Trajectory-Based Skill Learning for Overhead Construction Robots Using Generalized Cylinders with Orientation

Ci-Jyun Liang, Ph.D., S.M.ASCE[1]; Vineet R. Kamat, Ph.D., M.ASCE[2]; Carol C. Menassa, Ph.D., A.M.ASCE[3]; and Wes McGee[4]

Abstract: Overhead work involving the construction and maintenance of civil infrastructure (e.g., tunnels, overpasses, and buildings) is strenuous and fatigue-inducing for human workers and is particularly well-suited for co-robotization. Such work is typically quasi-repetitive, and on-site robots must adapt to unexpected workface conditions. Methods such as learning from demonstration can leverage human experts' demonstration to let robots directly learn new skills to perform tasks. This paper proposes a generalized cylinders with orientation approach to teach robots how to perform quasi-repetitive overhead construction tasks from human demonstration. The demonstration trajectories are first used to construct a generalized cylinder and generate the robot trajectory. To ensure that the construction component (e.g., tunnel lining segment, building ceiling tile) being installed can satisfy the geometric constraints of the workspace, orientation constraints need to be determined, and the robot must follow such constraints. A trajectory adaptation and human-in-the-loop refinement approach are developed to refine the robot trajectory. The proposed method was evaluated in a robot simulator with variable workspace. The results showed that the proposed approach achieves an improved success rate (82.0%) compared to that demonstrated in previous work (71.3%) and enables overhead construction robots to readily adapt to new worksite conditions. **DOI: 10.1061/(ASCE)CP.1943-5487.0001004.** *© 2021 American Society of Civil Engineers.*

Author keywords: Construction robots; Learning from demonstration; Generalized cylinders; Critical orientation; Trajectory adaptation; Human-robot collaboration; Buildings; Tunnels.

Introduction

Overhead construction work is particularly stressful and fatigue-inducing for human workers. Such work typically requires significant arm-raising movements that can cause chronic musculoskeletal disease in workers (e.g., difficulty reaching arms or lifting objects) (Dong et al. 2011). Overhead construction is common in several types of projects. For example, ceiling tiles are commonly installed overhead in residential and commercial buildings. Similarly, in the construction and maintenance of infrastructures such as bridges, overpasses, and tunnels (e.g., installing lining segments), human workers often perform overhead assembly and repair tasks in challenging work environments.

The adoption of construction robots on job sites has demonstrated improvement in the safety, the productivity of projects, and the quality of work (Pan et al. 2018). Similar to applications in disparate fields such as manufacturing and surgery, where a robot can assist with repetitive or precise work in narrow workspaces, robots on construction sites can assist with physically demanding and repetitive construction tasks. Co-robotization can be particularly beneficial in strenuous work such as overhead construction. However, unlike manufacturing or surgery robots, where the robots are typically placed at stationary locations to perform work through preprogramming or tele-operation, 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 (Lundeen et al. 2017). It is therefore impractical to preprogram a robotic construction work plan or define it as an optimization problem. In addition, tele-operated robots to complete construction tasks requires significant training of operators that must include both construction experts as well as robot technicians (Chi et al. 2012).

Imitation Learning for Construction Robots

Imitation learning or learning from demonstration (LfD) methods eliminate the requirement of preprogramming 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 (Argall et al. 2009). The human worker and the robot coexist in the workspace to teach and perform construction tasks 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 unidentical circumstances. In the performance phase, the robot first observes the scene to determine the start and target locations through scene understanding methods or human

[1] Research Associate, Dept. of Civil and Environmental Engineering, Univ. of Michigan, 2350 Hayward St., 2105 G.G. Brown Bldg., Ann Arbor, MI 48109 (corresponding author). ORCID: https://orcid.org/0000-0002-0213-8471. Email: cjliang@umich.edu

[2] Professor, Dept. of Civil and Environmental Engineering, Univ. of Michigan, 2350 Hayward St., 2105 G.G. Brown Bldg., Ann Arbor, MI 48109. ORCID: https://orcid.org/0000-0003-0788-5588. Email: vkamat@umich.edu

[3] Associate Professor, Dept. of Civil and Environmental Engineering, Univ. of Michigan, 2350 Hayward St., 2105 G.G. Brown Bldg., Ann Arbor, MI 48109. ORCID: https://orcid.org/0000-0002-2453-0386. Email: menassa@umich.edu

[4] Associate Professor, Dept. of Architecture and Urban Planning, Univ. of Michigan, 2000 Bonisteel Blvd., Ann Arbor, MI 48109. Email: wesmcgee@umich.edu

Note. This manuscript was submitted on June 2, 2021; approved on October 7, 2021; published online on December 3, 2021. Discussion period open until May 3, 2022; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

instructions (Lundeen et al. 2017, 2019). Then, the robot uses the model to reproduce the task based on the encountered circumstance under the human worker's supervision (Liang et al. 2020a). 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 (Grytnes et al. 2018). 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 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.

In our previous work, we have developed a visual LfD method for teaching robots quasi-repetitive construction tasks (Liang et al. 2020a). We first adapted the context translation model to extract the context from demonstration videos and translate it to the target scene (Liu et al. 2018). We then applied a reinforcement learning method, specifically the trust region policy optimization (TRPO) (Schulman et al. 2015), to generate the robot control policy and complete a ceiling tile installation process. That study demonstrated the applicability of the LfD method to construction tasks but required a sizeable number and variety of demonstration videos in order to achieve acceptable accuracy and adapt to unforeseen scenarios. In an attempt to address this limitation, our prior work also developed an immersive virtual reality (VR)-based online digital twin for human workers to create rapid demonstrations of construction tasks in various configurations (Wang et al. 2021).

With the VR system, demonstration data can be collected either by virtual cameras or by directly tracking the pose of manipulated objects in the virtual environment. First, the use of virtual demonstration data requires additional steps for application to physical robots, such as simulation-to-reality transfer (Zhao et al. 2020) or a combination of physical and virtual demonstrations (Liang et al. 2019a). Second, using the pose of manipulated objects and generating trajectories as demonstration data can mitigate the requirement of scene understanding and the correspondence problem in the visual demonstration data (Argall et al. 2009; Bach and Aggarwal 2012). The pose of manipulated objects can be recorded directly in the virtual environment to ensure the quality of the demonstration data, instead of applying additional processes, such as visual demonstration that require computer vision methods to extract features from video. Trajectory-based learning from demonstration approaches utilize trajectory demonstration data to learn robot paths and have the advantages of minimum parameter tuning requirement and minimum demonstration data requirement (Ahmadzadeh et al. 2017). Therefore, it is easier for construction workers to teach robot apprentices construction tasks using the trajectory demonstration.

Research Contribution

In this paper, we develop a trajectory-based learning from demonstration method for robot apprentices to learn specific construction tasks based on trajectory demonstration data. Ceiling tile installation in preinstalled grids is a common activity in tunnels, subways, and buildings and is thus chosen as the target quasi-repetitive overhead construction task to evaluate the robot imitation and compare the obtained results with the previous state-of-the-art visual LfD method (Liang et al. 2020a). The ceiling tile installation process is intuitive to human workers but poses several challenges to construction robots. The proposed method is implemented in the robot operating system (ROS) and the Gazebo (Koenig and Howard 2004; Quigley et al. 2009) standard physics digital twins for rapid evaluation and subsequent deployment to physical robots (Liang et al. 2020b). In terms of the demonstration data, human workers demonstrate the task using the VR system or control the virtual robot inside the simulator to complete the task while recording the trajectory of the construction component (e.g., tile).

During the human-robot collaboration process, the robot first gathers the tiles from the material staging area and navigates to the installation workspace using available localization and mapping algorithms (Xu et al. 2019, 2020). Then, the robot uses sensors such as cameras or laser scanners to measure the layout of the suspended grids and acquires the correct geometric information through adopted scene understanding methods (Lundeen et al. 2017). After gathering all required information and materials, the human worker indicates the target grid and the installation sequence to the robot. With such instruction, the robot can apply the trajectory-based learning from demonstration method to manipulate and place tiles at the target grid locations. Finally, the human worker inspects the tile alignment and performs any necessary adjustments. This workflow allows the robot to perform all physically demanding construction tasks while the human worker demonstrates the tasks beforehand and supervises the robot's work process from a safe place. The quality of the robot's work is also monitored by the human supervisor.

The remainder of this paper is organized as follows. First, existing construction robotics literature and robot learning from demonstration literature are reviewed, and the knowledge gaps are identified. Second, a trajectory-based learning from demonstration method, i.e., generalized cylinders with orientation (GCO) approach, is introduced to teach robot apprentices various construction tasks. Finally, ceiling tile installation experiments are conducted in the standard physics Gazebo robot simulator to evaluate the proposed method and compare its performance with a state-of-the-art visual LfD method that is based on integrating a context translation model with reinforcement learning.

Related Work

This section discusses the state-of-the-art in construction robotics and motivates the need for human-robot collaboration on construction sites. Second, we review existing work on robot imitation learning methods and discuss trajectory-based learning from demonstration methods for construction applications.

Robotics in Construction

Construction robots have been sporadically used on construction sites to assist with heavy-duty tasks (Chung et al. 2010) or navigate to hazardous or narrow locations to perform construction work (Beckett and Ross 2017). Each robot has typically been designed for specific functionality and the performance of a single-task (Bock and Linner 2016), such as bricklaying (Feng et al. 2015; Yu et al. 2009), welding (Nagata et al. 1997), or beam assembly (Chu et al. 2013; Jung et al. 2013; Liang et al. 2017). In order to analyze the existing research on construction robotics and highlight the knowledge gaps, we have previously proposed a taxonomy for human-robot collaboration in construction (Liang et al. 2021).

Preprogramming, adaptive manipulation, learning from demonstration, improvisatory control, and full autonomy are the five primary groups in the taxonomy categorized based on the level of robot autonomy and human effort in the performance of work.

In the first two levels (i.e., preprogramming and adaptive manipulation), the human worker plans and controls the robot to complete a construction task, whereas the robot executes the predefined work plan or uses sensors to make minor adjustments to the work plan. For example, the tile placement robot, bricklaying robot, or three-dimensional (3D) printing robot (contour crafting) (Carneau et al. 2020; Khoshnevis 2004; Vantyghem et al. 2020) are programmed with the robot code generated by the designed pattern and executed on-site (King et al. 2014; Yu et al. 2009). Sensors such as a laser profiler (Lundeen et al. 2017, 2019), force sensor (Yousefizadeh et al. 2019), or camera (Feng et al. 2015) can help adjust the preprogrammed work plan or receive remote control commands from the human worker to resolve any minor issues resulting from as-designed versus as-built discrepancies. Most robots designed to work in civil infrastructure environments are included in these two categories [e.g., tunnel or bridge inspection robots (Bolourian and Hammad 2020; Victores et al. 2011), bridge beam assembly robots (Yang et al. 2018)]. However, if the quality of the robot-built component is unacceptable, the human worker has to demolish the work performed and reconstruct it manually. Any unforeseen situations such as arbitrary obstructions or major discrepancies resulting from loose construction tolerances will also prevent a robot from accomplishing a task (Liang et al. 2020a).

In the last two levels (i.e., improvisatory control and full autonomy), the robot is expected to plan and execute the work sequence to complete a task with the human worker supervising the process and switching to manual control if necessary. For example, a drilling robot that has been utilized for landslide consolidation can autonomously operate, supervised by human workers remotely, and switch to tele-operation mode when necessary (Molfino et al. 2008). Autonomous navigation robots are used in built environments or construction sites for maintenance and construction applications without human control or intervention (Asadi et al. 2018; Tsuruta et al. 2019), especially for indoor or GPS-denied environments (Xu et al. 2020). However, these types of robots are unable to perform routine construction tasks such as drywall installation or ceiling tile fitting, which require guidance from human workers. In addition, when a 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 is intrinsically unaware of how human workers resolve a situation and requires human assistance every time it encounters similar circumstances.

Based on the previous categorization and the associated literature review, it is apparent that prior work in construction robotics has not extensively explored the idea of transferring knowledge and problem-solving skills from human workers to robots assisting in the performance of construction work. Such knowledge transfer can potentially enable robots to learn work tasks directly from humans and resolve problems that arise in the performance of quasi-repetitive construction tasks that cannot be readily solved through optimization approaches alone. For example, the process of ceiling tile installation requires complex manipulation trajectories that could be different for various tiles to pass through the grid area while avoiding collision with utilities that may be present above the suspended grids. Such encountered workface conditions can be easily resolved by experienced human workers, and the prospect of transferring such knowledge to robots through learning from demonstration or imitation learning methods (Liang et al. 2020a) offers promise in enabling the robots to perform such work with increased autonomy.

Robot Learning from Demonstration

Robot learning from demonstration or imitation learning methods enable a robot to acquire new skills by imitating observed demonstrations from human experts (Argall et al. 2009) and is an advantageous approach when the involved skills can neither be preprogrammed nor expressed as optimization problems (Ravichandar et al. 2020). LfD methods are typically applied to manipulation tasks or assembly tasks (Schwenkel and Guo 2019; Zhu and Hu 2018) that can be easily demonstrated by human experts. Existing LfD methods can be categorized based on the demonstration methods or learning methods (Liang et al. 2020a; Ravichandar et al. 2020; Torabi et al. 2019). The demonstration methods are concerned with how the skills are demonstrated to the robot, including trajectory demonstration and passive observation. Conversely, 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.

Demonstration Method: Trajectory Demonstration

The trajectory demonstration method belongs in the first group of demonstration methods, which directly provide the trajectory of the task to the robot. These types of methods include kinesthetic demonstration, tele-operated demonstration, and extended reality (XR) demonstration to record the demonstration trajectory. Fig. 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 (Calinon et al. 2006). The onboard sensors or external sensors are used to record the trajectory data, including joint angles, end-effector poses, and motor torques (Kormushev et al. 2011; Maeda et al. 2017; Song et al. 2021a; Zahedi et al. 2020). The kinesthetic demonstration provides an intuitive way to interact with the robot and eliminates the correspondence problem (Argall et al. 2009), but is limited to lightweight and small object manipulation (Zhu and Hu 2018). Thus, it is challenging to apply kinesthetic demonstration to construction tasks where the objects being manipulated are typically heavy and oversized.

In the tele-operated demonstration, the human expert controls the robot with a remote controller to demonstrate the task (Kukliński et al. 2014; Mandlekar et al. 2020). 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 (Kormushev et al. 2011). The tele-operated demonstration supports broader applications such as autonomous helicopter learning (Abbeel et al. 2010), mobile robot positioning (Argall et al. 2008), and hierarchical tasks learning (Mohseni-Kabir et al. 2015) 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 XR demonstration, the use of virtual reality or augmented reality (AR) provides various scenarios for human experts

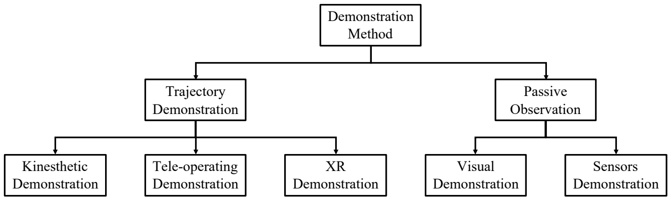


Fig. 1. Categories of the demonstration method.

to demonstrate tasks to the robot (Luebbers et al. 2019; Wang et al. 2021). The human expert can either control the virtual robot to demonstrate the task (Zhang et al. 2018) or directly demonstrate the task in the virtual environment and record the video (Dyrstad and Mathiassen 2017; Koganti et al. 2018). 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 XR, the human expert can demonstrate different construction tasks inside the virtual environment with different components and backgrounds.

Demonstration Method: Passive Observation

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 (Fitzgerald et al. 2015; Liu et al. 2018). Visual demonstration is susceptible to typical computer vision-related challenges such as occlusion (Liang et al. 2019b).

Motion capture systems provide accurate human whole-body motion data and can record various human motions, such as lifting objects for demonstration (Skoglund et al. 2010). 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 (Edmonds et al. 2017; Kukliński et al. 2014). 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 (Edmonds et al. 2017; Song et al. 2021b; Zahedi et al. 2020). 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.

Learning Method

The probabilistic approach, dynamic system approach, and reward-based approach are three subgroups of the learning method, as shown in Fig. 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 (Calinon et al. 2006, 2010; Zhu and Hu 2018). In addition, the 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 trajectories (Calinon et al. 2006; Jaquier et al. 2019; Song et al. 2021a). However, these methods usually require extensive parameter tuning to get robust manipulation (Ahmadzadeh and Chernova 2018). In order to minimize the parameter tuning process, Ahmadzadeh and Chernova (2018) developed the generalized cylinders–based LfD method to generate trajectories within the geometry. The robot trajectory is sampled inside the generalized cylinder (GC) with three degrees-of-freedom (DOF) world coordinates (X, Y, Z), and the orientation is randomly determined when applying inverse kinematics to obtain robot joint angles and control policy. However, in construction tasks, it is important to consider the manipulation orientation in order to pass (i.e., traverse through) some specific geometric constraints, especially for overhead construction. Therefore, this paper proposes the orientation constraint to significantly advance the GC approach for construction manipulation tasks.

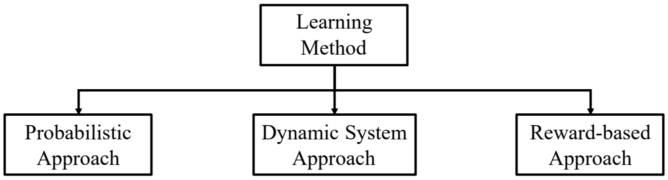


Fig. 2. Categorization of the learning method.

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 (Ijspeert et al. 2013; Pastor et al. 2009) but also requires an extensive 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 (Khansari-Zadeh and Billard 2011).

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 (Ravichandar et al. 2020). 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 minimal (Bain and Sammut 1999; Ravichandar et al. 2019). 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 (Abbeel et al. 2010; Ng and Russell 2000), 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) (Ho and Ermon 2016; Kinose and Taniguchi 2020) is one example of combining IRL with generative adversarial networks (GAN) (Goodfellow et al. 2014).

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 state-action demonstrations (i.e., visual observation with expert's action) (Liu et al. 2018; Torabi et al. 2019). The challenge of the IfO is how to extract actions from demonstration states. The dynamics model is the first type of the IfO method (Edwards et al. 2019; Pavse et al. 2020) that learns the action from state demonstration using forward or inverse dynamics models. For example, one of the inverse dynamics methods extracts actions a_t from state transitions (s_t, s_{t+1}) in the demonstration video to train the model, then uses it to determine the actions for robots to follow (Nair et al. 2017).

Reinforcement learning is the second type of IfO method that applies GAN to learn the policy (Merel et al. 2017) or manually defines the reward function (Finn et al. 2017; Liu et al. 2018; Sermanet et al. 2018). The visual demonstration and reward-based IfO method has been adapted and applied to quasi-repetitive construction tasks, i.e., ceiling tile installation process, in previous work (Liang et al. 2019a, 2020a). In this paper, we develop a trajectory demonstration and probabilistic method, i.e., generalized cylinders with orientation approach, for teaching robots similar types of processes and advancing the state of the art for comparison of the results with previous work. The trajectory demonstration and probabilistic method requires fewer demonstration data from human workers and results in a faster training process than the other learning method. The high accuracy and the ability to overcome uncertainty indicated by literature are also suitable for construction tasks.

In summary, the proposed generalized cylinders with orientation approach eliminates the requirement of the extensive parameter tuning process and large amounts of demonstration data associated with the probabilistic approach. In addition, the GCO approach also solves the issues of precise orientation manipulation in the GC method by using orientation constraints, heavy and large material in the kinesthetic demonstration, and unintuitive controlling issues in the tele-operated demonstration by XR demonstration.

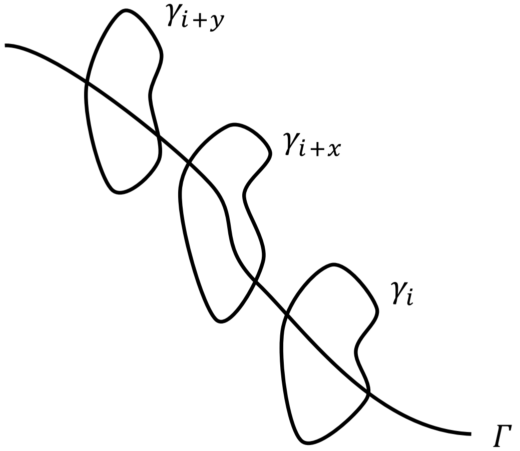


Fig. 3. Example of the generalized cylinder. Each cross section γ is perpendicular to each other along the center axis Γ.

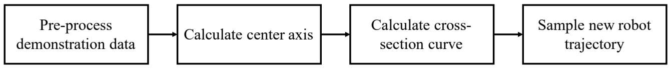

Pre-process demonstration data
Calculate center axis
Calculate cross-section curve
Sample new robot trajectory

Fig. 4. Procedure of the generalized cylinder method for robot learning from demonstration.

Generalized Cylinders with Orientation Approach

The generalized cylinder is a generic representation of an arbitrary cylinder. The center axis of the cylinder is defined as arbitrary spline curve $\Gamma(s) = (x(s), y(s), z(s))$ and the cross-section boundary of the cylinder is a closed curve $\gamma(r,s) = (x(r,s), y(r,s))$ with different shapes. Each cross section along the center axis is perpendicular to each other. Fig. 3 illustrates an example of GC with the center axis and three cross sections. The GC can be represented as

$$G(r,s) = \Gamma(s) + x(r,s)\nu(s) + y(r,s)\xi(s) \tag{1}$$

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 (Martínez-Salvador et al. 2003), mapping and state estimation (Özaslan et al. 2018), and learning from demonstration (Ahmadzadeh and Chernova 2018). In this research, we modify the GC for LfD to suit basic construction manipulation tasks and further propose a new generalized cylinders with orientation approach to perform complex construction manipulation tasks. The strategies of handling unforeseen situations and obstacle avoidance are also developed for human-robot collaboration. Details of each of these elements are discussed in the following subsections.

Generalized Cylinders for Robot Learning from Demonstration

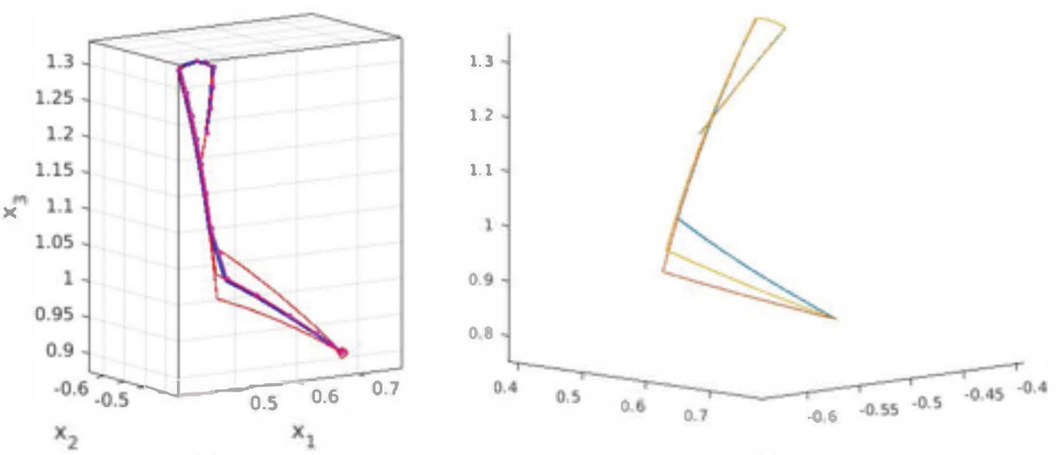
For the learning from demonstration approach, the GC is constructed from demonstration data and determines the robot trajectory within the GC space. Fig. 4 shows the detailed procedure of the GC for LfD. First, the demonstration data is preprocessed to obtain aligned data. Second, the center axis curve 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.

In the first step, the demonstration data is processed and aligned. The demonstration data is captured using the virtual robot simulator in Liang et al. (2020a), 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 $D^i = (D^i_x, D^i_y, D^i_z)$ where $i = 1, \ldots, m$ represents ith demonstration data set, i.e., robot's end-effector pose in 3D Cartesian coordinates.

Each demonstration data set has a different number of data points and requires the alignment process. We first apply the Ramer–Douglas–Peucker (RDP) algorithm to simplify and remove the redundant points in the demonstration data. Only key points remain in the simplified trajectory. The RDP algorithm takes the first and the last data points to find the farthest data point from the line segment and eliminates the data point whose distance to the line segment is smaller than the predefined threshold. By iterating this process, the RDP algorithm can preserve the keypoints in the demonstration data. Then, we resample the trajectory with n new data points, including key points from the simplified trajectory. Finally, we apply the dynamic time warping (DTW) algorithm to align each resampled demonstration data. The resulting demonstration data becomes $\widehat{D^i} = (\widehat{D^i_x}, \widehat{D^i_y}, \widehat{D^i_z})$, where $\hat{D} \in \mathbb{R}^{3\times n\times m}$ represents the set of demonstration having m different demonstration trajectories where each trajectory has n data points in 3D Cartesian coordinates. Fig. 5 shows the original and the processed demonstration data. On the left side [Fig. 5(a)] is the original demonstration data with three trajectories and different numbers of data points in each trajectory. On the right side [Fig. 5(b)] is the processed demonstration data, where all three trajectories have the same number of data points and are aligned with each other.

In the second step, the center axis Γ of the GC is calculated using processed demonstration data. We simply compute the average location of the demonstration data at the sth data point and assign to $\Gamma(s)$

$$\Gamma(s) = (x(s), y(s), z(s)) = \mathrm{average}(\hat{D}(s)) \tag{2}$$



(a)

(b)

Fig. 5. Example of (a) original demonstration data; and (b) processed demonstration data.

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 sth data point, we take all corresponding points from processed demonstration data $\hat{D}(s)$ and apply cubic spline interpolation to fit the data with the closed curve (Ahmadzadeh and Chernova 2018). Fig. 6 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. (1). Fig. 7 shows an example of the GC constructed by three demonstration data.

In the final step, a new robot trajectory has to be sampled within the GC. We follow the skill reproduction process in Ahmadzadeh and Chernova (2018) 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, we first project the initial pose p_0 onto S_1 and get the new pose p_1'. We use p_t to represent the current pose and p_{t+1} to represent the new pose in Eqs. (3)–(5) to keep consistency $(t = 0, 1, \ldots n-1)$

$$p'_{t+1} = T_t^{t+1} p_t \tag{3}$$

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, we project the center axis point $\Gamma(t)$ 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:

$$\eta = \frac{|\overline{p_t \Gamma(t)}|}{|\overline{g_t \Gamma(t)}|} \tag{4}$$

After obtaining the new pose p'_{t+1}, we again project the center axis point $\Gamma(t+1)$ 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

$$p_{t+1} = (\eta |\overline{g_{t+1} \Gamma(t+1)}|) p'_{t+1} \tag{5}$$

The entire process is repeated through every cross section to generate the new robot trajectory. Fig. 8 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 Fig. 9.

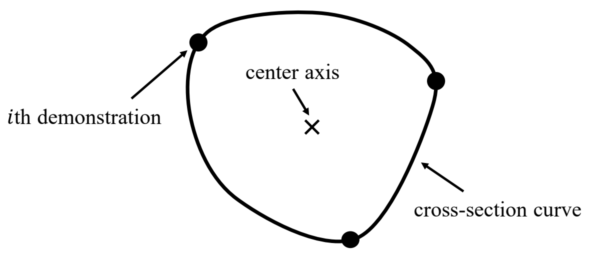


Fig. 6. Cross-section curve is defined by three demonstration data and a center axis data point.

Set 1

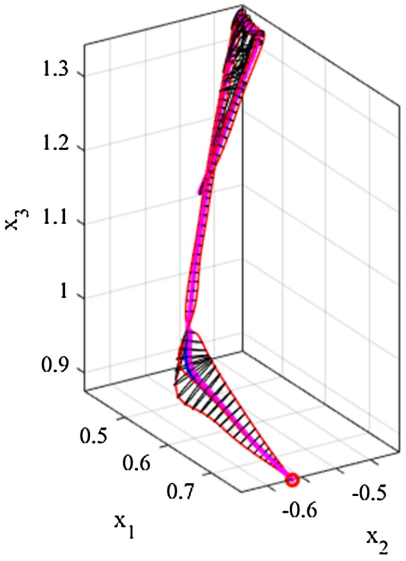


Fig. 7. Example of the GC constructed by three demonstration data.

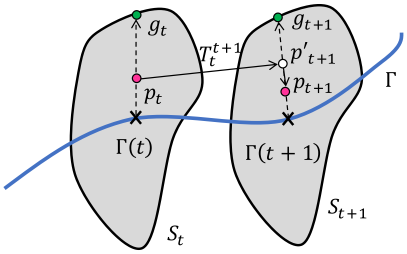
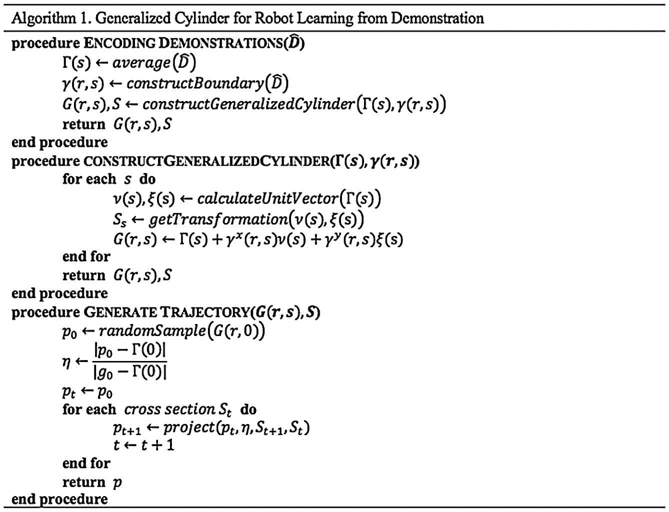


Fig. 8. Process of sampling new pose from the cross-section plane S_t to the next plane S_{t+1}.



```
Algorithm 1. Generalized Cylinder for Robot Learning from Demonstration
procedure ENCODING DEMONSTRATIONS($\hat{D}$)
    $\Gamma(s) \leftarrow average(\hat{D})$
    $\gamma(r,s) \leftarrow constructBoundary(\hat{D})$
    $G(r,s), S \leftarrow constructGeneralizedCylinder(\Gamma(s), \gamma(r,s))$
    return $G(r,s), S$
end procedure
procedure CONSTRUCTGENERALIZEDCYLINDER($\Gamma(s), \gamma(r,s)$)
    for each $s$ do
        $v(s), \xi(s) \leftarrow calculateUnitVector(\Gamma(s))$
        $S_s \leftarrow getTransformation(v(s), \xi(s))$
        $G(r,s) \leftarrow \Gamma(s) + \gamma^x(r,s)v(s) + \gamma^y(r,s)\xi(s)$
    end for
    return $G(r,s), S$
end procedure
procedure GENERATE TRAJECTORY($G(r,s), S$)
    $p_0 \leftarrow randomSample(G(r,0))$
    $\eta \leftarrow \frac{|p_0 - \Gamma(0)|}{|g_0 - \Gamma(0)|}$
    $p_t \leftarrow p_0$
    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
```

Fig. 9. Algorithm of the generalized cylinder for robot learning from demonstration.

Orientations Constraint

After obtaining the robot learned trajectory from the GC, the robot control policy can be determined using inverse kinematics. The robot trajectory is in 3D Cartesian coordinates 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. Fig. 10 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. We define the demonstration data points with minimum distance to each other 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. Fig. 11 shows an example of the insertion point and the critical orientation.

We propose the new algorithm called generalized cylinders with orientation approach using the GC method and orientation constraint method. Fig. 12 shows the procedure of the orientation constraint for the GC method. First, we find the cross section with the minimum area and the insertion point. The demonstration data points on this cross section are closest to each other.

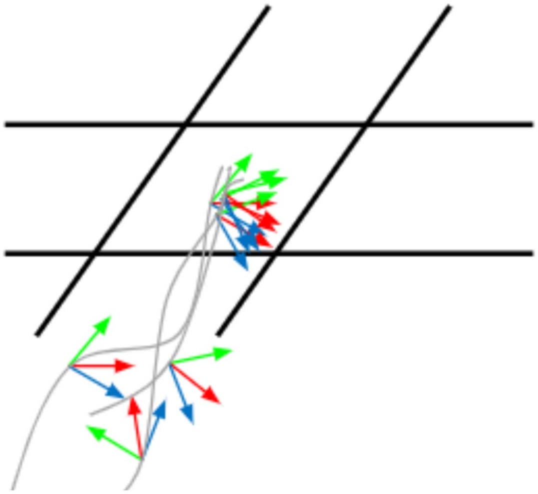


Fig. 10. Orientation information of the ceiling tile installation manipulation. The orientations near the grid are similar to each other.

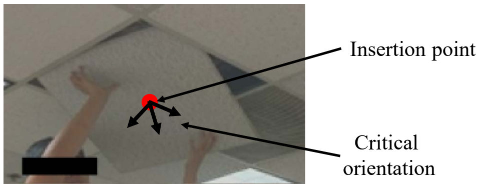

[redacted]

Fig. 11. Example of the insertion point and the critical orientation.

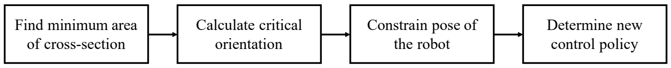

Find minimum area of cross-section
Calculate critical orientation
Constrain pose of the robot
Determine new control policy

Fig. 12. Procedure of the orientation constraint for the GC method.

Second, we calculate the critical orientation 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.

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. (1), the cross-section curve at the sth data point is $\gamma(r,s) = (x(r,s), y(r,s))$. The area of the cross-section curve can be calculated by

$$\text{area}(s) = \int_0^p \left(x\frac{dy}{dr} - y\frac{dx}{dr} \right) dr \tag{6}$$

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

In the second step, the critical orientation is calculated at the inserting point $\hat{s}$. The orientation of the demonstration data is represented using quaternions. We define the critical orientation as the average of the quaternions. Based on the definition, the average quaternion is the argument of the minima of the following equation (Markley et al. 2007):

$$\bar{q} = \underset{q \in \mathbb{S}^3}{\text{argmin}} \sum_i^n \omega_i \| A(q) - A(q_i) \|_F^2 \tag{7}$$

```
Algorithm 2. Orientation Constraint for the Generalized Cylinder Approach
procedure ORIENTATION CONSTRAINT($\hat{D}, G(r,s)$)
        for each $s$ do
                $area(s) \leftarrow \int_0^p \left(x\frac{dy}{dr} - y\frac{dx}{dr}\right) dr$
        end for
        $\hat{s} \leftarrow findMinimumArea(area(s))$
        $M \leftarrow constructQuaternionMatrix(\hat{D}(\hat{s}))$
        $\bar{q} \leftarrow findMaximumEigenvector(M)$
        return $\bar{q}$
end procedure
```

Fig. 13. Algorithm of the orientation constraint for the generalized cylinder approach.

Using Eq. (7), we can further derive the average quaternion by

$$\bar{q} = \underset{q\in\mathbb{S}^3}{\operatorname{argmax}}\, q^T M q \tag{8}$$

$$M = \sum_{i}^{n} \omega_i q_i q_i^T \tag{9}$$

Therefore, the average quaternion is the normalized eigenvector corresponding to the maximum eigenvalue of M. We calculate the average quaternion using Eq. (9) and the eigendecomposition process. In the final step, we apply inverse kinematics with the critical orientation constraint and the last orientation data (the orientation for fitting the grid) to find the robot control policy. The pseudocode of the GCO can be found in Fig. 13.

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 obstructions. 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 (Ahmadzadeh and Chernova 2018). Since construction tasks are quasi-repetitive and subject to various start and target locations, we propose a trajectory adaptation approach to refine the robot trajectory based on the new start and target locations.

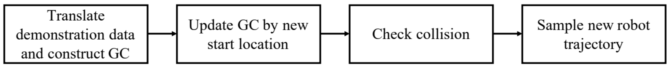

Translate demonstration data and construct GC
Update GC by new start location
Check collision
Sample new robot trajectory

Fig. 14. Procedure of the trajectory adaptation approach to refine the robot trajectory.

Fig. 14 shows the procedure of the trajectory adaptation approach. First, we translate each demonstration data to the new scene and match the target data point with the new target location. Then, we construct the GC using the translated demonstration data. Second, we update the GC by the new start location. Third, the collision of the new GC is checked by the collision detection algorithm. If the collision exists, we will update the GC to avoid the obstacle. Finally, we sample a new robot trajectory within the new GC with the orientation constraint and apply inverse kinematics to determine the robot control policy.

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 $\hat{D}$ to the new scene and matches the new target location p_t. Next, we construct the GC using the translated demonstration data $\widehat{D'}$ and the algorithm in Fig. 9 before sampling a new trajectory. In the second step, we update the GC 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, we can simply sample the new trajectory starting from the cross section of the new start location p_0 to the new target location p_t, as shown in Fig. 15(a). If the new start location p_0 is outside the GC space and coplanar with the first cross-section plane of the GC, we directly connect the new start location to the previous start location, as shown in Fig. 15(b). By following this process, the robot can maneuver on the first cross-section plane S_0 and follow the same trajectory afterward.

If the new start location p_0 is outside the GC space and not coplanar with the first cross section of the GC, we connect the new start location p_0 to every start waypoint of the demonstration data $\hat{D}'(0)$ with straight lines. Then, we construct a new GC $G'(r,s)$ using these updated straight-line demonstration data

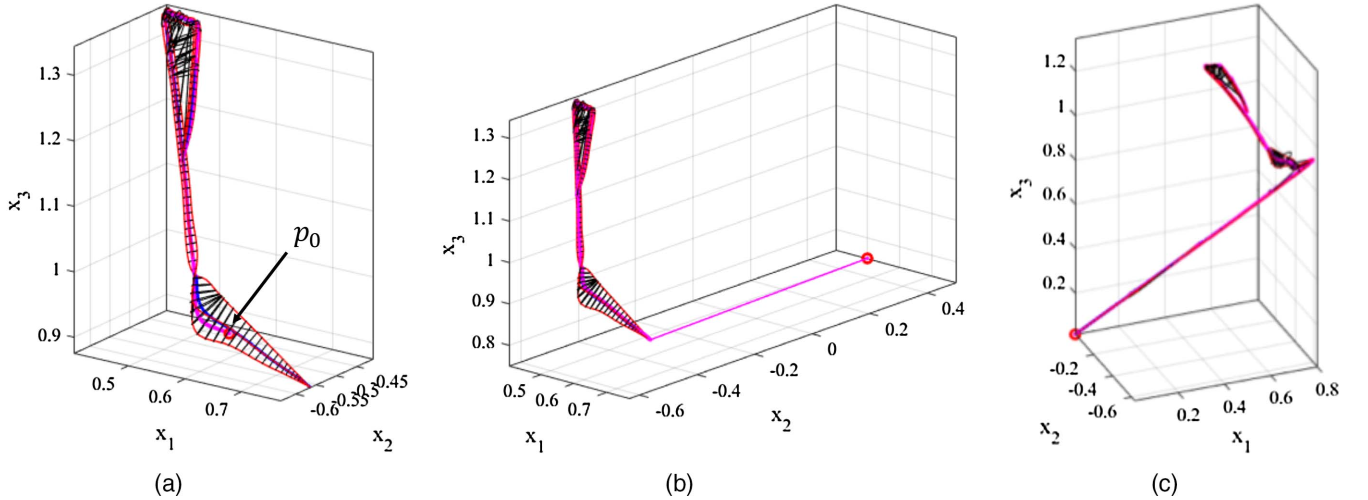
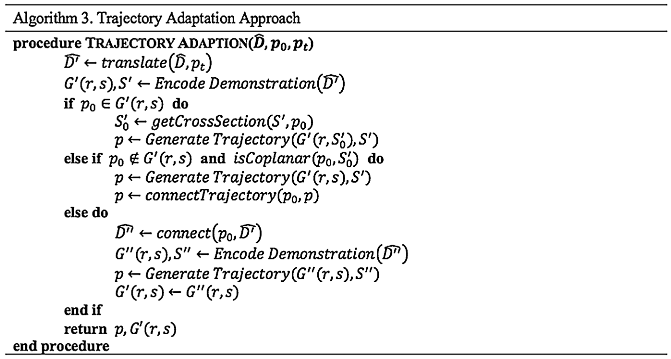


Fig. 15. Updated GC with different start and target locations: (a) inside GC; (b) outside GC and coplanar with the first cross section; and (c) outside GC and not coplanar with the first cross section.



```
Algorithm 3. Trajectory Adaptation Approach
procedure TRAJECTORY ADAPTION($\hat{D}, p_0, p_t$)
        $\widehat{D'} \leftarrow translate(\hat{D}, p_t)$
        $G'(r,s), S' \leftarrow Encode\ Demonstration(\widehat{D'})$
        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
```

Fig. 16. Algorithm of the trajectory adaptation approach.

$\widehat{D''}(0)$ and resample the robot trajectory. Because all updated demonstration data $\hat{D}''$ 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, we randomly sample a waypoint on the cross-section plane S_1 and utilize it as the second robot waypoint p_1. Then, we can repeat the generating trajectory procedure in Fig. 9 to find the new robot trajectory. Fig. 15(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. Fig. 16 shows the pseudo-code of the trajectory adaptation approach.

In the third step, we apply the collision detection algorithm to validate the GC. The bounding box algorithm is used to create bounding boxes around each geometry in the environment. We assume that the robot has all information of the surrounding environment using an approach described in previous work to collect and synchronize the geometry data (Liang et al. 2020b; Lundeen et al. 2017) 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 collisions. Existing methods used the adaptive ratio and deformation function to avoid the obstacle intersecting with the GC (Ahmadzadeh and Chernova 2018). We propose a human-in-the-loop refinement approach 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, we connect the center axis $\Gamma(1)$ to the new demonstration data $\widehat{D^m}(1)$ with a straight line and define a shift ratio ρ to select the waypoint

$$\rho = \frac{n}{|\overline{\Gamma(1)\widehat{D^m}(1)}|} \tag{10}$$

where n represents the total number of 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. Fig. 17 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, we can sample the rest of the robot trajectory using the updated GC and the algorithm in Fig. 9.

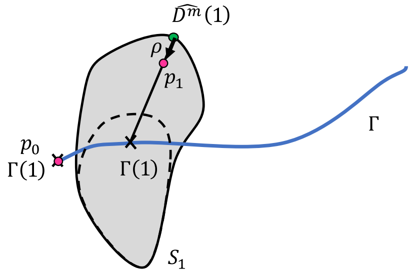
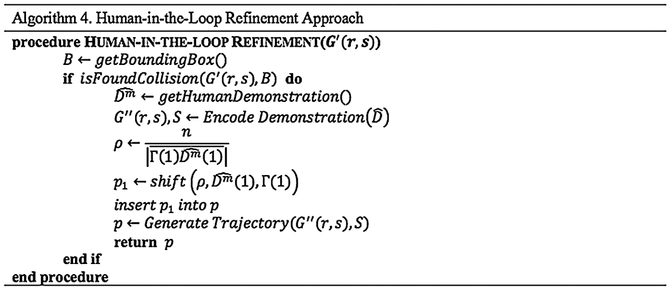


Fig. 17. Process of determining the robot waypoint p_1 on the first cross-section plane S_1.



```
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
                $\widehat{D^m} \leftarrow getHumanDemonstration()$
                $G''(r,s), S \leftarrow Encode\ Demonstration(\hat{D})$
                $\rho \leftarrow \frac{n}{|\overline{\Gamma(1)\widehat{D^m}(1)}|}$
                $p_1 \leftarrow shift\left(\rho, \widehat{D^m}(1), \Gamma(1)\right)$
                $insert\ p_1\ into\ p$
                $p \leftarrow Generate\ Trajectory(G''(r,s), S)$
                return $p$
        end if
end procedure
```

Fig. 18. Algorithm of the human-in-the-loop refinement approach.

Conversely, 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 collision, we include more critical orientation data points near the installation location. 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.

We propose two steps 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 a collision still occurs, we will apply the human-in-the-loop refinement approach 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. Fig. 18 shows the pseudo-code of the human-in-the-loop refinement approach.

Experiments and Results

We conducted a series of experiments to evaluate the proposed trajectory-based learning from demonstration approach. The ceiling tile installation process was chosen as the target quasi-repetitive construction task to teach and evaluate the robot. The standard physics robot simulator ROS Gazebo was used to build the robot's

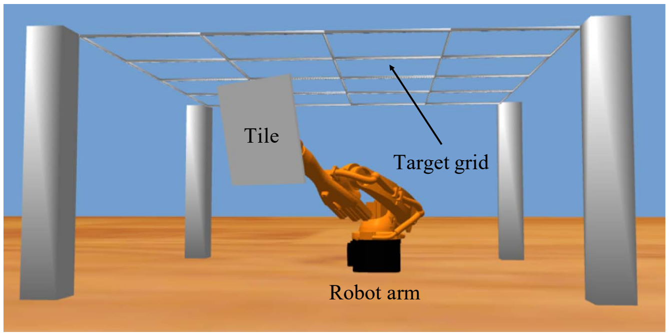


Fig. 19. ROS gazebo robot simulator. The KUKA industrial robot arm with a ceiling tile mounted on the end-effector and suspended grids were implemented in the simulator.

work environment, collect demonstration data, and evaluate the robot's performance (Liang et al. 2020b). The success rate of the installation is used as the evaluation metric for direct comparison with the same metric used in a previous work (Liang et al. 2020a), which is defined by whether the tile is placed at the desired location within the tolerated threshold.

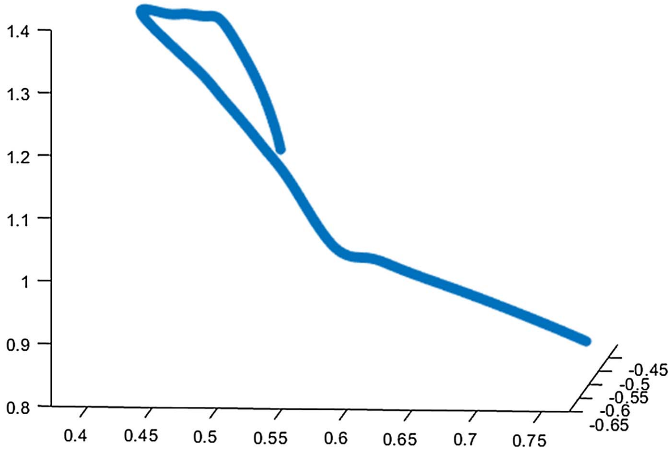


Fig. 20. One of the processed demonstration trajectories with 1,500 waypoints.

Experimental Setup

We first collected the demonstration data in the ROS Gazebo environment. A KUKA 6-DOF industrial robot arm was implemented in Gazebo and RViz (Kam et al. 2015; Koenig and Howard 2004) along with the suspended grids and tiles. Fig. 19 shows the ROS Gazebo robot simulator and the experimental environment. 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, we defined three different target locations, i.e., three different grids, and demonstrated four different trajectories for each target location from similar start locations (total 12 sets of demonstration trajectories). The proposed method applies cubic spline interpolation to fit data with the closed curve on each cross section, which requires at least three demonstration data points. We choose to use four demonstration data in our system to add additional constraints to the generalized cylinder.

We assumed that the robot had picked up the tile, and thus the tile was secured on the robot's end-effector. The demonstration data were preprocessed using the method discussed in the section "Generalized Cylinders for Robot Learning from Demonstration" to smoothen and align the trajectories. Each demonstration trajectory was resampled to 1,500 waypoints, as suggested in Ahmadzadeh and Chernova (2018). The number of the waypoints affects the computational time, the trajectory smoothness, and the collision checking ability. Fig. 20 shows one of the processed demonstration trajectories.

We implemented the generalized cylinders with orientation approach in MATLAB version R2020b and sent the control policy to the Gazebo robot. The advantage of using a robot simulator such as ROS Gazebo is that it enables rapid implementation and evaluation of newly proposed algorithms under varying circumstances that replicate the physical environment (Liang et al. 2020a). For example, we can create several obstacles in the simulator to test the trajectory adaptation and human-in-the-loop refinement approach. Furthermore, using the ROS framework, the virtual robot can directly communicate with the physical robot for implementation in the physical environment (Liang et al. 2020b).

We designed the experiments 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. We defined 50 start locations 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. We defined 10 different start locations

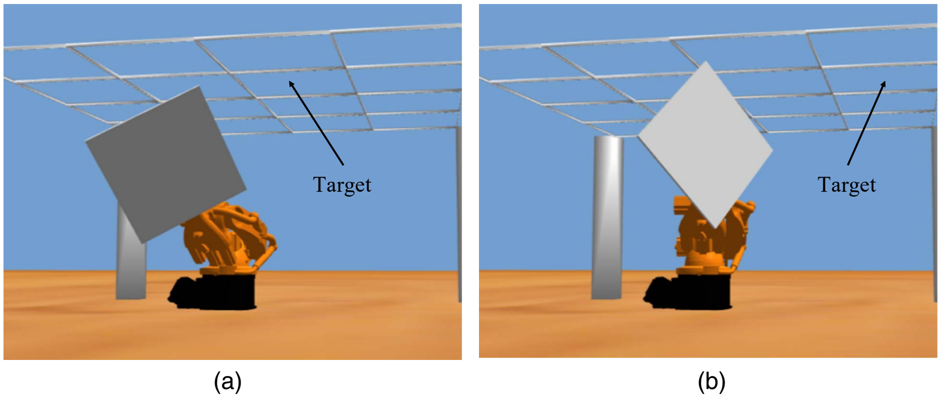


Fig. 21. Two examples of different start locations and target grids.

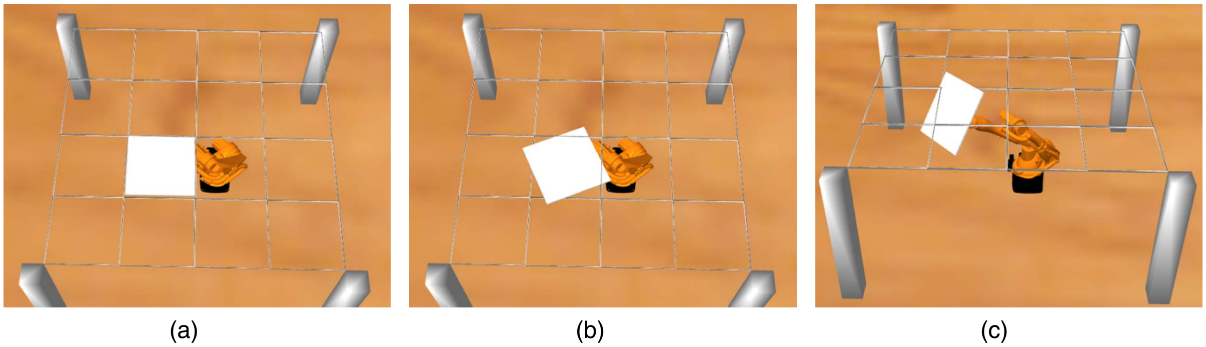


Fig. 22. Examples of (a) the success; (b) unable to pass the grid; and (c) collide with the grid cases.

and 10 different target grids to test the trajectory adaptation approach. We also captured the image of the start locations for the imitation from observation approach to compare with the GCO approach. Fig. 21 shows two examples of different start locations and target grids.

For the evaluation, we used the volumetric success rate as the evaluation metric, which is the same metric used in a previous work (Liang et al. 2020a). We calculate the volumetric difference of the final tile placement and desired tile placement, then compare the difference with the predefined threshold. If the volumetric difference exceeds the threshold, the robot execution is considered as failed. Based on the ceiling tile manual (Ceiling Tile UK 2018), the 60 cm^2 tile ($60 \times 60 \times 1$ cm) allows for 0.5 cm tolerance between the tile and the grid. Therefore, we define the threshold as $60 \times 1 \times 0.5\ \mathrm{cm} = 30\ \mathrm{cm}^3$. Fig. 22 shows examples of the success and the failed cases.

Table 1. Results of the GCO approach, GC approach, and CTRL method for ceiling tile installation

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

Results

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) (Liang et al. 2020a). Table 1 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.

Fig. 23 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,

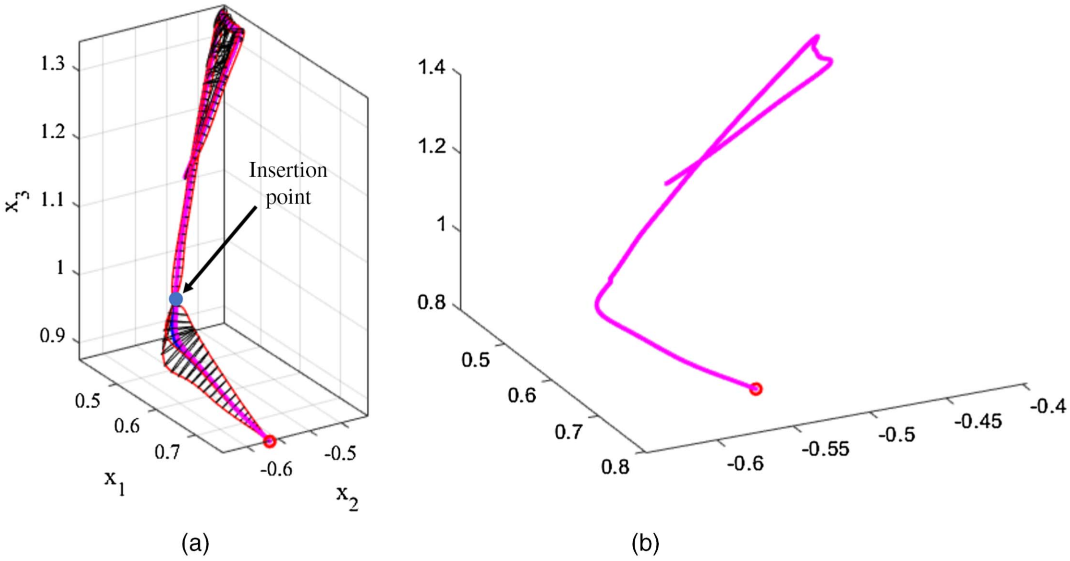


Fig. 23. Results of the GCO approach and the generated robot trajectory.

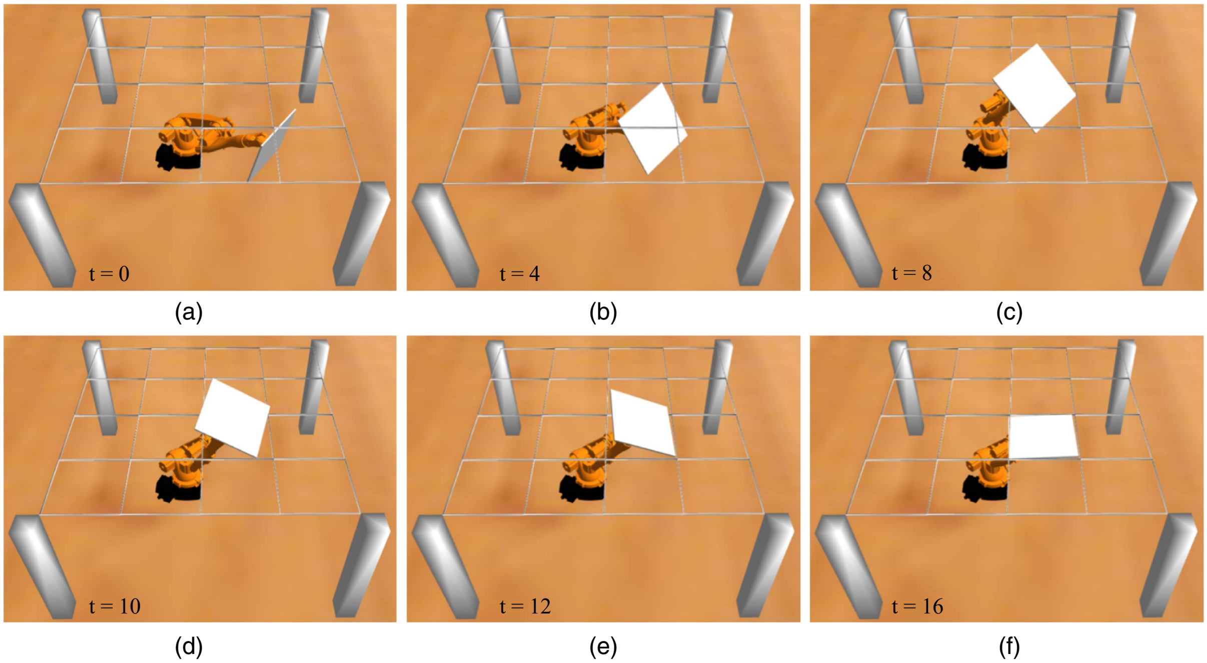


Fig. 24. Sequence of the robot executing the ceiling tile installation process using the GCO approach.

and the insertion point (critical orientation) is shown as the middle dot. The robot will manipulate from the start location p_0 to the target location p_t. For the failed cases in this experiment, the robot was 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. We applied the multiple critical orientations approach and resolved 34 failed cases. Only three cases still collided with the grid due to the inaccurate demonstration orientation recording (98.0% success rate). Fig. 24 shows one of the sequences of the robot executing the ceiling tile installation process.

To further evaluate the multiple critical orientations, we compare different numbers of critical orientations with average computation time and success rate. The GCO approach and the first experiment scenario are used in this evaluation. The computation time is defined as the elapsed time from constructing GC with processed demonstration data to finding robot control policy with inverse kinematics. Table 2 shows the results of the different number of critical orientations. The success rate continually increases when including more critical orientations, and the average computation time also increases because of the additional average quaternion calculation. The 204 critical orientations are the maximum number for resolving failed cases in the experiment, and the remaining three failed cases are caused by the error in the demonstration data.

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 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 insertion process. Even with the multiple critical orientations, the tile could still 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 three 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.

Fig. 25 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. Fig. 26 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.

Table 2. Results of the multiple critical orientations, average computation time, and success rate using GCO approach

Number of critical orientations	Average computation time (s)	Failure	Success rate (%)
1	11.25	37	75.3
10	19.63	33	78.0
100	65.47	19	87.3
204	184.91	3	98.0

Table 3. 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

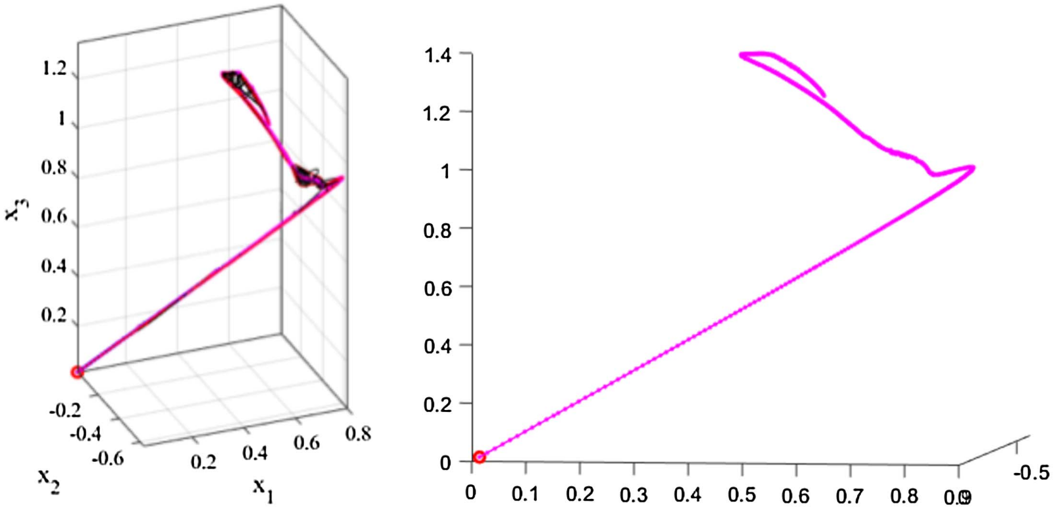


Fig. 25. Results of the GCOT approach and the generated robot trajectory.

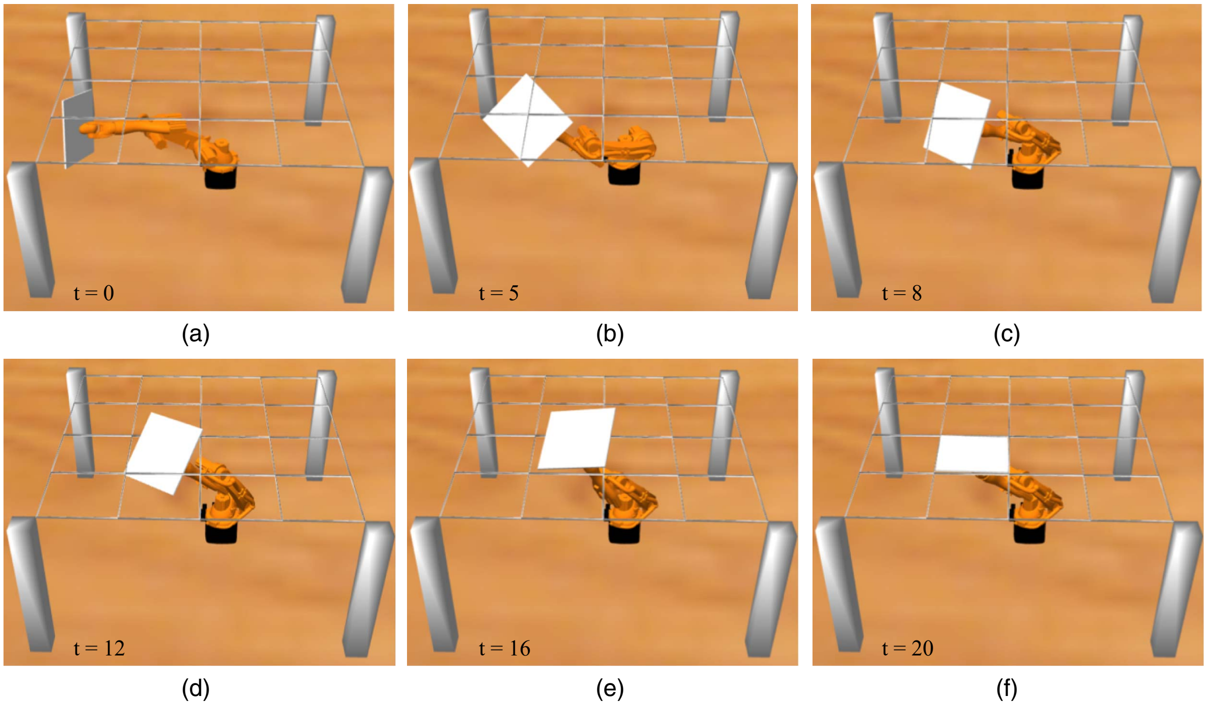


Fig. 26. Sequence of the robot executing the ceiling tile installation process using the GCOT approach with the new start and target locations.

Discussion

To evaluate the proposed trajectory-based learning from demonstration approach, we compared the demonstration data and the experiment results with the imitation learning from observation methods. For the demonstration data, the IfO methods typically require thousands of visual demonstration data (3,000 demonstration data in our experiment) (Liang et al. 2020a; Yu et al. 2019), whereas the proposed trajectory-based approach only needs a few-shot demonstration (four sets of demonstrations in our implementation).

In the CTRL method, the camera viewpoint is fixed for the demonstration data and the robot execution scene in order to increase the success rate. If camera viewpoints are arbitrary for the demonstration data and the robot execution scene, we have to include a greater variety of demonstration data, e.g., provide more demonstration videos with different scenes using the parametric VR system (Wang et al. 2021). In addition, to avoid unforeseen obstacles or improvise the robot's process, we have to apply additional algorithms such as meta reinforcement learning (Finn et al. 2017). Conversely, the GCO approach requires a 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, we can also collect demonstration data in the physical environment using the object pose estimation method (Billings and Johnson-Roberson 2018; Liang et al. 2019b) or markers such as AprilTag and Kanade-Lucas-Tomasi Enhanced by Global constraints (KEG) algorithm (Feng and Kamat 2013; Olson 2011) 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 varied visual demonstration data. The experiment results indicate that the success rate continually increased with multiple critical orientations until the errors were caused by the noisy demonstration data. In the future, the noisy demonstration data can be reduced by including more demonstration data and applying the random sample consensus (RANSAC) method (Fischler and Bolles 1981) to exclude noisy outliers in the data.

In terms of the field application scenario, the proposed method consists of two phases: the learning phase and the performance phase. In the learning phase, the human workers demonstrate and record the task using the VR system or control the virtual robot to complete the task. The recorded data is then processed and learned by the robot in the virtual simulator. This process requires at least one human worker to demonstrate the task, and with more human workers involved, the demonstration data will have a wider variety. In real practice, more demonstration data will yield more detailed generalized cylinder and orientation constraints but increase computational time. After the robot processes and learns the tasks from the demonstration data, the human worker will validate the robot performance in the virtual simulator using the test cases in the experiment to ensure the robot can generate the trajectory and achieve similar performance.

In the performance phase, the physical robot is deployed on the construction site and teams up with a human worker. The human worker will indicate the installation sequence, and the robot will show the planned result in the virtual simulator first and ask for human approval. The human worker can intervene at any point to adjust the trajectory using the proposed human-in-the-loop refinement approach. The physical robot will execute the work plan after receiving approval. A timely advantage of using the proposed approach is the facilitation of social distancing on construction sites as the method only requires one human worker to team up with a robot apprentice and can separate themselves from other human workers on construction sites.

The proposed trajectory-based learning from demonstration method for teaching robots construction tasks also has some limitations. First, the proposed method is evaluated in the virtual simulator and might encounter additional challenges when it is applied to physical robots on real construction sites. For example, the ceiling tiles and the suspended grids might not match each other due to loose tolerances and design-build discrepancies. This issue requires human-in-the-loop adaptation to help the robot adjust the component and provide additional instructions.

Second, the environment feedback is assumed to be collected by additional sensors and registered to the virtual simulator (Lundeen et al. 2017; Xiao et al. 2018). 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 twin can ensure state synchronization between the physical and the virtual environment (Liang et al. 2020b). Third, the proposed method used trajectory demonstration to learn the construction skill. However, some of the construction tasks require different types of demonstrations to learn the skill. 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.

For future work, 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 first be conducted to evaluate the human-in-the-loop refinement approach. Second, the deployment to the physical robot using digital twins for bidirectional communication between the virtual and physical robots will be developed. Third, 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 such as actual tunnel or bridge construction sites.

Conclusions

This research proposed a trajectory-based learning from demonstration method to train robots to perform overhead quasi-repetitive construction tasks. The generalized cylinder 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 standard physics robot simulator ROS Gazebo with ceiling tile installation demonstration trajectories collected from the human-controlled robot simulator and compared with the visual demonstration method (context translation and reinforcement learning method). The results showed that the GCO and GCOT could achieve 98.0% and 82.0% success rates with different start and target locations, which is suitable for training robots to perform overhead construction work alongside human workers.

Data Availability Statement

All data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

The work presented in this paper was supported financially by United States National Science Foundation Awards (Nos. 2025805 and 2128623). Any opinions, findings, and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the United States National Science Foundation.

References

Abbeel, P., A. Coates, and A. Y. Ng. 2010. "Autonomous helicopter aerobatics through apprenticeship learning." *Int. J. Rob. Res.* 29 (13): 1608–1639. https://doi.org/10.1177/0278364910371999.

Ahmadzadeh, S. R., R. M. Asif, and S. Chernova. 2017. "Generalized cylinders for learning, reproduction, generalization, and refinement of robot skills." In *Proc., Robotics: Science and Systems (RSS)*, 1–10. Cambridge, MA: Robotics: Science and Systems Foundation.

Ahmadzadeh, S. R., and S. Chernova. 2018. "Trajectory-based skill learning using generalized cylinders." *Front. Rob. AI* 5 (Dec): 132. https://doi.org/10.3389/frobt.2018.00132.

Argall, B. D., B. Browning, and M. Veloso. 2008. "Learning robot motion control with demonstration and advice-operators." In *Proc., 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 399–404. New York: IEEE.

Argall, B. D., S. Chernova, M. Veloso, and B. Browning. 2009. "A survey of robot learning from demonstration." *Rob. Auton. Syst.* 57 (5): 469–483. https://doi.org/10.1016/j.robot.2008.10.024.

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

Bach, W., and J. K. Aggarwal. 2012. *Motion understanding: Robot and human vision*. New York: Springer.

Bain, M., and C. Sammut. 1999. "A framework for behavioural cloning." *Mach. Intell*. 15: 103–129.

Beckett, A., and R. Ross. 2017. "PyroShield—A HVAC fire curtain testing robot." *Autom. Constr.* 81 (Sep): 234–239. https://doi.org/10.1016/j.autcon.2017.06.009.

Billings, G., and M. Johnson-Roberson. 2018. "Silhonet: An RGB method for 6D object pose estimation." Preprint, submitted September 18, 2018. https://arxiv.org/abs/1809.06893.

Bock, T., and T. Linner. 2016. *Construction robots: Elementary technologies and single-task construction robots*. New York: Cambridge University Press.

Bolourian, N., and A. Hammad. 2020. "LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection." *Autom. Constr.* 117 (Sep): 103250. https://doi.org/10.1016/j.autcon.2020.103250.

Calinon, S., F. Guenter, and A. Billard. 2006. "On learning the statistical representation of a task and generalizing it to various contexts." In *Proc., 2006 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2978–2983. New York: IEEE.

Calinon, S., E. L. Sauser, A. G. Billard, and D. G. Caldwell. 2010. "Evaluation of a probabilistic approach to learn and reproduce gestures by imitation." In *Proc., 2010 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2671–2676. New York: IEEE.

Carneau, P., R. Mesnil, N. Roussel, and O. Baverel. 2020. "Additive manufacturing of cantilever—From masonry to concrete 3D printing." *Autom. Constr.* 116 (Aug): 103184. https://doi.org/10.1016/j.autcon.2020.103184.

Ceiling Tile UK. 2018. "Suspended ceiling tiles." Accessed May 5, 2019. https://www.ceilingtilesuk.co.uk/sizes-of-ceiling-tile/.

Chi, H.-L., Y.-C. Chen, S.-C. Kang, and S.-H. Hsieh. 2012. "Development of user interface for tele-operated cranes." *Adv. Eng. Inf.* 26 (3): 641–652. https://doi.org/10.1016/j.aei.2012.05.001.

Chu, B., K. Jung, M.-T. Lim, and D. Hong. 2013. "Robot-based construction automation: An application to steel beam assembly (Part I)." *Autom. Constr.* 32 (Jul): 46–61. https://doi.org/10.1016/j.autcon.2012.12.016.

Chung, J., S. H. Lee, B.-J. Yi, and W. K. Kim. 2010. "Implementation of a foldable 3-DOF master device to a glass window panel fitting task." *Autom. Constr.* 19 (7): 855–866. https://doi.org/10.1016/j.autcon.2010.05.004.

Dong, X. S., X. Wang, C. Daw, and K. Ringen. 2011. "Chronic diseases and functional limitations among older construction workers in the United States: A 10-year follow-up study." *J. Occup. Environ. Med.* 53 (4): 372–380. https://doi.org/10.1097/JOM.0b013e3182122286.

Dyrstad, J. S., and J. R. Mathiassen. 2017. "Grasping virtual fish: A step towards robotic deep learning from demonstration in virtual reality." In *Proc., 2017 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 1181–1187. New York: IEEE.

Edmonds, M., F. Gao, X. Xie, H. Liu, S. Qi, Y. Zhu, B. Rothrock, and S.-C. Zhu. 2017. "Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles." In *Proc., 2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 3530–3537. New York: IEEE.

Edwards, A. D., H. Sahni, Y. Schroecker, and C. L. Isbell. 2019. "Imitating latent policies from observation." Preprint, submitted May 21, 2018. https://arxiv.org/abs/1805.07914.

Feng, C., and V. R. Kamat. 2013. "Plane registration leveraged by global constraints for context-aware AEC applications." *Comput.-Aided Civ. Infrastruct. Eng.* 28 (5): 325–343. https://doi.org/10.1111/j.1467-8667.2012.00795.x.

Feng, C., Y. Xiao, A. Willette, W. McGee, and V. R. Kamat. 2015. "Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites." *Autom. Constr.* 59 (Nov): 128–138. https://doi.org/10.1016/j.autcon.2015.06.002.

Finn, C., T. Yu, T. Zhang, P. Abbeel, and S. Levine. 2017. "One-shot visual imitation learning via meta-learning." Preprint, submitted September 14, 2018. https://arxiv.org/abs/1709.04905.

Fischler, M. A., and R. C. Bolles. 1981. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography." *Commun. ACM* 24 (6): 381–395. https://doi.org/10.1145/358669.358692.

Fitzgerald, T., K. McGreggor, B. Akgun, A. Thomaz, and A. Goel. 2015. "Visual case retrieval for interpreting skill demonstrations." In *Proc., Int. Conf. on Case-Based Reasoning (ICCBR)*, edited by E. Hüllermeier and M. Minor, 119–133. Cham, Switzerland: Springer.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. "Generative adversarial networks." In *Proc., Int. Conf. on Neural Information Processing Systems (NIPS), NIPS'14*, 2672–2680. Cambridge, MA: MIT Press.

Grytnes, R., M. Grill, A. Pousette, M. Törner, and K. J. Nielsen. 2018. "Apprentice or student? The structures of construction industry vocational education and training in Denmark and Sweden and their possible consequences for safety learning." *Vocations Learn.* 11 (1): 65–87. https://doi.org/10.1007/s12186-017-9180-0.

Ho, J., and S. Ermon. 2016. "Generative adversarial imitation learning." In *Proc., Int. Conf. on Neural Information Processing Systems (NIPS), NIPS'16*, 4572–4580. Red Hook, NY: Curran Associates.

Ijspeert, A. J., J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. 2013. "Dynamical movement primitives: Learning attractor models for motor behaviors." *Neural Comput.* 25 (2): 328–373. https://doi.org/10.1162/NECO_a_00393.

Jaquier, N., D. Ginsbourger, and S. Calinon. 2019. "Learning from demonstration with model-based Gaussian process." Preprint, submitted October 11, 2019. https://arxiv.org/abs/1910.05005.

Jung, K., B. Chu, and D. Hong. 2013. "Robot-based construction automation: An application to steel beam assembly (Part II)." *Autom. Constr.* 32 (Jul): 62–79. https://doi.org/10.1016/j.autcon.2012.12.011.

Kam, H. R., S.-H. Lee, T. Park, and C.-H. Kim. 2015. "RViz: A toolkit for real domain data visualization." *Telecommun. Syst.* 60 (2): 337–345. https://doi.org/10.1007/s11235-015-0034-5.

Khansari-Zadeh, S. M., and A. Billard. 2011. "Learning stable nonlinear dynamical systems with gaussian mixture models." *IEEE Trans. Rob.* 27 (5): 943–957. https://doi.org/10.1109/TRO.2011.2159412.

Khoshnevis, B. 2004. "Automated construction by contour crafting—Related robotics and information technologies." *Autom. Constr.* 13 (1): 5–19. https://doi.org/10.1016/j.autcon.2003.08.012.

King, N., M. Bechthold, A. Kane, and P. Michalatos. 2014. "Robotic tile placement: Tools, techniques and feasibility." *Autom. Constr.* 39 (Apr): 161–166. https://doi.org/10.1016/j.autcon.2013.08.014.

Kinose, A., and T. Taniguchi. 2020. "Integration of imitation learning using GAIL and reinforcement learning using task-achievement rewards via probabilistic graphical model." *Adv. Rob.* 34 (16): 1055–1067. https://doi.org/10.1080/01691864.2020.1778521.

Koenig, N., and A. Howard. 2004. "Design and use paradigms for Gazebo, an open-source multi-robot simulator." In *Proc., 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2149–2154. New York: IEEE.

Koganti, N., A. Rahman Hafiz Abdul Ghani, Y. Iwasawa, K. Nakayama, and Y. Matsuo. 2018. "Virtual reality as a user-friendly interface for learning from demonstrations." In *Proc., Extended Abstracts of the 2018 CHI Conf. on Human Factors in Computing Systems, CHI EA '18*, 1–4. New York: Association for Computing Machinery.

Kormushev, P., S. Calinon, and D. G. Caldwell. 2011. "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input." *Adv. Rob.* 25 (5): 581–603. https://doi.org/10.1163/016918611X558261.

Kukliński, K., K. Fischer, I. Marhenke, F. Kirstein, V. Maria, D. Sølvason, N. Krüger, and T. R. Savarimuthu. 2014. "Teleoperation for learning by demonstration: Data glove versus object manipulation for intuitive robot control." In *Proc., 2014 16th Int. Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 346–351. New York: IEEE.

Liang, C.-J., V. Kamat, and C. Menassa. 2019a. "Teaching robots to perform construction tasks via learning from demonstration." In *Proc., Int. Symp. on Automation and Robotics in Construction (ISARC)*, 1305–1311. Oulu, Finland: International Association for Automation and Robotics in Construction.

Liang, C.-J., V. R. Kamat, and C. C. Menassa. 2020a. "Teaching robots to perform quasi-repetitive construction tasks through human demonstration." *Autom. Constr.* 120 (Dec): 103370. https://doi.org/10.1016/j.autcon.2020.103370.

Liang, C.-J., S.-C. Kang, and M.-H. Lee. 2017. "RAS: A robotic assembly system for steel structure erection and assembly." *Int. J. Intell. Rob. Appl.* 1 (4): 459–476. https://doi.org/10.1007/s41315-017-0030-x.

Liang, C.-J., K. M. Lundeen, W. McGee, C. C. Menassa, S. Lee, and V. R. Kamat. 2019b. "A vision-based marker-less pose estimation system for articulated construction robots." *Autom. Constr.* 104 (Aug): 80–94. https://doi.org/10.1016/j.autcon.2019.04.004.

Liang, C.-J., W. McGee, C. Menassa, and V. Kamat. 2020b. "Bi-directional communication bridge for state synchronization between digital twin simulations and physical construction robots." In *Proc., Int. Symp. on Automation and Robotics in Construction (ISARC)*, 1480–1487. Oulu, Finland: International Association for Automation and Robotics in Construction.

Liang, C.-J., X. Wang, V. R. Kamat, and C. C. Menassa. 2021. "Human-robot collaboration in construction: Classification and research trends." *J. Constr. Eng. Manage.* 147 (10): 03121006. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002154.

Liu, Y., A. Gupta, P. Abbeel, and S. Levine. 2018. "Imitation from observation: Learning to imitate behaviors from raw video via context translation." In *Proc., 2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1118–1125. New York: IEEE.

Luebbers, M. B., C. Brooks, M. J. Kim, D. Szafir, and B. Hayes. 2019. "Augmented reality interface for constrained learning from demonstration." In *Proc., 2nd Int. Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 11–14. New York: IEEE.

Lundeen, K. M., V. R. Kamat, C. C. Menassa, and W. McGee. 2017. "Scene understanding for adaptive manipulation in robotized construction work." *Autom. Constr.* 82 (Oct): 16–30. https://doi.org/10.1016/j.autcon.2017.06.022.

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

Maeda, G. J., G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters. 2017. "Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks." *Auton. Robots* 41 (3): 593–612. https://doi.org/10.1007/s10514-016-9556-2.

Mandlekar, A., D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese. 2020. "Human-in-the-loop imitation learning using remote teleoperation." Preprint, submitted December 12, 2020. https://arxiv.org/abs/2012.06733.

Markley, F. L., Y. Cheng, J. L. Crassidis, and Y. Oshman. 2007. "Averaging quaternions." *J. Guidance Control Dyn.* 30 (4): 1193–1197. https://doi.org/10.2514/1.28949.

Martínez-Salvador, B., M. Pérez-Francisco, and A. P. Del Pobil. 2003. "Collision detection between robot arms and people." *J. Intell. Rob. Syst.* 38 (1): 105–119. https://doi.org/10.1023/A:1026252228930.

Merel, J., Y. Tassa, D. TB, S. Srinivasan, J. Lemmon, Z. Wang, G. Wayne, and N. Heess. 2017. "Learning human behaviors from motion capture by adversarial imitation." Preprint, submitted July 7, 2017. https://arxiv.org/abs/1707.02201.

Mohseni-Kabir, A., C. Rich, S. Chernova, C. L. Sidner, and D. Miller. 2015. "Interactive hierarchical task learning from a single demonstration." In *Proc., 10th Annual ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI), HRI '15*, 205–212. New York: Association for Computing Machinery.

Molfino, R. M., R. P. Razzoli, and M. Zoppi. 2008. "Autonomous drilling robot for landslide monitoring and consolidation." *Autom. Constr.* 17 (2): 111–121. https://doi.org/10.1016/j.autcon.2006.12.004.

Nagata, M., N. Baba, H. Tachikawa, I. Shimizu, and T. Aoki. 1997. "Steel frame welding robot systems and their application at the construction site." *Comput.-Aided Civ. Infrastruct. Eng.* 12 (1): 15–30. https://doi.org/10.1111/0885-9507.00043.

Nair, A., D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine. 2017. "Combining self-supervised learning and imitation for vision-based rope manipulation." In *Proc., 2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2146–2153. New York: IEEE.

Ng, A. Y., and S. J. Russell. 2000. "Algorithms for inverse reinforcement learning." In *Proc., Int. Conf. on Machine Learning (ICML), ICML '00*, 663–670. San Francisco: Morgan Kaufmann Publishers.

Olson, E. 2011. "AprilTag: A robust and flexible visual fiducial system." In *Proc., IEEE Int. Conf. on Robotics and Automation (ICRA)*, 3400–3407. New York: IEEE.

Özaslan, T., G. Loianno, J. Keller, C. J. Taylor, and V. Kumar. 2018. "Spatio-temporally smooth local mapping and state estimation inside generalized cylinders with micro aerial vehicles." *IEEE Rob. Autom. Lett.* 3 (4): 4209–4216. https://doi.org/10.1109/LRA.2018.2861888.

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

Pastor, P., H. Hoffmann, T. Asfour, and S. Schaal. 2009. "Learning and generalization of motor skills by learning from demonstration." In *Proc., 2009 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 763–768. New York: IEEE.

Pavse, B. S., F. Torabi, J. Hanna, G. Warnell, and P. Stone. 2020. "RIDM: Reinforced inverse dynamics modeling for learning from a single observed demonstration." *IEEE Rob. Autom. Lett.* 5 (4): 6262–6269. https://doi.org/10.1109/LRA.2020.3010750.

Quigley, M., B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Y. Ng. 2009. "ROS: An open-source robot operating system." In *Proc., IEEE Int. Conf. on Robotics and Automation (ICRA)*, 5. New York: IEEE.

Ravichandar, H., S. R. Ahmadzadeh, M. A. Rana, and S. Chernova. 2019. "Skill acquisition via automated multi-coordinate cost balancing." In *Proc., 2019 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 7776–7782. New York: IEEE.

Ravichandar, H., A. S. Polydoros, S. Chernova, and A. Billard. 2020. "Recent advances in robot learning from demonstration." *Annu. Rev. Control Rob. Auton. Syst.* 3 (1): 297–330. https://doi.org/10.1146/annurev-control-100819-063206.

Schulman, J., S. Levine, P. Abbeel, M. Jordan, and P. Moritz. 2015. "Trust region policy optimization." Preprint, submitted February 19, 2015. https://arxiv.org/abs/1502.05477.

Schwenkel, L., and M. Guo. 2019. "Optimizing sequences of probabilistic manipulation skills learned from demonstration." In Vol. 100 of *Proc., Conf. on Robot Learning (CoRL)*. Brookline, MA: Microtome Publishing.

Sermanet, P., C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain. 2018. "Time-contrastive networks: Self-supervised learning from video." In *Proc., 2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1134–1141. New York: IEEE.

Skoglund, A., B. Iliev, and R. Palm. 2010. "Programming-by-demonstration of reaching motions—A next-state-planner approach." *Rob. Auton. Syst.* 58 (5): 607–621. https://doi.org/10.1016/j.robot.2009.12.003.

Song, J., Q. Chen, and Z. Li. 2021a. "A peg-in-hole robot assembly system based on Gauss mixture model." *Rob. Comput. Integr. Manuf.* 67 (Feb): 101996. https://doi.org/10.1016/j.rcim.2020.101996.

Song, R., F. Li, W. Quan, X. Yang, and J. Zhao. 2021b. "Skill learning for robotic assembly based on visual perspectives and force sensing." *Rob. Auton. Syst.* 135 (Jan): 103651. https://doi.org/10.1016/j.robot.2020.103651.

Torabi, F., G. Warnell, and P. Stone. 2019. "Recent advances in imitation learning from observation." In *Proc., Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 6325–6331. Shanghai, China: International Joint Conferences on Artificial Intelligence.

Tsuruta, T., K. Miura, and M. Miyaguchi. 2019. "Mobile robot for marking free access floors at construction sites." *Autom. Constr.* 107 (Nov): 102912. https://doi.org/10.1016/j.autcon.2019.102912.

Vantyghem, G., W. De Corte, E. Shakour, and O. Amir. 2020. "3D printing of a post-tensioned concrete girder designed by topology optimization." *Autom. Constr.* 112 (Apr): 103084. https://doi.org/10.1016/j.autcon.2020.103084.

Victores, J. G., S. Martínez, A. Jardón, and C. Balaguer. 2011. "Robot-aided tunnel inspection and maintenance system by vision and proximity sensor integration." *Autom. Constr.* 20 (5): 629–636. https://doi.org/10.1016/j.autcon.2010.12.005.

Wang, X., C.-J. Liang, C. Menassa, and V. Kamat. 2021. "Interactive and immersive process-level digital twin for collaborative human–robot construction work." *J. Comput. Civ. Eng.* 35 (6): 04021023. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000988.

Xiao, Y., Y. Taguchi, and V. R. Kamat. 2018. "Coupling point cloud completion and surface connectivity relation inference for 3D modeling of indoor building environments." *J. Comput. Civ. Eng.* 32 (5): 04018033. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000776.

Xu, L., C. Feng, V. R. Kamat, and C. C. Menassa. 2019. "An occupancy grid mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments." *Autom. Constr.* 104 (Aug): 230–245. https://doi.org/10.1016/j.autcon.2019.04.011.

Xu, L., C. Feng, V. R. Kamat, and C. C. Menassa. 2020. "A scene-adaptive descriptor for visual SLAM-based locating applications in built environments." *Autom. Constr.* 112 (Apr): 103067. https://doi.org/10.1016/j.autcon.2019.103067.

Yang, Y.-Y., C.-M. Chang, and S.-C. Kang. 2018. "Framework of automated beam assembly and disassembly system for temporary bridge structures." In *Proc., Int. Symp. on Automation and Robotics in Construction (ISARC)*, 230–235. Oulu, Finland: International Association for Automation and Robotics in Construction.

Yousefizadeh, S., J. de Dios Flores Mendez, and T. Bak. 2019. "Trajectory adaptation for an impedance controlled cooperative robot according to an operator's force." *Autom. Constr.* 103 (Jul): 213–220. https://doi.org/10.1016/j.autcon.2019.01.006.

Yu, S.-N., B.-G. Ryu, S.-J. Lim, C.-J. Kim, M.-K. Kang, and C.-S. Han. 2009. "Feasibility verification of brick-laying robot using manipulation trajectory and the laying pattern optimization." *Autom. Constr.* 18 (5): 644–655. https://doi.org/10.1016/j.autcon.2008.12.008.

Yu, T., D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. 2019. "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning." Preprint, submitted October 24, 2019. https://arxiv.org/abs/1910.10897.

Zahedi, E., F. Khosravian, W. Wang, M. Armand, J. Dargahi, and M. Zadeh. 2020. "Towards skill transfer via learning-based guidance in human-robot interaction: An application to orthopaedic surgical drilling skill." *J. Intell. Rob. Syst.* 98 (3): 667–678. https://doi.org/10.1007/s10846-019-01082-2.

Zhang, T., Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. 2018. "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation." In *Proc., 2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 5628–5635. New York: IEEE.

Zhao, W., J. P. Queralta, and T. Westerlund. 2020. "Sim-to-real transfer in deep reinforcement learning for robotics: A survey." In *Proc., 2020 IEEE Symp. Series on Computational Intelligence (SSCI)*, 737–744. New York: IEEE.

Zhu, Z., and H. Hu. 2018. "Robot learning from demonstration in robotic assembly: A survey." *Robotics* 7 (2): 17. https://doi.org/10.3390/robotics7020017.