Modeling Cooperative Navigation in Dense Human Crowds

Anirudh Vemula[1], Katharina Muelling[1] and Jean Oh[1]

Abstract— For robots to be a part of our daily life, they need to be able to navigate among crowds not only safely but also in a socially compliant fashion. This is a challenging problem because humans tend to navigate by implicitly cooperating with one another to avoid collisions, while heading toward their respective destinations. Previous approaches have used hand-crafted functions based on proximity to model human-human and human-robot interactions. However, these approaches can only model simple interactions and fail to generalize for complex crowded settings. In this paper, we develop an approach that models the joint distribution over future trajectories of all interacting agents in the crowd, through a local interaction model that we train using real human trajectory data. The interaction model infers the velocity of each agent based on the spatial orientation of other agents in his vicinity. During prediction, our approach infers the goal of the agent from its past trajectory and uses the learned model to predict its future trajectory. We demonstrate the performance of our method against a state-of-the-art approach on a public dataset and show that our model outperforms when predicting future trajectories for longer horizons.

I. INTRODUCTION

There is an increasing need for robots to operate in the midst of human crowds. This requires the robot to be able to navigate through a crowd in a socially compliant way, i.e., the robot needs to collaboratively avoid collisions with humans and adapt its trajectories in a human predictable manner. To date, the majority of existing works in the area of social navigation has focused on the prediction of individual motion patterns in the crowd to improve the navigation performance [7], [8], [4]. However, even in the case of perfect prediction, these approaches can lead to severely suboptimal paths [1]. The primary reason for such underperformance is that these approaches do not capture the complex and often subtle interactions that take place among humans in a crowd; that is, these approaches model each agent independently of the others. This observation leads to the insight that agents in crowds engage in joint collision avoidance.

Humans navigate through dense crowds by adapting their trajectories to those of other people in the vicinity. Figure 1 shows three examples of such behavior where people pass through, slow down, or go around when they are near other pedestrians. In order to learn to navigate in a socially compliant way, it is key to capture such human-human interactions observed in a crowd. Pioneering works by [9], [16] propose hand-crafted functions to model such

[1]A. Vemula, K. Muelling and J. Oh are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA 15213. Email: avemula1@andrew.cmu.edu, {kmuelling, jeanoh}@nrec.ri.cmu.edu
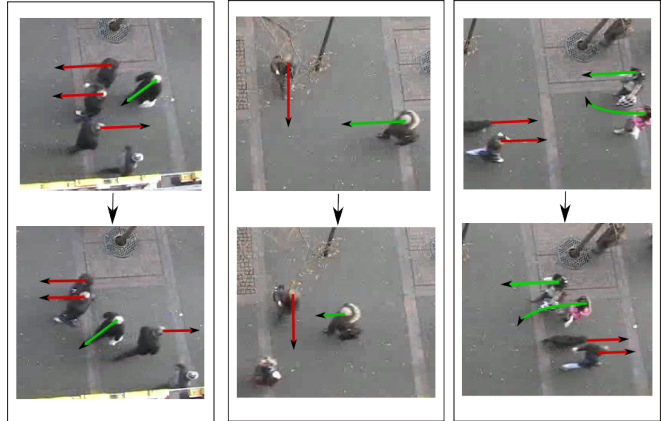
Fig. 1: Examples of pedestrians exhibiting cooperative behavior. In each image, the velocity of the pedestrian is shown as an arrow where the length of each arrow represents the speed. (Left) The pedestrian (with green arrow) anticipates the open space between the other three (with red arrow) and doesn't slow down. (Middle) The pedestrian (with green arrow) slows down to allow the other pedestrian (with red arrow) to pass through. (Right) The two pedestrians (with green arrows) make way for the oncoming agents (with red arrow) by going around them.

interactions based on proximity. Such functions are, however, limited in the complexity of interactions that they can model and fail to generalize for crowded settings. Trautman et. al. [1] proposed an approach that explicitly models human-human and human-robot interactions to enable a robot to safely navigate in a dense crowd. The trajectories of the robot and the humans are jointly predicted with a hand-crafted potential term to model interactions. Because the potential term is hand-crafted, it is possible that the robot trajectories generated may not resemble socially compliant human behavior. In this paper, we learn the interaction model from real-world pedestrian trajectories in order to predict human-like trajectories.

In safety-critical applications like robotics, where robots are expected to navigate safely among humans, we need to account for uncertainty in our predictions. Those approaches that learn interaction models but that do not deal with uncertainty as in [2] can lead to over-confident predictions which could result in awkward and disruptive behavior. Our approach considers the uncertainty regarding intentions (or goals) of the pedestrians and results in accurate predictions.

The summary of our contributions in this paper are as follows: We develop a new algorithm for enabling robots to move naturally through dense crowds. Following the key insight that agents in crowd engage in *joint collision avoidance*, we develop an approach that models the distribution over future trajectories of all interacting agents through a joint

density model that captures the idea of *cooperative* planning, i.e., agents cooperating with each other to achieve their goals and avoid collisions. To capture collision avoidance behavior, we learn a local interaction model that encodes how agents move based on how populated their vicinity is from real human trajectory data. During prediction, our model infers the goal of an agent from its past trajectory and uses the learned model to predict its future trajectory. Finally, we demonstrate that our model is capable of predicting robot trajectories that are natural and human-like by reporting the experimental results on the ETH pedestrian video dataset [11].

In the remainder of this paper, we will give an overview of relevant existing work in section II. The notation and the problem is defined in section III. Section IV will describe our approach in detail and its evaluation on a real world dataset is presented in section V. The results of the evaluation are discussed in section VI and the conclusions are presented in section VII, along with directions for future work.

II. RELATED WORK

A. Navigation in Uncertain Dynamic Environments

Common approaches to robot navigation in uncertain, populated environments typically compute a path to the goal without taking into account the collaborative collision avoidance behavior exhibited by humans. Most of the approaches instead rely on local reactive collision avoidance methods [5], [6]. Although these methods effectively avoid collisions with humans, the planned trajectories are usually suboptimal, compared to the trajectory a human would take for the same start and goal, and not socially compliant due to evasive maneuvers.

Naively modeling unpredictability in the trajectories of humans, using linear models like Kalman filters, leads to increasing uncertainty that makes safe and efficient navigation difficult [1]. Several works have focused on controlling the uncertainty in these predictive models by developing accurate human motion models [7], [8], but these approaches do not account for interactions between humans and thus cannot model joint collision avoidance.

B. Modeling human interactions

The social forces model proposed by [9] models motion of pedestrians in terms of forces that drive humans to reach a goal and to avoid obstacles. Subsequently, approaches have been proposed that use learning methods to fit the social forces model to observed crowd behavior [14], [15]. The social forces model has been shown to perform well in predicting trajectories in simulated crowds, but fails when it comes to predicting movements of pedestrians in real dense crowds as it uses a hand-crafted potential term based on distances, and doesn't learn human-human interactions from real data. Using a hand-crafted potential term results in a model that can capture simple interactions, like repulsion and attractions, but may fail to account for more complex crowd behavior. [10] models pedestrian motion behavior using dynamic potential fields that guide people to avoid collisions and move towards the goal, but this model does not

explicitly account for interactions like cooperation between agents in the crowd.

Hall [16] introduces a theory on human proximity relationships which have been used in potential field methods to model human-human interactions [12], [13]. These models capture interactions to avoid collisions, but do not model human-robot cooperation. However, models of cooperation are necessary for safe and efficient navigation in dense crowds [1], because in cases where the crowd density is high, the robot believes there is no feasible path in the environment unless it accounts for cooperation from the crowd. Reciprocal Velocity Obstacles (RVO), [26], and Velocity Obstacles (VO), [27], account for interactions between agents by computing joint collision-free velocities assuming constant velocities and shared collision avoidance behaviors. However, these approaches cannot handle stochastic behavior in pedestrian motions and do not train the model from real observed data.

Trautman et. al. [1] proposed *Interacting Gaussian processes* (IGP) to explicitly model the human-robot cooperation. Similar to the work presented in this paper, IGP models the trajectories of all interacting agents jointly which results in a probabilistic model that can capture joint collision avoidance behavior. However, the IGP model assumes that the final destinations of all pedestrians are known, which is not the case in a realistic prediction task. Another drawback of IGP is the use of hand-crafted interaction potential term to model cooperative behavior which may result in robot trajectories that are not socially compliant. In this paper, we learn the interaction model from observations of real pedestrian trajectory data in the hope that we achieve more human-like and socially compliant trajectories.

The works of [18], [19] are also closely related to our work. These approaches explicitly model human-robot cooperation and jointly predict the trajectories of all agents, using feature-based representations. Unlike our proposed approach, they use *maximum entropy inverse reinforcement learning* (IRL) to learn an interaction model from human trajectory database using carefully designed features such as clearance, velocity, or group membership. However, their approach has been tested in scripted environments with no more than four humans. In our work, we deal with crowded scenarios with an average of six humans in a single scene. Very recently, [25] have extended the maximum entropy approach to unseen and unstructured environments by using a receding horizon motion planning approach.

C. Trajectory prediction

A large body of works exist in the domain of computer vision and video surveillance that deal with predicting motion of people in videos, that are relevant to our work. [20], [21] learn motion patterns using Gaussian processes and cluster human trajectories into these motion patterns. But these approaches ignore human-human interactions. IRL has also been used in the domain of activity forecasting to predict human paths in static scenes, [22] and more recently, [2] used Long Short-Term Memory networks (LSTM) to jointly

reason across multiple agents to predict their trajectories in a scene. However, most of these approaches have been used in the context of prediction and have not been extended to navigation.

III. PROBLEM DEFINITION

A. Notation

We follow the notation of [1]. Let the index $i \in \{1, 2, \cdots, N\}$ specify the agent, where N is the number of individuals in the crowd and $i = R$ indicates the robot. The trajectory of agent $i \in \{R, 1, 2, \cdots, N\}$ is given by $\mathbf{f}^{(i)} = (f_1^{(i)}, f_2^{(i)}, \cdots, f_T^{(i)})$, where T is the length of the trajectory and $f_t^{(i)} = (x_t^{(i)}, y_t^{(i)}) \in \mathbb{R}^2$ is the location of agent i at time-step t. The observed locations of pedestrian i until time-step t is denoted as $\mathbf{z}_{1:t}^{(i)} = (z_1^{(i)}, z_2^{(i)}, \cdots, z_t^{(i)})$. We denote the set of all pedestrian trajectories by $\mathbf{f} = \{\mathbf{f}^{(i)}\}_{i=1:N}$, the robot trajectory by $\mathbf{f}^{(R)}$, and the set of all pedestrian observations until time-step t by $\mathbf{z}_{1:t} = \{\mathbf{z}_{1:t}^{(i)}\}_{i=1:N}$. We assume a fixed number of goals g in the environment are known and denote the set of goals by $\mathbf{G}$.

B. Planning using the joint density

Following the assumption that people engage in a joint collision avoidance when moving through a dense crowd as in [9], [1], the robot does not only have to respond to the observed trajectories of the pedestrians, but also has to account for the adaptive behavior of the humans. To capture this cooperative behavior, it has been suggested by [1] to use the joint density of both the robot and the crowd, denoted by $P(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t})$. Planning the path for the robot using this density corresponds to finding the maximum-a-priori (MAP) assignment for the following posterior:

$$(\mathbf{f}^{(R)}, \mathbf{f})_* = \arg\max_{\mathbf{f}^{(R)}, \mathbf{f}} P(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t}). \quad (1)$$

C. Problem

In this work, we assume that at each time-step t, we receive the observation $\mathbf{z}_t$ of the locations of all agents in the crowd. Given the current and past observations $\mathbf{z}_{1:t}$, we tackle the problem of estimating the joint posterior distribution of the future trajectories of all agents. Formally, we seek to model the density given by,

$$P(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t}).$$

Planning the robot's trajectory then corresponds to taking the MAP assignment for $\mathbf{f}^{(R)}$ and executing it until the next observation is received. At time-step $t + 1$, we receive a new observation $\mathbf{z}_{t+1}$ and update the above joint posterior density to $P(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t+1})$. This process is repeated until the robot reaches its destination. In contrast to [1], who tackle a similar problem, we aim to predict more natural and human-like robot trajectories by learning the model from pedestrian trajectory data.
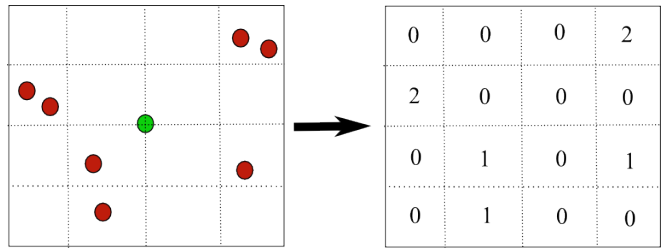

Fig. 2: Occupancy grid construction. (Left) A configuration of other agents (red) around the current agent (green). (right) 4x4 occupancy grid is constructed using the number of agents in each grid cell

IV. APPROACH

We exploit the observation that humans navigating in dense crowds adapt their trajectories based on the presence of agents in their vicinity [2]. We first explain the construction of *occupancy grids*, which account for the presence of other agents within an agent's local neighborhood (Section IV-A). We formulate the problem of learning the social interaction model as a Gaussian process regression problem, where we predict the agent's velocities at each time-step as a function of their occupancy grids and intended goal. Given the preprocessed training trajectories (Section IV-B), we train the GP model by maximizing its marginal likelihood to learn the hyperparameters of the kernel (Section IV-C). At prediction time, we use the learned model to infer the goal of each agent and jointly predict future trajectories of all interacting agents in the crowd (Section IV-D).

A. Constructing occupancy grids

To capture the local interactions of an agent with its neighbors, we construct an occupancy grid for each agent i at each time-step t that is similar to the social pooling grid used in [2]. The occupancy grid is constructed by discretizing the neighborhood of the agent's current location into a spatial grid of size $M \times M$. We then count the number of surrounding agents within each grid cell. Formally, we define occupancy grid as a vector of length M^2 given by,

$$\mathbf{O}_t^{(i)}(a+M(b-1)) = \sum_{j \in \mathcal{N}^{(i)}} \mathbb{I}_{a,b}[x_t^{(j)} - x_t^{(i)}, y_t^{(j)} - y_t^{(i)}]. \quad (2)$$

$\mathcal{N}^{(i)}$ denotes the set of all agents $j \neq i$, who are within the neighborhood of agent i. The indicator function $\mathbb{I}_{a,b}[x, y]$ defines if (x, y) is located in the (a, b) cell of the occupancy grid. In the remainder of the paper, $\mathbf{O}^{(i)}$ and $\mathbf{O}$ will denote the set of all occupancy grids at every time-step of an agent i and that of all agents, respectively.

B. Preparing training data for learning

Before training our model, we preprocess the training data $\mathbf{f}$, which correspond to the trajectories of all agents. First, we construct occupancy grids $\mathbf{O}^{(i)}$ and $\mathbf{O}$ along $\mathbf{f}$, as described in Section IV-A. Second, we process all the trajectories to obtain the velocities of agents at each time-step $(\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t})$. Third, since we have access to the entire trajectory at training, we can compute the goals of all the pedestrians at training. Note that this information about the true goals is used only during training and is not assumed in the prediction phase.

After this preprocessing, we have $(\mathbf{O}, \frac{\Delta\mathbf{x}}{\Delta t}, \frac{\Delta\mathbf{y}}{\Delta t})$ for all agents at every time-step and their corresponding goals $\{g^{(i)}\}$.

C. Training the local interaction model

To learn across different pedestrians traversing in different regions of the environment, we model the velocity of an agent at a specific time-step as a function of their intended goal and their occupancy grid at that time-step. Formally, we seek to estimate the distribution $P(\frac{\Delta\mathbf{x}}{\Delta t}|\mathbf{O}, g)$ and $P(\frac{\Delta\mathbf{y}}{\Delta t}|\mathbf{O}, g)$, for each goal g in $\mathbf{G}$, from the training data obtained in Section IV-B.

We start by modeling the interactions as a Gaussian Process (GP) regression problem, where the noisy data to be interpolated is $(\mathbf{O}, \frac{\Delta\mathbf{x}}{\Delta t})$ and $(\mathbf{O}, \frac{\Delta\mathbf{y}}{\Delta t})$. Note that we use a separate GP for each goal g to learn the mapping from occupancy grids to velocities. We use a *squared exponential automatic relevance determination* (SE-ARD) kernel (with additive noise) for these GPs [3]. The SE-ARD kernel learns a different lengthscale for each input dimension, which in our problem are the dimensions of the occupancy grid vector, i.e., the grid cells. Since, the kernel can learn the relevance of each input dimension by learning the lengthscale parameter [3] (dimensions with large lengthscales are less relevant than those with small lengthscales), the kernel will capture the relevance of each grid cell for a specific goal and effectively ignore irrelevant cells.

The SE-ARD kernel with additive noise is given by,

$$K_S(\mathbf{O}, \mathbf{O}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{M^2}\frac{(\mathbf{O}(d) - \mathbf{O}'(d))^2}{\ell_d^2}\right) + \sigma_n^2\delta(\mathbf{O}, \mathbf{O}') \quad (3)$$

where $\delta(\mathbf{O}, \mathbf{O}') = 1$ if $\mathbf{O}$ is equal to $\mathbf{O}'$ and zero otherwise, and $\mathbf{O}(d)$ is the value of the dth dimension in the vector $\mathbf{O}$. The hyperparameters of this kernel are σ_f (signal variance), $\{\ell_d\}_{d=1}^{M^2}$ (lengthscales) and σ_n (noise variance).

This construction results in a total of $2G$ GPs because there is a pair of GPs (in x- and y-direction) for each of the G goals; thus, we have $2G$ sets of hyperparameters to be learned. We denote the set of hyperparameters corresponding to the GP associated with goal g by Θ_x^g and Θ_y^g. To learn the hyperparameters Θ_x^g, we isolate the tuples $B_g = \{(\mathbf{O}^{(i)}, \frac{\Delta\mathbf{x}}{\Delta t}^{(i)})\}_i$ from training data, corresponding to the set of pedestrians i whose goal is g, and maximize the log marginal likelihood of the GP [3] given by,

$$\log P\left(\frac{\Delta\mathbf{x}}{\Delta t}|\mathbf{O}\right) = -\frac{1}{2}\frac{\Delta\mathbf{x}}{\Delta t}^T K_S(\mathbf{O}, \mathbf{O})^{-1}\frac{\Delta\mathbf{x}}{\Delta t} - \frac{1}{2}\log|K_S(\mathbf{O}, \mathbf{O})| - \frac{n_g}{2}\log 2\pi \quad (4)$$

where $\frac{\Delta\mathbf{x}}{\Delta t}$ and $\mathbf{O}$ are vectors constructed by concatenating elements of B_g, and n_g is the number of elements in $\frac{\Delta\mathbf{x}}{\Delta t}$.

We can learn Θ_y^g and all the other sets of hyperparameters for all goals $g \in \mathbf{G}$ in a similar fashion.

D. Prediction

During prediction we are given an unseen crowd with N_p pedestrians, a robot, and their observations $\mathbf{z}_{1:t}$ until time t. Our task is to predict their trajectories $\mathbf{f}$ and $\mathbf{f}^{(R)}$ for H time-steps into the future, using the learned model. We cannot directly use the GP predictive distribution because we do not know the goals of the pedestrians during prediction. Note that we know the goal of the robot $g^{(R)}$ since it is user-defined.

1) Infer goal of a pedestrian: Given observations $\mathbf{z}_{1:t}^{(i)}$ of pedestrian i until time t and the set of goals $\mathbf{G}$, we seek to infer the goal $g^{(i)} \in \mathbf{G}$ of the pedestrian. We assume a uniform prior $P(g^{(i)})$ over all goals, in the absence of any observations for agent i (A more informative prior over the goals can be found by analyzing the environment). Hence, we have:

$$P(g^{(i)}|\mathbf{z}_{1:t}^{(i)}) = \frac{P(\mathbf{z}_{1:t}^{(i)}|g^{(i)})P(g^{(i)})}{P(\mathbf{z}_{1:t}^{(i)})} \propto P(\mathbf{z}_{1:t}^{(i)}|g^{(i)}). \quad (5)$$

That is, we evaluate the likelihood that the observation sequence $\mathbf{z}_{1:t}^{(i)}$ is true conditioned on the fact that $g^{(i)}$ is the goal of agent i. Similar approaches have been explored in [22] and [24], for inferring destination of an agent given its previous path.

To compute the likelihood, we first compute, from the observations $\mathbf{z}_{1:t}$, the velocities $\{(\frac{\Delta\mathbf{x}}{\Delta t}, \frac{\Delta\mathbf{y}}{\Delta t})\}_{1:t-1}$ and the occupancy grids $\mathbf{O}_{1:t-1}$ of all pedestrians at each time-step until $t - 1$. For each possible goal $g \in \mathbf{G}$, we take the corresponding set of trained hyperparameters Θ_x^g and Θ_y^g, and evaluate the log marginal likelihood of the GP for each agent i (using equation 4). Hence, for each agent i and each goal $g^{(i)} \in \mathbf{G}$, we obtain the likelihood that its observed data $\mathbf{z}_{1:t}^{(i)}$ is generated from the GP conditioned on the goal $g^{(i)}$. Normalizing the likelihoods across all goals, we get the likelihood $P(\mathbf{z}_{1:t}^{(i)}|g^{(i)})$ for every goal $g^{(i)} \in \mathbf{G}$.

2) Predicting future trajectories: Now that we have a distribution over the goals $g^{(i)}$ for all agents i in the crowd, we can use the trained model to predict future locations. The joint posterior density can be decomposed as

$$P(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t}) = \sum_{\mathbf{g}} P(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{g}, \mathbf{z}_{1:t})P(\mathbf{g}|\mathbf{z}_{1:t}) \quad (6)$$

where $\mathbf{g} = \{g^{(i)}\}_{i=R,1:N}$ are the goals of all agents including the robot. We can assume that the goals of the pedestrians are independent of each other (and that we know the goal of the robot with certainty) given their respective observations. Then, we can write the distribution of a goal given a history of observations as:

$$P(\mathbf{g}|\mathbf{z}_{1:t}) = \prod_{i=1}^{N} P(g^{(i)}|\mathbf{z}_{1:t}^{(i)}) \quad (7)$$

where $P(g^{(i)}|\mathbf{z}_{1:t}^{(i)})$ is given by equation 5. We approximate the joint distribution $P(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{g}, \mathbf{z}_{1:t})$ by using the velocities and occupancy grids obtained from observations $\mathbf{z}_{1:t}$ (as done

in Section IV-D.1),

$$P(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{g}, \mathbf{z}_{1:t}) \approx P(\mathbf{f}^{(R)}, \mathbf{f}|\{\frac{\Delta \mathbf{x}}{\Delta t}, \frac{\Delta \mathbf{y}}{\Delta t}\}_{1:t-1}, \mathbf{O}_{1:t}, \mathbf{g})$$
(8)

The predictions for different agents are coupled through the occupancy grid which contains the configuration of other agents around each agent locally. This enables our model to capture local interactions, like joint collision avoidance and cooperation.

Since the task is to predict the future locations of all agents for the next H time-steps, $\mathbf{f}^{(i)}$ suffices to represent the next H locations of agent i after time t, in addition to previous locations.

3) Multi-step prediction: Future locations can be predicted using the learned model from Section IV-C. For each agent i, we fit a separate pair of GPs (with the learned hyperparameters Θ_x^g, Θ_y^g for goal g) to the observed tuples $(\mathbf{O}_{1:t}^{(i)}, \{\frac{\Delta \mathbf{x}}{\Delta t}\}_{1:t-1}^{(i)})$ and $(\mathbf{O}_{1:t}^{(i)}, \{\frac{\Delta \mathbf{y}}{\Delta t}\}_{1:t-1}^{(i)})$. Using their corresponding GPs, each agent can predict their velocities and compute the location for the next time-step by adding it to the current location.

This can be done exactly for time $t + 1$, i.e., we can predict $(\frac{\Delta \mathbf{x}}{\Delta t})_{t+1}^{(i)}$ and $(\frac{\Delta \mathbf{y}}{\Delta t})_{t+1}^{(i)}$ for each agent i, since we know the value of the occupancy grid at time t, $\mathbf{O}_t^{(i)}$. But for future time-steps, we need to estimate the occupancy grid at the previous time step using previous predictions. Instead of computing the distribution over future locations in an exact form (which can be extremely difficult), we use Monte Carlo sampling to approximate the distribution as shown in Algorithm 1.

At time $t + 1$, we compute the GP predictive distribution [3] for each $(\frac{\Delta \mathbf{x}}{\Delta t})_{t+1}^{(i)}$ and $(\frac{\Delta \mathbf{y}}{\Delta t})_{t+1}^{(i)}$ (line 2). We then proceed to sample S points from each of these distributions (line 6), and estimate S samples for the location $\mathbf{f}_{t+1}^{(i)}$ for all agents i (line 7). These sets of samples approximate the distribution $P(\mathbf{f}_{t+1}^{(R)}, \mathbf{f}_{t+1}|\{\frac{\Delta \mathbf{x}}{\Delta t}, \frac{\Delta \mathbf{y}}{\Delta t}\}_{1:t-1}, \mathbf{O}_{1:t}, \mathbf{g})$.

Since, for each sample, we have locations of all agents at time $t + 1$, we can compute occupancy grids $\mathbf{O}_{t+1}^{(i)}$ for each agent (line 9). Thus, we get S samples for $\mathbf{O}_{t+1}$. Now, to estimate location at time $t + 2$, we compute the mean of the S samples to get the set of occupancy grids, $\mathbf{O}_{t+1}$ (line 10). Using the mean, we predict the velocities at time $t + 2$ (line 12), and repeat the above process until H time-steps into the future. At every time-step t' ($t' \geq t + 1, t' \leq t + H$), we get a set of S samples corresponding to the locations $\mathbf{f}_{t'}$ that approximate the distribution

$$\{\mathbf{f}_{t'}^{(R)}, \mathbf{f}_{t'}\}_{j=1:S} \approx P(\mathbf{f}_{t'}^{(R)}, \mathbf{f}_{t'}|\{\frac{\Delta \mathbf{x}}{\Delta t}, \frac{\Delta \mathbf{y}}{\Delta t}\}_{1:t-1}, \mathbf{O}_{1:t}, \mathbf{g})$$

As we let the value of S grow, we get a better approximation.

V. EVALUATION

A. Setup

We evaluate our model on a publicly available human-trajectory dataset released by ETH, [11]. The dataset contains a video recorded from above a busy doorway of a university

---

Algorithm 1 Multi-step prediction through Sampling

1: **for** each agent i **do**
2:    Compute distributions of $(\frac{\Delta \mathbf{x}}{\Delta t})_{t+1}^{(i)}$, $(\frac{\Delta \mathbf{y}}{\Delta t})_{t+1}^{(i)}$ given $\mathbf{O}_t^{(i)}$
3: **end for**
4: **for** $t' = t + 2 \to t + H$ **do**
5:    **for** each agent i **do**
6:       Sample S points from distributions of $(\frac{\Delta \mathbf{x}}{\Delta t})_{t'-1}^{(i)}$, $(\frac{\Delta \mathbf{y}}{\Delta t})_{t'-1}^{(i)}$
7:       Compute S estimates for $\mathbf{f}_{t'}^{(i)}$ from sampled velocities
8:    **end for**
9:    Compute S samples for $\mathbf{O}_{t'}$ from estimates of $\mathbf{f}_{t'}$
10:   Set $\mathbf{O}_{t'}$ to be the mean of the S samples from above
11:   **for** each agent i **do**
12:      Compute distributions of $(\frac{\Delta \mathbf{x}}{\Delta t})_{t'}^{(i)}$, $(\frac{\Delta \mathbf{y}}{\Delta t})_{t'}^{(i)}$ given $\mathbf{O}_{t'}^{(i)}$
13:   **end for**
14: **end for**
15: **return** $\{\{\mathbf{f}_{t'}^{(R)}, \mathbf{f}_{t'}\}_{j=1:S}\}_{t'=t+1:t+H}$

Fig. 3: Example snapshot of the dataset with goals indicated by red dots

building with the pedestrian trajectories tracked and annotated. This video contains scenes with real world crowded settings, hundreds of trajectories and high crowd density. An example snapshot from the video (with goals marked) is shown in figure 3. Each time-step in the video is six frames long and amounts to 0.4 seconds. The average trajectory length of a pedestrian is 25 time-steps. The total number of pedestrians in the video is 360 and there are four goals in the environment. The resolution of the video is 640×480. Each pixel in the video frame corresponds to 0.042 metres (slightly varies across the frame, as the camera is angled and not exactly top-down).

We evaluate our model by choosing a pedestrian in the crowd randomly as our robot and use his start and goal state to plan a path through the crowd. The resulting path is compared to the true path that the pedestrian has taken in the video. Comparing the true path and the predicted path gives us a evaluation of how closely our prediction resembles human-like behavior. A similar evaluation was done in [1].

We compare our approach against IGP [1], as it deals with the same problem of jointly modeling trajectories of robot and pedestrians, and has shown good results in real robot navigation scenario among dense crowds [17], [23]. We will

use both our approach and IGP to predict the path until H time-steps into the future and compare it with the true trajectory.

Note that the original IGP needs to know the true final destination of each pedestrian at prediction time, which would give it an unfair advantage over our algorithm. Hence, we use a variant of IGP which doesn't need the true final destinations and use that in our comparison. At prediction time, we compute the average heading of the pedestrian for the last 5 time-steps and estimate the goal location in the computed heading direction. The estimated goal is used in the original IGP algorithm in place of the true final destination of the pedestrian.

To compare the path predicted by the two algorithms and the true path of the pedestrian, we consider two metrics:

1) *Average displacement error*: Introduced in [11], this metric computes the mean squared error over all estimated points at each time-step in the predicted trajectory and the true trajectory.
2) *Final displacement error*: Introduced in [2], this metric computes the mean distance between the final predicted location after H time-steps and the true location after H time-steps, where H is our prediction horizon.

B. Model Parameters

We construct occupancy grids around each agent of size 4×4 (i.e. $M = 4$) covering a space of 80×80 pixels in the video. In each video, we train the model on the trajectories of the first 50 pedestrians. At prediction time, we chose a scenario with 11 pedestrians (from the remaining part of the video, not used in training), one of whom is used as a robot in our model. We predict the future locations for a range of prediction horizons $H = 1, 2, 5, 10, 20$ time-steps. If the path of the pedestrian (or robot) ends in less than H time-steps, we will predict only until the end of his path. For multistep prediction, we use $S = 100$ samples to approximate the distribution over future locations. We implement the Gaussian process regression model using the GPML toolbox, [3].

C. Results

To test the prediction accuracy of both approaches, we have chosen scenarios with 11 pedestrians in the crowd where crowd density is high and some pedestrians head into and through the crowd. To get an unbiased estimate, we chose 5 such scenarios in the video. In each scenario, we assume each pedestrian to be the robot, one at a time, and compute their average displacement error and final displacement error, averaged over all time-steps. This results in 11 sets of error values and we compute the average over all sets to give the mean errors over all pedestrians. We repeat the experiment for different H values to get both short-range and long-range prediction accuracies of both approaches. The results are averaged over all 5 scenarios and are presented in Table I. Note that the errors are listed in pixels.

In Figure 4, we show an example scene where our approach predicts cooperative behavior. The set of pedestrians

Fig. 4: Example prediction by our model. For each pedestrian, we predict his future locations (which are plotted) for the next 5 timesteps. The bottom set of pedestrians are progressing towards a goal at the top centre of the image, but they go around the other set of pedestrians making way for them cooperatively

TABLE I: Prediction errors (in pixels) on the dataset for IGP and our approach

Metric	Prediction horizon (H)	IGP	Our Approach
	1	3.42	4.42
	2	5.66	6.14
Avg. Disp. Error	5	15.75	12.09
	10	21.59	21.52
	20	41.51	34.63
	1	3.42	4.42
	2	7.12	7.78
Final Disp. Error	5	23.18	19.77
	10	38.75	36.25
	20	67.41	54.2

going up give way to the set of pedestrians going down. To visualize what our local interaction model (from section IV-C) learned, we give it some example occupancy grids and goals, and observe the predicted velocities. Figure 5 shows that the model learns collision avoidance as it predicts velocities away from grid cells which are occupied and towards unoccupied grid cells. When an agent's vicinity is heavily populated in the direction of its goal, the magnitude of predicted velocity is very low, i.e., the agent moves slowly. If instead, its vicinity is populated in the opposite direction of its goal, the velocity of the agent doesn't get affected by the surrounding agents (as they are not obstructing its path).

To verify this observation, we have examined the values of the learned hyperparameters of the SE-ARD kernel. The lengthscales for grid cells that are not in the direction of the pedestrian's intended goal, are given high values, thus reducing their relevance in the velocity prediction. For example, in Figure 5 the lengthscales for the bottom grid cells in the left bottom occupancy grid, are set to values higher than 7 whereas the lengthscales for the top grid cells in the same grid are set to values lower than 1. This shows that our model learns how neighbors affect a pedestrian's path based on their relative spatial orientation.
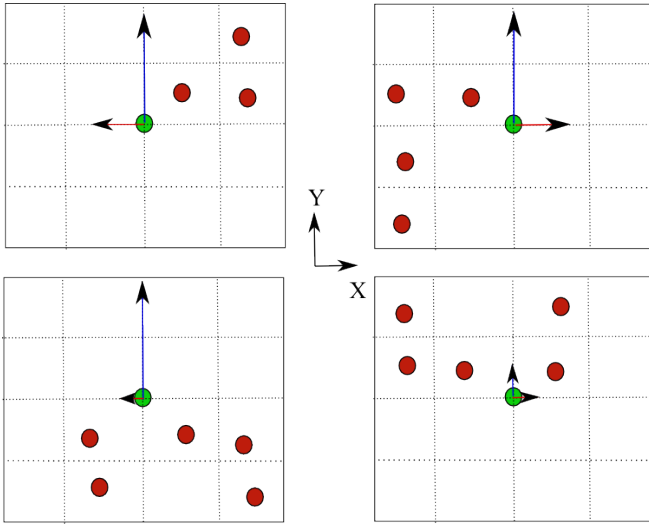
Fig. 5: Velocities predicted by our trained model for example occupancy grids. In each case, the goal of the pedestrian is right above in the Y-direction. Predicted mean y-velocity is shown in blue and predicted mean x-velocity is shown in red.

VI. DISCUSSION

From the results presented in Table I, we can observe that our approach performs better than IGP at predicting human trajectories for longer prediction horizons and worse for shorter horizons. This is mainly because IGP models trajectories directly by predicting future locations based on previously observed locations. This results in very accurate predictions in situations where there are no surrounding agents (and hence, no interactions) and for shorter prediction horizons, as it extrapolates the trajectory to future time-steps smoothly. Our model, on the other hand, models velocities at each time-step and needs to estimate the future location based on velocity predictions for the previous time-step. Thus, for shorter prediction horizons, our model has a higher variance associated with predicted locations. But for longer horizons, our model has higher accuracy in prediction as it reasons about the intended goal of the agent and captures local interactions at each time-step. IGP fails at longer horizons as the smooth extrapolation, coupled with the handcrafted interaction potential term, is unable to account for all the interactions and cooperative behavior among dense crowds. More importantly, our approach has a higher performance than IGP because it learns the local interaction model from real human trajectory data whereas the interaction model in IGP is hand-crafted.

Accurate predictions for longer horizons is important in social navigation as it results in a more globally optimal behavior. In cases where the prediction is accurate in a short horizon but poor for longer horizons, the resulting paths are locally optimal and can potentially lead to a non-socially compliant and reactive behavior.

Upon careful examination of predictions of our approach in crowded scenarios, we observed that it learns the behavior of slowing down (see bottom right of Figure 5) when its vicinity is heavily populated, which is a common behavior among humans. Also, as observed from the values of the

learned lengthscales for the SE-ARD kernel, our model learns how humans decide their velocity based on the relative spatial configuration of other agents in their neighborhood. As shown in Figure 5, our trained local interaction model captures collision avoidance based on an agent's occupancy grid by learning from human trajectory data without any hand-crafted potential term.

Although we present results for predicting trajectories of every agent in the crowd, this approach can be extended to robot navigation by treating the robot as an agent in the crowd. Planning the path of the robot in this model reduces to inference in the joint density as shown in Section III-B. The resulting path taken by the robot is the most likely path predicted according to the learned model. Recent work by [25] has shown that as long as pedestrians interact with the robot naturally (as one of them), such an interaction-aware modeling approach is significantly better than a reactive approach.

VII. CONCLUSION AND FUTURE WORK

In this work, we present a new approach to modeling cooperative behavior among humans in dense crowds. While most existing approaches use hand-crafted models to capture interactions among humans in the crowd, we take a data-driven approach and learn an interaction model from real human trajectory data. We propose a nonparametric statistical model that uses Gaussian processes to model velocities of agents in the crowd as a function of how populated their vicinity is. We show how our model can be used to predict future trajectories of pedestrians and compute the path of a robot through a dense crowd. The future trajectories are computed using a Monte Carlo sampling approach to multi-step prediction. Lastly, the efficacy of our approach is demonstrated by predicting trajectories of agents in a real world pedestrian dataset. Our results show that the model captures important aspects of human crowd behavior such as cooperative navigation and collision avoidance.

Currently, the model doesn't account for static obstacles which play a very important role in modeling navigation behavior. An interesting future direction would be to explore ways to account for both the dynamic pedestrians and static obstacles in the environment, while predicting future trajectories. Another important drawback of our approach is the assumption of known goals in the environment. This restricts the generalizability of the approach to previously seen environments and a separate model needs to be trained for a new environment.

As a part of future work, we plan to validate and verify our approach on a real robot placed in a dense human crowd. The task would be, given a start and goal location, the robot should be able to navigate safely and efficiently through the crowd. In addition to validating our approach, we intend to tackle the drawbacks of the approach as stated before. We are also looking to extend the model by coming up with latent representations of trajectories that encode time-varying information such as pedestrian's intention, planned future

path and velocities, that can be used instead of an occupancy grid in our approach.

ACKNOWLEDGMENTS

REFERENCES

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

[2] Alexandre Alahi and Kratarth Goel and Vignesh Ramanathan and Alexandre Robicquet and Li Fei-Fei and Silvio Savarese. "Social LSTM: Human Trajectory Prediction in Crowded Spaces." In IEEE International Conference on Computer Vision and Pattern Recognition 2016

[3] Williams, Christopher KI, and Carl Edward Rasmussen. "Gaussian processes for machine learning." The MIT Press, 2006.

[4] Large, Frdric, Dizan Vasquez, Thierry Fraichard, and Christian Laugier. "Avoiding cars and pedestrians using velocity obstacles and motion prediction." In Intelligent Vehicles Symposium, pp. 375-379. IEEE, 2004.

[5] Philippsen, R., and R. Siegwart. "Smooth and efficient obstacle avoidance for a tour guide robot." In Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on, vol. 1, pp. 446-451. IEEE, 2003.

[6] Burgard, Wolfram, Armin B. Cremers, Dieter Fox, Dirk Hhnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. "Experiences with an interactive museum tour-guide robot." Artificial intelligence 114, no. 1 (1999): 3-55.

[7] Thompson, Simon, Takehiro Horiuchi, and Satoshi Kagami. "A probabilistic model of human motion and navigation intent for mobile robot path planning." In Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on, pp. 663-668. IEEE, 2009.

[8] Bennewitz, Maren, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. "Learning motion patterns of people for compliant robot motion." The International Journal of Robotics Research 24, no. 1 (2005): 31-48.

[9] Helbing, Dirk, and Peter Molnar. "Social force model for pedestrian dynamics." Physical review E 51, no. 5 (1995): 4282.

[10] Treuille, Adrien, Seth Cooper and Zoran Popovic. Continuum crowds. ACM Trans. Graph. 25 (2006): 1160-1168.

[11] Pellegrini, Stefano, Andreas Ess, Konrad Schindler, and Luc Van Gool. "You'll never walk alone: Modeling social behavior for multi-target tracking." In 2009 IEEE 12th International Conference on Computer Vision, pp. 261-268. IEEE, 2009.

[12] Svenstrup, Mikael, Thomas Bak, and Hans Jrgen Andersen. "Trajectory planning for robots in dynamic human environments." In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 4293-4298. IEEE, 2010.

[13] Pradhan, Ninad, Timothy Burg, and Stan Birchfield. "Robot crowd navigation using predictive position fields in the potential function framework." In Proceedings of the 2011 American Control Conference, pp. 4628-4633. IEEE, 2011.

[14] Helbing, Dirk, and Anders Johansson. Pedestrian, crowd and evacuation dynamics. Springer New York, 2011.

[15] Johansson, Anders, Dirk Helbing, and Pradyumn K. Shukla. "Specification of the social force pedestrian model by evolutionary adjustment to video tracking data." Advances in complex systems 10, no. supp02 (2007): 271-288.

[16] Hall, Edward T. "A system for the notation of proxemic behavior." American anthropologist 65, no. 5 (1963): 1003-1026.

[17] Trautman, Peter, Jeremy Ma, Richard M. Murray, and Andreas Krause. "Robot navigation in dense human crowds: the case for cooperation." In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp. 2153-2160. IEEE, 2013.

[18] Kuderer, Markus, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. "Feature-Based Prediction of Trajectories for Socially Compliant Navigation." In Robotics: science and systems. 2012.

[19] Kretzschmar, Henrik, Markus Spies, Christoph Sprunk, and Wolfram Burgard. "Socially compliant mobile robot navigation via inverse reinforcement learning." The International Journal of Robotics Research (2016): 0278364915619772.

[20] Kim, Kihwan, Dongryeol Lee, and Irfan Essa. "Gaussian process regression flow for analysis of motion trajectories." In 2011 International Conference on Computer Vision, pp. 1164-1171. IEEE, 2011.

[21] Joseph, Joshua, Finale Doshi-Velez, Albert S. Huang, and Nicholas Roy. "A Bayesian nonparametric approach to modeling motion patterns." Autonomous Robots 31, no. 4 (2011): 383-400.

[22] Kitani, Kris M., Brian D. Ziebart, James Andrew Bagnell, and Martial Hebert. "Activity forecasting." In European Conference on Computer Vision, pp. 201-214. Springer Berlin Heidelberg, 2012.

[23] Trautman, Peter, Jeremy Ma, Richard M. Murray and Andreas Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation. I. J. Robotics Res. 34 (2015): 335-356.

[24] Ziebart, Brian D., Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J. Andrew Bagnell, Martial Hebert, Anind K. Dey, and Siddhartha Srinivasa. "Planning-based prediction for pedestrians." In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pp. 3931-3936. IEEE, 2009.

[25] Pfeiffer, Mark, Ulrich Schwesinger, Hannes Sommer, Enric Galceran, and Roland Siegwart. "Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models." In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, pp. 2096-2101. IEEE, 2016.

[26] Van den Berg, Jur, Ming Lin, and Dinesh Manocha. "Reciprocal velocity obstacles for real-time multi-agent navigation." In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pp. 1928-1935. IEEE, 2008.

[27] Fiorini, Paolo, and Zvi Shiller. "Motion planning in dynamic environments using velocity obstacles." The International Journal of Robotics Research 17, no. 7 (1998): 760-772.