerican Society of Digital Graphics, Joint Conference, Porto, Portugal, 2019: pp. 189–196. https://doi.org/10.5151/proceedings-ecaadesigradi2019_488.

[290] Y. Cai, Y. Wang, M. Burnett, Using augmented reality to build digital twin for reconfigurable additive manufacturing system, Journal of Manufacturing Systems. In Press (2020). https://doi.org/10.1016/j.jmsy.2020.04.005.

[291] T. Linner, A. Shrikathiresan, M. Vetrenko, B. Ellmann, T. Bock, Modeling and operating robotic environments using Gazebo/ROS, in: Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Seoul, Korea, 2011: pp. 957–962. https://doi.org/10.22260/ISARC2011/0177.

[292] L. Vasey, B. Felbrich, M. Prado, B. Tahanzadeh, A. Menges, Physically distributed multi-robot coordination and collaboration in construction, Construction Robotics. 4 (2020) 3–18. https://doi.org/10.1007/s41693-020-00031-y.

[293] R.A. Light, Mosquitto: server and client implementation of the MQTT protocol, Journal of Open Source Software. 2 (2017) 265. https://doi.org/10.21105/joss.00265.

[294] Beckhoff, TwinCAT ADS, Beckhoff, 2021. https://github.com/Beckhoff/ADS (accessed February 6, 2021).

[295] D.T. Coleman, I.A. Sucan, S. Chitta, N. Correll, Reducing the barrier to entry of complex robotic software: a MoveIt! case study, Journal of Software Engineering for Robotics. 5 (2014) 3–16. https://doi.org/10.6092/JOSER_2014_05_01_p3.

[296] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Robot programming by demonstration, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer, Berlin, Heidelberg, 2008: pp. 1371–1394. https://doi.org/10.1007/978-3-540-30301-5_60.

[297] A. Hussein, M.M. Gaber, E. Elyan, C. Jayne, Imitation learning: a survey of learning methods, ACM Computing Surveys. 50 (2017) 21:1-21:35. https://doi.org/10.1145/3054912.

[298] J.Y. Chai, M. Çakmak, C.L. Sidner, Teaching robots new tasks through natural interaction, in: K.A. Gluck, J.E. Laird (Eds.), Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions, MIT Press, 2018. https://doi.org/10.7551/mitpress/11956.003.0013.

[299] G. Michalos, S. Makris, J. Spiliotopoulos, I. Misios, P. Tsarouchi, G. Chryssolouris, Robo-partner: seamless human-robot cooperation for intelligent, flexible and safe operations in

the assembly factories of the future, Procedia CIRP. 23 (2014) 71–76. https://doi.org/10.1016/j.procir.2014.10.079.

[300] J.-G. Ge, Programming by demonstration by optical tracking system for dual arm robot, in: Proceedings of the IEEE ISR, IEEE, Seoul, South Korea, 2013: pp. 1–7. https://doi.org/10.1109/ISR.2013.6695708.