Prediction-based Path Planning for Safe and Efficient Human-Robot Collaboration in Construction via Deep Reinforcement Learning

Jiannan Cai, Ph.D., A.M.ASCE,[1*] Ao Du, Ph.D., A.M.ASCE,[2]

Xiaoyun Liang, S.M.ASCE[3], and Shuai Li, Ph.D., A.M.ASCE[4]

[1]School of Civil & Environmental Engineering, and Construction Management, The University of Texas at San Antonio. One UTSA Circle, San Antonio, TX 78249. Email: jiannan.cai@utsa.edu
[2]School of Civil & Environmental Engineering, The University of Texas at San Antonio. One UTSA Circle, San Antonio, TX 78249. Email: ao.du@utsa.edu
[3]School of Civil & Environmental Engineering, and Construction Management, The University of Texas at San Antonio. One UTSA Circle, San Antonio, TX 78249. Email: xiaoyun.liang@utsa.edu
[4]Department of Civil and Environmental Engineering, The University of Tennessee, Knoxville. 851 Neyland Drive, Knoxvill, TN 37902. Email: sli48@utk.edu
* Corresponding author

Abstract

Robotics has attracted broad attention as an emerging technology in construction to help workers with repetitive, physically demanding, and dangerous tasks, thus improving productivity and safety. Under the new era of human-robot co-existence and collaboration in dynamic and complex workspaces, it is critical for robots to navigate to the targets efficiently without colliding with moving workers. This study proposes a new deep reinforcement learning (DRL)-based robot path planning method that integrates the predicted movements of construction workers, to achieve safe and efficient human-robot collaboration in construction. First, an uncertainty-aware long short-term memory network is developed to predict the movements of construction workers and associated uncertainties. Second, a DRL framework is formulated, where predicted movements of construction workers are innovatively integrated into the state space and the computation of the reward function. By incorporating predicted trajectories in addition to current locations, the proposed method enables proactive planning such that the robot could better adapt to human movements, thus ensuring both safety and efficiency. The proposed method is demonstrated and evaluated using simulations generated based on real construction scenarios. The results show that prediction-based DRL path planning achieves a 100% success rate (with a total of 10,000 episodes) for robots to achieve the destination along the near-shortest path. Furthermore, it reduces the collision rate with moving workers by 23% compared to the conventional DRL method which does not consider predicted information.

Introduction

The construction industry faces longstanding challenges including low productivity, high rates of injuries and facilities, as well as workforce aging, and labor shortage. Despite the high labor cost, construction productivity remained stagnant during the past decades, whereas that in manufacturing has nearly doubled (McKinsey&Company et al. 2015). Worker fatalities in the construction sector continuously account for 20% of total fatalities, highest among all private industries (OSHA 2020), and another 46% of construction workers suffer from work-related musculoskeletal disorder injuries (Dong et al. 2019). In addition, more than 20% of construction workers are aged 55 and older (The Center for Construction Research and Training 2018), with 430,000 more workers needed to fill the vacancy (Associated Builders and Contractors 2021). Under such situation, construction automation and robotics has emerged as a promising solution to assist in physically demanding and dangerous work, and has been introduced in various operations, such as earthwork (ASI 2019), laying bricks (Madsen 2019), and site inspection (Jacob-Loyola et al. 2021). The co-existence of and collaboration between human workers and robots may lead to potential collision risks, which requires the robots to proactively plan their motion based on the dynamics of human workers to avoid any potential collision while ensuring collaboration efficiency.

Several studies (Chen et al. 2022; Jeong et al. 2021; Kayhani et al. 2018; Kim et al. 2003) have developed methods for robotic motion planning on construction sites, while most of them only consider site configuration and static obstacles and neglect the impact of moving workers. For the few studies that can be used with moveable obstacles (e.g., Kim et al. (2003)), current obstacle locations are incorporated without any prediction of their future movement. In dynamic and complex environments that involve large uncertainties of movements, like construction sites, it is critical to have a reliable prediction of human behavior, and integrate their anticipated movement into robot path planning to generate safe and feasible trajectories (Fridovich-Keil et al. 2020; Zhou et al. 2022). In some pilot studies, Hu et al. (2020) modeled predicted locations of construction workers as static obstacles to generate a collision-free path, which does not consider the temporal evolvement and associated uncertainty of the prediction. Another previous study by the authors (Cai et al. 2021) modeled the risk of collision between construction robots and workers based on the uncertainty-aware predicted trajectories of construction workers, which, however, has not been effectively integrated with robot path planning algorithms.

To close these gaps, by integrating deep reinforcement learning (DRL) with trajectory prediction, this study proposes a new path planning method for safe and efficient human-robot collaboration on dynamic construction sites. The novelty and contributions of the proposed method are threefold. First, predicted trajectories of construction workers are innovatively introduced in the state space, and the computation of the reward function of the DRL framework. A new reward function is designed to ensure both safety and efficiency by integrating the current locations of the robot and workers, the target location of the robot, and the predicted locations of workers. Second, the proposed method is robust to different construction scenarios with a varying number of moving obstacles (e.g., workers and other machines). Furthermore, by leveraging high-level information (e.g., pre-processed locations) instead of raw sensing data, this method can be adaptive to various construction environments with different settings of sensors. Third, the proposed method is validated using simulations generated based on real construction scenarios, and is shown to outperform the conventional DRL-based path planning in both efficiency and safety (evaluated using quantitative metrics as detailed in the "Experiments and Results" section).

Background and Review of Related Studies

Robotic applications in construction

The applications of robotics in construction have been evolving over the past years, ranging from single-task construction robots, such as brick-laying robot (Madsen 2019), rebar-tying robot (Cardno 2018), robotic excavator (ASI 2019), painting robot (Asadi et al. 2018), to general-purpose robotic platforms equipped with multiple skills (e.g., sensing, navigation, manipulation) for more flexible human-robot collaboration (Kim et al. 2021). Particularly, mobile robots (both unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV)) have attracted increasing attention and are applied to various tasks, including site inspection and progress monitoring (Asadi et al. 2020; Freimuth and König 2018; Jacob-Loyola et al. 2021; Kim et al. 2019; Lu et al. 2021), material handling and object manipulation (Asadi et al. 2021; Wang et al. 2020), site layout drawing (Dusty Robotics 2022; Tsuruta et al. 2019), etc. Teleoperation is a common approach to control robots, where, conventionally, operators directly give commands via input devices such as joystick and tablets (Khasawneh et al. 2019; Okishiba et al. 2019). Methods have also been developed to control robots from intuitive motions of workers in either real or virtual reality (VR) environments (Gong et al. 2019; Wang et al. 2021; Zhou et al. 2022). With the advances in sensing, computer vision, and artificial intelligence (AI), many studies have been dedicated to enhancing machine intelligence for automatic robot control to further reduce workers' mental load and increase productivity. Examples include vision-based navigation and object manipulation (Asadi et al. 2021; Narazaki et al. 2022; Wang et al. 2020), and brain signal-based robot control (Liu et al. 2021a; b), etc.

Most existing studies focus on achieving specific construction tasks, with an underlying assumption that the robot can navigate to the target safely and efficiently across the unstructured and dynamic jobsites using various path planning algorithms developed in the general robotics domain (detailed in next section). Although a few studies devised new algorithms for mobile robot path planning to address challenges in construction sites, e.g., uneven terrain (Jeong et al. 2021), complex structures in congested spaces (Chen et al. 2022), they mainly considered static site configuration and obstacles, while neglecting the moving workers who work on various operations simultaneously and share the workspaces with robots. To achieve safe and efficient human-robot collaboration in construction, it is critical to incorporate the anticipated dynamics of workers in robot path planning.

Path planning for mobile robots

In the field of mobile robots, path planning is an essential task and has been extensively investigated by many studies. Path planning aims to find a sequence of actions to transform robots from a start to a final position, which typically follows a hierarchical approach, i.e., a combination of global and local path planning (Xiao et al. 2022). Global path planning is used to identify a coarse route from robot's current location to the target position. Graph-based methods, e.g., Dijkstra's (Dijkstra 1959) and A* (Hart et al. 1968) algorithms, and sampling-based algorithms, e.g., Rapidly-Exploration Random Tree (Lavalle 1998)), are classical methods for global path planning. These approaches generate the path plan based on the static configuration of an environment, and if the environment changes, methods such as D* algorithm (Stentz 1994) need to be used for dynamic re-planning. Local path planning generates a detailed motion plan for execution leveraging current observation of the environment obtained from onboard sensors. Since the current location of moving obstacles can be perceived in local path planning, it can be used for real-time collision avoidance in dynamic environments. Typical methods include artificial potential field (Ge and Cui 2002) and partial swarm optimization (Min et al. 2005), which can find a local path fast and efficiently, however, can be easily stuck in the local optimum (Zhang et al. 2018).

Despite the achievements, classical path planning algorithms heavily rely on mathematical models and expert experience to model the environment (both static configuration and dynamics of moving entities), and to solve for the optimal path given performance criteria. In dynamic and complex environments, especially with uncertain movement of entities, classic methods may not be sufficient to generate a reliable path (Xiao et al. 2022). To overcome these challenges, deep learning-based methods have been developed to learn robot motion plans directly from perceptual information of the environment in an end-to-end manner. Specifically, DRL has been increasingly explored in robot path planning (Ajeil et al. 2020; Botteghi et al. 2020; Xie et al. 2017). For instance, Gao et al. (2020) designed a DRL-based path planning method for collision-free autonomous navigation, The trained model was also transferred from 2D to a complex 3D environment, demonstrating the good generalizability of the DRL model. Yan et al. (2021) proposed a DRL-based method combined with a variant of the long short-term memory (LSTM) model to generate optimal angle motions in a marine environment for unmanned surface vehicles. Wen et al. (2020) leveraged a DRL-based active simultaneous localization and mapping (SLAM) framework to achieve autonomous navigation for the mobile robot in an environment with both static and moving obstacles, however, the number of moving obstacles is limited as one in their application.

Existing studies have shown the great potential of DRL as an effective way to generate collision-free paths in complex and dynamic environments. However, there remain three knowledge gaps. First, most studies only consider the current locations of moving agents without a reliable prediction of their future behavior. Given the sensing uncertainty as well as the unexpected movement of agents in complex environments, such an approach cannot guarantee safety (Fridovich-Keil et al. 2020). A pilot study (Fridovich-Keil et al. 2020) incorporated confidence-aware motion prediction in the path planning, however, it still relies on mathematical models (i.e., hidden Markov model) to capture human dynamics by assuming people take actions following Markovian fashion. To close this gap, this study leverages an LSTM-based model to predict human trajectory and associated uncertainties and incorporate the predicted movement into the design of DRL-base path planning. Second, in DRL, the design of the reward function has a significant impact on the convergence and performance of the model, which should reflect the objective of the planning context. Most studies focus on collision-free path planning in a general mobile robot application, and cannot be readily applied to construction scenarios. In this study, a new reward function is designed to incorporate both safety and efficiency requirements in construction applications, considering the predicted trajectory of construction workers. Third, most studies model the state space using raw sensor measurements and focus on the environment with only one moving obstacle, constraining the applicability to specific robot and environment configurations. In contrast, this study leverages high-level information in the state space and integrates information of the nearest obstacle, which makes it applicable in various environments with different robot configurations and varying numbers of obstacles.

Methodology

Problem formulation

In this study, the path planning problem is modeled as a Markov Decision Process (MDP) where a mobile robot selects its action at each time step (A_t), based on the environment states (S_t), such as the locations of the robot, destination, and moving workers. After executing the action, the environment is transmitted to the next state (S_{t+1}) and the robot receives a reward (r_{t+1}). Over a given time horizon T, the goal of the robot is to learn an optimal policy $\pi(A_t|S_t)$, to find a safe and efficient path from its initial location to a known destination on construction sites with moving workers, by maximizing the expectation of cumulative discounted long-term reward R_t (Eq. 1):

$$R_t = r(S_t, A_t) + \gamma r(S_{t+1}, A_{t+1}) + \cdots + \gamma^{T-t} r(S_T, A_T) = \sum_{i=t}^{t+T} \gamma^{i-t} r(S_i, A_i) \quad (1)$$

where γ is the discount factor in the range of 0 to 1. To ensure safety and efficiency, the reward function is designed to combine both the locations of obstacles (i.e., moving workers) and the destination to motivate the robot to find the shortest possible collision-free path. In contrast to most existing studies on DRL-based robot path planning that only consider current observations of the environment, the proposed method innovatively incorporates predicted trajectories of workers using an LSTM-based prediction model (detailed in the next section) to achieve more proactive planning. Figure 1 shows the overall process.

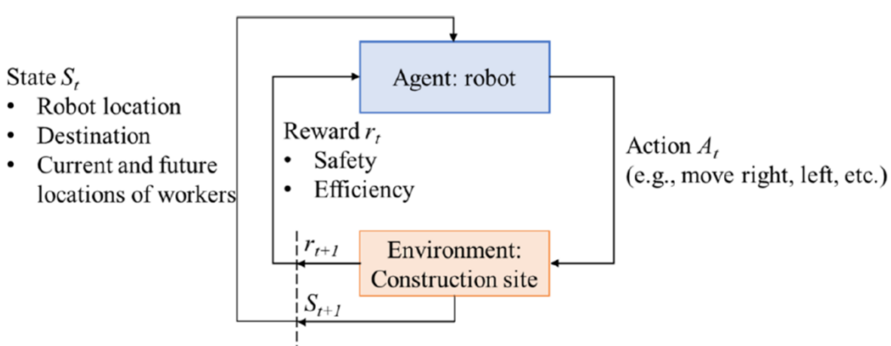

Figure 1. The MDP framework for robot-environment interaction on construction site

In this study, the construction site is discretized into a 2D grid map, following the common practice in robot path planning studies (Ajeil et al. 2020). Thus, the locations of workers, robot, and destination are represented by the grids they occupy. Discretizing the state space and the corresponding action space allows finite decisions and effective dimensionality reduction in the planning process, where there is a trade-off between precise path planning and computational efficiency. A small grid size enables the system to model the locations of each agent more accurately and to plan the robot's path more precisely with relatively high computational complexity. On the other hand, a large grid size leads to fewer states and can significantly improve computational efficiency. However, a grid is more likely to be occupied by multiple agents and results in collisions even if the distance between them is still large, which might cause unnecessary detours. In addition, coarse grids could better accommodate uncertainties: considering possible uncertainties of agents' locations (e.g., caused by sensing errors), the corresponding states are more likely to remain unchanged with coarse grids compared to fine grids.

Taking into account the above factors, as well as existing studies on construction workspace modeling (Zhang et al. 2015; Dong et al. 2018), a grid size of 0.5x0.5 m is adopted in this study. The path is represented by a sequence of 2D grid coordinates resulting from the action taken by the robot at each time step. Since this study focuses on path planning for mobile robots instead of low-level control of robot motion, only the time-series locations are considered, and it is assumed that the robot moves one grid each time step as a simplification. In addition, it is assumed that the robot has full knowledge of the environment, including its location and destination, and the locations of all workers, which could be obtained from external sensing or camera systems (e.g., Cai and Cai (2020)).

Uncertainty-aware worker trajectory prediction

In a previous study (Cai et al. 2020), an LSTM network with an encoder-decoder architecture was developed to predict future trajectories of construction entities, where the trajectory was predicted from a deterministic perspective without considering the associated uncertainty of the prediction. Such practice may pose a potential risk of robot-worker collision in the path planning problem. The above method is extended to incorporate uncertainty in this study. Inspired by Alahi et al. (2016), the trajectory is assumed to follow bivariate Gaussian distribution, and the parameters that characterize the distribution rather than absolute locations are estimated, see Figure 2.

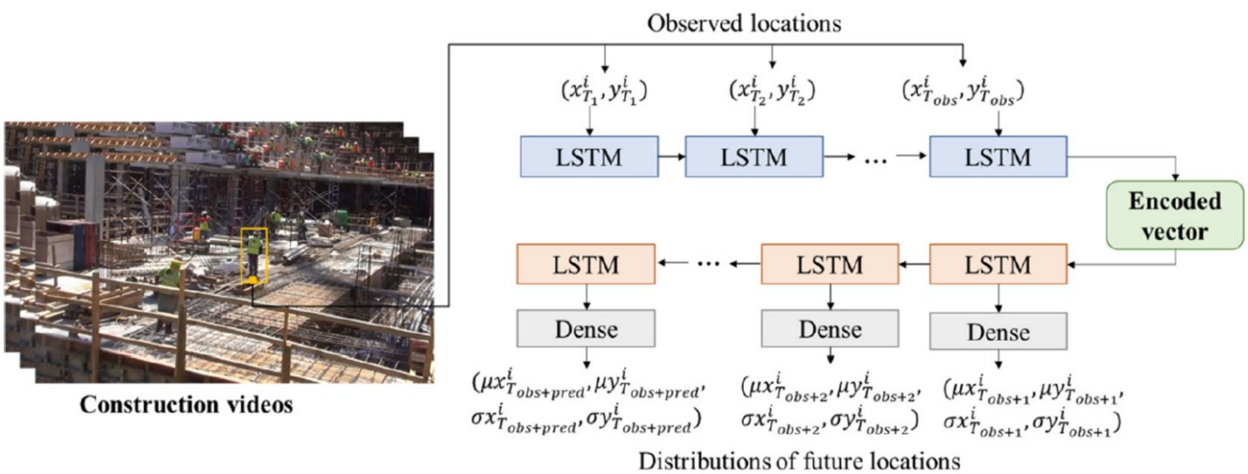

Figure 2. Uncertainty-aware LSTM-based trajectory prediction model

Specifically, the observed locations of workers are extracted from construction videos, represented as pixel coordinates of the mid-bottom point of the workers' bounding box. In this study, the bounding boxes are manually labeled to generate worker's movement data, which can also be obtained automatically via object tracking algorithms (e.g., Roberts and Golparvar-Fard (2019), Cai and Cai (2020)). Then, the observed trajectory from time T_1 to T_{obs} serve as inputs and are fed into the LSTM encoder, and the position distributions, $N(\boldsymbol{\mu}^i_t, \boldsymbol{\sigma}^i_t, \rho^i_t)$, from time T_{obs+1} to

$T_{obs+pred}$ are generated via LSTM decoder, where $\boldsymbol{\mu}_t^i = \left(\mu_x, \mu_y\right)_t^i$ is the mean, $\boldsymbol{\sigma}_t^i = \left(\sigma_x, \sigma_y\right)_t^i$ is the standard deviation, and ρ_t^i is the correlation coefficient (in this study, $\rho_t^i = 0$, assuming the movements in both directions are independent). Accordingly, the negative log-Likelihood is adopted as loss function: $L^i = -\sum_{t=T_{obs+1}}^{T_{obs+pred}} \log(P(x_t^i, y_t^i | \mu_t^i, \sigma_t^i))$. Readers are referred to Cai et al. (2020, 2021) for details of the network architecture. The resulting predicted trajectory is then included in the state space to provide additional information for robot path planning (detailed in the next section).

State space

The state space reflects the observations of the robots about the environment, which serves as the input of its decision process. In conventional studies on robot navigation and control (e.g., (Gao et al. 2020), (Yan et al. 2021)), raw sensory data (e.g., light detection and ranging (LiDAR) measurement, images from cameras) are used to model state space. In contrast, this study models state space using high-level information, including the current location and destination of robots, as well as current and predicted locations of construction workers, which can be processed from raw sensory data using various object localization and tracking algorithms (Roberts and Golparvar-Fard (2019), Cai and Cai (2020)). Through such an approach, the proposed method can be naturally extended to other systems with different sensors. As a result, at any time step *t*, state space is a 16-dimensional vector, denoted as $S^t = \left[x_r^t, y_r^t, x_d^t, y_d^t, x_o^{t,0}, y_o^{t,0}, x_o^{t,1}, y_o^{t,1}, \ldots, x_o^{t,5}, y_o^{t,5}\right]$, where,

- $[x_r^t, y_r^t]$ is the location of the robot in terms of 2D grid coordinates, varying at each time step after the robot takes an action. In practice, this information could be obtained via multiple sources, such as onboard GPS, cameras, and/or LiDAR.

- $[x_d^t, y_d^t]$ is the destination of the robot in terms of 2D grid coordinates, remaining unchanged during the decision horizon. This is reasonable because a robot is typically given a specific task with a known goal (i.e., destination) in practice.
- $[x_o^{t,0}, y_o^{t,0}, x_o^{t,1}, y_o^{t,1}, \ldots, x_o^{t,5}, y_o^{t,5}]$ represents the current location ($[x_o^{t,0}, y_o^{t,0}]$) and predicted locations ($[x_o^{t,1}, y_o^{t,1}, \ldots, x_o^{t,5}, y_o^{t,5}]$) of the robot's nearest worker. The past trajectories of workers can be observed from various sensors (e.g., cameras, radio-based sensors), based on which the future trajectories can be predicted using the method described in the previous section. In this study, locations in the next five steps are considered. It is noted that since the proposed LSTM-based uncertainty-aware model predicts the distribution of the future trajectory, the future locations in state space are sampled from the distribution. Besides, to generalize the proposed methods to environments with varying numbers of workers/obstacles, only the nearest obstacle is modeled in the state space instead of all workers on the jobsite, which is reasonable because the nearest obstacle is expected to have the largest impact on robot path in terms of collision avoidance.

In the above state space, all elements are represented in 2D grid coordinates, and have the same scale that is determined by the environment. Thus, it can be directly used as inputs for Deep *Q*-Networks (DQN) as detailed in the "DQN model" section.

Action space

Given the current and target locations in the 2D grid environment, a robot takes action to move from the current location to the target location along the grids. Existing studies typically define the action space of a mobile robot to include four discrete actions, i.e., up, down, left, and right (Xu et al. 2019; Zhang et al. 2019). In addition to the four actions, this study also introduces a fifth action, i.e., stay, to incorporate the scenario where the robot can choose to wait for workers to move away based on the predicted movement of workers. Note that, it is assumed that the robot moves at a fixed speed (one grid per time step), and thus, the updated location of the robot after it takes one specific action can be illustrated in Figure 3, which determines the transition rule of robot location.

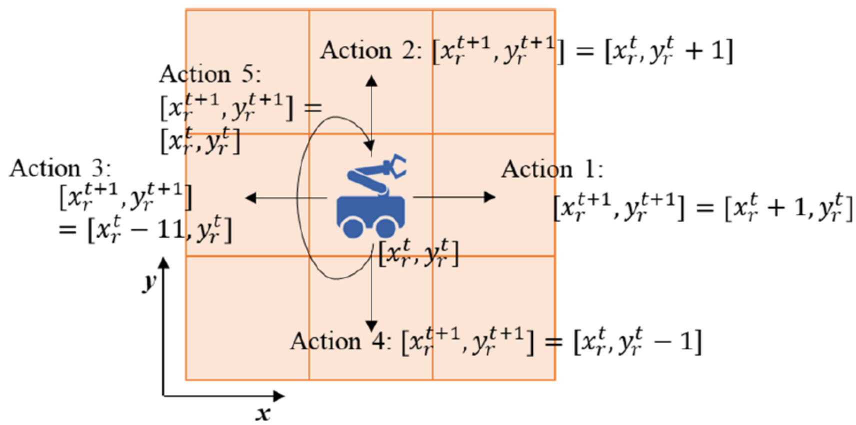

Figure 3. Updated location after taking specific action at time *t*

Reward Function

The reward function is critical to the success of DRL in solving complex tasks. In mobile robot path planning on construction sites, a robot is expected to move efficiently from its start point to its destination without collision with other workers moving on the jobsites. Therefore, the design of the reward function considers both efficiency and safety. For efficiency, three factors are incorporated: 1) at any time step, the robot should be motivated to move closer to the target position instead of moving away from the target. Therefore, after taking an action, if the distance between robot's location and desired position is smaller than their previous distance, a positive reward is given, otherwise, a negative reward (i.e., penalty) is applied. The value of this reward is denoted as r_{dist} . 2) By only applying the first reward, the robot may move aimlessly, exhibiting a tendency to move back and forth to avoid penalty while making minimum progress to the target. In this case, a negative reward should be given to prevent the aimless movement and force the robot to navigate to the target, which is denoted as $r_{aimless}$. 3) Eventually, the robot is given a large reward for reaching the target, denoted by r_{dest}.

For safety, two factors are incorporated: 1) the robot should not collide with moving workers. In this study, as both the current locations of the workers and a period of the predicted trajectory (i.e., 5-time steps) are considered, a deduction factor is applied for potential collision with the predicted positions of workers. Let r_{obs} denotes the penalty for colliding with a worker (i.e., the robot and worker are in the same grid) and γ is the deduction factor, then the penalty (negative reward) for colliding with a worker *n*-step later is denoted by $\gamma^n r_{obs}$ (no deduction for colliding at the current location). As a result, a negative reward is formulated for potential collision with future positions of workers. To avoid applying penalties repeatedly, only the earliest time when a collision may occur is considered, thus, $r_{collision} = \min\{r_{obs}, \gamma^1 r_{obs}, \dots \gamma^n r_{obs}\}$, where *n*=5 in this case. Specifically, if the robot does not collide with the worker at *t*, r_{obs} for that step is 0. In this way, the penalty is only applied to the earliest time step of a potential collision, and the later the collision, the smaller the penalty is; 2) Once the robot exceeds the boundary of the site, a large penalty, denoted as $r_{boundary}$, is applied if they go beyond the boundary to ensure the robot move within its workspace. As a result, the reward function of the proposed method is $r = r_{dist} + r_{aimless} + r_{dest} + r_{collision} + r_{boundary}$.

DQN model

The MDP problem is solved via DRL, a modern approach to overcome the limitation of traditional Q-learning by improving computational efficiency and convergence, leveraging advanced function approximators (e.g., neural networks) instead of *Q*-table (Du and Ghavidel 2022). Various DRL algorithms have been developed, including DQN (Mnih et al. 2013), Deep Deterministic Policy Gradients (DDPG) (Lillicrap et al. 2015), and Actor-Critic methods (Mnih et al. 2016). In this study, the proposed DRL-based robot path planning method is implemented based on a typical DRL algorithm, DQN, which employs neural networks to approximate the Q values. Bellman equation (Eq. 2) is adopted for value iteration in the training of the neural network $Q(s, a; \theta)$ as an action-value function approximator.

$$Q^*(s,a) = E_{s'\sim S}\left[r(s,a) + \gamma \max_{\alpha'} Q^*(s',\alpha')\right] \tag{2}$$

Where Q^* denotes the optimum action-value (Q) function, θ denotes the weights of the neural network. To achieve supervised learning for DQN, temporal difference (TD) training is adopted which introduces a target network to provide target Q values to act as the "ground truth". The target network is initialized with the same network structure and weights as the action-value network, and the weights of both networks will be updated using common stochastic gradient descent algorithms. However, the target network is updated in a delayed fashion compared to the action-value network that is updated in every training step, and such stability could also improve the convergence.

Specifically, a fully connected (FC) neural network with two hidden layers (24 nodes for each layer) is used as approximators of both action-value function and target function of the DQN (Du and Ghavidel 2022). The 16-dimensional state vector is considered as input to the neural network. The Rectified Linear Unit (ReLU) activation function is implemented for the two hidden layers and a linear activation function is implemented for the output layer, illustrated in Figure 4.

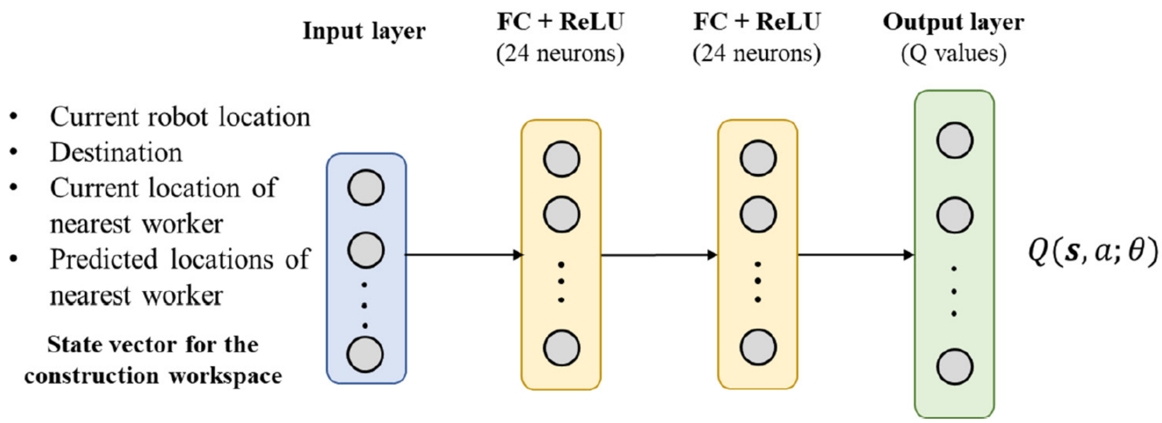

Figure 4. Architecture of DQN

Experiments and Results

Environment setup

The proposed prediction-based DRL framework for robot path planning is demonstrated and evaluated using simulations that are generated based on real construction scenarios, detailed as follows.

First, construction videos from three real construction projects (two recorded by authors, and one collected from (YouTube 2019)) were used to train the uncertainty-aware LSTM model for worker trajectory prediction, which was then used to generate simulation environments to validate the proposed method for robot path planning. The videos consist of a total of 84 workers conducting various construction activities, including site preparation, material handling, formwork, etc. The videos were taken from various angles at height, similar to the perspective of surveillance cameras. Figure 5 illustrates some sample images of the collected videos. All videos were down-sampled to 2fps, compatible with other studies (Alahi et al. 2016) on video-based human trajectory prediction. The videos were pre-processed and bounding boxes of workers were manually labeled to extract entity positions, serving as inputs of the prediction model. A total of 2,544 frames consisting of 241 trajectories with various lengths were extracted (Cai et al. 2020), where a trajectory was terminated if the worker was severely occluded by other objects. In this study, the observation duration of worker movements was 3s (i.e., 6 frames) and the prediction duration was 5s (i.e., 10 frames), following relevant studies (Alahi et al. 2016) on human trajectory prediction. Therefore, the 241 trajectories were further split into 16-frame tracks using a sliding window with a stride of 2 frames, resulting in 3640 tracks for training and testing the prediction model. Consistent with previous studies (Cai et al. 2020, 2021), the dataset was randomly split into training set (80%), validation set (10%), and testing set (10%). The network was trained with *Adam* optimizer (Kingma and Ba 2014), with a learning rate of 0.001, batch size of 20, and dropout rate of 0.5. The proposed method was implemented on a desktop with Intel i7-9700 CPU, 32GB, and NVIDIA GeForce GTX 2060 GPU using Keras library within Tensorflow platform.

Figure 5. Sample images from the dataset: (a) and (b) from two building projects photographed by authors; (c) and (d) from a hospital project (YouTube 2019)

Figure 6 visualizes an example of predicted trajectories with associated uncertainties, where red lines represent the actual trajectories of construction workers. In Figure 6, the predicted trajectories closest to the actual one have the highest probability (yellow color), while those farther away exhibit relatively low probability (blue color). This establishes a reliable basis to incorporate predicted trajectories and their associated uncertainties for robot path planning.

Figure 6. Example of predicted distributions of future locations over 10 timesteps

Second, the above construction scenario (shown in Figure 6) was selected to simulate the environment to train the DRL model for robot path planning. To establish a realistic environment, the construction site and worker locations were converted from image plane to ground plane via projective transformation (Hartley and Zisserman 2003), where the transformation matrix was estimated from known dimensions of construction equipment. Such approximation may not influence the simulation of robot path planning because the locations of different workers and the boundary of the jobsite were estimated from the same set of parameters. The construction site, after being converted to the ground plane, is a 16m x 11m area, and was further divided into a 32x22 grid map with a grid size of 0.5 m.

Third, to train a generalized DRL model, a large number of training data are needed. To achieve this, extensive simulations were conducted where construction environments with different configurations were generated based on the above construction scenario (shown in Figure 6). Specifically, in each training episode of the DRL model, 1) the initial and target locations of the robot were randomly generated within the environment (i.e., 32x22 grid map obtained in the previous step); 2) the number of workers was randomly selected, ranging from 1 to 5; 3) the initial movements of workers were randomly extracted from all trajectories in the video to reflect real construction scenarios, and their following trajectories during the training process were sampled from predicted trajectory distribution obtained from the proposed LSTM model in the first step.

Model training

In the proposed path planning method, the parameter values in the reward function, especially the relative magnitudes of different components, have a significant influence on the model performance and convergence, and should be determined considering the planning context and objective. For instance, the relative magnitude between the reward of moving towards the destination (i.e., r_{dist}) and the penalty of collision (i.e., $r_{collision}$) indicates the relative importance of efficiency and safety. For human-robot collaboration in construction, safety is the priority, and thus the penalty of collision is set to be much larger than the reward of moving towards the destination. It implies that the robot tends to take a detour to avoid potential collisions with the workers. In terms of safety, collision at an earlier time step is more urgent and severe than at more future time steps. Besides, due to the increasing uncertainty of trajectory prediction at larger prediction horizons, the discount for the penalty of collision at future time steps should be set as a large value (i.e., γ should be relatively small) to ensure that the robot can efficiently achieve the goal, and the model can converge.

For efficiency, the penalty for aimless movement ($r_{aimless}$) should be compatible with the reward (or penalty) of the robot moving close to (or far away from) the destination. Otherwise, the robot tends to move back and forth instead of moving towards the goal. The reward for achieving the destination should be the largest so that the robot has the motivation to complete the task within the planning horizon. In addition, the boundary of the workspace is considered a hard constraint, and the episode is terminated if the robot exceeds the boundary. Therefore, the penalty for exceeding the boundary ($r_{boundary}$) should be the largest so that the robot can rapidly learn the boundary states and avoid exceeding the boundary.

Given the above principles and after experiments with various combinations of values, the parameters in the reward function were determined as follows: $r_{dest} = 100$, indicating the robot is given a +100 reward for reaching the destination; $r_{dist} = \pm 1$, representing the robot receives a +1 reward or -1 penalty each step for moving close to or far away from the destination; if the distance between robot and destination remains unchanged in two steps, $r_{dist} = 0$; $r_{aimless} = -1$, indicating the robot receives a -1 penalty each step for moving aimlessly; $r_{boundary} = -20$, indicating the robot receives a -20 penalty for exceeding the boundary; $r_{collision} = \min\{r_{obs}, \gamma^1 r_{obs}, \dots \gamma^n r_{obs}\}$, where $r_{obs} = -20$ *and* $\gamma = 0.3$, representing the robot is given a -20 penalty for potential collision with workers at the current time step, and a deduction factor of 0.3 is applied for collisions at future time steps.

Exploration-exploitation strategy, i.e., the ε-greedy, was used to achieve a good balance between exploration (to explore those unvisited sequences) and exploitation (to leverage the policy that has been learned so far), where the exploration probability ε was initially set to 1.0, and exponentially decays to a minimum value of 0.01 along with the training episodes. The Adam optimizer was implemented for the neural network training with an initial learning rate of 0.01 and exponential learning rate decay. 50,000 training episodes were considered for the DRL model training. For each training episode, the process was terminated if either of the three criteria is satisfied: 1) the step reaches 100; 2) the robot reaches its destination; 3) the robot exceeds environment boundaries. The convergence of the training curve is shown in Figure 7.

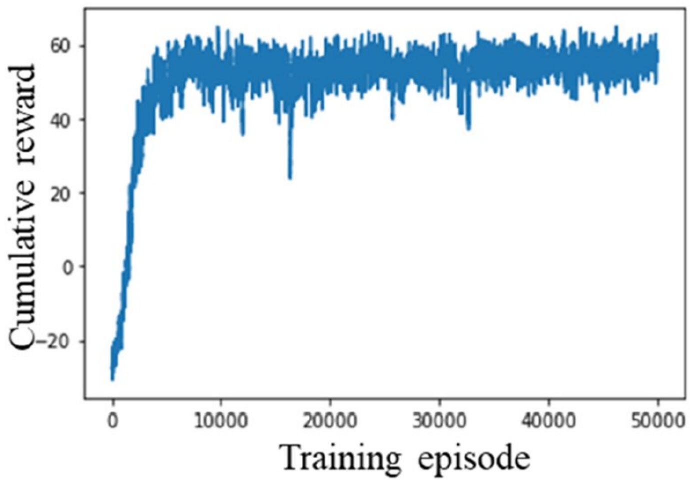

Figure 7. Training curve of the proposed DRL model

Performance of the DRL model

The trained DRL agent was further tested using 10,000 random episodes (100 steps per episode) to evaluate the learning performance. In addition, a conventional DRL agent that does not consider the predicted locations of workers was trained and tested for comparison. Specifically, in the conventional DRL method, the observation space only contained the current and target locations of the robot, and the current location of the nearest worker. The proposed reward function and corresponding parameters were also used in the conventional DRL agent for consistency, except for the exclusion of penalty for potential collision at future time steps.

Quantitative metrics were used for performance evaluation in terms of both efficiency and safety. For efficiency, two metrics were used: 1) success rate, computed as the ratio between the number of successful cases and the total number of testing episodes, where a successful case is when the robot achieves the desired destination before the end of the episode; 2) average ratio between the actual path and shortest path, where for each episode, the actual path is learned from the DRL agent, and the shortest path is computed as $abs(x_r^0 - x_d) + abs(y_r^0 - y_d)$.

For safety, collision rate was used for evaluation, where one collision case is counted if the robot collides with a worker (at the current time step) at least once during an episode. To examine the efficacy of the proposed model in proactive planning, collision rates at anticipatory locations of workers were also examined, the smaller of which offers a more reliable and safer path and can accommodate potential sensing uncertainties in practice. Table 1 shows the results of the above metrics for both models.

Table 1 shows that the proposed prediction-based DRL model outperforms the conventional model that does not consider the predicted locations of workers in both efficiency and safety. The proposed method achieves a 100% success rate with an average ratio between actual and shortest paths close to 1, which means that the robot can successfully reach the destination in all 10,000 episodes along the near-shortest path. It represents the high efficiency of the planned paths. Figure 8 shows the visualization of the robot's planned path in one example, where the site was zoomed to the central area with moving workers and the robot.

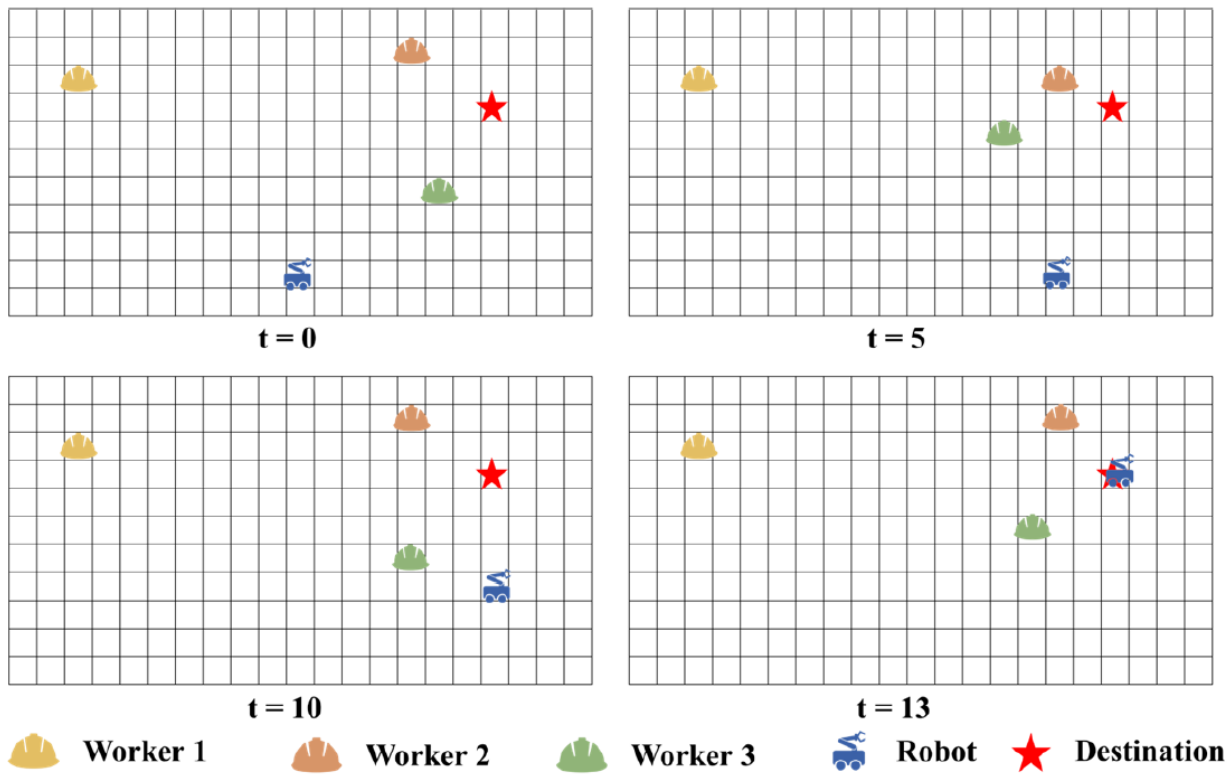

Figure 8. Example of robot's path learned using proposed DRL model

In contrast, the robot agent that is learned using the conventional DRL model fails to reach the destination in 2.64% of total cases (or 264 cases). In almost all unsuccessful cases, the robot ends up moving back and forth without heading to the destination, which has never occurred in the robot agent learned using the proposed model. One possible reason is that by introducing predicted locations of workers and associated penalties for future collision, the robot agent has more knowledge not only of the current situation but also of the situation in near future. It provides the robot with more information and incentive to make decisions to move to the destination. The less chance for the robot to collide with workers (both at the current time and in the near future), as shown in Table 1, also proves the advantage of the proposed method in terms of safety.

Table. 1 Performance comparison between proposed and conventional DRL models (a total of 10,000 episodes)

Model	Success rate	Average ratio between actual and shortest paths (ideally 1)	Collision rate (at current location)	Collision rate (at 1-step predicted location)	Collision rate (at 2-step predicted location)	Collision rate (at 3-step predicted location)	Collision rate (at 4-step predicted	Collision rate (at 5-step predicted
Conventional DRL (without prediction)	97.36%	1.08 (1.005)	3.34%	3.46%	3.43%	3.22%	3.27%	3.30%
Prediction-based DRL (proposed in this study)	100%	1.0003	2.56%	2.47%	2.54%	2.54%	2.50%	2.58%

Note: 1) the value in the bracket is the average ratio between actual and shortest paths for successful episodes using the conventional DRL model.; 2) the scenarios when the workers occupy the same cell with the destination were excluded when determining the number of collisions

Conclusions

This study proposes a new prediction-enabled path planning method for construction robots considering the predicted trajectories of onsite workers. Specifically, an LSTM-based model is developed to predict the trajectory distribution of workers instead of absolute trajectories to incorporate the uncertainty. DRL framework is used for path planning where the predicted trajectory is innovatively introduced to observation space and integrated into the computation of reward function to ensure both safety and efficiency. Extensive simulations generated from real construction scenarios are conducted to validate the proposed framework, which is also compared with conventional DRL-based path planning that does not consider prediction information. The results show that the proposed method generates safer and more efficient paths: 1) it achieves a 100% success rate for the robot to move to the destination along the near-shortest path; 2) it reduces the collision rate with moving workers by 23% compared to conventional DRL method. By leveraging high-level information (e.g., pre-processed locations) instead of raw sensing data, this method is adaptive to various robotics and application scenarios with different settings of sensors. In addition, by incorporating location information of the nearest neighbor instead of all obstacles in the environment, the proposed method is robust to scenarios with a varying number of moving obstacles and can be extended to other environments (e.g., search and resecure, manufacturing).

The proposed framework is critical for human-robot collaboration on unstructured and dynamic construction sites because it allows proactive adjustment of robot's movement to avoid collisions without interrupting normal operation. It endows the mobile robot with the capability to automatically generate a safe and efficient path considering site dynamics, which is an essential prerequisite of multiple robotic construction operations, such as material handling, site inspection, etc. With intelligent path planning that takes into account anticipatory movements of site entities, it is safer and more feasible to introduce construction robots to on-site operations and share the workspaces with other workers who do not interact with them directly on a single task. Furthermore, in terms of human-robot collaborative tasks, human operators are released to only provide high-level commands that are related to the intended tasks (e.g., assign target areas to be inspected or materials to be handled), without having to specify a detailed path, which will significantly reduce the mental load of workers and increase the productivity.

There remain some limitations to be addressed in the future study. First, despite the high successful rate, worker trajectories in only one real-world scenario (captured in construction video) were extracted and augmented by adding randomness in the simulated environments. In future study, more diverse scenarios will be generated to train the agent, which will be further tested on both robotic simulation and real robot implementation. Second, as this study focuses on moving workers with uncertain predicted trajectories, static obstacles are not modeled in the framework. However, the proposed framework can be easily extended to include static obstacles by integrating locations of static obstacles in the observation space and the computation of reward function, similar to moving obstacles except that only fixed locations instead of time-series locations need to be considered. Third, since the focus of this study is high-level robot path planning, the speed of the robot is assumed constant as a simplification. Such assumption could be relaxed in future study by incorporating speed into action space to accommodate different requirements on robot's motion in various applications.

Data Availability Statement

Some data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request, including 1) data and codes for generating DRL environment, and training and testing DRL path planning model, 2) codes for training and testing LSTM-based trajectory prediction model.

Acknowledgments

This research was partially funded by the University of Texas at San Antonio (UTSA), Office of the Vice President for Research, Economic Development, and Knowledge Enterprise, and the U.S. National Science Foundation (NSF) via Grants 2138514 and 2129003. The authors gratefully acknowledge UTSA's and NSF's supports.

References

Ajeil, F. H., I. K. Ibraheem, A. T. Azar, and A. J. Humaidi. 2020. "Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments." *Sensors (Switzerland)*, 20 (7). https://doi.org/10.3390/s20071880.

Alahi, A., K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. 2016. "Social LSTM: Human trajectory prediction in crowded spaces." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 961–971.

https://doi.org/10.1109/CVPR.2016.110.

Asadi, E., B. Li, and I. M. Chen. 2018. "Pictobot: A Cooperative Painting Robot for Interior Finishing of Industrial Developments." *IEEE Robotics and Automation Magazine*, 25 (2): 82–94. https://doi.org/10.1109/MRA.2018.2816972.

Asadi, K., V. R. Haritsa, K. Han, and J.-P. Ore. 2021. "Automated Object Manipulation Using Vision-Based Mobile Robotic System for Construction Applications." *Journal of Computing in Civil Engineering*, 35 (1): 04020058. https://doi.org/10.1061/(asce)cp.1943-5487.0000946.

Asadi, K., A. Kalkunte Suresh, A. Ender, S. Gotad, S. Maniyar, S. Anand, M. Noghabaei, K. Han, E. Lobaton, and T. Wu. 2020. "An integrated UGV-UAV system for construction site data collection." *Automation in Construction*, 112. https://doi.org/10.1016/j.autcon.2019.103068.

ASI. 2019. "Robotic Excavators." Accessed October 12, 2019. https://www.asirobots.com/mining/excavator/.

Associated Builders and Contractors. 2021. "Construction Spending and Employment: History and Forecast Terms and Sources." Accessed June 2, 2021. https://www.abc.org/Portals/1/News Releases/explainer-2021.pdf?ver=2021-03-23-162431-727.

Botteghi, N., B. Sirmacek, K. A. A. Mustafa, M. Poel, and S. Stramigioli. 2020. "On Reward Shaping for Mobile Robot Navigation: A Reinforcement Learning and SLAM Based Approach." *arXiv preprint arXiv:2002.04109*.

Cai, J., and H. Cai. 2020. "Robust Hybrid Approach of Vision-Based Tracking and Radio-Based Identification and Localization for 3D Tracking of Multiple Construction Workers." *Journal of Computing in Civil Engineering*, 34 (4): 04020021. American Society of Civil Engineers. https://doi.org/10.1061/(asce)cp.1943-5487.0000901.

Cai, J., A. Du, and S. Li. 2021. “Prediction-Enabled Collision Risk Estimation for Safe Human-Robot Collaboration on Unstructured and Dynamic Construction Sites.” *Computing in Civil Engineering 2021*, 34–41. https://doi.org/10.1061/9780784483893.005.

Cai, J., Y. Zhang, L. Yang, H. Cai, and S. Li. 2020. “A context-augmented deep learning approach for worker trajectory prediction on unstructured and dynamic construction sites.” *Advanced Engineering Informatics*, 46. https://doi.org/10.1016/j.aei.2020.101173.

Cardno, C. A. 2018. “Robotic Rebar-Tying System Uses Artificial Intelligence.” *Civil Engineering Magazine Archive*, 88 (1): 38–39. https://doi.org/10.1061/ciegag.0001260.

Chen, Z., K. Chen, C. Song, X. Zhang, J. C. P. Cheng, and D. Li. 2022. “Global path planning based on BIM and physics engine for UGVs in indoor environments.” *Automation in Construction*, 139: 104263. https://doi.org/10.1016/j.autcon.2022.104263.

Dijkstra, E. W. 1959. “A note on two problems in connexion with graphs.” *Numerische Mathematik*, 1 (1): 269–271. https://doi.org/10.1007/BF01386390.

Dong, X. S., E. Betit, A. M. Dale, G. Barlet, and Q. Wei. 2019. *Trends of Musculoskeletal Disorders and Interventions in the Construction Industry*. https://www.cpwr.com/wp-content/uploads/2020/06/Quarter3-QDR-2019.pdf.

Du, A., and A. Ghavidel. 2022. “Parameterized deep reinforcement learning-enabled maintenance decision-support and life-cycle risk assessment for highway bridge portfolios.” *Structural Safety*, 97: 102221. https://doi.org/10.1016/j.strusafe.2022.102221.

Dusty Robotics. 2022. “Build better with BIM-drien layout.” Accessed June 2, 2022. https://www.dustyrobotics.com/.

Freimuth, H., and M. König. 2018. “Planning and executing construction inspections with unmanned aerial vehicles.” *Automation in Construction*, 96: 540–553.

https://doi.org/10.1016/j.autcon.2018.10.016.

Fridovich-Keil, D., A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin. 2020. "Confidence-aware motion prediction for real-time collision avoidance." *The International Journal of Robotics Research*, 39 (2–3): 250–265. SAGE Publications Sage UK: London, England. https://doi.org/10.1177/0278364919859436.

Gao, J., W. Ye, J. Guo, and Z. Li. 2020. "Deep reinforcement learning for indoor mobile robot path planning." *Sensors (Switzerland)*, 20 (19): 1–15. https://doi.org/10.3390/s20195493.

Ge, S. S., and Y. J. Cui. 2002. "Dynamic motion planning for mobile robots using potential field method." *Autonomous Robots*, 13 (3): 207–222. https://doi.org/10.1023/A:1020564024509.

Gong, D., Y. Wang, J. Yu, and G. Zuo. 2019. "Motion mapping from a human arm to a heterogeneous excavator-like robotic arm for intuitive teleoperation." *2019 IEEE International Conference on Real-Time Computing and Robotics, RCAR 2019*, 493–498. https://doi.org/10.1109/RCAR47638.2019.9044131.

Hart, P. E., N. J. Nilsson, and B. Raphael. 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." *IEEE Transactions on Systems Science and Cybernetics*, 4 (2): 100–107. https://doi.org/10.1109/TSSC.1968.300136.

Hartley, R., and A. Zisserman. 2003. *Multiple view geometry in computer vision*. Cambridge university press. ISBN:0521540518.

Hu, D., S. Li, J. Cai, and Y. Hu. 2020. "Toward Intelligent Workplace: Prediction-Enabled Proactive Planning for Human-Robot Coexistence on Unstructured Construction Sites." *Winter Simulation Conference 2020*, 2412–2423. IEEE. https://doi.org/10.1109/WSC48552.2020.9384077.

Jacob-Loyola, N., F. Muñoz-La Rivera, R. F. Herrera, and E. Atencio. 2021. "Unmanned aerial vehicles (Uavs) for physical progress monitoring of construction." *Sensors*, 21 (12). https://doi.org/10.3390/s21124227.

Jeong, I., Y. Jang, J. Park, and Y. K. Cho. 2021. "Motion Planning of Mobile Robots for Autonomous Navigation on Uneven Ground Surfaces." *Journal of Computing in Civil Engineering*, 35 (3): 04021001. https://doi.org/10.1061/(asce)cp.1943-5487.0000963.

Kayhani, N., H. Taghaddos, and S. BehzadiPour. 2018. "Construction equipment collision-free path planning using robotic approach." *ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things*. https://doi.org/10.22260/isarc2018/0169.

Khasawneh, A., H. Rogers, J. Bertrand, K. C. Madathil, and A. Gramopadhye. 2019. "Human adaptation to latency in teleoperated multi-robot human-agent search and rescue teams." *Automation in Construction*, 99: 265–277. https://doi.org/10.1016/j.autcon.2018.12.012.

Kim, P., J. Park, Y. K. Cho, and J. Kang. 2019. "UAV-assisted autonomous mobile robot navigation for as-is 3D data collection and registration in cluttered environments." *Automation in Construction*, 106. https://doi.org/10.1016/j.autcon.2019.102918.

Kim, S. K., J. S. Russell, and K. J. Koo. 2003. "Construction robot path-planning for earthwork operations." *Journal of Computing in Civil Engineering*, 17 (2): 97–104. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:2(97).

Kim, S., M. Peavy, P. C. Huang, and K. Kim. 2021. "Development of BIM-integrated construction robot task planning and simulation system." *Automation in Construction*, 127. https://doi.org/10.1016/j.autcon.2021.103720.

Kingma, D. P., and J. L. Ba. 2014. "Adam: A Method for Stochastic Optimization." *arXiv preprint arXiv:1412.6980*.

Lavalle, S. M. 1998. "Rapidly-Exploring Random Trees: A New Tool for Path Planning." https://www.cs.csustan.edu/~xliang/Courses/CS4710-21S/Papers/06 RRT.pdf.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. 2015. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971*.

Liu, Y., M. Habibnezhad, and H. Jebelli. 2021a. "Brain-computer interface for hands-free teleoperation of construction robots." *Automation in Construction*, 123. https://doi.org/10.1016/j.autcon.2020.103523.

Liu, Y., M. Habibnezhad, and H. Jebelli. 2021b. "Brainwave-driven human-robot collaboration in construction." *Automation in Construction*, 124. https://doi.org/10.1016/j.autcon.2021.103556.

Lu, T., S. Tervola, X. Lü, C. J. Kibert, Q. Zhang, T. Li, and Z. Yao. 2021. "A novel methodology for the path alignment of visual SLAM in indoor construction inspection." *Automation in Construction*, 127. https://doi.org/10.1016/j.autcon.2021.103723.

Madsen, A. J. 2019. "The SAM100: Analyzing Labor Productivity." Accessed July 5, 2022. https://digitalcommons.calpoly.edu/cmsp/243/.

McKinsey&Company, S. Changali, A. Mohammad, and M. Van Nieuwland. 2015. "The construction productivity imperative." *McKinsey Quarterly*. Accessed June 14, 2021. https://www.mckinsey.com/business-functions/operations/our-insights/the-construction-productivity-imperative.

Min, H. Q., J. H. Zhu, and X. J. Zheng. 2005. "Obstacle avoidance with multi-objective optimization by PSO in dynamic environment." *2005 International Conference on Machine Learning and Cybernetics, ICMLC 2005*, 2950–2956. https://doi.org/10.1109/icmlc.2005.1527447.

Mnih, V., A. P. Badia, L. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu. 2016. “Asynchronous methods for deep reinforcement learning.” *33rd International Conference on Machine Learning, ICML 2016*, 4 (February 2016): 2850–2869.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. 2013. “Playing Atari with Deep Reinforcement Learning.” *arXiv preprint arXiv:1312.5602*, 135–160.

Narazaki, Y., V. Hoskere, G. Chowdhary, and B. F. Spencer. 2022. “Vision-based navigation planning for autonomous post-earthquake inspection of reinforced concrete railway viaducts using unmanned aerial vehicles.” *Automation in Construction*, 137: 104214. https://doi.org/10.1016/j.autcon.2022.104214.

Okishiba, S., R. Fukui, M. Takagi, H. Azumi, S. Warisawa, R. Togashi, H. Kitaoka, and T. Ooi. 2019. “Tablet interface for direct vision teleoperation of an excavator for urban construction work.” *Automation in Construction*, 102: 17–26. https://doi.org/10.1016/j.autcon.2019.02.003.

OSHA. 2020. “Commonly Used Statistics.” Accessed March 25, 2020. https://www.osha.gov/oshstats/commonstats.html.

Roberts, D., and M. Golparvar-Fard. 2019. “End-to-end vision-based detection, tracking and activity analysis of earthmoving equipment filmed at ground level.” *Automation in Construction*, 105. https://doi.org/10.1016/j.autcon.2019.04.006.

Stentz, A. 1994. “Optimal and efficient path planning for partially-known environments.” *Proceedings - IEEE International Conference on Robotics and Automation*, 3310–3317. Publ by IEEE. https://doi.org/10.1007/978-1-4615-6325-9_11.

The Center for Construction Research and Training. 2018. *The Construction Chart Book*.

https://www.cpwr.com/wp-content/uploads/publications/5th-Edition-Chart-Book-Final.pdf.

Tsuruta, T., K. Miura, and M. Miyaguchi. 2019. "Mobile robot for marking free access floors at construction sites." *Automation in Construction*, 107. https://doi.org/10.1016/j.autcon.2019.102912.

Wang, X., C.-J. Liang, C. C. Menassa, and V. R. Kamat. 2021. "Interactive and Immersive Process-Level Digital Twin for Collaborative Human–Robot Construction Work." *Journal of Computing in Civil Engineering*, 35 (6): 04021023. https://doi.org/10.1061/(asce)cp.1943-5487.0000988.

Wang, Z., H. Li, and X. Yang. 2020. "Vision-based robotic system for on-site construction and demolition waste sorting and recycling." *Journal of Building Engineering*, 32. https://doi.org/10.1016/j.jobe.2020.101769.

Wen, S., Y. Zhao, X. Yuan, Z. Wang, D. Zhang, and L. Manfredi. 2020. "Path planning for active SLAM based on deep reinforcement learning under unknown environments." *Intelligent Service Robotics*, 13 (2): 263–272. https://doi.org/10.1007/s11370-019-00310-w.

Xiao, X., B. Liu, G. Warnell, and P. Stone. 2022. "Motion planning and control for mobile robot navigation using machine learning: a survey." *Autonomous Robots*. https://doi.org/10.1007/s10514-022-10039-8.

Xie, L., S. Wang, A. Markham, and N. Trigoni. 2017. "Towards Monocular Vision based Obstacle Avoidance through Deep Reinforcement Learning." *arXiv preprint arXiv:1706.09829*.

Xu, H., N. Wang, H. Zhao, and Z. Zheng. 2019. "Deep reinforcement learning-based path planning of underactuated surface vessels." *Cyber-Physical Systems*, 5 (1): 1–17. https://doi.org/10.1080/23335777.2018.1540018.

Yan, N., S. Huang, and C. Kong. 2021. "Reinforcement Learning-Based Autonomous Navigation and Obstacle Avoidance for USVs under Partially Observable Conditions." *Mathematical Problems in Engineering*, 2021. https://doi.org/10.1155/2021/5519033.

YouTube. 2019. "Hospital construction." Accessed April 7, 2019. https://www.youtube.com/channel/UCEKwrM78pRv8WRcKvZNtE1w.

Zhang, H. Y., W. M. Lin, and A. X. Chen. 2018. "Path planning for the mobile robot: A review." *Symmetry*, 10 (10). https://doi.org/10.3390/sym10100450.

Zhang, X., C. Wang, Y. Liu, and X. Chen. 2019. "Decision-making for the autonomous navigation of maritime autonomous surface ships based on scene division and deep reinforcement learning." *Sensors (Switzerland)*, 19 (18). https://doi.org/10.3390/s19184055.

Zhou, T., Q. Zhu, Y. Shi, and J. Du. 2022. "Construction Robot Teleoperation Safeguard Based on Real-Time Human Hand Motion Prediction." *Journal of Construction Engineering and Management*, 148 (7): 04022040. https://doi.org/10.1061/(asce)co.1943-7862.0002289.