Proximity Prediction of Mobile Objects to Prevent Contact-Driven Accidents in Co-Robotic Construction

Daeho Kim[1], SangHyun Lee[2], and Vineet R. Kamat[3]

Abstract

Robotic solutions have garnered increased attention from the construction industry as an effective means of improving construction safety and productivity. However, in deploying such robots to real fields many safety concerns have remained untackled, particularly contact-driven accidents that can be potentially escalated by mobile robots. To address this issue, the authors develop a fully automated framework that enables predicting the proximity between mobile objects, leveraging a camera-mounted unmanned aerial vehicle (UAV), computer vision, and deep neural networks, and conduct a field test to evaluate its validity. In the test, the framework showed a promising result: it achieved average proximity error of 0.95 meters in predicting 5.28 seconds future proximity between a worker and a truck. The major contribution of this study is in predicting the risk of impending collision in advance, thereby making pro-active interventions possible. Computationally, the predictive functionality based on computer vision and deep neural network including convolutional neural network and generative adversarial network would allow robots to examine alternative multiple paths beforehand and enable providing advance alerts to workers. These pro-active interventions would effectively reduce the chances of impending collisions between mobile robots and construction workers.

Keywords: *Autonomous robot; Contact-driven accident; Proximity prediction; Unmanned aerial vehicle; Deep neural network.*

[1] Ph.D. Candidate, Dept. of Civil and Environmental Engineering, Univ. of Michigan, MI, 48109.
[2] Professor, Dept. of Civil and Environmental Engineering, Univ. of Michigan, MI, 48109,
E-mail: shdpm@umich.edu. (Corresponding author)
[3] Professor, Dept. of Civil and Environmental Engineering, Univ. of Michigan, MI, 48109.

Introduction

Construction industries around the world are gradually gearing up for robotic automation (Tsuruta et al. 2019; Lattanzi and Miller 2017; Veha et al. 2013). A growing number of construction companies are embracing robotic solutions to reap the benefits of improved productivity and safety (Kim et al. 2019a; Bock 2015). Notably, the global construction robot market is rapidly growing; it is expected to reach around $190 million by 2025, at a compound annual growth rate (CAGR) of over 20% (Research and Market 2019). According to a market report from Tractica, U.S., more than 7,000 autonomous (or semi-) robots are expected to be deployed to construction fields in the U.S. between 2018 and 2025 (Tractica 2019).

A wide range of construction robots are under development or in the early stage of deployment: for example, to name a few, there are structure robots for 3D printing (Liu and Li 2018), bricklaying (Moon et al. 2018), welding (Tavares et al. 2019), modular building (Yang et al 2019), finishing robots for drywall installation (Yu et al. 2016), façade painting and cleaning (Vega-Heredia et al. 2019), and infrastructure robots for demolition (Li et al. 2019; Zheng et al. 2018), and rebar tying (Cardno 2018). Assisting physically demanding, highly repetitive, and hazardous tasks in construction, such robots are expected to be the main driver that transforms future construction into a more productive and safer industry.

However, in deploying such robots to real fields, many safety concerns have remained untackled (Guiochet et al. 2017). The authors specifically cast light on the contact-driven hazards that could be escalated by mobile robots. Construction generally takes place in a highly unstructured and dynamic environment. Workspace (e.g., terrain and structure) evolves over time and multiple entities (e.g., workers, equipment, and robots) are bound to share a limited workspace. In such a unstructured and evolving space, the chances of contact-driven accidents (e.g., struck-by and caught-in/between) by motorized resources can arise easily, frequently, and unexpectedly (Kim et al. 2019b; Teizer et al. 2010).

In co-robotic construction, where mobile robots are closely involved in the field built for human labor, workers will be assuming greater risk for such accidents. Any movement caused by misperception of a situation (e.g., approaching, deviating, and reversing) can pose a fatal threat to

nearby workers. However, it is unknown how mobile robots' situational intelligence—such as the capacity for understanding, reasoning, and improvisational decision making—rises to the dynamically evolving situations of construction. In mobile robots' navigation and behavior in such uncertain situations, there could be unexpected errors, which could pose a greater risk of forcible contact to nearby workers.

A major research question for this problem has been attuned to the process of detecting the proximity between workers and robots (or mobile equipment) in time (Kim et al. 2019b; Park et al. 2016; Teizer 2015; Teizer et al. 2010). Several collision avoidance technologies, such as those using proximity sensors or computer vision methods, have been explored to this end. Most existing technologies have been used to monitor or detect the proximity at a current time-step, relying upon present sensory data. In many cases, however, prediction is far more important and effective for contact-driven accident prevention (Kim et al. 2019c). This is principally because the sooner robot and worker are informed of their proximity to each other, the more likely they are to avoid the potential collision. Nevertheless, few studies have attempted to address it.

With this background, the authors develop deep neural networks (DNNs)-based framework that enables proximity prediction of mobile objects. In this framework, a camera-mounted unmanned aerial vehicle (UAV) monitors associated entities, serving as the third eye of robots and workers, which has a wider line-of-sight (Figure 1). Inputting the UAV-captured imagery data, the framework powered by DNNs for object detection (Figure 1-A) and trajectory prediction (Figure 1-B) performs proximity prediction (Figure 1-C) in a fully automated way.

The major contribution of this work is to enable predicting the risk of impending collision in advance, thereby making pro-active safety interventions possible. Specifically, the proximity prediction would assist mobile robots' predictive path planning and rerouting. Also, via wearable devices (e.g., wrist band and smart safety glasses), it would enable providing an advance alert to workers, helping them to take timely evasive action. These pro-active interventions would effectively reduce the chances of impending collisions between mobile robots (or mobile equipment) and construction workers.

Moreover, the authors apply a generative adversarial network (GAN) to trajectory prediction, which opens a new possibility of GAN for potential construction applications.

Existing Collision Avoidance Technologies and Challenges in Construction Applications

There has been a wide range of collision avoidance technologies, such as those based on proximity sensors or computer vision methods. This section provides a holistic view of these technologies, discussing their pros and cons. In addition, the authors detail a major challenge that the technologies would have in construction field applications.

Collision Avoidance Technologies: Proximity Sensors

Based on operation principles, proximity sensors can be largely categorized into two types: (i) time-of-flight (TOF)-based sensor and (ii) tag-based sensor. The TOF-based sensor, installed on a robot, measures the distance of surroundings (e.g., geographic features, obstacles, and workers) by emitting a certain form of energy and reading its time-of-flight. As well-recognized sensors, sound navigation and ranging (SONAR), radio detection and ranging (RADAR), and light detection and ranging (LIDAR) are included in this category.

SONAR (or ultrasonic sensor) measures distances to physical objects by transmitting a high-frequency sound wave and measuring the TOF of its echo reflected from the target objects. A sound wave requires a certain medium to travel. Its propagation, therefore, involves many disturbances by the medium's physical conditions (e.g., temperature and pressure), and it can be more so particularly in the case of longer-range detection (Varghese and Boone 2015). Accordingly, the application of SONAR in mobile robots has been limited to short-range detection—typically less than 3 meters (e.g., reverse parking) (Ducarme 2019).

On the other hand, RADAR uses radio signal (300 MHz - 40 GHz), a kind of electromagnetic wave, which does not require a certain medium to travel. It thus functions in many wild conditions (e.g., rain, fog, snow, and dust) and has a long-range of reading—generally more than 30 meters (Ducarme 2019). In addition, using Doppler Effect (Chen et al. 2006), it can also detect the speed of moving

objects as well as its proximity (Varghese and Boone 2015). However, the performance of RADAR can vary by reflectors. This is because the radio signal can be easily dispersed, particularly when encountering unfavorable reflectors such as plastics, dry wood, or objects with large flat surfaces (Ruff 2006).

LIDAR also uses a kind of electromagnetic wave, the beam of light (or laser). It is able to not only measure distances to objects but also scan 3D surroundings with multi-axis lasers. The more lasers a LIDAR transmits, the denser 3D world can be reconstructed (Ducarme 2019; Varghese and Boone 2015). Of stand-alone sensors, LIDAR is often cited as the most accurate proximity sensor (Gargoum et al. 2018). Also, the 3D readout is potentially used as the primary source for the path planning of many autonomous navigating robots (Kim et al. 2018). However, LIDAR, as with other TOF-based sensors, cannot distinguish what the detected objects are. To distinguish objects, it needs additional object classification software (Ducarme 2019).

Distinctive to these TOF-based sensors, tag-based sensors utilize an energy field (e.g., electromagnetic field) and detect proximity via the signal communication between a reader mounted to a robot and tags worn by workers. With this principle, many kinds of sensors have been devised, including radio frequency identification (RFID), magnetic field (MF), and Bluetooth low energy (BLE). As the tag-based sensors don't rely on the TOF measurement, they are less affected by the line-of-sight (Ducarme 2019). However, the tag-based sensors have hardly gained a competitive edge over the TOF-based sensors in terms of accuracy and fidelity. According to a test conducted by Park et al. (2016), the proximity errors of RFID, MF, and BLE sensors were up to 5.0, 3.4, and 2.6 meters, respectively, with the standard deviation of 2.1, 0.3, and 1.8 meters. Although the tag-based sensors still have the potential to complement other technologies (e.g., SONAR, RADAR, and LIDAR), the prerequisite that all targets need to be attached with a tag hinders their application in construction (Memarzadeh et al. 2013; Park et al. 2012).

The proximity sensors have been widely applied in robotics to assist the robots' collision avoidance (Cui et al. 2019). However, the effectiveness, availability, and functionality of the existing proximity sensors could be challenged in a highly unstructured and dynamic construction site. For

example, the TOF-based sensors (e.g., SONAR, RADAR, and LIDAR) could be frequently blinded by physical barriers; while the performances of tag-based sensors (e.g., RFID, MF, and BLE) are susceptible to deterioration due to the jamming caused by metallic or wooden objects, both of which are common in construction sites.

Above all, this study highlights the existing technologies' limited scope of application in construction. The application of sensor-based technologies have been limited to detecting or monitoring proximity at current time-step. However, it may not be as effective in many impending situations. In a dynamic and unstructured construction site, contact-driven accidents occur spontaneously in unexpected ways. In such an impending situation, mere detecting or monitoring proximity would not be effective because the near-sighted measure wouldn't allow enough time for the involved robot (and equipment operator) and worker to take prompt evasive action. In this sense, to better prevent contact-driven accidents in co-robotic construction, collision avoidance technology needs to be equipped with the prediction functionality for potential accidents.

Collision Avoidance Technologies: Computer Vision-based Methods

Over recent years, computer vision-based methods have demonstrated great potential as a supplementary technology to proximity sensors (Zhu et al. 2017; Park et al. 2016; Memarzadeh et al. 2013; Park et al. 2012; Brilakis et al. 2011). It uses one or more imaging devices (e.g., digital camera) to capture multiple targets and stream the digital images to a computer. In turn, it utilizes the computing power to conduct object detection and proximity measurement. With the improvement of computing power, the potential of the computer vision continues to grow. This growth is evidenced by the number of construction studies that have explored computer vision-based collision avoidance technologies. For example, Memarzadeh et al. (2013) developed an algorithm to detect multi-class construction objects by integrating histogram of oriented gradient (HOG) and histogram of hue-saturation-value (HSV); Kim et al. (2016) proposed a proximity monitoring framework that employs Gaussian mixture model (GMM)-based object detection; Kim et al. (2017) introduced another proximity monitoring framework using multi-view cameras and object detection based on HOG and support vector machine (SVM). The

158 previous studies have greatly contributed to examining the potential of computer vision-based collision
159 avoidance technologies. However, there are several drawbacks of the computer vision-based methods,
160 which need to be addressed for construction applications.

161 A major imaging device widely used is stationary cameras such as tripod-mounted or
162 surveillance cameras (Zhu et al. 2017; Park et al. 2016; Brilakis et al. 2011). These cameras are cheap,
163 readily available, and easy to apply. However, this technology can involve frequent occlusions of targets
164 (i.e., the situation that targets are occluded by physical barriers and so become invisible) particularly on
165 construction sites where a number of obstacles to the camera's line-of-sight are scattered (Kim et al.
166 2019b). The problem is that such occlusions are fatal to any computer vision-based object detection
167 because the computer vision is bound to rely on the visible information of a target (e.g., the target's
168 pixel values and configuration). Therefore, the application of mobile imaging devices which have a
169 wider line-of-sight and mobility, thereby reducing such occlusions (e.g., UAVs), must be considered.

170 Many earlier studies applied one or more hand-crafted features—such as HOG, HSV, scale
171 invariant feature transform (SIFT), and speeded-up robust features (SURF)—to object detection.
172 However, using such features naturally involves a heavy computation due to pre-processing and
173 multiple steps for feature extraction, resulting in slow processing speed (Kim et al. 2019b). Recently,
174 DNN-based object detection has made large progress in terms of speed and accuracy by leveraging
175 parallel computing and finer-level learned features. Accordingly, an increasing number of studies have
176 attempted to apply the DNN-based object detection framework for construction applications. For
177 example, Fang et al. (2018), Luo et al. (2018), Son et al. (2019), and Yan et al. (2019) applied faster
178 region-based convolutional neural network (Faster R-CNN, Ren et al. 2017) for construction objects
179 detection; Kim et al. (2018) and Alipour et al. (2019) applied region-based fully convolutional network
180 (R-FCN, Dai et al. 2016). The studies applying DNNs proved to greatly improve the speed and accuracy
181 of construction object detection. However, since the DNNs (i.e., Faster R-CNN and R-FCN) rely on
182 two-stage inferences (region proposal and classification) by two separated networks, they involve a
183 high computational cost and couldn't achieve the real-time operation—30 frame per second (*FPS*). The

real-time operation is definitely critical in assisting collision avoidance. Computer vision-based methods, therefore, must demonstrate real-time operation for real-world applications.

Despite the drawbacks, computer vision-based methods have immense potential to supplement sensor-based technologies. With increasingly published vision datasets, advanced DNN architectures, and enhanced computing power, both speed and accuracy of computer vision-based methods continue to improve. Also, it involves less hardware installation and enables classification as well as the detection of multiple objects. However, its scope of application, as with the aforementioned sensor-based technologies, has been limited to proximity monitoring at current time-step. To more pro-actively assist collision avoidance, the prediction of future proximity and potential hazard needs to be addressed.

DNN-based Framework for Proximity Prediction

To address the above challenges, the authors develop a fully automated framework that enables real-time proximity prediction of mobile objects, leveraging a camera-mounted UAV, object detection DNN [you only look once-v3 (YOLO-V3, Redmon and Farhadi 2018)], and trajectory prediction DNN [social GAN (S-GAN, Gupta et al. 2018)]. This framework consists of two main modules: (i) a trajectory observation module that monitors targets' locations and records their past trajectories and (ii) a trajectory prediction module that predicts the target's future trajectories and estimates their future proximity. This section details each module's functionality and development process as well as presents its validation result.

Module 1: Trajectory Observation

The first module monitors targets' locations and records their past trajectories, which are the primary input for trajectory prediction (Figure 2). This module first detects targets on a UAV-captured input image and estimates their center location as image coordinates (i.e., x-y pixel coordinates) using an object detection model based on YOLO-V3 (Figure 2-A). In turn, this module rectifies the coordinates to the world coordinates through geometric transformation using a reference object since the image coordinates can neither reflect the true scene scale nor be accurate due to a projective distortion inherent

on a 2D image captured by a UAV (Figure 2-B). This module runs the object detection and the coordinate rectification at every input image, thereby continuing to update true-to-scale, distortion-free locations of targets. Based on the location information, it records the targets' past trajectories (from 3.96 seconds earlier to current, Figure 2-C) and streams those to the second module for trajectory prediction.

The primary role of Module 1 is the trajectory observation of mobile construction objects but it can also conduct real-time proximity monitoring. In a prior study (Kim et al. 2019b), the authors demonstrated this module's performance on proximity monitoring—0.26 meters average displacement error (i.e., average of Euclidean distance between a target's ground truth and estimated positions) and 0.61 meters average proximity error (i.e., average of absolute difference between a pair of targets' ground truth proximity and estimated proximity). The details of Module 1's proximity monitoring performance can be found in our prior study (Kim et al. 2019b).

Object Detection using YOLO-V3

To develop an object detection model, the authors leveraged YOLO-V3, which demonstrated outstanding performances in terms of both speed and accuracy. The YOLO-V3 realizes a one-stage operation by leveraging end-to-end convolutional layers and grid-based value encoding. As a result, it could reduce the network complexity and computational cost, achieving real-time operation (35 *FPS*) (Redmon and Farhadi 2018). Also, taking advantage of multi-scale inference, the YOLO-V3 improves reasoning capability. Consequently, it could show superior accuracy on common objects in context (COCO) object detection challenge [55.3% mean average precision (*mAP*)] over other one-stage object detection DNNs [e.g., single shot multibox detector (SSD, Liu et al. 2016): 45.4% *mAP* and deconvolutional single shot detector (DSSD, Fu et al. 2017): 46.1% *mAP*] (Redmon and Farhadi 2018).

In this work, the authors started from a YOLO-V3 model developed in our prior study (Kim et al. 2019b) which was trained with COCO (Lin et al. 2014) and construction dataset (N=4,114) and updated through additional fine-tuning with larger construction dataset (N=13,147). As a result, the updated model demonstrated a promising detection performance on a test dataset (N=547): it showed 97.23% *mAP* (Equation 1) and 83.54% average intersection over union (Avg. *IoU*, Equation 2) for

238 excavator, wheel loader, truck, worker, and reference objects (e.g., square, rectangular, and pentagonal
239 concrete footings).
240 While it would have been ideal to train this model with a mobile construction robots dataset
241 as well, the lack of imagery data for mobile construction robots made such work impossible to complete
242 at this time. This model would have another chance to fine-tune its process once sufficient datasets for
243 mobile construction robots become available.
244

$$mAP = \frac{1}{n} * \sum_{1}^{n} \left(\frac{1}{11} * \sum_{r=0.0,0.1,\ldots,1.0} MP_r \right) \quad \text{Equation 1}$$

Note: n=the total number of object classes; MP_r=maximum precision at a certain recall value r (i.e., 0, 0.1, 0.2, …, 1.0) on precision-recall curve of 50% IoU.

245

$$\text{Avg. } IoU = \frac{1}{k} * \sum_{1}^{k} \left(\frac{AoO}{AoU} \right) \quad \text{Equation 2}$$

Note: k=the total number of detected objects; AoO=area of overlap; AoU=area of union.

246

247 ***Coordinate Rectification using Geometric Transformation***

248 While a camera maps a 3D world onto a 2D image, the real scene scale is lost and a projective distortion
249 arises (Figure 2-B). Therefore, the image coordinates of an object can neither reflect the true scene scale
250 nor be accurate. Module 1 thus performs coordinate rectification following the object detection, thereby
251 recording true-to-scale, distortion-free coordinates of objects. To this end, this module uses a geometric
252 transformation algorithm using a reference object, which was developed in our prior study (Kim et al.
253 2019b). This algorithm detects a reference object's edges, contours, and four vertexes and estimates the
254 transformation matrix by matching the vertexes to the known reference dimensions (Figure 2-B). In
255 turn, it converts image coordinates to true-to-scale, distortion-free world coordinates using the
256 transformation matrix. In a prior study (Kim et al. 2019b), the authors validated the effect of the

geometric transformation algorithm on improving distance measurement accuracy: in a lab test, the algorithm improved the percentage accuracy of proximity measurement from 68.32% to 93.33% at the maximum. The details of the geometric transformation algorithm and its evaluation result can be found in our prior study (Kim et al. 2019b).

Figure 2 illustrates an example of the geometric transformation where a square concrete footing is used as a reference object (Figure 2-B). However, any objects having four or more vertexes—such as quadrangle, pentagonal, or hexagonal objects—can be used as a reference object if its dimensions are known.

Module 2: Trajectory Prediction

The second module (i.e., trajectory prediction) takes a set of target's past trajectories as input (from 3.96 seconds earlier to current, Figure 3-A) and predicts their future trajectories for up to 5.28 seconds (Figure 3-B), using a trajectory prediction model based on S-GAN. The set of future trajectories informs where the targets will be located for the next 5.28 seconds at an interval of 0.66 seconds. Lastly, based on the targets' predicted locations, this module estimates the targets' proximity for the next 5.28 seconds—the proximity after 0.66, 1.32, 1.98, 2.64, 3.30, 3.96, 4.62, and 5.28 seconds (Figure 3-C).

Trajectory prediction studies have been dominated by data-driven learning approaches. This is basically because the movement of an entity (e.g., people) is so diverse and uncertain that it is extremely challenging to model through hand engineering. In an early stage, there are several studies to use hand-crafted features-based learning (Yamaguchi et al. 2011; Antonini et al. 2006; Helbing and Molnar 1995) or statistical learning such as polynomial regression (Rashid and Behzadan 2017), Gaussian process (Trautman et al. 2015; Tay and Laugier 2008), and hidden Markov model (Rashid and Behzadan 2017). However, many contemporary studies are motivated to use a DNN, following the trajectory of many other data-driven studies. In recent years, several DNN architectures for trajectory prediction have been released: for example, there are social long short-term memory (S-LSTM, Alahi et al. 2016), crowd interaction DNN (Xu et al. 2018), interaction aware DNN (Pfeiffer et al. 2018), and S-GAN (Gupta et al. 2018). Of these, the S-GAN, incorporating several distinctive features,

284 demonstrated a state-of-the-art performance over others (Gupta et al. 2018). It enables a model to learn
285 social behavior (e.g., collision avoidance) as well as an entity's moving pattern by integrating an LSTM
286 encoder-decoder and a social pooling layer (Gupta et al. 2018). By realizing GAN architecture (i.e.,
287 coupling discriminator to generator) and adversarial training, it enhances the capability to learn
288 complicated distributions of mobile objects' trajectories and improves reliability of prediction output.
289 For this reason, this study applied the S-GAN and developed a trajectory prediction model through
290 transfer learning.
291
292 ***Network Architecture of S-GAN***
293 The S-GAN has two main components: (i) generator that predicts targets' future trajectories (Figure 4-
294 A) and (ii) discriminator that inspects the quality of the predictions (Figure 4-B).
295 • Generator (Figure 4-A): the generator takes past trajectories of targets as input and predicts their
296 future trajectories through network integrating social pooling layer into the middle of LSTM
297 encoder-decoder. The generator first converts the input trajectories to fixed-length vectors via
298 multilayer perceptron (MLP, Figure 4-AA) and feeds it to LSTM units of encoder (figure 4-AB).
299 The LSTM units then encode the targets' movement patterns individually and forward the encoded
300 features to social pooling layer which infers the targets' social interactions and generates pooled
301 tensor for each target (Figure 4-AC). Lastly, the decoder interprets the interconnected hidden state
302 of input trajectories with multiple LSTM units and generates socially plausible future trajectories
303 of the targets (Figure 4-AD). Here, the decoder initializes itself with input trajectories so that it can
304 generate future trajectories that better conform to the past ones.
305 • Discriminator (Figure 4-B): the discriminator inspects the predicted trajectories' quality and
306 conformity to the past trajectories. It takes both of past and future trajectories together as input and
307 encodes their conformity features through LSTM units (Figure 4-BA). In turn, it calculates the
308 predicted trajectories' conformity score via MLP (Figure 4-BB) and inspects them whether they are
309 plausible or not (i.e., classifies whether real or fake). The prediction that successfully fools the
310 discriminator is selected as the final outcome.

Transfer Learning of S-GAN

The authors developed a trajectory prediction model through transfer learning of the S-GAN. The following details were specifically considered: (i) parameter initialization, (ii) fine-tuning, and (iii) hyper-parameter tuning. This work started from the S-GAN model, which is pre-trained with the two benchmark datasets: (i) Eidgenossische Technische Hochschule Zurich (ETH, Pellegrini et al. 2010) and (ii) University of Cyprus (UCY, Leal-Taixe et al. 2014). As the most widely benchmarked datasets in trajectory prediction studies, the two datasets in total contain 1,536 human trajectories. They reflect various movement patterns such as crossing each other, collision avoidance, group forming, and dispersing (Alahi et al. 2016). Having such diverse data in pre-training was intended to prevent overfitting in the following fine-tuning process.

From that starting point (i.e., pre-learned weights), the fine-tuning with construction dataset was conducted to better fit the pre-trained model to construction settings. Specifically, the authors fine-tuned it with the integrated dataset (i.e., ETH + UCY + the construction dataset), rather than only with the construction dataset, so as to minimize the possibility of overfitting. In this tuning, the trajectories of construction mobile resources (e.g., worker, wheel loader, and excavator), annotated from 916 UAV-captured images, were used.

The farther prediction is achieved, the earlier safety intervention can be made. The authors thus modified the original prediction length (3.96 seconds=12 time-steps x 0.33 seconds) to 5.28 seconds (16 time-steps x 0.33 seconds) and particularly examined how observation-related hyper-parameters affects the model's final performance. Trajectory prediction is primarily based on the interpretation of targets' previous movement patterns. Thus, the properties of past trajectory must have a significant impact on the model's final performance. In this sense, this task additionally tuned the two observation-related hyper-parameters (i.e., observation length and sampling interval) with the following reasons.

- Observation length: a target's future trajectory is highly attributed to its previous movement pattern. The length of observation (i.e., how long observation the model will consume) must thus have a significant impact on a model's prediction performance. Thus, three different observation lengths

were considered in this work: (i) 2.64 seconds (80 frames), (ii) 3.96 seconds (120 frames), and (iii) 5.28 seconds (160 frames).

- Sampling interval: the other hyper-parameter selected was sampling interval. This is because it controls the minuteness of input and output trajectories. With a denser sampling interval, the model can have finer input, but should take the burden of outputting denser prediction as well. On the other hand, with a sparser sampling interval, the model should have coarser input but can avoid such complexity. To examine which level of sampling interval would better fit for our problem, the authors considered four different sampling intervals: (i) 0.17 seconds (5 frames), (ii) 0.33 seconds (10 frames), (iii) 0.66 seconds (20 frames), and (iv) 1.33 seconds (40 frames).

Test Result

For comparative evaluation of the twelve tuned models, the test on a construction dataset was followed. In this test, a total of 397 UAV-captured images was used and the trajectories of three object classes were considered: (i) worker, (ii) wheel loader, and (iii) excavator (Figure 5). As evaluation metrics, average displacement error (*ADE*) and final displacement error (*FDE*), the typical two evaluation metrics to access trajectory prediction accuracy, were applied (Alahi et al. 2016; Gupta et al. 2018). The *ADE* is the average value of displacement errors (*DEs*, Euclidean distances) between ground truths and predictions over all predicted time-steps (i.e., average of $DE@1^{st}$~8^{th}, Figure 5) meanwhile the *FDE* is the distance between the predicted final destination and the ground truth destination at the end of the prediction period (i.e., $DE@8^{th}$, Figure 5). This test was intended to evaluate the pure performances of the tuned models, so the authors fed the models the ground truth of observation trajectories.

Table 1 summarizes the *ADE* and *FDE* results. Overall, the tuned models showed a promising prediction accuracy: all the *ADEs* were less than 0.9 meters and the *FDEs* were less than two meters. It was shown that the model of 0.66 seconds (20 frames) sampling interval and 3.96 seconds (120 frames) observation length has the highest accuracy in terms of both *ADE* and *FDE*: this model achieved the *ADE* of 0.45 meters and the *FDE* of 0.79 meters in this test. Considering this result, the authors adopted the model that showed the least error as the trajectory prediction module.

365 **Field Test**

366 A field test was conducted to demonstrate the validity of the overall framework. It would have been
367 ideal to test the proposed framework with mobile construction robots, since the robots are hardly
368 available to date, this test employed a truck which is similar looking to an autonomous truck. Figure 6
369 illustrates the test environments and settings. In this test, the authors simulated the three types of
370 movement patterns between a worker and a truck: (i) moving forward side by side (movement pattern
371 #1); (ii) crossing each other side by side (movement pattern #2); and (iii) crossing each other in curves
372 (movement pattern #3), as shown in Figure 6. The worker and the truck set off at the same time at the
373 designated origins and followed the ground lines at a constant velocity (1.5 *meters/second*) until arriving
374 at the designated destinations. The movement patterns were simulated three times per each. During this
375 test, the authors flew a camera-mounted UAV over the testbed and ran the developed framework to
376 predict the proximity between the targets (i.e., the metric distance between the worker and the truck).
377 Lastly, the accuracy of the proximity outputs was evaluated by comparing it to the corresponding ground
378 truth proximity.
379
380 ***Measurement of Ground Truth Proximity***
381 To measure the ground truth proximity over all time-steps, the authors intentionally used ground lines
382 and markers (Figure 6). The targets were ordered to follow a reference line at a constant velocity.
383 Therefore, the origin-destination locations and times of a target were known so that the target's in-
384 between locations and times could be measured by interpolation. In doing so, the authors measured all
385 the ground truth locations of the targets over all time-steps and their ground truth proximity accordingly.
386
387 ***Evaluation Metrics***
388 To evaluate the accuracy of targets' predicted locations, the two displacement errors, average
389 displacement error (*ADE*) and final displacement error (*FDE*), were applied. While the *ADE* and *FDE*
390 represent the accuracy of predicted trajectory for each individual target, it does not directly represent
391 the accuracy of predicted proximity between a pair of targets. Thus, in addition to the *ADE* and *FDE*,

this test also evaluated average proximity error (*APE*) and final proximity error (*FPE*). The *APE* is the average value of the absolute differences between predicted proximity and ground truth proximity over all time-steps (Equation 3). Meanwhile, the *FDE* is the absolute difference between predicted proximity and ground truth proximity at the end of the prediction period (Equation 4). Lastly, this test also measured each module's operating time to evaluate its computational efficiency.

$$APE = \frac{1}{n} * \sum_{i=1}^{n} |P_g - P_p| \quad \text{Equation 3}$$

Note: n=the number of cases; P_g=ground truth proximity; P_p=predicted proximity.

$$FPE = |P_{gf} - P_{pf}| \quad \text{Equation 4}$$

Note: P_{gf}=ground truth proximity at the end of prediction period (i.e., after 5.28 seconds); P_{pf}=predicted proximity at the end of prediction period (i.e., after 5.28 seconds).

Proximity Prediction Result

In terms of *ADE* and *FDE*, the developed framework showed promising results. Overall it achieved the *ADEs* for both the worker and the truck less than two meters, the *FDEs* less than 3.5 meters (Table 2). The *ADE* and *FDE* for the worker were 1.64 and 3.39 meters overall and those for the truck were 1.99 and 2.99 meters (Table 2). In line with the *ADE* and *FDE* results, the *APE* and *FPE* results were also promising. Overall the framework achieved 0.95 meters *APE* and 1.71 meters FPE between the worker and the truck (Table 3). Also, the *APEs* between the worker and the truck for all three movement patterns were less than 1.5 meters, the *FPEs* less than 2.5 meters (Table 3).

Notably, it was determined that to predict farther time-step is more challenging. Figure 7 illustrates the trend of proximity error (i.e., absolute difference between predicted proximity and ground truth proximity) as prediction time-step increases. As shown in Figure 7, for all movement patterns, the proximity errors continued to rise as the prediction time-step increases: on average, the framework showed the proximity error of 0.53 meters at 0.66 seconds prediction, but the error continued to climb

as prediction time-step went farther, reaching to 1.71 meters (=the overall *FPE*, Table 3) at 5.28 seconds prediction (Figure 7).

Operating Time

Figure 8 illustrates the operating time of Modules 1 and 2. With a single graphic processing unit (GPU, NVIDIA Tesla K40), Module 1 (i.e., trajectory observation) spent 0.28 seconds per a frame (Figure 8-A) and Module 2 (i.e., trajectory prediction) spent 0.12 seconds per a cycle (i.e., from taking a set of past trajectories to generating a set of future trajectories, Figure 8-B). Given that this framework runs Module 1 at every 0.66 seconds (i.e., at 20 frames interval), it was able to perform trajectory observation with zero time-lag in computation. And overall, the framework demonstrated that it can update the future proximity for the next 5.28 seconds at every 0.66 seconds with 0.40 seconds time-lag in computation (i.e., 0.28 seconds for Module 1 + 0.12 seconds for Module 2, Figure 8-C). It means that the framework can update future proximity for the next 4.88 seconds at every 0.66 seconds continuously (i.e., 5.28 seconds prediction – 0.40 seconds time-lag in computation).

Discussions

As shown in the field test, the developed framework demonstrated a promising performance of proximity prediction in terms of both accuracy and speed. On the basis of the result, in this section, the authors present how this framework can better assist the collision avoidance between workers and mobile robots (or mobile equipment) at unstructured and dynamic construction sites. In addition, the authors discuss the implication of using GAN-based trajectory prediction DNN and lastly present potential improvement points for future studies.

Real-World Applications to Prevent Contact-driven Accidents by Mobile Objects

The framework showed that it can continuously update future proximity for the next 5.28 seconds at every 0.66 seconds within one-meter proximity error on average (computing time per update=0.40 seconds). This prediction performance can have a far-reaching significance beyond the detection of

current proximity in accident prevention in that it enables pro-active safety interventions. For example, if a robot can be informed of whether a worker will be on the path or inside the action radius of itself in the future, the robot can make pro-active path planning and rerouting in advance. Likewise, it is also possible to provide an advance alert to workers via wearable devices (e.g., wrist band and smart safety glasses) so that the workers can take timely evasive action. Assuming that an autonomous truck is approaching a worker at five meters per second, the framework can inform the worker and the autonomous truck of their potential collision 5.28 seconds before it happens. The worker then has around 25 meters of physical distance from the autonomous truck to easily avoid the collision without strain. These pro-active interventions would effectively reduce the chances of an impending collision between mobile robots and construction workers.

In addition, the developed framework also can be readily applied to other mobile objects such as motorized equipment and vehicle. This framework can detect mobile objects, such as excavator, wheel loader, and truck, and also, the scope of targets can be easily expanded through tuning of the object detection model with the additional training dataset. The framework can thus provide equipment operators and vehicle drivers with an alert in advance as well, helping to avoid a potential collision with workers and mobile robots.

In real-world applications, however, the quality and speed of network connection need to be further investigated and improved. The developed framework uses a camera-mounted UAV (or UAVs) to stream imagery input data to a computing device (e.g., a cloud server). Also, it needs wireless communication with robots and wearable devices to timely feedback. Therefore, in real-world applications, it is critical to ensure rapid data transmission from a computing device to a UAV (or UAVs), wearable devices, and robots. Leveraging 5G wireless network and internet of thing (IoT) cloud platform can be a promising solution to this end. The 5G wireless network would support real-time data transmission at data transfer rate of several gigabytes per second. Also, with the high-speed network connection, an IoT cloud platform could connect multiple UAVs, robots, and wearable devices to a cloud server, which would enable near real-time operation of proximity prediction as well as rapid communication with workers and robots.

To the fully automated operation of the proposed framework, the strategies for UAV operations need to be further studied. In the framework, UAV (or UAVs) plays a vital role in tracking target and reference objects. Therefore, future studies on how to capture mobile target objects and a stationary reference object simultaneously and continuously must be done. To this end, operating multiple UAVs and realizing real-time image stitching could be considered as a possible solution. Also, thorough field experiments need to be conducted in order to investigate how the elevation of UAV can impact proximity monitoring and prediction performance. The higher elevation a UAV flies at, the wider the scene can be monitored. However, it can cause target objects to be seen too small, which can affect object detection performance and accordingly proximity monitoring and prediction accuracy.

Implications of Using GAN-based DNN for Trajectory Prediction

GAN is basically an unsupervised generative model that makes plausible data from a noise input (e.g., Gaussian noise) based on probability distribution learned from real data (Goodfellow et al. 2014). The uniqueness of GAN that yields a highly competitive edge over other generative models (e.g., naïve Bayes, hidden Markov model, and Markov random fields) is the adversarial training between generator and discriminator. In GAN training, the generator tries to minimize min-max loss whereas the discriminator counteracts to maximize it (Equation 5). In this min-max game, both generator and discriminator get to improve while competing with each other. This adversarial training is known to better fit to understanding complex distributions of real data (e.g., images and speeches) than using a certain loss (objective) function manually devised.

$$\text{Min} - \text{Max Loss} = E_x[\log D(x)] + E_z[\log(1 - D(G(z)))] \quad \text{Equation 5}$$

Note: D(x)=discriminator's estimate of the probability that real data instance x is real; E_x=expected value over all real data instances; G(z)=generator's output when given input z; D(G(z))=discriminator's estimate of the probability that a fake instance is real; E_z=expected value over all inputs to the generator.

488 The interesting fact is that the GAN can also be used for trajectory prediction which is basically
489 a supervised learning problem. The S-GAN incorporates the GAN architecture and uses adversarial
490 training so that it can enhance the capability to learn hidden distribution of mobile objects' diverse
491 trajectories. More noticeably, the S-GAN leverages the GAN architecture in a conditional way such that
492 it can still take prior information (i.e., past trajectory) as input and consume ground truth for network
493 supervision. That is, instead of using noise input, it takes past trajectories and initializes the decoder
494 with the prior information, thereby generating future trajectories more conformed to the past. Moreover,
495 it uses *L_2 loss* (Equation 6) in addition to the min-max loss so that it can condition the decoder to
496 generate the prediction closer to the ground truth. In these ways, the S-GAN could take advantage of
497 both adversarial training and supervised learning, consequently resulting in a promising performance
498 of trajectory prediction.
499

$$L_2 \text{ Loss} = \sum_{i=1}^{n} (Y_g - Y_p)^2 \qquad \text{Equation 6}$$

Note: n=dimension of output vector; Y_g=ground truth trajectory; Y_p=predicted trajectory.

500
501 However, the application of S-GAN presents several challenges, particularly in training. The
502 adversarial training between generator and discriminator can be often stuck at local minima and in
503 general takes a longer period than the training of normal DNNs. The single most important reason
504 behind such challenges is the imbalance between generator and discriminator. For example, if the
505 discriminator is too strong, then the generator training can easily fail due to vanishing gradients. On the
506 other hand, if the generator easily defeats the discriminator, it tends to produce the most plausible output
507 repeatedly, which can make the discriminator permanently trapped (called mode collapse).
508 Compared to dominant DNN architectures such as convolutional neural network (CNN) and
509 recurrent neural network (RNN), GAN is a new kind of DNN. Certainly, there are still many chances
510 to improve its trainability, which may include regularization using noise addition (Arjovsky and Bottou
511 2017), penalization of discriminator weights (Roth et al. 2017), and the use of advanced min-max loss

(e.g., *Wasserstein loss*). The application of such advanced techniques would provide us with a better chance to leverage S-GAN (or other GAN-based DNNs) and to have a higher accuracy of proximity prediction thereby.

Another way to improve the prediction accuracy would include post-processing incorporating construction-specific knowledge. The S-GAN showed a promising accuracy of trajectory prediction in this study; however, it would not cover all the possible scenarios that can happen on construction sites and the prediction accuracy can deteriorate in those cases. The post-processing incorporating construction-specific knowledge, such as the average or maximum velocity of each robot (or equipment), construction robots' pre-programmed collision avoidance behavior, and construction workers' collision avoidance behavior, can likely be used to refine predicted trajectory's velocity and direction, which could improve the overall accuracy of proximity prediction.

Conclusions

In this study, the authors developed a DNN-based framework for proximity prediction, leveraging a camera-mounted UAV, object detection DNN (YOLO-V3) and trajectory prediction DNN (S-GAN). Also, the authors demonstrated the framework's validity in a field test: the framework achieved 0.95 meters average proximity error (APE) and 1.71 meters final proximity error (*FPE*) in predicting 5.28 seconds future proximity. During construction operations, contact-driven hazards by mobile robots (or mobile equipment and vehicle) can easily arise in various scenarios: for example, a navigating robot suddenly change in direction or an autonomous vehicle could reverse into a blind spot. In such unpredictable situations, the proximity prediction would enable advance detection of impending collisions, thereby making pro-active interventions possible. Specifically, the predictive functionality would allow robots to make alternative path planning and rerouting beforehand and enable providing advance alerts to workers via wearable devices. These pro-active interventions would contribute to mitigating the chances of impending collisions between mobile robots (or mobile equipment and vehicle) and construction workers. Moreover, the authors apply GAN to trajectory prediction, which opens a new possibility of GAN for potential construction applications.

Data Availability

Some or 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 a National Science Foundation Award (No. IIS-1734266, *'Scene Understanding and Predictive Monitoring for Safe Human-Robot Collaboration in Unstructured and Dynamic Construction Environments'*). 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 National Science Foundation. Lastly, the authors wish to specially thank Weston Tanner, John McGlennon, and Chris Klaft from WALSH Construction Co. and Steve La Cava and Andy Thelen from TOEBE Construction LLC. for their considerate assistance in collecting onsite data.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. 2016. "Social lstm: Human trajectory prediction in crowded spaces." In *Proc., 2016 IEEE Conference on Computer Vision and Pattern Recognition*., 961-971. Las Vegas, NV: IEEE.

Antonini, G., Bierlaire, M., and Weber, M. 2006. "Discrete choice models of pedestrian walking behavior." *Transportation Research Part B: Methodological*. 40 (8): 667-687. https://doi.org/10.1016/j.trb.2005.09.006.

Arjovsky, M., and Bottou, L. 2017. "Towards principled methods for training generative adversarial networks." In Proc., *5th Internaltional Conference on Learning Representations*. Toulon, France. arXiv:1701.04862.

Awolusi, I., Marks, E., and Hallowell, M. 2018. "Wearable technology for personalized construction safety monitoring and trending: Review of applicable devices." *Automation in construction*, 85 (Jan): 96-106. https://doi.org/10.1016/j.autcon.2017.10.010.

Bock, T. 2015. "The future of construction automation: Technological disruption and the upcoming

566 ubiquity of robotics." *Automation in Construction*. 59 (Nov): 113-121.
567 https://doi.org/10.1016/j.autcon.2015.07.022.
568 Cardno, C. A. 2018. "Robotic Rebar-Tying System Uses Artificial Intelligence." *Civil Engineering*
569 *Magazine Archive*. 88 (1): 38-39. https://doi.org/10.1061/ciegag.0001260.
570 Chen, V. C., Li, F., Ho, S.-S., and Wechsler, H. 2006. "Micro-Doppler effect in radar: phenomenon,
571 model, and simulation study." *IEEE Transactions on Aerospace and electronic systems*. 42 (1):
572 2-21. https://doi.org/10.1109/TAES.2006.1603402.
573 Cui, J., Liew, L. S., Sabaliauskaite, G., and Zhou, F. 2019. "A review on safety failures, security attacks,
574 and available countermeasures for autonomous vehicles." *Ad Hoc Networks*, 90 (Jul): 101823.
575 https://doi.org/10.1016/j.adhoc.2018.12.006.
576 DuCarme, J. 2019. "Developing effective proximity detection systems for underground coal mines."
577 *Advances in Productive, Safe, and Responsible Coal Mining*. 101-119.
578 https://doi.org/10.1016/B978-0-08-101288-8.00003-1.
579 Gargoum, S. A., Karsten, L., El-Basyouny, K., and Koch, J. C. 2018. "Automated assessment of vertical
580 clearance on highways scanned using mobile LiDAR technology." *Automation in Construction*.
581 95 (Nov): 260-274. https://doi.org/10.1016/j.autcon.2018.08.015.
582 Guiochet, J., Machin, M., and Waeselynck, H. 2017. "Safety-critical advanced robots: A survey."
583 *Robotics and Autonomous Systems*. 94 (Aug): 43-52.
584 https://doi.org/10.1016/j.robot.2017.04.004.
585 Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. 2018. "Social gan: Socially acceptable
586 trajectories with generative adversarial networks." In *Proc., 2018 IEEE Conference on*
587 *Computer Vision and Pattern Recognition.*, 2255-2264. Salt Lake City, UT: IEEE.
588 Helbing, D., and Molnar, P. 1995. "Social force model for pedestrian dynamics." *Physical review E*, 51
589 (5): 4282. https://doi.org/10.1103/PhysRevE.51.4282.
590 Kim, D., Goyal, A., Newell, A., Lee, S., Deng, J., and Kamat, V. R. 2019a. "Semantic relation detection
591 between construction entities to support safe human-robot collaboration in construction." *2019*
592 *ASCE International Conference on Computing in Civil Engineering*., 265-272. Atlanta, GA:

ASCE.

Kim, D., Liu, M., Lee, S., and Kamat, V. R. 2019b. "Remote proximity monitoring between mobile construction resources using camera-mounted UAVs." *Automation in Construction*. 99 (Mar): 168-182. https://doi.org/10.1016/j.autcon.2018.12.014.

Kim, D., Liu, M., Lee, S., and Kamat, V. R. 2019. "Trajectory prediction of mobile construction resources toward pro-active struck-by hazard detection." In *Proc., International Symposium on Automation and Robotics in Construction*., 982-988. Banff, AB, Canada.

Kim, P., Chen, J., and Cho, Y. K. 2018. "SLAM-driven robotic mapping and registration of 3D point clouds." *Automation in Construction*. 89 (May): 38-48. https://doi.org/10.1016/j.autcon.2018.01.009.

Lattanzi, D., and Miller, G. 2017. "Review of robotic infrastructure inspection systems." *Journal of Infrastructure Systems*. 23 (3): 04017004. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000353.

Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., and Savarese, S. 2014. "Learning an image-based motion context for multiple people tracking." In *Proc., 2016 IEEE Conference on Computer Vision and Pattern Recognition*., 3542-3549. Las Vegas, NV: IEEE.

Li, J., Wang, Y., Zhang, K., Wang, Z., and Lu, J. 2019. "Design and analysis of demolition robot arm based on finite element method." *Advances in Mechanical Engineering*. 11 (6): 1687814019853964. https://doi.org/10.1177/1687814019853964.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. 2014. "Microsoft coco: Common objects in context." In *Proc., European conference on computer vision.*, 740-755. Zurich, Swiss: Springer

Liu, J., and Li, G. 2018. "Research on the development of 3D printing construction industry based on diamond model." *Innovative Technology and Intelligent Construction.*, 164-176. Reston, VA: ASCE.

Loop, C., and Zhang, Z. 1999. "Computing rectifying homographies for stereo vision." In *Proc., 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.,* 125-131.

620 Fort Collins, CO: IEEE.
621 Memarzadeh, M., Golparvar-Fard, M., and Niebles, J. C. 2013. "Automated 2D detection of
622 construction equipment and workers from site video streams using histograms of oriented
623 gradients and colors." *Automation in Construction*. 32 (Jul): 24-37.
624 https://doi.org/10.1016/j.autcon.2012.12.002.
625 Moon, S., Becerik-Gerber, B., and Soibelman, L. 2019. "Virtual Learning for Workers in Robot
626 Deployed Construction Sites." *Advances in Informatics and Computing in Civil and*
627 *Construction Engineering*., 889-895.
628 Park, J., Marks, E., Cho, Y. K., and Suryanto, W. 2015. "Performance test of wireless technologies for
629 personnel and equipment proximity sensing in work zones." *Journal of Construction*
630 *Engineering and Management*. 142 (1): 04015049. https://doi.org/10.1061/(ASCE)CO.1943-
631 7862.0001031.
632 Park, M.-W., and Brilakis, I. 2012. "Construction worker detection in video frames for initializing vision
633 trackers." *Automation in Construction*. 28 (Dec): 15-25.
634 https://doi.org/10.1016/j.autcon.2012.06.001.
635 Pellegrini, S., Ess, A., and Van Gool, L. 2010. "Improving data association by joint modeling of
636 pedestrian trajectories and groupings." In *Proc., European conference on computer vision.* 452-
637 465. Crete, Greece: Springer.
638 Pfeiffer, M., Paolo, G., Sommer, H., Nieto, J., Siegwart, R., and Cadena, C. 2018. "A data-driven model
639 for interaction-aware pedestrian motion prediction in object cluttered environments." In *Proc.,*
640 *2014 IEEE International Conference on Robotics and Automation.,* 1-8. Brisbane, QLD,
641 Australia: IEEE.
642 Redmon, J., and Farhadi, A. 2018. "Yolov3: An incremental improvement." *arXiv preprint*
643 *arXiv:1804.02767*.
644 Research and Markets. 2019. "Global construction robot market - drivers, restraints, opportunities,
645 trends, and forecast up to 2025." (URL: https://www.researchandmarkets.com, accessed on
646 Sept. 08 2019)

647 Ruff, T. 2006. "Evaluation of a radar-based proximity warning system for off-highway dump trucks."
648 *Accident Analysis & Prevention*. 38 (1): 92-98. https://doi.org/10.1016/j.aap.2005.07.006.
649 Salimans, T., and Kingma, D. P. 2016. "Weight normalization: A simple reparameterization to accelerate
650 training of deep neural networks." In *Proc., 30th Conference on Neural Information Processing*
651 *Systems.*, 901-909. Barcelona, Spain: NIPS.
652 Tavares, P., Costa, C. M., Rocha, L., Malaca, P., Costa, P., Moreira, A. P., Sousa, A., and Veiga, G. .2019.
653 "Collaborative Welding System using BIM for Robotic Reprogramming and Spatial
654 Augmented Reality." *Automation in Construction*. 106 (Oct): 102825.
655 https://doi.org/10.1016/j.autcon.2019.04.020.
656 Tay, M. K. C., and Laugier, C. 2008. "Modelling smooth paths using gaussian processes." In *Proc.,*
657 *Field and Service Robotics*., 381-390.
658 Teizer, J. 2015. "Wearable, wireless identification sensing platform: self-monitoring alert and reporting
659 technology for hazard avoidance and training (SmartHat)." *Journal of Information Technology*
660 *in Construction*. 20 (19): 295-312.
661 Teizer, J., Allread, B. S., Fullerton, C. E., and Hinze, J. 2010. "Autonomous pro-active real-time
662 construction worker and equipment operator proximity safety alert system." *Automation in*
663 *construction*. 19 (5): 630-640. https://doi.org/10.1016/j.autcon.2010.02.009.
664 Tractica. 2019. "Construction & demolition robots - robot assistants and structure, finishing, and
665 infrastructure robots: global market analysis and forecast." (URL:
666 https://www.tractica.com/research/construction-demolition-robots, accessed on Sept. 08 2019)
667 Trautman, P., Ma, J., Murray, R. M., and Krause, A. 2015. "Robot navigation in dense human crowds:
668 Statistical models and experimental studies of human–robot cooperation." *The International*
669 *Journal of Robotics Research*. 34 (3): 335-356. https://doi.org/10.1177/0278364914557874.
670 Tsuruta, T., Miura, K., and Miyaguchi, M. 2019. "Mobile robot for marking free access floors at
671 construction sites." *Automation in Construction*. 107 (Nov): 102912.
672 https://doi.org/10.1016/j.autcon.2019.102912.
673 Vähä, P., Heikkilä, T., Kilpeläinen, P., Järviluoma, M., and Gambao, E. 2013. "Extending automation

of building construction—Survey on potential sensor technologies and robotic applications." *Automation in Construction*. 36 (Dec): 168-178. https://doi.org/10.1016/j.autcon.2013.08.002.

Varghese, J. Z., and Boone, R. G. 2015. "Overview of autonomous vehicle sensors and systems." In *Proc., International Conference on Operations Excellence and Service Engineering*., 178-191.

Vega-Heredia, M., Mohan, R. E., Wen, T. Y., Siti'Aisyah, J., Vengadesh, A., Ghanta, S., and Vinu, S. 2019. "Design and modelling of a modular window cleaning robot." *Automation in Construction*. 103 (Jul): 268-278. https://doi.org/10.1016/j.autcon.2019.01.025.

Wang, M.-z., Luo, M., Cen, Y.-w., and Huang, J.-z. 2018. "Research on Space Pose and Hydraulic System Stability of Remote-Controlled Demolition Robot." In *Proc., 5th International Conference on Information Science and Control Engineering.,* 962-967.

Więckowski, A. 2017. "“JA-WA”-A wall construction system using unilateral material application with a mobile robot." *Automation in Construction.* 83 (Nov): 19-28. https://doi.org/10.1016/j.autcon.2017.02.005.

Xu, Y., Piao, Z., and Gao, S. 2018. "Encoding crowd interaction with deep neural network for pedestrian trajectory prediction." In *Proc., IEEE Conference on Computer Vision and Pattern Recognition.*, 5275-5284. Salt Lake City, UT: IEEE.

Yamaguchi, K., Berg, A. C., Ortiz, L. E., and Berg, T. L. 2011. "Who are you with and where are you going?" In *Proc., IEEE Conference on Computer Vision and Pattern Recognition.,* 1345-1352. Colorado Springs, CO: IEEE.

Yang, Y., Pan, M., and Pan, W. 2019. "Co-evolution through interaction’of innovative building technologies: The case of modular integrated construction and robotics." *Automation in Construction*. 107 (Nov): 102932. https://doi.org/10.1016/j.autcon.2019.102932.

Yu, S. N., Lee, S. Y., Han, C. S., Lee, K. Y., and Lee, S. H. 2007. "Development of the curtain wall installation robot: Performance and efficiency tests at a construction site." *Autonomous Robots.* 22 (3): 281-291.

Table 1. *ADE/FDE* of tuned trajectory prediction models (unit: meters)

Sampling interval (unit: seconds)	Observation length (unit: seconds)		
	2.64	3.96	5.28
0.17	0.85/1.70	0.76/1.63	0.87/1.93
0.33	0.88/1.83	0.45/0.88	0.55/1.14
0.66	0.67/1.38	**0.45/0.79**	0.45/0.81
1.33	0.80/1.59	0.68/1.07	0.56/0.89

Note: left/right values are ADE/FDE, respectively; ADE/FDE in this table are average values of worker, wheel loader, and excavator; prediction lengths of all the models are 5.28 seconds.

Table 2. *ADE* and *FDE* for truck and worker (unit: meters)

Category	*ADE*		*FDE*	
	Worker	Truck	Worker	Truck
Movement pattern #1	1.76	1.84	3.06	2.32
Movement pattern #2	1.44	1.58	2.42	2.21
Movement pattern #3	1.73	2.54	4.68	4.45
Overall	**1.64**	**1.99**	**3.39**	**2.99**

Note: prediction length=5.28 seconds; ADEs and FDEs in this table are the average values for the three trials; overall values are the average for three movement patterns.

Table 3. *APE* and *FPE* between truck and worker (unit: meters)

Category	*APE*	*FPE*
Movement pattern #1	0.44	0.81
Movement pattern #2	1.23	1.94
Movement pattern #3	1.18	2.37
Overall	**0.95**	**1.71**

Note: prediction length=5.28 seconds; APEs and FPEs in this table are the average values for the three trials; overall values are the average for three movement patterns.

Figure Captions