Advanced Engineering Informatics 46 (2020) 101173

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

A context-augmented deep learning approach for worker trajectory prediction on unstructured and dynamic construction sites

Jiannan Cai[a], Yuxi Zhang[b], Liu Yang[b], Hubo Cai[b,*], Shuai Li[c]

[a] Department of Construction Science, The University of Texas at San Antonio, 501 W César E Chávez Blvd, San Antonio, TX 78207, United States
[b] Lyles School of Civil Engineering, Purdue University, 550 Stadium Mall Drive, West Lafayette, IN 47907, United States
[c] Department of Civil and Environmental Engineering, University of Tennessee, Knoxville. 851 Neyland Drive, Knoxville, TN 37996, United States

ARTICLE INFO

Keywords:
Trajectory prediction
Struck-by accident
Deep learning
Contextual information
Long short-term memory (LSTM)

ABSTRACT

Predicting workers' trajectories on unstructured and dynamic construction sites is critical to workplace safety yet remains challenging. Existing prediction methods mainly rely on entity movement information but have not fully exploited the contextual information. This study proposes a context-augmented Long Short-Term Memory (LSTM) method, which integrates both individual movement and workplace contextual information (i.e., movements of neighboring entities, working group information, and potential destination information) into an LSTM network with an encoder-decoder architecture, to predict a sequence of target positions from a sequence of observations. The proposed context-augmented method is validated using construction videos and the prediction accuracy achieved is 8.51 pixels in terms of final displacement error (FDE), with an observation time of 3 s and prediction time of 5 s—5.4% smaller than using the position-based method. Compared to conventional one-step-ahead predictions, the proposed sequence-to-sequence method predicts trajectories over multiple steps to avoid error accumulation and effectively reduces the FDE by 70%. In addition, qualitative analysis is conducted to provide insights to select appropriate prediction methods given different construction scenarios. It was found that the context-aware model leads to better performance comparing to the position-based method when workers are conducting collaborative activities.

1. Introduction

The construction industry is one of the most dangerous industries: it employs only 5% of the US workforce [1] but accounts for 21.1% (1008 deaths) of the total worker fatalities in 2018 [2]. The struck-by accident is a major cause, leading to 804 worker fatalities (18%) in construction from 2011 to 2015 [3]. It is also a single leading cause for non-fatal injuries, accounting for 34% of cases of injuries from 2011 to 2015 [4]. To prevent struck-by accidents, previous studies [5–7] focused on determining the proximity between workers and equipment using sensing technologies and comparing the proximity to predefined thresholds to detect struck-by hazards. Low detection accuracy and reliability are the main challenges attributed to the difficulty in predicting the future movements of jobsite entities while considering the uncertainties of their movements on the unstructured and dynamic construction sites. For instance, warning systems can raise 59% false alarms due to the uncertainty in proximity analysis [8]. As a result, workers may lose confidence in and ignore the alarms, which hinders the efficacy of struck-by prevention systems. According to Luo et al. [9], the estimated response rate of proximity warning systems for generic hazards is about 0.528. Under such a situation, the accurate prediction of worker trajectory provides additional information and is critical to achieving a proactive and informative struck-by prevention system.

Existing studies have created a few methods to predict trajectories of construction resources. Zhu et al. [10] proposed a novel Kalman filter to predict the movements of workers and mobile equipment using positions obtained from multiple video cameras. Dong et al. [11] and Rashid et al. [12] modeled the worker movements as a Markov process to predict their trajectories based on historical records. However, one main challenge in the trajectory prediction of construction entities is the low accuracy over large time horizons because of two interrelated reasons. First, it is insufficient to only consider the previous movements of individual entities when predicting their future trajectories. Since multiple entities co-exist on the construction site, forming various working groups to accomplish different activities [13], their behavior will be influenced by each other and the specific activities they are

* Corresponding author.
E-mail addresses: jiannan.cai@utsa.edu (J. Cai), zhan2889@purdue.edu (Y. Zhang), yang1233@purdue.edu (L. Yang), hubocai@purdue.edu (H. Cai), sli48@utk.edu (S. Li).

https://doi.org/10.1016/j.aei.2020.101173
Received 27 March 2020; Received in revised form 29 August 2020; Accepted 4 September 2020
Available online 14 September 2020
1474-0346/

involved in. To accurately predict worker trajectory, such contextual information must be incorporated. Second, due to the complex and dynamic jobsite context, it is not adequate to capture the worker movement using a pre-defined model with hand-crafted features that may only fit particular scenarios.

A few recent studies [14,15] attempted to predict the construction entity trajectory through a data-driven approach given the advances in deep learning techniques. Despite the promise of deep learning, the rich contextual information regarding working groups and involved activities on construction jobsites have not been fully exploited to better predict worker's trajectory under various construction scenarios. Towards that end, this study proposes a long short-term memory (LSTM)-based, context-augmented deep learning model that integrates both individual movement information and contextual information, including movements of neighboring entities, working group information, and potential destination information. In addition, the proposed method adopts a sequence-to-sequence (seq2seq) neural network architecture that allows the elimination of error accumulation in prediction trajectories over multiple time steps.

The remainder of the paper is outlined as follows. Section 2 describes related studies and limitations. Section 3 introduces the proposed method for context-aware trajectory prediction. Section 4 describes the experiments used to evaluate the technical approaches and analyzes the results. Section 5 summarizes the study, highlights the contribution, and discusses the future direction.

2. Review of related studies

In this section, related studies on proximity-based struck-by prevention and trajectory prediction are reviewed and their limitations are outlined.

2.1. Related studies on proximity-based struck-by prevention

Struck-by accident is one of the leading causes of construction fatalities and has attracted increasing research interest. Many studies developed prevention mechanisms to provide alerts when workers and equipment are too close to each other, as shown in Table 1. Most of them compare the proximity information detected via various real-time locating systems (RTLS) with a pre-defined threshold or statistical hazard zones and provide early warnings when the distance is less than the threshold [5–7,16]. But these approaches only focus on proximity at a snapshot while overlooking the dynamic nature of workers and equipment. Another group of studies [17–21] integrates proximity with more risk factors (e.g., equipment workspace, blind spot information, velocity) to determine the hazard zone. These approaches consider the dynamic and complexity of construction work. However, current approaches detect struck-by hazards and take actions "just" before potential accidents might happen with limited prediction ability, which has a large chance of interrupting normal operation and making incorrect warnings. Therefore, there is a critical need for accurate prediction of worker trajectory, which paves the way for a proactive and

Table 1
Related studies on proximity-based struck-by prevention.

Factors used to detect struck-by-hazards	Hazard zone modeling	Reference
Proximity	Pre-defined threshold	[5–7]
Proximity considering sensor accuracy	Statistical hazard zones	[16]
Proximity and equipment workspace	Line segment intersection algorithm	[17]
Proximity, blind spot information, and velocity	Network-based model	[18]
Proximity and crowdedness	Fuzzy inference method	[19]
Proximity, direction, and velocity	Rule-based model	[20,21]

Table 2
Related studies on trajectory prediction.

Category	Input Features	Model	Application Scenario(s)	Reference
Bayesian filtering	Position, velocity, acceleration	Kalman Filter	Movement of construction workers and equipment/Moving objects	[10,22–24]
	Position, velocity, acceleration considering different motion states (walking and stop)	Switching Linear Dynamical System	Pedestrian behavior	[25]
Probabilistic planning	Latent segments of trajectories	Hidden Markov Model	Construction worker movement	[12]
	Position and change of moving direction with two states (walking and working)	Markov Model	Construction worker movement	[11]
	Positions considering the environment (e.g. obstacles)	Markov Decision Process	Pedestrian behavior	[26,27]
	Position, speed, orientation considering the semantic map and goals	Jump Markov Process	Pedestrian behavior	[28]
	Position, speed, orientation considering goals and social force	Joint Sampling Markov Decision Process	Human motion	[29]
Data-Driven approaches	Position	Three stacked layers of LSTM	Pedestrian behavior	[30]
	Position and occupancy map	Social-LSTM	Human motion in crowded space	[31]
	Position, occupancy map, and scene features	Social-Scene-LSTM	Pedestrian/human motion in crowded space	[32,33]
	Position considering the social interaction via social pooling layer	Social Generative Adversarial Network (GAN)	Movement of construction workers and equipment	[15]
	Position, occupancy map and entity type	Encoder-decoder LSTM	Movement of construction workers and equipment	[14]

informative struck-by prevention mechanism.

2.2. Related studies on trajectory prediction

Trajectory prediction is an essential yet challenging task in the computer vision community and has been increasingly studied in applications such as pedestrian behavior analysis due to the emergence of autonomous vehicles. There are typically three types of approaches in trajectory prediction, i.e., Bayesian filtering, probabilistic planning, and data-driven approaches. Table 2 summarizes related studies on trajectory prediction, including the features and models used for prediction as well as the application scenarios.

Bayesian filtering methods [10–12,22–25] explicitly model the movement dynamics as mathematical models, such as Kalman/Particle Filters and Hidden Markov Models, and are traditionally applied to predict trajectories. However, these approaches often result in physically impossible locations (e.g., behind walls, within obstacles). Additionally, Bayesian filtering methods rely on simplified models and hand-crafted states with parameters estimated from historical records/observations, which may only fit particular scenarios and simple movements. Probabilistic planning methods [26–29] treat entities as intelligent agents who actively plan their motion/path to achieve a goal. The problem is formulated as a path planning or optimal control task, such as the Markov decision process (MDP). The optimal policy is determined by maximizing some inherent reward functions. These approaches can incorporate context information, such as a semantic map and social force, but they still use hand-crafted features to model states and reward functions that are suitable to particular settings.

Recently, with the advances in deep learning techniques, the data-driven approach [14,15,30–33] has been increasingly used given that it does not require explicitly modeling movement dynamics and that it can be generalized to various scenarios. The problem is usually formulated as a time-series regression problem. Traditionally, only past movements of individual entities are used as inputs to predict future trajectory [30], which is insufficient to capture human behavior under different scenarios, especially when human behavior is influenced by the environment. Recent studies in the computer vision community have recognized the significance of context information and considered various contextual features to predict pedestrian trajectory. For instance, Alahi et al. [31] created a social-LSTM model and proved that the pedestrian trajectory can be better predicted by incorporating the interaction among multiple pedestrians. Xue et al. [32] and Syed and Morris [33] incorporated the occupancy map and scene features in the trajectory prediction.

Very few studies have incorporated the contextual information in trajectory prediction in the construction domain. Kim et al. [15] applied a hyper-parameter tuned Social GAN to predict trajectories of construction entities in 5 s. Tang et al. [14] developed an LSTM network that integrates entity type (i.e., worker and equipment) and occupancy maps of the construction site to predict entity trajectory in up to 2 s. Despite these pilot studies, the trajectory was predicted only in one specific job setting with entities conducting a specific activity. There remains a critical need to exploit the contextual cues that are effective to predict the entity trajectory under general construction jobsite scenarios. To close this gap, this study proposes an LSTM-based, context-augmented model that integrates both individual movement information and contextual information, including movements of neighboring entities, relationship with neighboring entities (i.e., within one group or not in one group), and potential destination, to accurately predict the trajectory of construction workers.

3. Methodology

In this study, a context-aware LSTM-based method has been designed to predict worker trajectories using visual data that contain rich contextual information. Entity movement and contextual information are incorporated in the LSTM-based seq2seq neural network for trajectory prediction. Fig. 1 illustrates the overall framework. This method consists of two major steps: Step 1—contextual information formulation and Step 2—LSTM-based seq2seq trajectory prediction.

In the first step, contextual information regarding the interaction between the entity and its nearest neighbor, and the potential destination is considered. Specifically, the contextual information is represented by three features, the neighbor position, the relationship with the neighbor (i.e., group/not a group), and the distance from potential destination. In our previous studies [13,34], it was found that the interactions among construction entities can be modeled using positional and attentional cues and further used to reason about the construction working group and corresponding group activity. This forms the technical foundations to formulate the contextual features in this study. In the second step, the above features are concatenated and fed into an LSTM encoder that encodes the information regarding both entity movements and jobsite contexts during the observation time. The encoded information is then fed into an LSTM decoder that generates a sequence of estimated positions during the prediction period. In this way, the proposed method takes into account the construction job contextual information and avoids the error accumulation when predicting trajectory over multiple time steps.

3.1. Problem formulation

Construction sites are complex and dynamic, where multiple entities coexist and form different working groups to collaborate on various activities. Fig. 2 illustrates a real construction scenario with potential struck-by hazard, where three workers (in blue dotted bounding boxes) are guiding the bulldozer (in yellow dashed bounding box) to roll over a path while two workers (in red solid bounding boxes) are walking across the workplace. Their moving directions, indicated by the arrows, present a potential conflict with the bulldozer. As construction workers may be distracted by their allocated tasks and surrounding noises, they

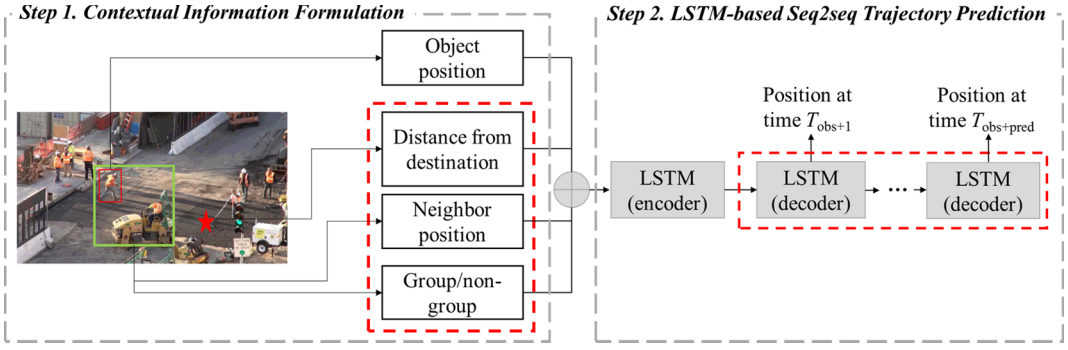


Fig. 1. Overall Framework.

Fig. 2. Construction scenario with potential struck-by hazard.

may fail to recognize the approach of other entities. Therefore, given the current positions of construction entities and the jobsite context, it is important to predict entity future movements so that the potential collision between entities can be proactively detected and avoided.

Construction videos are used as the data source for trajectory prediction given its increasing availability on jobsites and its capability of providing rich contextual information. Entity position is captured by the mid-bottom point of its bounding box on the 2D image plane. As a result, at any time step t, the i^{th} entity on the jobsite is represented by its pixel coordinates on the image plane, i.e., (x_t^i, y_t^i). The inputs are the observation of site dynamics from time step 1 to time step T_{obs}, including trajectories of all entities, i.e., $(x_{1:T_{obs}}^{1:N}, y_{1:T_{obs}}^{1:N})$, and the jobsite contexts, i.e., $\mathbf{f}_{1:T_{obs}}^{context}$, where N is the total number of entities in the scene, and the subscript represents the trajectory or context during the specific time period. The objective is to predict the future trajectory of target entity i from time step T_{obs+1} to $T_{obs+pred}$, denoted as $(x_{T_{obs+1}:T_{obs+pred}}^{i}, y_{T_{obs+1}:T_{obs+pred}}^{i})$. Inspired by [15], the prediction time is set as 5 s assuming it would be enough for entities to take action. The observation time is set as 3 s. The ratio of prediction and observation time will also vary in the experiments to further analyze the influence of prediction time.

Different from previous studies [14,31] which only observe entity positions and implicitly incorporate the interactions among entities using hidden states learned from deep neural networks, this study explicitly models the contextual information $\mathbf{f}_{1:T_{obs}}^{context}$ (including entity interaction and potential destination) on the jobsite, as detailed in Section 3.2. Note that it is assumed the visual data are first preprocessed to obtain entity positions and contextual features, consistent with most of the related studies [14,15,31,32].

3.2. Contextual information formulation

Construction entities (including both workers and equipment) interact with each other, constituting working groups to accomplish assigned tasks. It is expected that the worker's behavior will be influenced by other entities as well as the involved construction activity. The rationale is that construction workers tend to avoid obstacles to prevent potential collisions, while staying close to their co-workers or group members to conduct the activity collaboratively. Meanwhile, the worker's movement is typically within the workspace specified by their involved activity, which indicates their potential destination. The specific contextual features considered in this study include neighbor position, group relationship with the neighbor, and distance to potential destination.

3.2.1. Neighbor position

It is not uncommon that the positions of other entities in the scene are incorporated to reflect their interactions with the target entity when predicting its trajectory. A conventional approach is to construct an occupancy map of the scene or within a certain area of the target entity to represent the existence of other entities [14,31]. The main drawback is that if the grid size is large, resulting in coarse occupancy map, the dynamic changes of entity positions cannot be effectively reflected, especially when entity movement is not substantial across consecutive time steps, such as on construction sites; if the grid size is small, resulting in fine occupancy map, only a few grids will be occupied by entities, which leads to very sparse occupancy map, i.e., most values are zero.

In contrast, this study directly uses neighbor position information as one contextual feature. Note that, only the position of the entity's

Fig. 3. Pixel coordinates of construction entities (Entity i is the target, j is its nearest neighbor).

nearest neighbor is considered in order to ensure the same dimensional features in different scenarios. It is reasonable as entities are more likely to be affected by others who are spatially closer to them. Fig. 3 illustrates an example of entity locations in the image coordinate system, where the positions of construction entities are represented by the pixel coordinates of the mid-bottom points of their bounding boxes. At any time step t, positions of all entities (from 1 to N) are observed, denoted as (x_t^k, y_t^k), $k \in 1. ..N$. Then, the distance between any two of the entities is calculated as the Euclidian distance between their pixel coordinates. As a result, the position of the nearest neighbor of Entity i can be easily denoted as $(x_t^j, y_t^j), j = \arg\min \|(x_t^i - x_t^k, y_t^i - y_t^k)\|$, $k \in 1. ..N$, $k \neq i$. The locations of construction entities can be automatically obtained using vision-based object tracking methods created in some existing studies [35,36]. However, in this study, in order to exclude the impact of the possible errors in object tracking, the construction images are manually annotated to draw the bounding boxes and extract the pixel coordinates.

3.2.2. Group relationship with neighbor

In addition to the neighbor position, the relationship between an entity and its neighbor in terms of whether they belonging to the same working group also influences entity movement. For instance, workers tend to avoid entities that are not in the same group to prevent potential conflict, while they tend to have similar movement patterns with their co-workers. However, such scenarios are not differentiated, and the group information has been overlooked in current studies. The group relationship between an entity and its nearest neighbor is considered as a second contextual feature. Two entities are considered belonging to one working group if they are interacting with each other during the construction, and the group relationship feature is set as "1". Otherwise, they are considered not belonging to the same group with feature value being "0".

The working group is identified by integrating positional and attentional cues via an LSTM-based method created in our previous study [13]. The workflow is as follows.

1. Spatial and attentional states of construction entities from construction videos are represented as numerical values, as shown in Fig. 4(a). The spatial state refers to an entity's real-time position on the image plane, represented by the pixel coordinates of the central point of the bounding box. The attentional state refers to the direction of an entity's visual attention, captured by head pose, body orientation, and body pose. Specifically, the worker's head yaw and body orientation are categorized into eight discrete classes: north (N) – 1, south (S) – 2, east (E) – 3, west (W) – 4, northeast (NE) – 5, northwest (NW) – 6, southeast (SE) – 7, and southwest (SW) – 8, as shown in Fig. 4(b) and (c). The head pitch is categorized into three discrete classes: looking up (U) – 2, looking horizontally (H) – 1, and looking down (D) – 0, as shown in Fig. 4(d). Note that the equipment is simplified as rigid objects, and the main cab is treated as its "head". Thus, for equipment, the body orientation is identical to the head yaw and the head pitch always remains horizontal.
2. Positional and attentional cues are computed from the spatial and attentional states of two entities to model their interaction, which are critical features for working group identification. Five positional cues are modeled: 1) distance relationship—modeled as the topological relationship between the bounding boxes of two entities using the 9-Intersection model [37] (see Fig. 5(a)), where numerical value for each relationship is assigned based on topological distance [38].; 2) directional relationship—modeled as eight regions to measure the relative direction between two entities based on the project-based model [39] (see Fig. 5(b)); 3) difference in speed—computed as $\Delta v_t^{i,j} = abs(v_t^i - v_t^j)/max(v_t^i, v_t^j)$, where v_t^i is the speed of entity i at time t, computed as $v_t^i = \sqrt{(x_{t+1}^i - x_t^i)^2 + (y_{t+1}^i - y_t^i)^2}$; 4) difference in moving direction—computed as $\Delta\theta^{i,j} = \min\{abs(\theta^i - \theta^j), 8 - abs(\theta^i - \theta^j)\}$, where θ^i is the moving direction of entity i, represented as the numerical values in Fig. 5(b), and 5) difference between moving direction and relative direction—computed similarly to the previous cue but measures the degree of entity i moving towards entity j.
   In addition, six attentional cues are modeled: 1) difference between head yaw and relative direction—to measure the gaze exchange between two entities, where head yaw is represented based on Fig. 4(b) and relative direction is represented using Fig. 5(b). 2) difference in head yaw—to measure the joint attention of two entities, 3) difference between head yaw and moving direction, 4) difference between head yaw and body orientation—both 3) and 4) are used to model the change of visual attention of an individual entity, 5) head pitch, and 6) body pose—both 5) and 6) are special cues on construction jobsites to reflect worker's visual attention, where the head pitch is modeled as Fig. 4(d), and body pose is considered as either standing – "1" or bending – "2" for workers.
3. The above positional and attentional cues are concatenated into time-series features and fed into an LSTM network followed by a two-node fully connected layer for working group identification. The readers are referred to [13] for the detailed method.

3.2.3. Distance to potential destination

On construction sites, worker behavior is goal-based and purposeful, motivated by their involved activities. It is expected that the worker will inherently move towards the potential destination. Thus, the distance between worker's current position and the potential destination is treated as a third contextual feature, illustrated in the construction image in Fig. 1, where the red bounding box represents the target entity, and the "star" sign represents the destination. It is assumed the destination is time-invariant during a short period of time. Given time step t, the distance from the target to the destination is used as a contextual feature to incorporate the temporal dynamics, denoted as $(\Delta x_t^i, \Delta y_t^i) = (|x_t^i - x^{dest}|, |y_t^i - y^{dest}|)$, where (x_t^i, y_t^i) is the entity location, (x^{dest}, y^{dest}) is the pixel coordinates of the destination. This study simplifies the destination as prior knowledge to examine its influence on worker trajectory prediction. In practice, the potential destination can be inferred from the involved activity and the corresponding

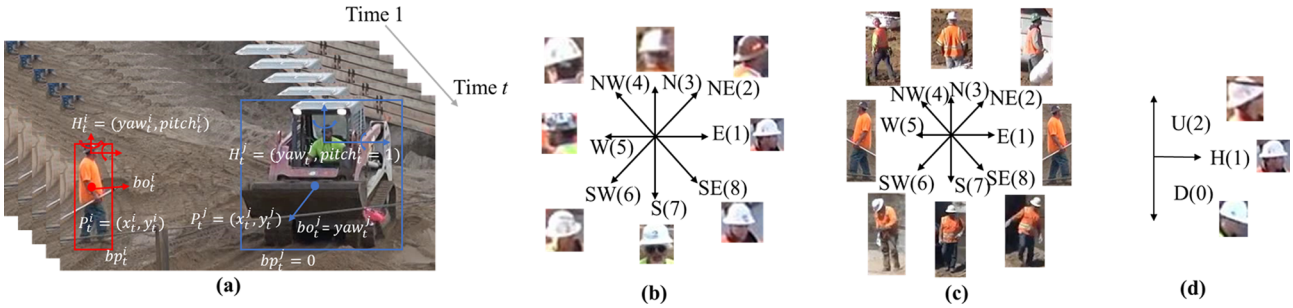


Fig. 4. Construction entity state representation: (a) Example of spatial and attentional state, (b) head yaw, (c) body orientation, (d) head pitch.

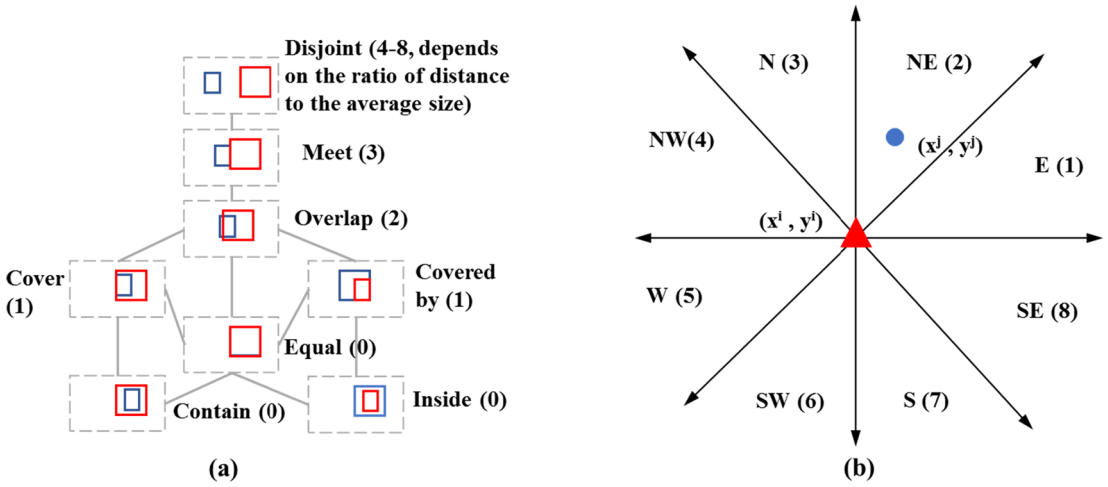


Fig. 5. Numerical representation of topological and directional relationships between two entities: (a) topological relation, (b) directional relation.

workspace, where ongoing activity can be automatically learned from visual data and workspace can be acquired from site layout or building information model.

3.3. LSTM-based sequence-to-sequence (seq2seq) trajectory prediction

LSTM network [40] is a typical recurrent neural network (RNN) and can be used to model temporal dependency among sequential features. It has been successfully applied to many sequential problems such as natural language translation and activity recognition. Fig. 6 illustrates a typical LSTM network that takes time-series features $\{x_1, x_2, ...,x_n\}$ as input. The LSTM network consists of several cells ordered sequentially, each of which has the same structure with three gates, i.e., input gate, forget gate, and output gate, to control the information flow within the cell. At time step t, the cell state is determined by both the input of the current time step and the output from the previous time step, updated using Eq. (1).

$$\begin{cases} i_t = \delta(W_{xi}x_t + V_{hi}h_{t-1} + b_i) \\ f_t = \delta(W_{xf}x_t + V_{hf}h_{t-1} + b_f) \\ o_t = \delta(W_{xo}x_t + V_{ho}h_{t-1} + b_o) \\ g_t = \tanh(W_{xc}x_t + V_{hc}h_{t-1} + b_c) \\ c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \\ h_t = o_t \otimes \tanh(c_t) \end{cases} \quad (1)$$

where x_t is the input, i_t, f_t, o_t are the input gate, forget gate, and output gate at time t respectively. $h_t$is the hidden state with N hidden units ($N = 25$ in this study) and is also the output of this cell, and c_t is the cell state. g_t is the input modulation that adds information to the cell state. δ is the sigmoid function and $\otimes$ represents element-wise multiplication. W_{xi}, W_{xf}, W_{xo}, W_{xc}, V_{hi}, V_{hf}, V_{ho}, W_{hc}, b_i, b_f, b_o, b_c, are the learnable parameters for each LSTM cell that control the level of information transferred from previous time steps as well as the level of information taken from the current time step.

Recently, LSTM network has been widely used in data-driven trajectory prediction. As shown in Fig. 7, a conventional approach [30,31] is that 1) in the training process, the model is fed with time-series inputs and trained to output one-step prediction; and 2) in the inference process, the observations from time step 1 to T_{obs} are fed into the trained model and the position in the next time step T_{obs+1} is estimated. Then, the estimated position at time T_{obs+1} is used as input along with observations from time 2 to T_{obs}, to predict for time T_{obs+2}, which happens recursively till $T_{obs+pred}$. Under such a case, the model only predicts one step each time and the predicted result is used as inputs recursively in order to generate a sequence of positions over multiple time steps. This practice leads to large error accumulation.

To solve this problem, this study adopts the LSTM encoder-decoder architecture, which allows the generation of a sequence with arbitrary length from a given sequence and was first introduced in machine translation tasks [41]. Fig. 8 illustrates the proposed model. In this method, the entity position during observation time and the corresponding contextual features (discussed in Section 3.2) are concatenated into a 7-dimensional feature vector, denoted by $\mathbf{X}_t = [obj\_x, obj\_y, dis\_x, dis\_y, neighbor\_x, neighbor\_y, group]$, where first two dimensions represent object (target) positions in x and y directions; third and fourth dimensions represent the distance from the destination in x and y directions; fifth and sixth dimensions represent neighbor positions; and the last dimension indicates the group information. This feature vector describes the object position and the jobsite context at any given time. The time-series feature is constructed by chaining a series of time-variant feature vectors over a time period, denoted by $\{\mathbf{X}_t, \mathbf{X}_{t+\Delta t}, \mathbf{X}_{t+2\Delta t}, ...,\mathbf{X}_{t+T}\}$, where t is the starting time, Δt is the temporal resolution and T is the time duration of observation. In this study, features that represent position and distance information are

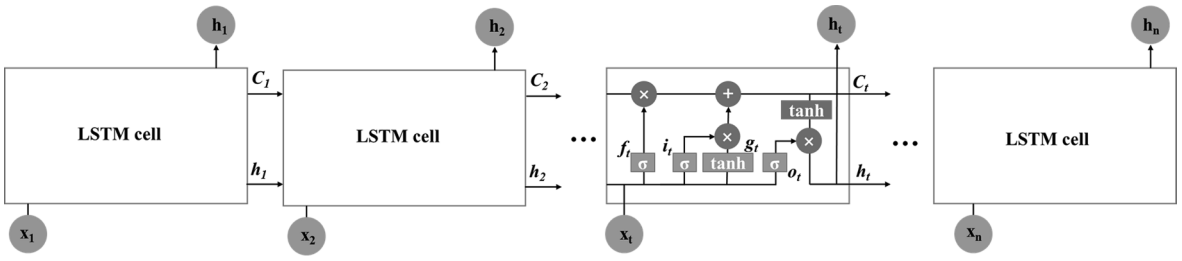


Fig. 6. LSTM network and LSTM cell.

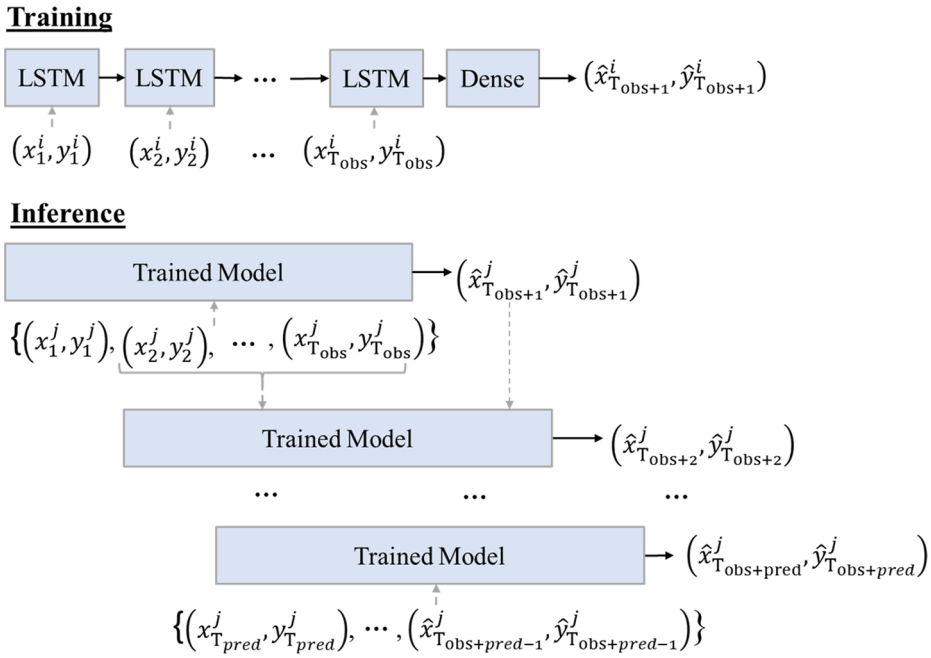


Fig. 7. Conventional LSTM-based recursive approach for multi-step prediction.

in pixels with the range depending on image size, while the group information is binary (either 0 or 1). The time-series features are normalized to the range [0, 1] in data processing to ensure the same scale of the features, and serve as the inputs of LSTM encoder.

The encoder outputs an encoded vector (i.e., the hidden state of the final encoder LSTM cell) that encapsulates the information from the observed movements and jobsite context. The encoded vector is used to initialize the states in LSTM decoder which allows the integration of previous information for better prediction of future trajectory. The hidden state of each LSTM cell in the decoder is considered as the output of the corresponding time step, which is further fed into a dense layer with two nodes. The dense layer essentially performs a linear regression, resulting in estimated positions from time T_{obs+1} to $T_{obs+pred}$.

Similar to Saleh et al. [30], the network is trained by minimizing one of the most commonly used loss functions, i.e., mean squared error (MSE) loss function [42], using *Adam* optimizer [43]. The MSE is computed as $MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2$, where N is the size of training data, $\hat{Y}_i$ and Y_i are the predicted and actual i^{th} trajectory.

4. Implementation and results

4.1. Implementation

The dataset used to test the proposed method is introduced and the implementation details are described. Two evaluation metrics are also explained to assess the prediction performance.

4.1.1. Data description

To demonstrate the proposed method, ten construction videos were collected from three projects: a hospital construction project from the publicly-available website – YouTube [44], and two building projects videotaped by the authors, respectively. The videos consist of a total of 84 workers in different construction scenarios, conducting various activities in different working groups. All videos were down-sampled to 2fps, similar to other studies [31,32] on pedestrian trajectory prediction using surveillance video. Fig. 9 illustrates some images from the dataset.

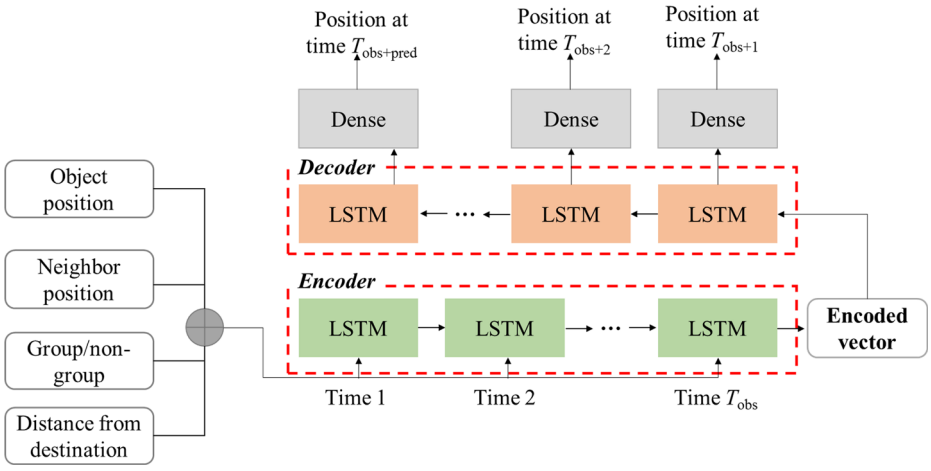


Fig. 8. Context-aware LSTM-based seq2seq model.

Fig. 9. Sample images: (a)-(b) from hospital project, (c) from building project 1, (d) from building project 2.

4.1.2. Data preparation

Visual data were pre-processed to extract entity positions and contextual features, which are then used as inputs to train and test the proposed method. First, all entities (workers and equipment) are manually annotated using bounding boxes with pixel coordinates of the mid-bottom points representing their positions on the images. Second, the nearest neighbor of each worker is identified by computing the distances between any two entities. It is noted that although neighboring entities may include both workers and equipment, only workers are considered as target entities for trajectory prediction because of the data constraint—most of our dataset involves only workers. However, the proposed method can be easily extended to equipment by training a different model using equipment movement data. As the movement patterns for workers and equipment are expected to be different, it is better to treat them separately [14]. In future study, we will implement the proposed method for equipment trajectory prediction by extending the dataset with more equipment movements.

Third, the group information is manually labeled based on a period of observations. Two entities are considered belonging to one working group if they are interacting during the construction, and are labeled as "1". Otherwise, they are considered working independently, labeled as "0". As explained in Section 3.2.2, this information can be automatically obtained from positional and attentional cues using the method created in our previous study [13]. Note that it is also possible to use construction planning and schedules to extract group information. However, in reality, workers may not always follow what is planned due to the complexity and uncertainty of construction work and identifying the working group in an automatic approach provides real-time information. In this study, we use manually annotated group information to exclude the possible errors in an automatic approach and focus on evaluating the influence of contextual information. A promising method is to integrate the planned and the actual information to determine the workspace and group work.

Finally, the potential destination of workers, simplified as prior knowledge in this study, is determined as their final position in the scene, based on which the dynamic distance from worker to the potential destination is computed in both x and y directions. Because the focus of this study is trajectory prediction by integrating position and contextual information, preprocessed higher-level information (i.e., extracted location and contextual features) is used to exclude the impact from possible errors caused by automatic worker localization and group identification. This practice also aligns with relevant studies on trajectory prediction in both construction and other domains [14,15,31].

As a result, a total of 241 trajectories with various lengths were obtained for 84 workers. The length of observation was set as 3 s (i.e., 6 frames) and prediction length as 5 s (i.e., 10 frames), which is consistent with relevant studies ([31,32]) on pedestrian trajectory prediction. Correspondingly, the 241 trajectories were trimmed into tracks using a sliding window with a fixed length of 8 s (i.e., 16 frames). To augment the dataset, the sliding window starts from every other frame of the original trajectory, resulting in 3640 tracks (tracks that are less than 16 frames were excluded).

4.1.3. Implementation details

The proposed method is implemented using Keras library on top of Tensorflow platform, on a desktop with 3.6 GHz Intel i9-9900 K CPU, 32 GB, and NVIDIA GeForce GTX 2080 Ti GPU. The dataset is randomly split into training set (80%), validation set (10%), and testing set (10%). The network is trained with *Adam* optimizer [43], with a learning rate of 0.001, batch size of 20, and dropout of 0.5. In the experiments, different combinations of the above hyperparameters, as well as the number of hidden units, were used and the optimal ones that result in the highest accuracy in validation set were selected. To prevent overfitting, early stopping criterion is used, i.e., if the total loss on validation set does not decrease for 100 epochs, then the model will be terminated and the checkpoint that leads to the smallest loss on the validation set will be saved; otherwise, the model will stop after 1000 epochs. Moreover, the model is trained on the training set, evaluated on the validation set for early stopping and optimal hyperparameter selection, and tested on the testing set to assess the performance of the proposed method.

4.1.4. Evaluation metrics

Two evaluation metrics – final displacement error (FDE) and average displacement error (ADE) – are selected because they are the most widely used evaluation metrics in trajectory prediction studies in the construction domain [14,15] as well as other applications such as pedestrian analysis [31–33]. FDE is the MSE between the final predicted location and the final actual location of all testing data, computed as $FDE = \frac{\sum_{i=1}^{N} \| \hat{y}_T^i - y_T^i \|}{N}$, where N is data size, $\hat{y}_T^i$ is the final predicted location for i^{th} data, and y_T^i is the final actual location for i^{th} data. It measures the accuracy in predicting an entity's final location, which is critical in predicting the proximity between two entities and detecting potential collisions. ADE is the MSE over all locations of predicted trajectories and the actual trajectories, computed as $ADE = \frac{\sum_{i=1}^{N} \sum_{t=0}^{t=T} \| \hat{y}_t^i - y_t^i \|}{N \times T_{pred}}$, where T_{pred} is the prediction duration. It measures how close the predicted and actual trajectories are and is critical to ensure the accuracy of the overall predicted trajectory.

In this study, the entity position is captured by the mid-bottom point of its bounding box on the 2D image plane. Therefore, the predicted positions and ground truth positions are represented in pixel coordinates, resulting in FDE and ADE in pixels values. The 2D pixel coordinates on the image plane can be projected onto the world plane (i.e., ground plane) via a projective transformation (i.e., homography). To compute the transformation matrix between two planes, at least four pairs of corresponding points are needed in both planes using the Direct Linear Transformation (DLT) algorithm [45]. In this work, because the construction videos are collected from different sources including public website, the actual point locations on the jobsites are not available, and thus the FDE and ADE in pixels are used for evaluation. In our future study, the dataset will be expanded to include videos with known ground control points to predict the trajectory in the world coordinate system.

4.2. Results

The result of the proposed method is compared with that obtained

Table 3
Three LSTM-based models for comparison.

Model	Input features	Rationale for multi-step forecasting
Position (recursive)	Time-series positions	The model can only predict one-step ahead, and achieve multi-step prediction by conducting inference process recursively (see Fig. 7)
Position (seq2seq)	Time-series positions	Encoder-decoder architecture to enable multi-step forecasting (see Fig. 8)
Position + Context (seq2seq) (**proposed in this study**)	Time-series positions and contextual features	

Fig. 10. Example results of trajectory prediction.

using two other LSTM-based models: (1) a baseline model that recursively predicts trajectory based on object positions; and (2) a seq2seq model that predicts trajectory over multiple time steps simultaneously based on object positions. Table 3 lists the differences in three models.

Fig. 10 illustrates two example results of trajectory prediction. The proposed method results in the predicted trajectory being the closest to the ground truth. The position-based seq2seq model leads to a trajectory with a slightly larger discrepancy compared to the proposed method. In contrast, the position-based recursive model has the largest discrepancy from the ground truth trajectory due to the error accumulation.

4.2.1. Quantitative prediction results

Table 4 lists the quantitative results from the three models. The recursive approach leads to much larger errors in both FDE and ADE compared to the seq2seq approaches, which proves that the seq2seq model is an effective way to avoid error accumulation when predicting trajectory over multiple time steps. More specifically, the position-based seq2seq model results in a 68.2% and 41.9% reduction in FDE and ADE, respectively, and the proposed context-augmented seq2seq model leads to reduction of 70.0% and 41.6%, compared to the recursive model. The context-augmented model results in smaller FDE but a slightly larger ADE compared to the position-based model. This is because by incorporating contextual information, especially the potential destination information, the model is inherently trained to adapt more to the long-term goal, rather than accurate prediction of each step. It is reasonable because the final displacement is more critical in predicting the struck-by hazard in safety management.

4.2.2. Qualitative analysis

The results from two seq2seq models, i.e., position-based seq2seq model and context-augmented seq2seq model, are analyzed qualitatively to evaluate the impact of contextual information and identify the scenarios, under which integrating contextual information leads to better performance. Specifically, for each testing data, predicted trajectories obtained using context-aware and position-based methods are plotted against the ground truth trajectories that are manually annotated. Then, the scenarios are categorized based on whether or not context-aware method perform better than position-based method by visually inspecting each plot, examining the overall trend in the plot, and checking back with the corresponding construction videos. Some representative plots are shown in this section to illustrate the main findings.

Table 4
Quantitative results from three models.

Model	FDE (pixel)	ADE (pixel)
Position (recursive)	28.32	15.41
Position (seq2seq)	9.00	**8.95**
Position + Context (seq2seq)	**8.51**	9.00

It was found that when workers are walking continuously and not involved in specific collaborating activities, contextual information does not have a significant influence and both models result in relatively accurate prediction, as shown in Fig. 11. On the other hand, if the target is collaborating with others or involved in certain activities, incorporating contextual information leads to better prediction (see Fig. 12). In Fig. 12(a), the target intends to move towards his co-worker, who is working at the left-bottom corner of the image. With contextual information, especially the position and the relationship with the nearest neighbor, the context-aware model accurately predicts the behavior of the target moving towards his neighbor, resulting in a path closer to the actual trajectory. In contrast, the position-based model only considers individual movement patterns and is more likely to end up with a near-linear trajectory, which is farther from the actual trajectory. In Fig. 12(b), the target is conducting road paving activity with a roller and other co-workers. Although there remains some discrepancy with the actual trajectory, the context-aware model accurately predicts the trend of worker movement, whereas the position-based model predicts the movement in the opposite direction.

In some cases, however, the proposed method may fail. Fig. 13(a) illustrates when the status of target significantly changes during prediction time (e.g., from stationary to moving and vice versa), the movement cannot be accurately predicted. In addition, it is also very challenging when workers are conducting activities within a limited workspace without substantial movement, as shown in Fig. 13(b).

To sum up, when the target is continuously moving but not making interactions with other entities, the context information is mainly related to target's positions, and thus the proposed context-augmented

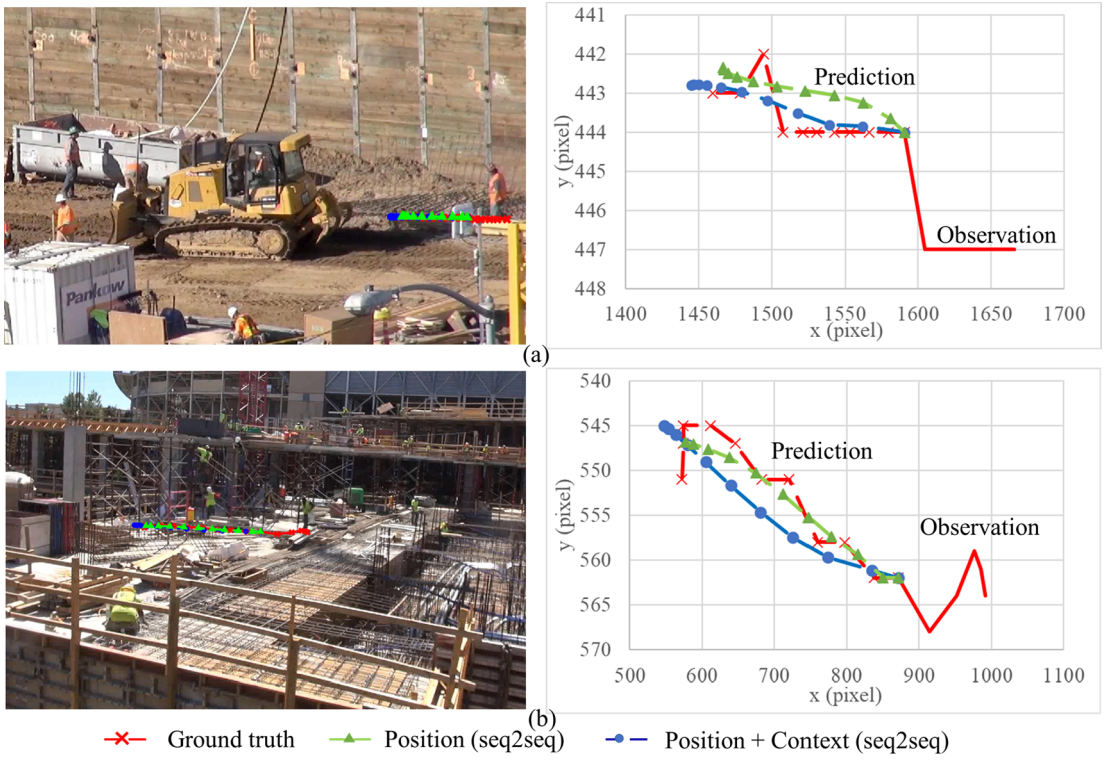


Fig. 11. Two seq2seq models lead to similar results under moving scenarios.

seq2seq model results in similar accuracy with the position-based seq2seq model. When there are interactions between targets and the surrounding entities, e.g., the target is collaborating with others or involved in certain activities, there is rich contextual information about the target's group relationship with the nearest neighbor, the neighbor's position, and the distance to the destination. By incorporating this context information, the model is provided with additional features and thus achieves better prediction compared to the position-based model.

On the other hand, a sudden change of target (e.g., from stationary to moving) during the prediction time would lead to the failure of both models. This is because, essentially, the seq2seq model is using a sequence of movements to predict the next sequence, while a sudden change will break the pattern learned in the observed sequence. Additionally, when workers are conducting activities within a limited workspace without substantial movement, it is very challenging for the models to differentiate a sequence of movements from near-stationary

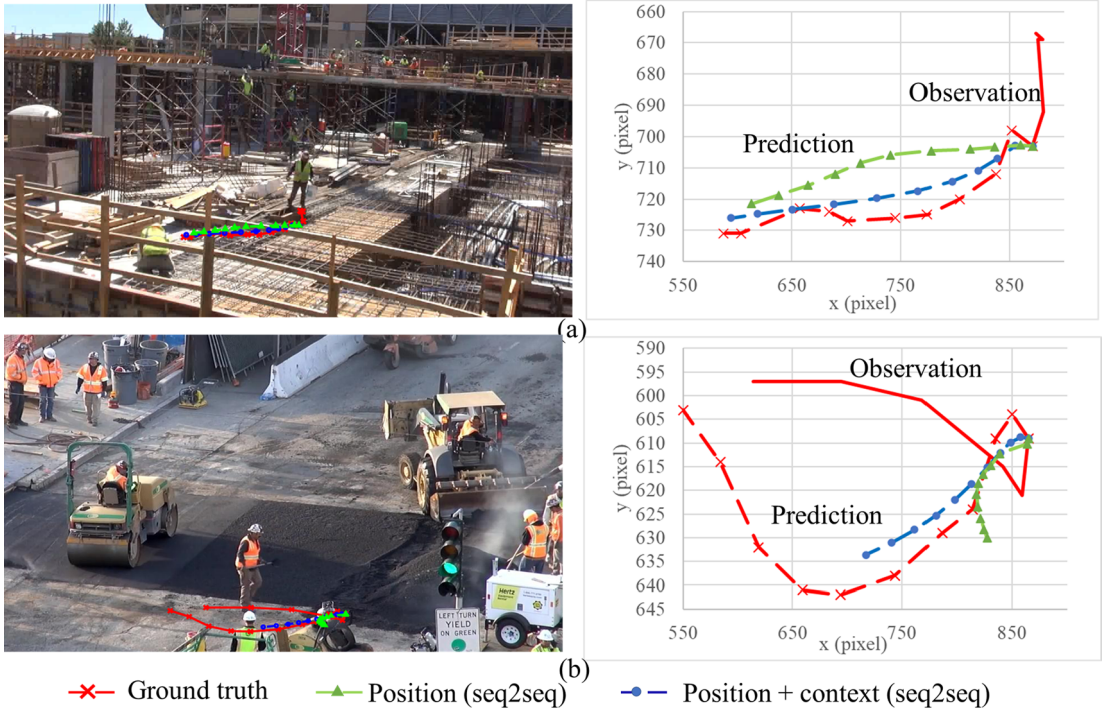


Fig. 12. Context-augmented model leads to better prediction.

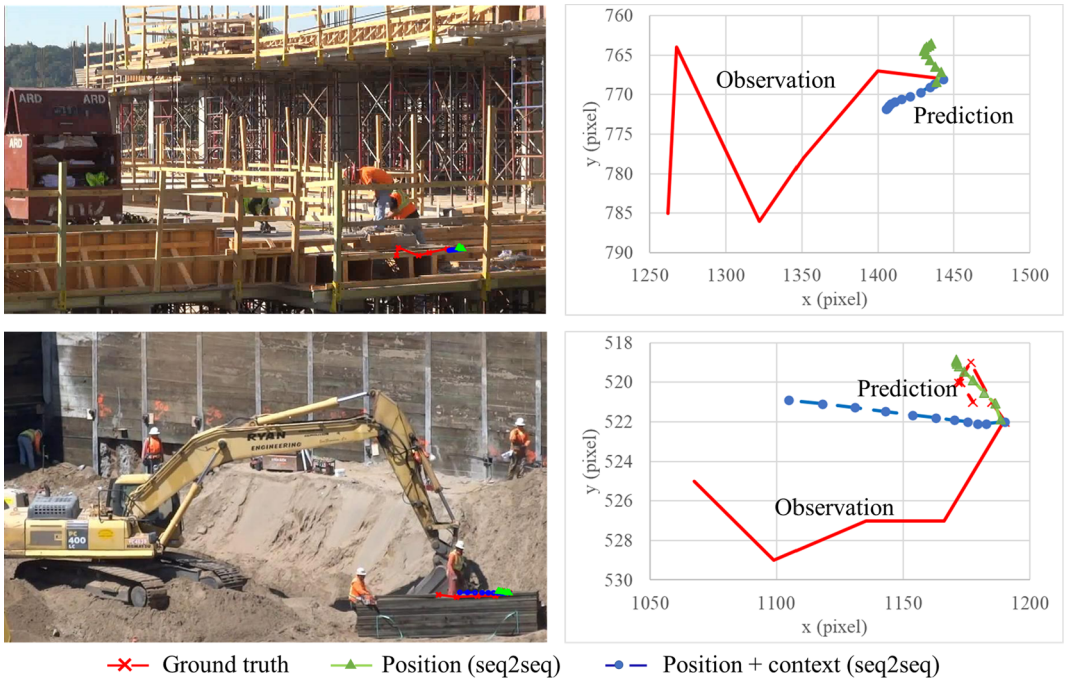


Fig. 13. Examples when context-augmented model fails.

status and make accurate predictions.

4.2.3. *Influence of prediction time*

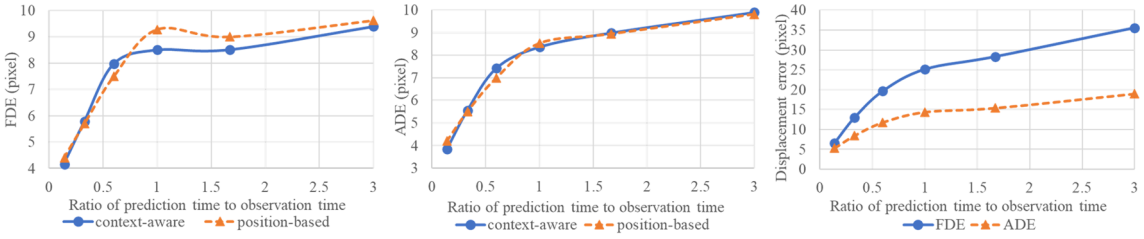
To evaluate the influence of prediction time on different methods, this study examines the prediction performance with respect to various ratios of prediction to observation length within the 8-s track prepared in the dataset. Specifically, the partition of observation time and prediction time varies as 7 s/1s, 6 s/2s, 5 s/3s, 4 s/4s, 3 s/5s (used in the previous experiment), and 2 s/6s. The results are illustrated in Fig. 14. It is not surprising that both FDE and ADE increase as the ratio of prediction to observation increases for all three prediction models, which further proves the challenge in long-term trajectory prediction (i.e., when prediction time is no less than observation time).

From Fig. 14(a), the context-aware model generally results in a smaller FDE compared to the position-based model, especially when the length of prediction is no less than the length of observation—the FDE of the context-aware model is 8.4%, 5.4%, and 2.4% smaller than that of the position-based model when the ratio of prediction to observation time is 1, 1.67, and 3, respectively. The two models lead to compatible ADE based on Fig. 14(b). From Fig. 14(c), the discrepancy between FDE and ADE for the position-based recursive model becomes much larger as the increase of the ratio, compared to those in two seq2seq models (Fig. 14(a) and (b)). It proves the advantage of seq2seq architecture in mitigating the error accumulation for long-term trajectory prediction. In the comparison of position-based and context-aware seq2seq models, the FDEs for both models are compatible in short-term prediction (i.e., when the ratio is less than 1). However, the context-aware method leads to lower FDE in long-term prediction.

5. Conclusions and discussion

Predicting workers' trajectories on unstructured and dynamic construction sites has great potential to improve workplace safety. It provides rich information and is critical to pro-actively prevent struck-by accidents, which has been a major cause of construction fatalities and a single leading cause for non-fatal injuries. This study proposed an LSTM model augmented by jobsite contextual information for construction worker trajectory prediction considering both individual movement information and jobsite contextual information. The contextual information is represented as movements of neighboring entities, working group information, and potential destination information. Experiments were conducted using videos collected from three different construction projects. The results show that the newly created method leads to a smaller final displacement error than the model relying solely on target movements, especially in long-term prediction when the length of prediction is no less than that of observation. The adopted sequence-to-sequence network architecture also significantly improves the performance in both final displacement error and average displacement error by eliminating error accumulation over multiple time steps.

In addition, qualitative analysis was conducted to identify scenarios when incorporating contextual information is worthwhile. It was found that when workers are conducting collaborative activities within an



(a) FDE of seq2seq models under different ratios of prediction to observation

(b) ADE of seq2seq models under different ratios of prediction to observation

(c) Prediction performance of recursive model under different ratios of prediction to observation

Fig. 14. Influence of prediction time on different models.

area, incorporating contextual information leads to better results. The context-aware prediction model should be selected when the construction scenario involves multiple entities collaborating on group activities. Both context-aware and position-based methods lead to relatively accurate predicted trajectories when workers move continuously and are not involved in collaborating activities. However, in such case, the position-based method is favorable. Although in this study, the training time for two models is almost the same (about 3 s per epoch), with more data in the future, the position-based method is expected to be less computational expensive considering the fewer features involved in training the model. Moreover, extracting contextual information involves much more complex computing process and may introduce additional errors. Both models may fail when entity states change significantly. In such case, it is not reliable to directly predict worker's trajectory and more information (e.g., activity type, entity posture) may be needed. As an exploratory study that integrates jobsite context in the prediction of workers' movements, the results and findings are obtained based on the limited construction scenarios. More construction videos in different scenarios need to be incorporated to further validate the proposed methods.

This study contributes to the body of knowledge by creating a novel context-augmented deep learning method for construction worker trajectory prediction. The proposed method not only considers spatial interaction between the target and neighboring entities, but also innovatively incorporate the semantic relationship between entities (i.e., whether or not within a working group) and the long-term goal (i.e., the potential destination). The results show that integrating the above contextual information outperforms the position-based prediction, especially for long-term prediction when prediction time is no less than observation time. The proposed context-aware trajectory prediction forms the base for a proactive struck-by prevention mechanism. In addition to the early warning when two entities are expected to get too close, the predicted trajectory also provides information to actively plan a safe path to avoid collisions while ensuring the smooth operation.

As construction videos are used as the data source for trajectory prediction, cameras are recommended to be installed on height to mitigate the occlusion, while maintaining adequate resolutions of entities in the image at the same time. In this study, construction videos are in two resolutions—1920 × 1080, and 1280 × 720, with average worker size around 60 × 120 and equipment size around 450 × 350. In practice, when monitoring construction operations on the complex jobsites, several cameras are needed to ensure the desired coverage and the optimal camera placement are determined by considering both camera coverage and total cost [46,47]. Moreover, to transfer pixel coordinates into world coordinates (e.g., in meters) on the jobsite, at least four ground control points (with known world coordinates) are needed to establish projective transformation between image plane and ground plane using DLT algorithm.

There remain a few limitations that deserve further research efforts. First, due to the availability of construction data, especially the annotated data, the data size used in the experiment is relatively small and thus poses a potential limitation to the representativeness of the proposed method. For possible application and adoption of the proposed approach, scenarios where workers have distinguishable movements and interactions with surrounding entities are recommended for better prediction results. To further justify the model performance, more construction videos will be collected and annotated to expand the existing construction dataset and statistical tests will be conducted. Besides, transfer learning can be adopted to leverage the public dataset in other domains (e.g., crowds datasets [48,49]) to overcome the limitation in the availability of annotated construction datasets. Second, this study used preprocessed worker position and contextual information to train the neural network. In practice, due to the complexity and dynamics in the construction operation, such information may not be acquired with perfect accuracy. In future study, we will work on automating the entire process and further exploit on how possible errors in feature estimation will influence the trajectory prediction performance. Third, only nearest neighbor was considered in the contextual information to reduce the feature dimension when training on small dataset. In future study, occupancy map will be adopted to capture all neighbors within an area to incorporate more comprehensive jobsite contexts. Fourth, the potential destination is simplified as prior knowledge to examine its influence on trajectory prediction. Future study will focus on developing new methods to infer workers' destinations based on their involved activities and the corresponding workspaces.

Data Availability

Some data, models, or code generated or used during the study are available from the corresponding author upon reasonable request, including construction videos and python codes for data processing and trajectory prediction.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study is partially funded by the U.S. National Science Foundation (NSF) through Grant 1850008. The support from NSF is acknowledged.

References

[1] U.S.Bureau of Labor Statistics, Employment, Hours, and Earnings from the Current Employment Statistics survey (National), (2018). https://beta.bls.gov/dataQuery/find?st=0&r=20&fq=survey:[ce]&more=0 (accessed March 25, 2020).
[2] OSHA, Commonly Used Statistics, (2018). https://www.osha.gov/oshstats/commonstats.html (accessed March 25, 2020).
[3] X.S. Dong, X. Wang, R. Katz, G. West, J. Bunting, Struck-by Injuries and Prevention in the Construction Industry, CPWR Q. Data Rep. Second Qua (2017). http://www.cpwr.com/sites/default/files/publications/Quarter1-QDR-2017.pdf (accessed April 12, 2019).
[4] US Department of Labor, Employer-reported workplace injuries and illnesses-2015, (2016). https://www.bls.gov/news.release/archives/osh_10272016.pdf (accessed June 3, 2019).
[5] J. Teizer, T. Cheng, Proximity hazard indicator for workers-on-foot near miss interactions with construction equipment and geo-referenced hazard areas, Autom. Constr. 60 (2015) 58–73, https://doi.org/10.1016/j.autcon.2015.09.003.
[6] E.D. Marks, J. Teizer, Method for testing proximity detection and alert technology for safe construction equipment operation, Constr. Manag. Econ. 31 (2013) 636–646, https://doi.org/10.1080/01446193.2013.783705.
[7] J. Teizer, B.S. Allread, C.E. Fullerton, J. Hinze, Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system, Autom. Constr. 19 (2010) 630–640, https://doi.org/10.1016/j.autcon.2010.02.009.
[8] T. Ruff, Evaluation of a radar-based proximity warning system for off-highway dump trucks, Accid. Anal. Prev. 38 (2006) 92–98, https://doi.org/10.1016/j.aap.2005.07.006.
[9] X. Luo, H. Li, F. Dai, D. Cao, X. Yang, H. Guo, Hierarchical bayesian model of worker response to proximity warnings of construction safety hazards: toward constant review of safety risk control measures, J. Constr. Eng. Manag. 143 (2017), https://doi.org/10.1061/(ASCE)CO.1943-7862.0001277.
[10] Z. Zhu, M.W. Park, C. Koch, M. Soltani, A. Hammad, K. Davari, Predicting movements of onsite workers and mobile equipment for enhancing construction site safety, Autom. Constr. (2016), https://doi.org/10.1016/j.autcon.2016.04.009.
[11] C. Dong, H. Li, X. Luo, L. Ding, J. Siebert, H. Luo, Proactive struck-by risk detection with movement patterns and randomness, Autom. Constr. 91 (2018) 246–255, https://doi.org/10.1016/j.autcon.2018.03.021.
[12] K.M. Rashid, S. Datta, A.H. Behzadan, R. Hasan, Risk-incorporated trajectory prediction to prevent contact collisions on construction sites, J. Constr. Eng. Proj. Manag. 8 (2018) 10–21.
[13] J. Cai, Y. Zhang, H. Cai, Two-step long short-term memory method for identifying construction activities through positional and attentional cues, Autom. Constr. 106 (2019) 102886, https://doi.org/10.1016/j.autcon.2019.102886.
[14] S. Tang, M. Golparvar-Fard, M. Naphade, M.M. Gopalakrishna, Video-based activity forecasting for construction safety monitoring use cases, in: Comput. Civ. Eng. 2019 Smart Cities, Sustain. Resil. - Sel. Pap. from ASCE Int. Conf. Comput. Civ. Eng. 2019,

2019: pp. 204–210. https://doi.org/10.1061/9780784482445.026.
[15] D. Kim, M. Liu, S. Lee, V.R. Kamat, Trajectory prediction of mobile construction resources toward pro-active struck-by hazard detection, in: Proc. 36th Int. Symp. Autom. Robot. Constr. ISARC 2019, 2019: pp. 982–988. https://doi.org/10.22260/isarc2019/0131.
[16] T. Edrei, S. Isaac, Construction site safety control with medium-accuracy location data, J. Civ. Eng. Manag. 23 (2017) 384–392, https://doi.org/10.3846/13923730.2016.1144644.
[17] F. Vahdatikhaki, A. Hammad, Dynamic equipment workspace generation for improving earthwork safety using real-time location system, Adv. Eng. Informatics. 29 (2015) 459–471, https://doi.org/10.1016/j.aei.2015.03.002.
[18] J. Wang, S. Razavi, Spatiotemporal network-based model for dynamic risk analysis on struck-by-equipment hazard, J. Comput. Civ. Eng. 32 (2018), https://doi.org/10.1061/(ASCE)CP.1943-5487.0000732.
[19] H. Kim, K. Kim, H. Kim, Vision-based object-centric safety assessment using fuzzy inference: monitoring struck-by accidents with moving objects, J. Comput. Civ. Eng. 30 (2015) 04015075, https://doi.org/10.1061/(asce)cp.1943-5487.0000562.
[20] J. Wang, S. Razavi, Two 4D models effective in reducing false alarms for struck-by-equipment hazard prevention, J. Comput. Civ. Eng. 30 (2016) 04016031, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000589.
[21] J. Wang, S.N. Razavi, Low false alarm rate model for unsafe-proximity detection in construction, J. Comput. Civ. Eng. 30 (2016), https://doi.org/10.1061/(ASCE)CP.1943-5487.0000470.
[22] T. Liu, P. Bahl, I. Chlamtac, Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks, IEEE J. Sel. Areas Commun. 16 (1998) 922–935, https://doi.org/10.1109/49.709453.
[23] C.G. Prévost, A. Desbiens, E. Gagnon, Extended Kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle, in: Proc. Am. Control Conf., 2007: pp. 1805–1810. https://doi.org/10.1109/ACC.2007.4282823.
[24] C. Hermes, A. Barth, C. Wöhler, F. Kummert, Object motion analysis and prediction in stereo image sequences, in: 8. Oldenburg. 3D-Tage, 2009.
[25] J.F.P. Kooij, F. Flohr, E.A.I. Pool, D.M. Gavrila, Context-based path prediction for targets with switching dynamics, Int. J. Comput. Vis. 127 (2019) 239–262, https://doi.org/10.1007/s11263-018-1104-4.
[26] B.D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J.A. Bagnell, M. Hebert, A.K. Dey, S. Srinivasa, Planning-based prediction for pedestrians, in: 2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009, 2009: pp. 3931–3936. https://doi.org/10.1109/IROS.2009.5354147.
[27] K.M. Kitani, B.D. Ziebart, J.A. Bagnell, M. Hebert, Activity forecasting, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2012: pp. 201–214. https://doi.org/10.1007/978-3-642-33765-9_15.
[28] V. Karasev, A. Ayvaci, B. Heisele, S. Soatto, Intent-aware long-term prediction of pedestrian motion, in: Proc. - IEEE Int. Conf. Robot. Autom., 2016, pp. 2543–2549. https://doi.org/10.1109/ICRA.2016.7487409.
[29] A. Rudenko, L. Palmieri, K.O. Arras, Joint Long-Term Prediction of Human Motion Using a Planning-Based Social Force Approach, in: Proc. - IEEE Int. Conf. Robot. Autom., 2018, pp. 4571–4577. https://doi.org/10.1109/ICRA.2018.8460527.
[30] K. Saleh, M. Hossny, S. Nahavandi, Intent prediction of pedestrians via motion trajectories using stacked recurrent neural networks, IEEE Trans. Intell. Veh. 3 (2018) 414–424, https://doi.org/10.1109/tiv.2018.2873901.
[31] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, S. Savarese, Social LSTM: Human trajectory prediction in crowded spaces, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2016, pp. 961–971. https://doi.org/10.1109/CVPR.2016.110.
[32] H. Xue, D.Q. Huynh, M. Reynolds, SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction, in: Proc. - 2018 IEEE Winter Conf. Appl. Comput. Vision, WACV 2018, 2018, pp. 1186–1194. https://doi.org/10.1109/WACV.2018.00135.
[33] A. Syed, B.T. Morris, SSeg-LSTM: Semantic scene segmentation for trajectory prediction, in: IEEE Intell. Veh. Symp. Proc., 2019, pp. 2504–2509. https://doi.org/10.1109/IVS.2019.8813801.
[34] J. Cai, Y. Zhang, H. Cai, Integrating positional and attentional cues for construction working group identification: A long short-term memory based machine learning approach, in: Comput. Civ. Eng. 2019 Data, Sensing, Anal., American Society of Civil Engineers Reston, VA, 2019, pp. 35–42. https://doi.org/10.1061/9780784482438.005.
[35] Z. Zhu, X. Ren, Z. Chen, Integrated detection and tracking of workforce and equipment from construction jobsite videos, Autom. Constr. 81 (2017) 161–171, https://doi.org/10.1016/j.autcon.2017.05.005.
[36] J. Cai, H. Cai, Robust hybrid approach of vision-based tracking and radio-based identification and localization for 3D tracking of multiple construction workers, J. Comput. Civ. Eng. 34 (2020) 4020021.
[37] M. Egenhofer, J. Herring, Categorizing binary topological relations between regions, lines, and points in geographic databases, Univ. Maine, Orono, Maine, Dept. Surv. Eng. Tech. Rep. (1992) 1–28. https://pdfs.semanticscholar.org/b303/39af3f0be6074f7e6ac0263e9ab34eb84271.pdf (accessed April 12, 2019).
[38] M. Nabil, A.H.H. Ngu, J. Shepherd, Picture similarity retrieval using the 2D projection interval representation, IEEE Trans. Knowl. Data Eng. 8 (1996) 533–539, https://doi.org/10.1109/69.536246.
[39] A. Isli, Integrating cardinal direction relations and other orientation relations in Qualitative Spatial Reasoning, Ann. Math. (2003). http://arxiv.org/abs/cs/0307048.
[40] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[41] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Adv. Neural Inf. Process. Syst., 2014: pp. 3104–3112.
[42] T. Lee, Loss Functions in Time Series Forecasting, Univ. Calif. 1 (2007) 1–14. http://www.faculty.ucr.edu/~taelee/paper/lossfunctions.pdf.
[43] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, ArXiv Prepr. ArXiv1412.6980. (2014). http://arxiv.org/abs/1412.6980.
[44] YouTube, Hospital construction, (2019). https://www.youtube.com/channel/UCEKwrM78pRv8WRcKvZNtE1w (accessed April 7, 2019).
[45] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.
[46] X. Yang, H. Li, T. Huang, X. Zhai, F. Wang, C. Wang, Computer-aided optimization of surveillance cameras placement on construction sites, Comput. Civ. Infrastruct. Eng. 33 (2018) 1110–1126, https://doi.org/10.1111/mice.12385.
[47] J. Kim, Y. Ham, Y. Chung, S. Chi, Systematic camera placement framework for operation-level visual monitoring on construction jobsites, J. Constr. Eng. Manag. 145 (2019), https://doi.org/10.1061/(ASCE)CO.1943-7862.0001636.
[48] S. Pellegrini, A. Ess, K. Schindler, L. Van Gool, You'll never walk alone: Modeling social behavior for multi-target tracking, in: Proc. IEEE Int. Conf. Comput. Vis., 2009, pp. 261–268. https://doi.org/10.1109/ICCV.2009.5459260.
[49] A. Lerner, Y. Chrysanthou, D. Lischinski, Crowds by example, Comput. Graph, Forum. 26 (2007) 655–664, https://doi.org/10.1111/j.1467-8659.2007.01089.x.