A Data-driven Model for Interaction-aware Pedestrian Motion Prediction in Object Cluttered Environments

Mark Pfeiffer, Giuseppe Paolo, Hannes Sommer, Juan Nieto, Roland Siegwart, and Cesar Cadena

Abstract— This paper reports on a data-driven, interaction-aware motion prediction approach for pedestrians in environments cluttered with static obstacles. When navigating in such workspaces shared with humans, robots need accurate motion predictions of the surrounding pedestrians. Human navigation behavior is mostly influenced by their surrounding pedestrians and by the static obstacles in their vicinity. In this paper we introduce a new model based on Long-Short Term Memory (LSTM) neural networks, which is able to learn human motion behavior from demonstrated data. To the best of our knowledge, this is the first approach using LSTMs, that incorporates both static obstacles and surrounding pedestrians for trajectory forecasting. As part of the model, we introduce a new way of encoding surrounding pedestrians based on a 1d-grid in polar angle space. We evaluate the benefit of interaction-aware motion prediction and the added value of incorporating static obstacles on both simulation and real-world datasets by comparing with state-of-the-art approaches. The results show, that our new approach outperforms the other approaches while being very computationally efficient and that taking into account static obstacles for motion predictions significantly improves the prediction accuracy, especially in cluttered environments.

I. INTRODUCTION

A vast amount of research has been done in the area of robotic navigation in static and known environments. Yet, when bringing robots from static or controlled environments to dynamic ones with agents moving in a varied set of patterns, many unsolved challenges arise [1]. In order to integrate robots in a smooth fashion into the workspace shared with other agents — in many cases pedestrians — their behavior needs to be well understood such that accurate predictions of their future actions can be made. While humans can rely on their "common sense" and experience for reading other agents' behavior, robots are restricted to pre-defined models of human behavior and interactions. Independent of the robotic platform, ranging from small service robots to autonomous cars, high accuracy of the other agents' motion predictions is crucial for efficient and safe motion planning.

The main factors influencing pedestrian motion are interactions among pedestrians, the environment, and the location of their destination. By taking into account the interaction between pedestrians, the accuracy of the motion models can be significantly increased [1]–[5]. It was shown that using interaction-aware motion models for dynamic agent
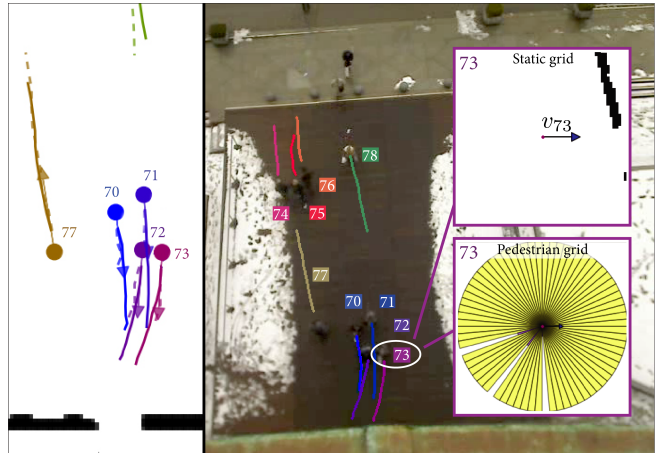
Fig. 1: Trajectories of pedestrians predicted with the presented motion model for interacting pedestrians on a real-world dataset. **Left**: Two-dimensional view of the navigating pedestrians, where the ground truth data (dashed) and the predicted trajectories (solid) are shown. **Main image**: Real-world environment with predicted trajectories projected into the image frame. **Right top**: Static obstacle grid of agent 73, which serves as one input to the model. Static obstacles are shown in black. **Right bottom**: Pedestrian grid of agent 73, which is another input to the model. Both grids are aligned with the agent's position and heading, indicated by the arrows in the center of the grids. Each prediction is purely based on such two grids per agent, and its current velocity (v_{73}).

prediction in motion planning applications makes robots more predictable for human agents [2] and therefore also more "socially compliant" [6]. Especially for cluttered environments it is also important to model the pedestrians' reactions to static obstacles in their close proximity [2], [3].

The existing approaches are limited by at least one of the following shortcomings: (i) The feature functions, which abstract agent trajectory information to an internal representation, are hand-crafted and therefore can only capture simple interactions. (ii) The approaches are not scalable to dense crowds since they use pairwise interactions between all agents [1]–[3], which leads to a quadratic complexity in number of agents, and therefore real-time computation is only feasible for a small number of agents. (iii) Static obstacles are neglected [1], [4], [5] and (iv) knowledge about a set of potential destinations is assumed [1], [3], [5].

In this work, we overcome all the above limitations with a novel data-driven approach to model interaction and motion of pedestrians based on a recurrent neural network (RNN) [7]. We treat the pedestrian prediction problem as a sequence modeling task; a research field where RNNs — especially Long-Short Term Memory (LSTM) neural networks [8] — have shown an outstanding performance within the last five years [9], [10]. To this end, we present a novel model architecture, which fuses three channels of information per pedes-

trian: The pedestrian's current velocity, information about the static obstacles around the pedestrian, and information about surrounding pedestrians. Given these inputs, the pedestrian's future motion is predicted in a receding horizon manner. Compared to other approaches, which do forward simulation through repetitive use of 1-step predictions provided by the model [4], [5], we already do the forecast for the whole prediction horizon, and therefore do not need a forward simulation. One possibility to deploy the presented model on a robotic platform is to predict trajectories of surrounding pedestrians and plan a collision-free path for the robot given the predicted trajectories.

With the presented architecture, abstract feature representations of the inputs can be learned from the demonstration data instead of handcrafting them. By using fixed size grid-based input formats, the evaluation time of the model is constant for a varying number of surrounding agents. This significantly benefits the approach regarding its scalability to more agents.

The main contribution of this paper is the introduction of a new model architecture based on LSTM neural networks for pedestrian trajectory prediction that does not suffer from the shortcomings (i)-(iv). Another major contribution is the way of encoding the information about the surrounding pedestrians in a specialized one-dimensional representation, which combines low dimensionality with a large infromation content. We provide an extensive evaluation and comparison against state-of-the-art approaches, both on simulated and real-world data, where we resort to a well-known publicly available dataset for pedestrian motion in public areas [11]. We hypothesize that by taking into account both pedestrian interactions and the static obstacles, the state-of-the-art prediction approaches can be outperformed.

This paper is structured as follows: Section II gives an overview of the related work. Section III introduces the specific problem and our approach to solve it. Section IV shows our experimental results before we discuss and conclude in Section V.

II. Related work

One of the first approaches to model interactions among pedestrians is the social forces model presented by Helbing et al. [12]. Their model is based on a potential field, or more specifically, attractive forces to model goal-driven behavior and repulsive forces to model obstacle avoidance and interactions among various agents. Later on, Helbing et al. adapted the model to interacting vehicles and traffic dynamics [13]. Similarly, Treuille et al. [14] use continuum dynamics to model naturally moving crowds. When using a good set of parameters, oscillations in the motion can be avoided [15] and they are reliable models for simulation purposes. However, they cannot be applied to motion forecasting without the knowledge of *all* agents' destinations.

Another well known approach for agent interaction forecasting of holonomic agents is the Reciprocal Velocity Obstacles (RVO) method introduced by van den Berg et al. [16]. The main advantages of the RVO approach are its computational efficiency and the guarantee for collision free paths of the agents which not only makes it applicable for prediction but also motion planning applications. The idea behind RVO is to compute a joint set of collision-free velocities based on the assumption of other agents moving with a constant velocity or of a joint collision avoidance effort among all agents. The original approach was extended by Alonso-Mora et al. [17] to non-holonomic platforms. However, by design, the RVO approach requires the knowledge of all agents' target velocities and is only a deterministic model.

When it comes to learning-based interaction modeling approaches, various techniques have been analyzed. Trautman et al. [1] formalized the so called "freezing robot problem" and pointed out why interaction modeling is an important factor for robot navigation. They proposed a model based on interacting Gaussian Process (GP)s, where an interaction potential combines multiple trajectories with each other; each one described by a GP itself. In order to take into account multimodal behavior for motion planning in human crowds, mixture models were introduced later on [18]. Another learning based interaction model was presented by Vemula et al. [5], where a GP model is used to forecast future agent velocities and destinations. The predictions are made based on a grid-based representation of the world which collects information about pedestrian positions and heading in the agent's surroundings. Yet a prior knowledge of destinations is required and no static obstacles are taken into account.

A significant amount of work has gone into modeling interacting pedestrians using maximum entropy Inverse Reinforcement Learning (IRL). Kuderer, Kretzschmar et al. [3], [6], [19] introduced a maximum entropy probability distribution for a joint set of continuous state-spaces, also inspired by Ziebart et al. [20] and Henry et al. [21]. The model is based on a set of hand-crafted feature functions capturing the interaction and collision avoidance behavior of pedestrians. In our previous work [2] we extended this approach to an interaction-aware motion planner and showcased the applicability of this approach in real-world experiments. Since the presented maximum entropy IRL models rely on sampling methods for expectation value computation during training, a common drawback is the training time. Wulfmeier et al. [22] proposed an approach based on deep maximum entropy IRL, where a traversability map of the environment can be learned from demonstration data. No hand-crafted features are required, yet dynamic obstacles — such as other pedestrians — are not taken into account explicitly. Another approach addressing socially aware motion planning was presented by Chen et al. [23]. They use deep Reinforcement Learning (RL) to learn motion planning policies instead of using imitation learning [24] of human or other expert demonstrations. In addition, they show how to induce social norms into the model, rather than learning them.

Alahi et al. [4] introduced the *Social-LSTM* framework. They use one LSTM neural network per agent and pool the information of other agents into a grid-based structure to capture the interactions between multiple agents. Two

versions are introduced: one, where only agent positions are pooled and the other, where complete hidden states of close-by agents' LSTM models are pooled. Fernando et al. [25] introduced another LSTM interaction model based on attention techniques. However, the number of agents that can be dealt with specifically is encoded in the architecture of the model. Both approaches based on LSTM neural networks only take into account dynamic agents and do not reason about the static obstacle avoidance. This limits the performance of the prediction models especially confined spaces with many obstacles.

Regarding sequence prediction, RNN models — including LSTMs [8] and Gated Recurrent Units (GRU) [26] — have successfully shown their performance in various applications in the last decade. They were used to forecast video frames [27], recognize speech [9] or for text translation [10]. In addition, Graves et al. [28] showed that RNN models can be used for sequence generation which is also our targeted application.

III. APPROACH

This section describes the underlying problem of pedestrian prediction and our approach to solve it.

A. Problem formulation

When navigating in cluttered environments, with other agents and / or static obstacles, humans show an outstanding capability to "read" the intentions of other people, perceive the environment and extract relevant information from it. Ultimately, we want robots to be well integrated into the environment in a socially compliant way such that they can safely navigate, do not get stuck or disturb other traffic participants — and that at a human like level. In order to reach this level of autonomy, robots need to have accurate models to forecast the evolution of the environment, including other agents. By taking into account the motion predictions of other agents, safe paths can be planned for the robot. The more accurate the motion predictions of the others, the more predictable will the robot be as the re-planning effort per timestep can be minimized [2].

In this paper we address the problem of finding a good motion and interaction model for pedestrians from recorded demonstrations such that it can be used for the prediction of pedestrians in motion planning applications. We want to find a model

$$(\mathbf{x}_{i,t+1:t+K_H}, \mathbf{H}_{t+1}) = \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{X}_t, \mathbf{G}_t, \mathbf{H}_t, i), \qquad (1)$$

that outputs predictions for the future states $\mathbf{x}_{i,t+1:t+K_H}$ of agent i with a prediction horizon K_H. The current state tuple

$$\mathbf{X}_t := [\mathbf{x}_{1,t}, \ldots, \mathbf{x}_{i,t}, \ldots, \mathbf{x}_{N,t}] \qquad (2)$$

comprises the state of all relevant N pedestrians (within a certain distance from the pedestrian i) at time t. $\mathbf{G}_t$ defines an occupancy grid containing information about the static obstacles encoded in a two-dimensional (2D) grid. $\mathbf{H}_t$ defines a hidden state of the model which propagates information through time and is updated at every timestep.

Since in a real-world application the destinations for other pedestrians cannot be assumed to be known, our model must not rely on a predefined set of target positions but only forecast the future motion of pedestrians according to the perceived data. Only the current information, i.e. at time t, about static obstacles and other agents, together with the hidden state $\mathbf{H}_t$ is used for forecasting the future trajectories. Importantly, in order to be applicable to a wide range of situations and environments, overfitting to a certain environment needs to be avoided.

B. Motion and interaction model

Pedestrian trajectory forecasting can be seen as a sequence modeling task. In order to model the complex associations between agent state, static and dynamic obstacles (i.e. pedestrians) over time, we resort to LSTM neural networks [8] that already successfully showed their high performance in sequence modeling and prediction tasks in the recent years.

With the model, we want to predict the future motion of the ith pedestrian, which in the following will be referred to as the *query agent*. If path predictions for all pedestrians are desired, the model has to be evaluated once for each pedestrian. By doing so, the time complexity of the prediction problem for all N pedestrians is $\mathcal{O}(N)$.
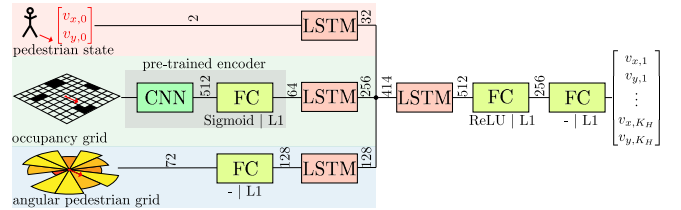

Fig. 2: Architecture of the LSTM model. The inputs are the query agent's state (velocity), the occupancy grid and the angular pedestrian grid (APG), all centered at the query agent's position and aligned with its coordinate frame. Each of the three different input channels gets processed separately through embedding, CNN and LSTM layers. The CNN / FC combination which pre-processes the occupancy grid is pre-trained with an auto-encoder. The concatenation of the extracted features from each channel is followed by another LSTM layer, a FC and an output layer. The output of the model is a sequence of velocities predicted for the query agent's future.

In our model, three channels of information are taken as inputs and fused by a joint LSTM in order to predict the future behavior of the query agent, as depicted in Figure 2. The first input, the state, is the query agent's velocity in its local Cartesian coordinate frame (v_x, v_y). Since all inputs are centered at the query agent's position and aligned with its heading, no position or heading information is required for the presented model. As pedestrians are holonomic agents, their velocity can differ from their heading. Therefore, both v_x and v_y need to be taken into account. It is assumed, that a pedestrian tracker is able to provide the positions and velocities of surrounding agents.

The second input is a 2D occupancy grid, encoding information about the static obstacles in the vicinity of the query agent. As a model input we do not use the full occupancy grid $\mathbf{G}_t$ but a local extract of it centered at the query agent's position and aligned with its current heading. When deployed on a mobile robot, such an occupancy grid can be typically
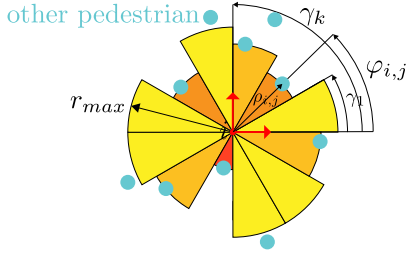
Fig. 3: Angular pedestrian grid (APG) construction from relative pedestrian positions ($\mathbf{X}_t$). The APG is centered at the query agent's position and aligned with its heading. The encoded value per grid cell is the minimum distance to the pedestrians that lie within the cone of the angular grid cell.

obtained based on range finder data. For this work, it is assumed to be provided.

Third, there is the information about the other agents surrounding the query agent, encoded in a special hybrid grid. In order to be able to capture well the dynamics of other pedestrians, a high resolution grid would be needed, increasing the dimensionality a lot. Therefore, we introduce a new way of encoding the information of other pedestrian agents with our angular pedestrian grid (APG), as shown in Figure 3. The output of the APG encoding is a one dimensional vector, $\mathbf{r}_i$, where each element $r_{i,k}$ is defined by:

$$r_{i,k} := \min\left(r_{\max}, \min\left(\{\rho_{i,j} \,|\, j \in \mathcal{N}_{i,k}\}\right)\right)$$
$$\mathcal{N}_{i,k} := \left\{j \in \{1, \ldots, N\} \setminus \{i\} \,|\, \varphi_{i,j} \in [\gamma_k, \gamma_{k+1}]\right\} \quad (3)$$

and $k \in \{1, \ldots, K\}$, where K defines the number of uniform angular cells, i.e. $\gamma_k := (k-1)\frac{2 \cdot \pi}{K}$. The $(\rho_{i,j}, \varphi_{i,j})$ represents the polar coordinates of pedestrian j in the frame of the query agent, i, at the query time t. Therefore $\mathbf{r}_i$ is a function of $\mathbf{X}_t$, (2), only.

Compared to a standard 2D grid, the resolution of the APG only influences the dimensionality of the input linearly, while still being able to capture radial distance changes with continuous resolution instead of discretized grid cells. In addition, the observability of angular position changes of surrounding pedestrians becomes more precise the closer they are to the query agent. One drawback of the APG representation is the fact that only the closest surrounding pedestrian in each angular cone can be captured, yet we assume that these are the ones that affect the query agents decisions the most. As in the occupancy grid case, the APG is centered and aligned with the pedestrian's position and heading, respectively.

The three inputs to the model are processed as shown in Figure 2. They get processed separately before being fused into a common LSTM layer. The architecture with three separate channels turned out to be more efficient in training than with two big consecutive LSTM blocks instead. Two FC layers reason about the future velocities, given the output of the common LSTM cell. The output of the model is a series of future velocity pairs (v_x, v_y) of length K_H. By predicting multiple future steps at once, forward simulation and re-evaluation of the model is not needed and computational efficiency during deployment is improved. The predicted

path for the query agent can be found via Euler forward integration of the velocities from the query agent's position.

1) Pre-training of the CNN: As the extraction of relevant features from the occupancy grid (grey box in Figure 2) is mostly independent of the rest of the overall LSTM model, we propose to pre-train the CNN by using a convolutional auto-encoder (AE) inspired by [29] and [30] as shown in Figure 4. By doing so, the overall training time can be significantly reduced since the weights of the CNN and FC encoding are directly affected and not only through backpropagation based on the overall loss.

In the encoding part, the 2D occupancy grid gets reduced to the latent space of the AE. This CNN encoding phase is incorporated into the full LSTM neural network structure from Figure 2. The second part of the AE is the decoding phase, where the latent space has to be decoded such that the output represents the 2D occupancy grid. The CNN weight parameters between encoding and decoding are shared and the training error is defined by

$$L(\mathbf{g}_{\text{in}}, \mathbf{g}_{\text{out}}) := \sum_{i=1}^{D_x} \sum_{j=1}^{D_y} \left(\mathbf{g}_{\text{in}}(i,j) - \mathbf{g}_{\text{out}}(i,j)\right)^2, \quad (4)$$

namely the squared error between the encoded-decoded output ($\mathbf{g}_{\text{out}}$) and the original input grid ($\mathbf{g}_{\text{in}}$), where i and j define the grid indices along the x- and y-axis. D_x and D_y represent the grid dimensions along the axes. By pre-training the feature extraction for the occupancy grid in this way, it can be assured that the latent space consists of meaningful features, since the original grid needs to be reconstructed from the latent space.
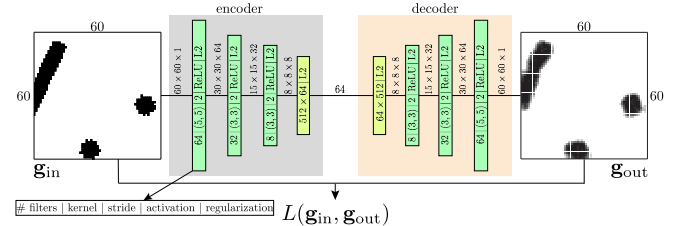

Fig. 4: Architecture of the convolutional AE network for grid feature extraction. By applying three convolutional layers and a FC layer, the original occupancy grid ($\mathbf{g}_{\text{in}}$) gets compressed to a latent space size of 64 during the encoding phase. The decoder reconstructs a grid ($\mathbf{g}_{\text{out}}$) which ideally perfectly matches the input grid. The encoding part of this AE network is later on used in the full LSTM motion and interaction model.

2) Model training: Supervised training is conducted for the motion and interaction model using pedestrian trajectory demonstrations as ground truth. Using the training data, the model is trained end-to-end, while keeping the CNN weights of the encoder (obtained from pre-training) fixed. The FC weights of the encoder are initialized with the values obtained during pre-training but then optimized with the rest of the network.

The full network with its LSTM cells is trained using back-propagation through time (BPTT) [31]. Since the demonstration trajectories have different length and the vanishing / exploding gradient problem [32], [33] needs to be avoided, the truncation depth is fixed to d_{trunc}. Sub-sequences of

length d_{trunc} of the demonstrated trajectories are used for training. For succeeding sub-sequences, the hidden states of the LSTMs are initialized with the final states of the previous sequence. In cases when there is no preceding sub-sequence, the hidden states of the LSTMs are initialized with zeros. Thus, information from the preceding sub-sequence can be forward propagated to the next one, however the optimization (BPTT) only affects the samples within the current sub-sequence. Figure 5 illustrates how the hidden state $\mathbf{H}$ gets propagated between the sub-sequences.
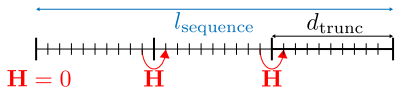

Fig. 5: Visualization of the truncated backpropagation through time during training. The whole sequence of length l_{sequence} gets split up in sub-sequences of length d_{trunc}. For the initial sub-sequence the hidden state of the LSTMs gets initialized with zeros, for succeeding ones with the last one of the preceding sequence.

The loss function used for end-to-end training of the model is defined as follows:

$$L(\mathbf{u}_i, \boldsymbol{\theta}) := \frac{1}{K_H} \sum_{l=1}^{K_H} ||\mathbf{u}_i(l) - \mathbf{v}_{\text{gt},i}(l)|| + \lambda \cdot \alpha(\boldsymbol{\theta}), \quad (5)$$

where $\mathbf{u}_i = [u_i(t+1), \ldots, u_i(t+H)]$ represents the predicted velocities by the model for the prediction horizon and $\mathbf{v}_{\text{gt},i}$ the ground truth velocities (from the demonstration data) for the same horizon. $\alpha(\boldsymbol{\theta})$ introduces the regularization terms and λ the regularization factor. The implemented regularization methods can be found in Figures 2 and 4.

3) Model query during forecasting: For deployment, at each timestep the current inputs are fed to the network and can be evaluated. The only difference to a standard feed-forward neural network is the necessity of feeding back the hidden states of the LSTMs after each query. Therefore, when a new pedestrian is detected, the hidden state is initialized with zeros, while in later steps the hidden state of the preceding step is used.

IV. EXPERIMENTS

The experiments presented in this section evaluate the added value of an interaction-aware motion model and the new capability to incorporate knowledge of static obstacles into the prediction.

As parameters for the static obstacle occupancy grid, we use a $6\,\text{m} \times 6\,\text{m}$ grid with a resolution of $0.1\,\text{m}$ per cell, which leads to a dimension of 60×60 cells. For the APG, we use a maximum range (r_{max}) of $6\,\text{m}$ and 72 (K) angular cells, which leads to an angular resolution of $5\,°$.

A. Data assessment

We use both simulated and publicly available real-world [11] datasets to train and evaluate the model. The simulation data is generated based on the well-known social forces model [12] in various environments. The simulation environments differ in size and the setup of the static obstacles, ranging from an empty corridor to an environment like in

Figure 6 (left). In each environment, up to 20 pedestrians are simulated simultaneously while they navigate between randomly changing target positions. Once a target position is reached, a new one is sampled for each pedestrian, while it is assured that they are not in collision with static obstacles. As suggested in [12], we add Gaussian noise ($\sigma = 0.3$) to the forces generated by the social forces model, which directly affect the acceleration of the pedestrians. The simulation does not explicitly account for group behavior of walking pedestrians.
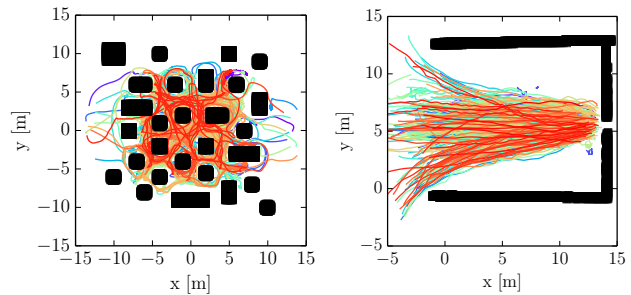

Fig. 6: **Left**: Training data generated with the simulation based on the social forces model [12]. For better visibility, only a subset of the trajectories is visualized. **Right**: Real-world training data recorded in a public environment [11]. The static obstacles are shown in black, the agent's trajectories in colors.

B. Evaluation and performance metrics

In order to assess the performance of our model, we compare it against three well known motion prediction approaches. Constant velocity (CV) and constant acceleration (CAcc) approaches are the baseline for any motion prediction method. These models simply assume each agent to continue with it's current velocity or acceleration, respectively. In addition, we use the social forces and an LSTM model without knowledge of the static grid for comparison. For the evaluation on the simulation data we assume the social forces model to significantly outperform all the other approaches used for comparison since the data was generated based on the same model. The only difference is the added noise on the acceleration.

The LSTM model unaware of the static grid is similar to the O-LSTM model, recently presented in [4]. Since the original implementation is not publicly available, we use our own version of the O-LSTM method for comparison. Instead of the discrete grid, that solely knows about a pedestrian count per cell, we also use our APG to encode the pedestrian information. This however, contains more detailed information than a standard 2D grid. In the following, our introduced model will be referred to as the LSTM model, the one unaware of static obstacles as LSTM-noGrid.

To evaluate the performance of all models quantitatively, we use the prediction error, i.e. the Euclidean distance between prediction and ground truth values, over time. We analyze the prediction errors for a time horizon of $3\,\text{s}$, which requires 10 prediction steps with a sampling time of $300\,\text{ms}$. In addition, we do a qualitative evaluation by visual inspection of the predicted trajectories for multiple examples, both for simulated and real-world data.

C. Simulation results

Since real-world data oftentimes contains tracking error or pedestrians who are standing still or walking back and forth, we will start our evaluation based on the simulation data. The simulation results are consistent up to the noise on the accelerations of the pedestrians and therefore are a good way to analyze whether motion policies of pedestrians can be learned by the presented model. In addition, compared to real data acquisition it is fairly easy to generate new datasets in different environments.

The model tested in simulation is trained in two different environments: First, it is pre-trained in an empty corridor environment with 10 pedestrians navigating between random target positions at the two sides of the corridor. With this dataset, the pure interactions with other pedestrians can be learned. Second, the main part of the training is conducted in the environment shown in Figure 6 (left) and comprises 20 pedestrians. It turned out that pre-training in a simplistic environment and afterwards continue training in the environment of choice speeds up the overall training time of the model. The corridor dataset contains less than an hour of data, while the main dataset contains about five hours of navigating pedestrians. The overall training time is about 3 h on a Nvidia GeForce GTX 980 Ti GPU[1] and requires 300'000 training steps.
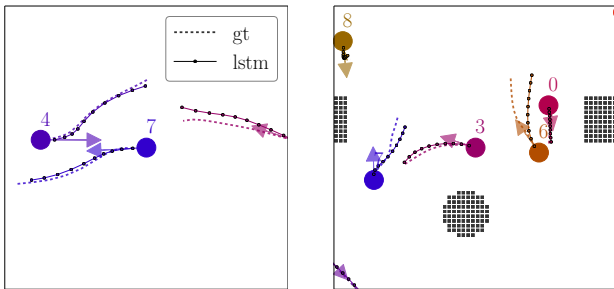

Fig. 7: Two examples for interaction- and obstacle-aware pedestrian trajectory prediction using the presented LSTM model. The big circles represent pedestrians (number specifies the agent id), the small ones the ten prediction steps. The current velocity of each pedestrian is indicated by the arrow. Static obstacles are shown as black grid cells. Both examples stem from the test data which was not used for training the model.

The evaluation is conducted on a test dataset, which uses a different environment (arrangement of static obstacles) than the one used for training. Like this, the generalization capabilities of the learned motion- and interaction model can be analyzed.

Figure 7 shows two snapshots of the trajectory predictions conducted by the model vs. the ground truth data. The left image clearly shows how two pedestrians are jointly avoiding each other. Although the agents' current velocities (arrows) do not show any avoidance measures yet, the LSTM model already predicts that agent 4 and 7 will both swerve to their left side in order to avoid a potential collision. The figure also indicates that the predicted avoidance maneuver is similar to the one conducted in the ground truth data, which shows that

the model is able to transfer its knowledge between different scenarios.

The right image shows a situation where the future pedestrian trajectories are clearly influenced by the occurrence of static objects. Although there are no pedestrians in the vicinity of agent 8, our model predicts that this agent will almost come to a stop because of the static obstacle ahead of it. Agent 7 is predicted to avoid the same static obstacle from the other side while agent 3 needs to do a sharp left turn in order not to collide with agent 7. Since the model's LSTM cells provide internal memory, it can keep track of the past motion of an agent. The internal memory is especially helpful since no target knowledge is available and the predictions only rely on past and current observations. Compared to the two agents in the left image, where the avoidance effort is shared, agents 0 and 6 in the right image do not share the avoidance. Agent 6 needs to perform the avoidance maneuver since the static obstacle is blocking agent 0 from moving to its left. Moreover, the qualitative performance of the predictions can be assessed in the associated video submission[2].

The quantitative analysis of the prediction error in the simulation dataset is shown in the upper part of Figure 8 and in Table I. As expected, the prediction error of the social forces model is significantly smaller than all other approaches, which stems from the fact that the model was also used for data generation. Moreover, the social forces model has perfect knowledge about the destinations of all agents, an element which is not required in the presented LSTM model. The prediction error in Figure 8 only stems from the acceleration noise in the social forces model. The baseline prediction methods are significantly worse than the LSTM models, since no knowledge about the environment can be included in the predictions. Comparing the two LSTM models, which were both trained on the same dataset, one can see, that adding static obstacles to the input of the model improves its prediction capabilities significantly. This supports our initial hypothesis. Especially for a longer prediction horizon, the difference between the model only based on the pedestrian-pedestrian interactions (LSTM-noGrid) and the one using both static obstacles and pedestrians (LSTM) becomes distinct.

Since the inputs to the model are encoded in grids, the evaluation complexity per agent is constant. This also results in an almost constant (depending on the other processes running on the computer) and predictable query time of the model. For this example, where 20 pedestrians need to be predicted all the time, the average evaluation time for all agents was 51 ms on a standard laptop with an Intel® Core™ i7-4810MQ CPU processor with 2.80GHz. This results in an average prediction time of 2.6 ms per pedestrian.

D. Real-world results

We use a publicly available pedestrian tracking dataset [11] to evaluate our model on real-world data. Since the
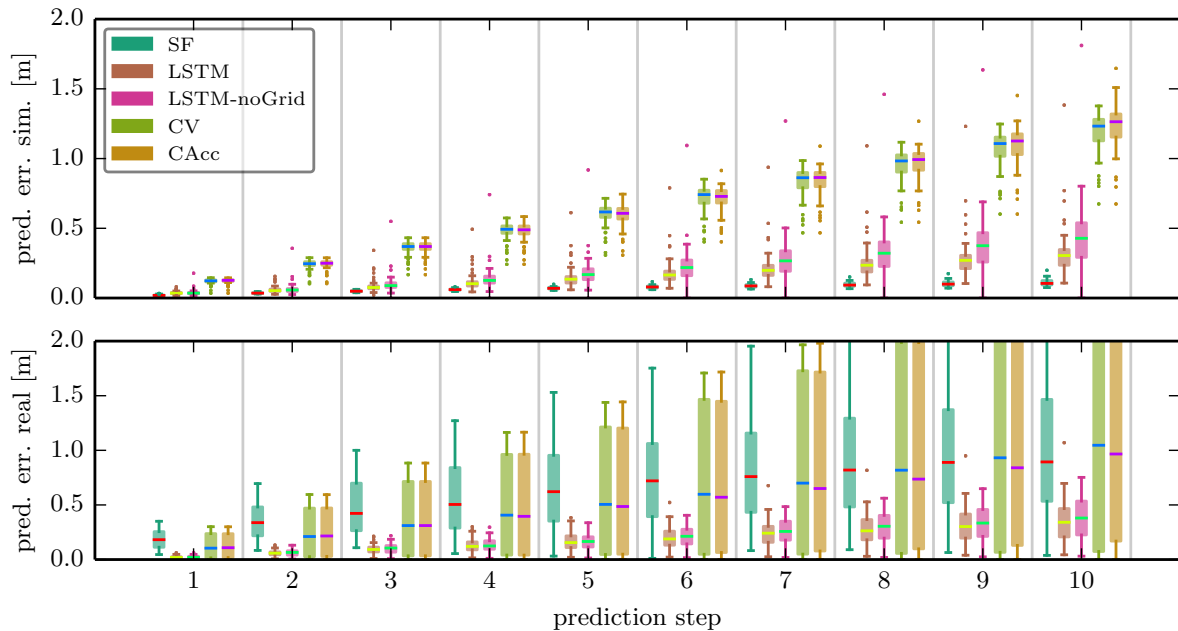
Fig. 8: This figure shows the evaluation of the prediction error of various models over a prediction horizon of 3 s with 10 prediction steps. The upper figure presents the prediction error on the synthetic / simulation-based dataset, the lower one the evaluation on the real-world data. On both datasets, the baseline methods are significantly outperformed by the learning based LSTM models. While the simulation data is generated with the social forces (SF) model (and therefore the prediction error is very small), the learning-based LSTM approaches outperform all the models in the real-world dataset. The results clearly show that the model taking into account static obstacles (LSTM) significantly outperforms the one unaware of static obstacles (LSTM-noGrid), once the environment is more cluttered. The presented LSTM model is not constrained to a single dataset, which is indicated by the similar prediction error for both simulation and real-world dataset.

TABLE I: Average prediction error in meter for the compared models on the simulated and real-world dataset.

	SF	LSTM	LSTM-noGrid	CV	CAcc
simulation	0.071	0.169	0.220	0.652	0.659
real-world	0.667	0.179	0.193	0.676	0.682

real-world dataset is much smaller than the one generated in simulation, it will also give a good indication of how well the model can deal with this amount of information and how prone it is to overfitting. During training, we use the pre-trained model (trained on the simulation data) and train it with the real-world dataset.

The dataset is recorded at two different locations, both very sparse in terms of static objects. As for the simulation results, we will use one environment for training while the other will be used for testing only. The real-world training data is shown in Figure 6 (right). The dataset also contains information about the group structure of the pedestrian motion. If pedestrians move close together, they are registered as a group. We also introduced this information into the social forces model, as described in [34]. Since the SF model requires destinations per agent we provided it always with the endpoint of the complete observed trajectory as the agent's target.

Figure 8 and Table I show the quantitative analysis of the prediction error in the test environment. As expected, the constant velocity / acceleration model can only serve as a lower bound for the motion predictions. The performance of the social forces model is largely affected by the considerable noise in the real-world data which significantly influences the short-term predictions. While the prediction error of the other models increases clearly with the horizon, the social forces prediction error only increases slowly over time which originates from the knowledge of the destinations of the agents.

Since both environments (real-world training and test) barely have any objects blocking the walking areas of the pedestrians, see Figure 6 (right), both the LSTM and the LSTM-noGrid model (with and without static objects) perform almost equally in this environment. However, especially for long term predictions, the prediction errors for the LSTM model aware of static obstacles is still smaller than the errors of the one unaware of static obstacles.

Compared to the social forces model, the prediction error can be reduced significantly — despite the fact that the former has more information available, namely the targets per pedestrian. Additionally, the prediction errors for the real data are in the same range as for the simulation data. This also shows that the LSTM model can learn to deal with noisy input data and can provide reliable interaction-aware predictions in a real-world environment.

A qualitative assessment of the predictions in the real-world environment is shown in Figure 1. Although agents 72 and 73 walk straight towards the static obstacle, the model is able to predict that they will walk through the open slot (which actually is a door). This goes along with the ground truth trajectories. More examples for the navigation performance in the real-world environment are provided in the associated video submission.

V. CONCLUSION

In this paper, we introduced a new approach to model pedestrian dynamics and interactions among them. The model architecture is based on an LSTM neural network which is trained from demonstration data. To the best of our knowledge, we present the first approach which is able to predict pedestrian-pedestrian interactions and the avoidance of static obstacles at the same time with a neural network model. In addition, we introduce a new way of handling dynamic objects, the angular pedestrian grid (APG), which encodes the information of the surrounding pedestrians. In a multi-pedestrian scenario, we use one LSTM network per pedestrian, and therefore the computational complexity only increases linearly with the number of agents. This is especially important when the crowd density gets higher. The presented model does not require a known destination per pedestrian or a predefined set of destination candidates, which makes it well applicable to many real-world applications.

The performance of the interaction- and obstacle-aware motion model is evaluated both on simulation and real-world data. Our experiments show, that the prediction accuracy of the model can be significantly increased by taking into account static obstacles, especially when the environment becomes more cluttered.

In an environment with only few static obstacles (like the real-world data we evaluated on) our proposed model still clearly outperforms the other, state-of-the-art approaches that we use for comparison. These include the social forces model, which cannot be applied without providing all agents' target positions and and interaction-aware LSTM model without static obstacle information.

Additional real-world data needs to be acquired in particular for cluttered environments to further evaluate the performance of our method in different scenarios. Since the model can make accurate and reliable predictions while being computationally efficient even for a multitude of pedestrians, it is very well suited to be integrated in a mobile platform and to be used for realtime motion planning applications, which will be done in future work.

REFERENCES

[1] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 797–803.

[2] M. Pfeiffer, *et al.*, "Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2016, pp. 2096–2101.

[3] H. Kretzschmar, *et al.*, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.

[4] A. Alahi, *et al.*, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.

[5] A. Vemula, *et al.*, "Modeling cooperative navigation in dense human crowds," *arXiv preprint arXiv:1705.06201*, 2017.

[6] H. Kretzschmar, *et al.*, "Learning to predict trajectories of cooperatively navigating agents," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4015–4020.

[7] I. Goodfellow, *et al.*, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[8] S. Hochreiter and J. Schmidhuber, "Long short term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 12 1997.

[9] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[10] D. Bahdanau, *et al.*, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[11] S. Pellegrini, *et al.*, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 261–268.

[12] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[13] D. Helbing and B. Tilch, "Generalized force model of traffic dynamics," *Physical review E*, vol. 58, no. 1, p. 133, 1998.

[14] A. Treuille, *et al.*, "Continuum crowds," in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 1160–1168.

[15] T. Kretz, "Oscillations in the social force model," *Physica A*, vol. 438, pp. 272–285, 2015.

[16] J. Van den Berg, *et al.*, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 1928–1935.

[17] J. Alonso-Mora, *et al.*, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 203–216.

[18] P. Trautman, *et al.*, "Robot navigation in dense human crowds: the case for cooperation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2153–2160.

[19] M. Kuderer, *et al.*, "Feature-based prediction of trajectories for socially compliant navigation," in *Proc. of Robotics: Science and Systems (RSS)*, 2012, p. 193.

[20] B. D. Ziebart, *et al.*, "Maximum entropy inverse reinforcement learning." in *AAAI*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[21] P. Henry, *et al.*, "Learning to navigate through crowded environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 981–986.

[22] M. Wulfmeier, *et al.*, "Large-scale cost function learning for path planning using deep inverse reinforcement learning," *The International Journal of Robotics Research*, p. 0278364917722396, 2017.

[23] Y. F. Chen, *et al.*, "Socially aware motion planning with deep reinforcement learning," *CoRR*, vol. abs/1703.08862, 2017. [Online]. Available: http://arxiv.org/abs/1703.08862

[24] M. Pfeiffer, *et al.*, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1527–1533.

[25] T. Fernando, *et al.*, "Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection," *arXiv preprint arXiv:1702.05552*, 2017.

[26] J. Chung, *et al.*, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[27] M. Ranzato, *et al.*, "Video (language) modeling: a baseline for generative models of natural videos," *arXiv preprint arXiv:1412.6604*, 2014.

[28] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[29] J. Masci, *et al.*, "Stacked convolutional auto-encoders for hierarchical feature extraction," *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 52–59, 2011.

[30] C. Cadena, *et al.*, "Multi-modal auto-encoders as joint estimators for robotics scene understanding." in *Robotics: Science and Systems*, 2016.

[31] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[32] R. Pascanu, *et al.*, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[33] Y. Bengio, *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[34] K. Yamaguchi, *et al.*, "Who are you with and where are you going?" in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1345–1352.