The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Unfreezing the Robot: Navigation in Dense, Interacting Crowds

Peter Trautman and Andreas Krause

Abstract— **In this paper, we study the safe navigation of a mobile robot through crowds of dynamic agents with uncertain trajectories. Existing algorithms suffer from the "freezing robot" problem: once the environment surpasses a certain level of complexity, the planner decides that all forward paths are unsafe, and the robot freezes in place (or performs unnecessary maneuvers) to avoid collisions. Since a feasible path typically exists, this behavior is suboptimal. Existing approaches have focused on reducing the predictive uncertainty for individual agents by employing more informed models or heuristically limiting the predictive covariance to prevent this overcautious behavior. In this work, we demonstrate that both the individual prediction and the predictive uncertainty have little to do with the frozen robot problem. Our key insight is that dynamic agents solve the frozen robot problem by engaging in "joint collision avoidance": They cooperatively make room to create feasible trajectories. We develop IGP, a nonparametric statistical model based on dependent output Gaussian processes that can estimate crowd interaction from data. Our model naturally captures the non-Markov nature of agent trajectories, as well as their goal-driven navigation. We then show how planning in this model can be efficiently implemented using particle based inference. Lastly, we evaluate our model on a dataset of pedestrians entering and leaving a building, first comparing the model with actual pedestrians, and find that the algorithm either outperforms human pedestrians or performs very similarly to the pedestrians. We also present an experiment where a covariance reduction method results in highly overcautious behavior, while our model performs desirably.**

I. INTRODUCTION

Navigation in cluttered, dynamic environments is a challenging problem whose solution would have numerous applications. In particular, assistive robots [22] in crowded environments (malls, cafeterias, and libraries are obvious examples) would benefit from a navigation algorithm which could operate safely amidst moving agents in close proximity to one another. Unfortunately, classical algorithms for navigation in static or dynamic environments typically ignore uncertainty [15], and thus do not generalize well to situations where the behavior of the environment is stochastic. Current approaches, such as [23], [2], [10], [14] develop human motion models to augment the prediction, in the hope that good prediction will lead to good navigation. Similary, in [27], more informed prediction is achieved by collecting large amounts of pedestrian data in an office space, and then modeling the goal-directed trajectories of pedestrians using maximum entropy inverse optimal control.

Peter Trautman is a student in the Control and Dynamical Systems option, California Institute of Technology, 1200 E. California Blvd, Pasadena CA 91125, USA. trautman@cds.caltech.edu

Andreas Krause is a faculty member in the Computer Science option, California Institute of Technology, 1200 E. California Blvd, Pasadena CA 91125, USA. krausea@caltech.edu

What these approaches have in common is that each agent is modeled *independently* of the others[1]; because of this modeling assumption, what always occurs, given dense enough crowds, is something colloquially known as the "freezing robot problem" (FRP). Intuitively, the FRP occurs when the robot believes the environment to be unsafe, i.e., every path is expected to collide with the an agent in the crowd due to massive uncertainty. Thus, the robot either makes no forward progress, or takes extreme evasive action to avoid collisions within the crowd.

One intuitive strategy to mitigate the FRP would be to use "more informed" models, i.e., those that attempt to limit the predictive uncertainty about the crowd motion to allow identification of feasible paths that are unlikely to lead to collisions. In fact, [6] goes one step further, (heuristically) holding the individual agent predictive covariance *constant* at a low value as a surrogate for near perfect prediction (in the hopes that as the robot gets close to the dynamic agents, the prediction will be good enough for safe navigation to occur). However, perhaps surprisingly, in Section II, we show that even under *perfect prediction*, i.e., each agent's trajectory is *known* to the planning algorithm, the FRP still occurs if the crowd density is high enough. Thus, obtaining more accurate predictive models for individual behavior alone cannot be expected to solve the FRP.

Given this observation, how is it possible that people can safely navigate through crowds (see, for example, Figure 1 showing pedestrians safely navigating past each other on a crowded sidewalk)? The key insight is that people typically engage in *joint* collision avoidance (called the *social forces model* in [8], [9], [7]): they adapt their trajectories to each other to make room for navigation. Further evidence from the multi-robot coordination community and specifically the work of van den Berg et al. ([26], [25]), show that robots programmed to jointly avoid each other are guaranteed to be collision free and show improved efficiency at joint navigation tasks.

This joint collision avoidance criteria has been exploited to improve the data association and target tracking of individuals in human crowds ([18], [19], [16]). To our knowledge, however, it has not been utilized to improve *navigation* in human crowds. Thus, our central idea is to explicitly model the interactions among the agents and between the robot and the crowd, for the purpose of navigation. To this end, we develop *interacting Gaussian processes* (IGP), a principled statistical model, based on dependent output

[1]The recent work of [11] does not assume independence since the cost function is *learned* from (simulated) crowd interaction data.

Gaussian Processes. IGP describes a probabilistic interaction between multiple navigating entities. Furthermore, efficient planning is naturally implemented in this model using a particle based approximate inference scheme.

Fig. 1. Crowded street scene

We evaluate our model on real world pedestrian data (see Figure 2 and Section V-B). Our results show that IGP leads to better (shorter and safer) paths than those taken by the observed pedestrians.

In summary, our main contributions are

- the formalization of the FRP, which is fundamental to navigation in cluttered, dynamic environments,
- the insight that modeling cooperative collision avoidance is required to solve the FRP,
- the development of a novel model for interaction based on coupled output Gaussian Processes (GPs), called *Interacting Gaussian Processes*,
- the development of an approximate inference algorithm for prediction and navigation of the IGP, and
- the demonstration of the effectiveness of this model under real world crowd conditions.

II. THE FREEZING ROBOT PROBLEM

A. Formalizing the FRP

We now formalize the FRP. Fix an agent i, where the index i can take values in the set $\{R, 1, 2, \ldots, n\}$, and $i = R$ indicates the robot. Suppose we have a prior distribution $p(\mathbf{f}^{(i)})$ over the agent's trajectory

$$\mathbf{f}^{(i)} = (f_1^{(i)}, \ldots, f_T^{(i)})$$

over T timesteps, where each $f_t^{(i)} = (x_t, y_t) \in \mathbb{R}^2$ is the planar location of agent i at time t. We also have a likelihood function $p(z_t^{(i)} \mid f_t^{(i)})$ for our observations. In the following, we will assume that the observations do not depend on the robot's actions.

After obtaining the first t observations $\mathbf{z}_{1:t}^{(i)}$, we can perform Bayesian inference to calculate the posterior $p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}^{(i)})$. Assuming all agents behave independently of each other, we have a factorizable prior joint distribution over trajectories

$$p(\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)}) = \prod_i p(\mathbf{f}^{(i)}).$$

Thus, the posterior remains independent

$$p(\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)} \mid \mathbf{z}_{1:t}) = \prod_i p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}^{(i)}),$$

and $\mathbf{z}_{1:t} = \left\{\mathbf{z}_{1:t}^{(i)}\right\}_{i=1}^{n}$ is the set of observations about all agents.

Our goal in dynamic navigation is to pick a policy π that adaptively chooses a path $\mathbf{f}^{(R)}$ for the robot based on its observations. The policy π is typically specified by stating which next location $f_{t+1}^{(R)}$ the robot should choose given observations $\mathbf{z}_{1:t}$.

Thus, for any complete sequence of observations $\mathbf{z}_{1:T}$, the robot can potentially end up choosing a different path $\mathbf{f}^{(R)} = \pi(\mathbf{z}_{1:T})$. The cost $J(\pi)$ of a policy π is the expected cost

$$J(\pi) = \int p(\mathbf{f}, \mathbf{z}_{1:T}) c(\pi(\mathbf{z}_{1:T}), \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)}) d\mathbf{f} d\mathbf{z}_{1:T}, \tag{II.1}$$

where, for a fixed robot trajectory $\mathbf{f}^{(R)}$, the cost function $c(\mathbf{f}^{(R)}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)})$ models the length of the path plus penalties for colliding with any of the agents. We use the shorthand notation $\mathbf{f} = (\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)})$.

Unfortunately, even if the observations $\mathbf{z}_{1:T}$ do not depend on the robot's actions[2], solving for the optimal policy π requires solving a continuous-state *Markov Decision Process* (MDP), where the dimensionality grows linearly with the number of agents, which is intractable.

The intractability of such MDPs is fairly common; in the path planning community, a state of the art, tractable approximation to the MDP is a method called *Receding Horizon Control* (RHC). RHC proceeds in a manner similar to MDPs, albeit online: as observations become available, RHC calculates, based on some cost function, the optimal *non-adaptive* action (i.e., fixed path) to take at that time. Indeed, if we let $J(\mathbf{f}^{(R)} \mid \mathbf{z}_{1:t})$ be the objective function which calculates the "cost" of each path $\mathbf{f}^{(R)}$ based on the observations $\mathbf{z}_{1:t}$, that is

$$J(\mathbf{f}^{(R)} \mid \mathbf{z}_{1:t}) = \int c(\mathbf{f}^{(R)}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)}) p(\mathbf{f} \mid \mathbf{z}_{1:t}) d\mathbf{f}, \tag{II.2}$$

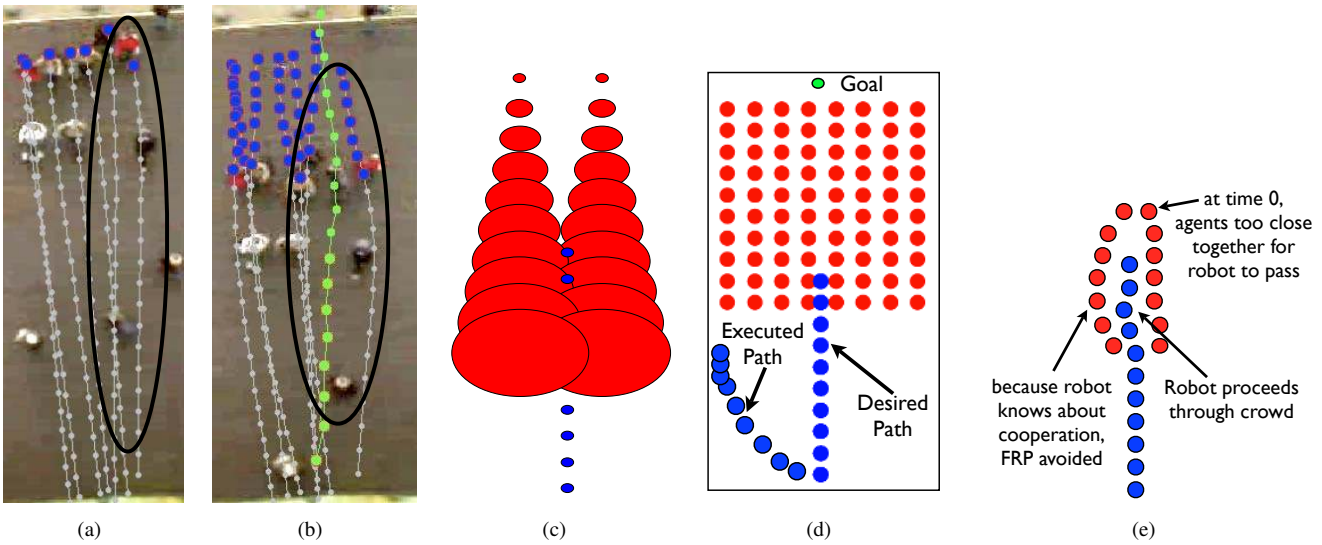
where $\mathbf{f}^{(R)}$ is the trajectory of the robot, then RHC finds $\mathbf{f}_t^*$, where

$$\mathbf{f}_t^* = \arg\min_{\mathbf{f}^{(R)}} J(\mathbf{f}^{(R)} | \mathbf{z}_{1:t}). \tag{II.3}$$

As each new observation $\mathbf{z}_\tau$ arrives, for $\tau > t$, a new path $\mathbf{f}_\tau^*$ is calculated and executed until another observation arrives.

Unfortunately, for crowded environments, $J(\mathbf{f}^{(R)} | \mathbf{z}_{1:t}) \gg 0$ for *any* path $\mathbf{f}^{(R)}$. Depending on the density of the crowd, $J(\mathbf{f}^{(R)} | \mathbf{z}_{1:t})$ can become arbitrarily large, causing the navigation algorithm to either freeze or take unnecessary evasive action. This is the Freezing Robot Problem (see Figure 2(d) for an illustration).

[2]If the observations depend on the actions, the problem becomes a continuous state-space POMDP which is even more intractable.



(a) (b) (c) (d) (e)

Fig. 2. **(a,b) Empirical evidence of joint collision avoidance:** blue circles (representing current position) over gray lines are pedestrians moving down, and black circles are the area of interest. In (a), the blue pedestrians have not yet seen the green person; their projected trajectories (in gray) are very narrow. In (b), green dots with red encircling are current position of the pedestrian moving up, and *all* of the pedestrians have adjusted their trajectories to create space—notice how wide the gray prediction has become. It is this joint collision avoidance behavior which we capture in this paper. **(c-e) Illustration of FRP.** Dynamic crowd agents in red traveling downward, robot we are trying to control in blue. The multiple dots indicate multiple points along one trajectory. (c) Uncertainty explosion due to uncorrected prediction. (d) Even with perfect prediction, room for robot navigation may not exist. (e) Modeling cooperative collision avoidance remedies the FRP.

B. Approaches for solving the FRP

In order to fix the FRP, one state of the art approach [6], called *partially closed loop receding horizon control* (PCLRHC), anticipates the observations (effectively hallucinating that a certain measurement sequence of the *entire* trajectory sequence has already taken place at time $t < T$); ultimately, the approach is motivated by the assumption that the culprit of the FRP is an uncertainty explosion, illustrated in Figure 2(c). The claim is that if you can control the covariance, then you can keep the value of $J(\mathbf{f}^{(R)}|\mathbf{z}_{1:t})$ low for some (short path length) trajectories $\mathbf{f}^{(R)}$, and thus solve the FRP (other approaches, which incorporate more accurate agent modeling, are similar in motivation to PCLRHC, since better dynamic models would reduce predictive covariance as well).

However, in severely crowded environments, even the *optimal solution* to the MDP suffers from freezing robot behavior. This is because we can lower bound the optimal MDP cost by

$$\mathbb{E}_{\mathbf{z}_{1:T}}[\min_{\mathbf{f}^{(R)}} J(\mathbf{f}^{(R)}|\mathbf{z}_{1:T})], \tag{II.4}$$

which is the expected cost in the case of *perfect* prediction (i.e., knowing the future). Unfortunately, in dense environments, such as those shown in Figures 2(d) and 2(a), this lower bound can still be prohibitively expensive, and lead even optimal planners (such as the MDP) to exhibit freezing robot actions. It follows that suboptimal methods, which work at improving the prediction or reducing the covariance, cannot be expected to solve the FRP.

This analysis suggests that the planning problem as introduced above is ill-posed. We thus revisit our probability density,

$$p(\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)} \mid \mathbf{z}_{1:t}), \tag{II.5}$$

and remark that a crucial element is missing—the agent motion model is agnostic of the navigating robot. One solution is thus immediately apparent: include an interaction between the robots and the agents (in particular, a joint collision avoidance) in order to lower the MDP cost in equation II.4. We additionally remark that the illustration in Figure 2(b), the crowd experiments catalogued in the research of [8], [9], [7], the multi-robot coordination theorems of [26], [25], and the tracking experiments of [18], [19], [16], all corroborate the argument that autonomous dynamic agents utilize joint collision avoidance behaviors for successful crowd navigation. We thus consider methods to incorporate such an interaction.

A naive approach to modeling interaction would be to model a *conditional* density $p(\mathbf{f} \mid \mathbf{z}_{1:t}, \mathbf{f}^{(R)})$, that encodes assumptions on how the agents react to the robot's actions, i.e., the idea that all agents will "give way" to the robot's trajectory. The problem with this approach is that it implicitly assumes that the robot has the ability to control the crowd. Thus, this approach would not only create an obnoxious robot, but an overaggressive and potentially dangerous one as well. This method is unsuitable for crowded situations.

The other alternative, which we advocate in this paper, is to consider a robot action as an *agent action* (i.e., the robot is modeled as one of the agents) and to model a joint distribution describing their interaction:

$$p(\mathbf{f}^{(R)}, \mathbf{f} \mid, \mathbf{z}_{1:t}). \tag{II.6}$$

This distribution encodes the idea of cooperative planning, and joint collision avoidance. Planning then corresponds to

computing $p(\mathbf{f}^{(R)} \mid \mathbf{z}_{1:t})$, i.e., inferring what the robot *should* do given observations of the other agents. In effect, the dynamic agents and the robot cooperate to simultaneously achieve the best (safest and shortest) paths. This idea is illustrated in Figure 2(e). Section III formalizes the density $p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t})$.

III. INTERACTION GP (IGP) MODEL

A. Gaussian process models for trajectories

A Gaussian process (GP) [20] is a distribution over (typically smooth) functions, and thus arguably well-suited to model trajectories. Formally, a GP is a collection of Gaussian random variables indexed by a set, in our case, the set of time steps $\{1, \ldots, T\}$, and parameterized by a mean function m (typically taken as zero without loss of generality) and covariance (or kernel) function k. The kernel parameterizes the smoothness of the functions, and can be learned from data. In a sense, GPs generalize linear models such as Kalman filters by replacing the Markov assumption with a (more general) smoothness assumption. This fact alone leads to less diffuse predictions than standard Kalman filters.

We start by modeling each agent's trajectory as an independent sample from a GP, $\mathbf{f}^{(i)} \sim GP(0, k)$. For simplicity of notation, we formalize the model for one-dimensional locations only – multiple dimensions are easily incorporated by modeling each dimension as a separate GP. Given the observations $\mathbf{z}_{1:t}^{(i)}$ through time t, we can calculate the individual posterior as $p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}^{(i)}) = GP(\mathbf{f}^{(i)}, m_t^{(i)}, k_t^{(i)})$, where

$$m_t^{(i)}(t') = \Sigma_{1:t,t'}^{\mathrm{T}}(\Sigma_{1:t,1:t} + \sigma^2\mathbf{I})^{-1}\mathbf{z}_{1:t}^{(i)}, \tag{III.1}$$

$$k_t^{(i)}(t_1, t_2) = k(t_1, t_2) - \Sigma_{1:t,t_1}^{\mathrm{T}}(\Sigma_{1:t,1:t} + \sigma^2\mathbf{I})^{-1}\Sigma_{1:t,t_2}. \tag{III.2}$$

Hereby, $\Sigma_{1:t,t'} = [k(1,t'), k(2,t'), \ldots, k(t,t')]$, and $\Sigma_{1:t,1:t}$ is the matrix such that the (i,j) entry is $\Sigma_{i,j} = k(i,j)$ and the indices (i,j) take values from $1:t$. Lastly, σ^2 is the measurement noise (which is assumed to be Gaussian).

B. Incorporating goal information

An advantage to the GP formalism is that it estimates the entire trajectory in a non-Markovian way. Indeed, this allows us to incorporate goal information (either probabilistic or exact) in a principled way, such that the resulting distribution over trajectories reflects the full impact of the additional data. Implementation-wise, we merely treat the goal information as a measurement on the final step of the trajectory, i.e., observing $\mathbf{z}_T^{(i)}$ to be the perceived goal. By varying the amount of noise in the measurement, we can encode how certain we are about the goal. For $\mathbf{z}_T^{(R)}$, i.e., the robot's goal, we set the noise very small. Furthermore, waypoints along the trajectory could be easily encoded in the same manner. Compare this with a Kalman filter, which has no way of naturally incorporating such information.

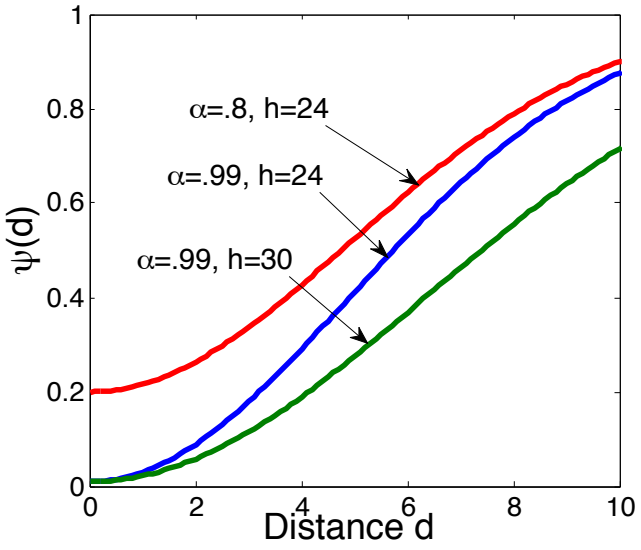


Fig. 3. The interaction potential $1 - \alpha \exp(-\frac{1}{2h^2}d^2)$, for various α, h.

C. Interaction GP (IGP) model for cooperative navigation

Our key modeling idea is to capture the dynamic interactions by introducing dependencies between the GPs. We begin with the independent GP priors

$$p(\mathbf{f}^{(R)} \mid \mathbf{z}_{1:t}),\ p(\mathbf{f}^{(1)} \mid \mathbf{z}_{1:t}),\ \ldots,\ p(\mathbf{f}^{(n)} \mid \mathbf{z}_{1:t}),$$

and couple them by multiplying in an *interaction potential*

$$\psi(\mathbf{f}^{(R)}, \mathbf{f}) = \psi(\mathbf{f}^{(R)}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)}),$$

so that

$$p_{\mathsf{IGP}}(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t}) = \frac{1}{Z}\psi(\mathbf{f}^{(R)}, \mathbf{f}) \prod_{i=R}^{n} p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}). \tag{III.3}$$

The product $\prod_{i=R}^{n}$ is meant to indicate that the robot is included in the calculation. In our experiments, we choose the interaction potential as:

$$\psi(\mathbf{f}^{(R)}, \mathbf{f}) = \prod_{i=R}^{n}\prod_{j=i+1}^{n}\prod_{\tau=t}^{T}\Big(1 - \alpha \exp\big(-\frac{1}{2h^2}|f_\tau^{(i)} - f_\tau^{(j)}|\big)\Big) \tag{III.4}$$

where $|f_\tau^{(i)} - f_\tau^{(j)}|$ is the Euclidean distance at time τ between agent i and agent j. The rationale behind our choice is that any specific instantiation of paths

$$\mathbf{f}_k^{(R)}, \mathbf{f}_k^{(1)}, \mathbf{f}_k^{(2)}, \ldots, \mathbf{f}_k^{(n)}$$

(where the subscript k indicates a point in path space, so that $\mathbf{f}_k^{(i)} \in \mathbb{R}^{2T}$ for all i) becomes very unlikely if, at any time τ, any two agents i and j are too close. Furthermore, the parameter h controls the "safety margin" of the repulsion, and $\alpha \in [0,1]$ the strength of the repulsion.

The parameter h was chosen to be the closest approach of two navigating pedestrians (out of the entire video sequence, approximately 10 pixels), while α was chosen such that $h =$

10 was enforced the vast majority of the time. See Figure 3 for an illustration.

IV. COOPERATIVE PLANNING AND INFERENCE

A. The Navigation Density

Our model $p_{\mathsf{IGP}}(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t})$ immediately suggests a natural way to perform navigation: at time t, find the *maximum a-posteriori* (MAP) assignment for the posterior

$$(\mathbf{f}^{(R)}, \mathbf{f})^* = \arg\max_{\mathbf{f}^{(R)}, \mathbf{f}} p_{\mathsf{IGP}}(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t}), \tag{IV.1}$$

and then take $f_{t+1}^{(R)*}$ as the next action in the path (where $t+1$ means the next step of the estimation). At time $t+1$, we receive a new observation of the agents and the robot, update the posterior to $p_{\mathsf{IGP}}(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t+1})$, find the MAP assignment again and choose $f_{t+2}^{(R)*}$ as the next step in the path. We repeat this process until the robot has arrived at its destination.

B. Importance sampling for approximate inference in IGP

While in GPs exact, efficient inference is possible, the introduction of the interaction potential makes the posterior $p_{\mathsf{IGP}}(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t})$ non-Gaussian and thus approximate inference is required. Standard approaches to approximate inference in models derived from GPs include Laplace approximation [3] and Expectation Propagation [17]. These methods approximate the non-Gaussian posterior by a Gaussian which has the same mode, or which minimizes the Kullback-Leibler divergence respectively. These methods are most effective if the posterior is unimodal (and can be well-approximated by a Gaussian). In IGP, however, the posterior is expected to be multimodal: In particular, for two agents moving towards each other in a straight line, evasion in either direction is equally likely. This is akin to people walking towards each other, flipping from one "mode" to the other while attempting to not collide.

To cope with the multimodality, we use an approximate inference technique based on importance sampling, a well understood approximate inference method for Bayesian statistics (for an introduction see [1], [4], [21], [24]; for a more detailed, up to date analysis of the method see [5], [13]). We implement importance sampling (see chapter 4 of [12] for a discussion of importance sampling) for estimation of the navigation density as follows:

- For all agents i, sample independent trajectories of agent i from the prior (see [20] for a discussion of sampling functions from a GP):

$$(\mathbf{f}^{(i)})_k \sim p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}), \tag{IV.2}$$

  where $p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t})$ is the sampling density (GP) for agent trajectory i.

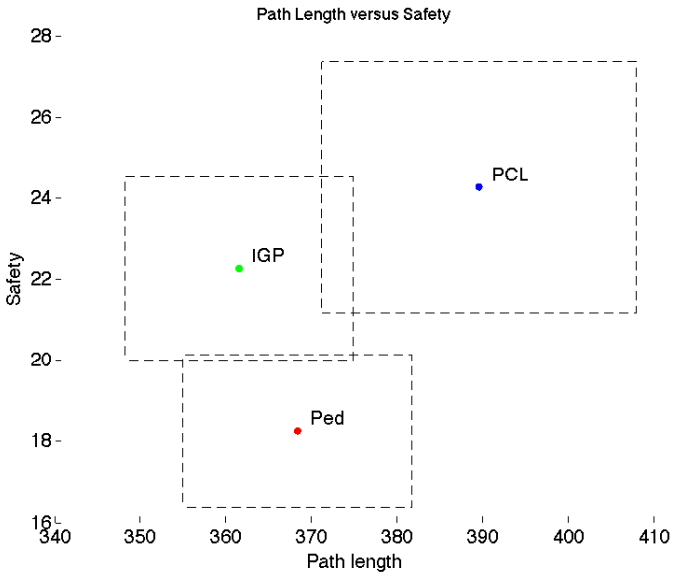


Fig. 4. Path length versus safety over 10 runs. IGP outperforms pedestrians in both safety and path length, while PCLRHC is inappropriate for this application.

- Evaluate the weight of each sample $(\mathbf{f}^{(R)}, \mathbf{f})_k$ using the rules of importance sampling:

$$w_k = \frac{p_{\mathsf{IGP}}((\mathbf{f}^{(R)}, \mathbf{f})_k \mid \mathbf{z}_{1:t})}{\prod_{i=R}^{n} p((\mathbf{f}^{(i)})_k \mid \mathbf{z}_{1:t})} \tag{IV.3}$$

$$= \frac{\psi((\mathbf{f}^{(R)}, \mathbf{f})_k) \prod_{j=R}^{n} p((\mathbf{f}^{(i)})_k \mid \mathbf{z}_{1:t})}{\prod_{i=R}^{n} p((\mathbf{f}^{(i)})_k \mid \mathbf{z}_{1:t})} \tag{IV.4}$$

$$= \psi((\mathbf{f}^{(R)}, \mathbf{f})_k). \tag{IV.5}$$

- The posterior is then approximated by the empirical sampling distribution,

$$p_{\mathsf{IGP}} \approx \sum_{k=1}^{N} w_k \delta[(\mathbf{f}^{(R)}, \mathbf{f})_k - (\mathbf{f}^{(R)}, \mathbf{f})], \tag{IV.6}$$

  where $\delta[(\mathbf{f}^{(R)}, \mathbf{f})_k - (\mathbf{f}^{(R)}, \mathbf{f})]$ is the delta function centered at sample $(\mathbf{f}^{(R)}, \mathbf{f})_k$.

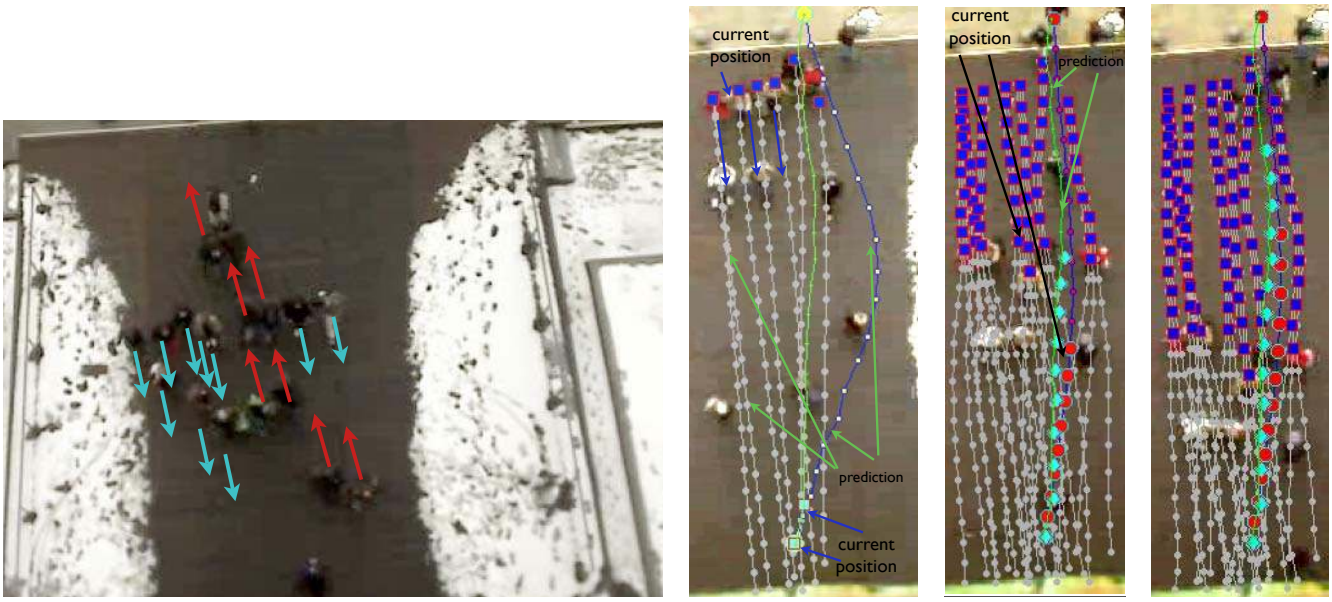
As we let the number N of samples grow, we approximate p_{IGP} to arbitrary accuracy, based on how much computation we are willing to dedicate. Note that all samples are independent of one another. Thus, the technique can be parallelized.

In practice, we found that as few as 100 particles were sufficient for navigation and was computable in about 0.1 seconds in Matlab. We found that solutions did not really improve beyond about 5000 particles, which took about 5 seconds to run in Matlab. Thus, our approach could realistically be run in real time on a robot.

V. EVALUATION

A. Experimental setup: crowded pedestrian data

We evaluate our approach on a data set of over 8 minutes of video recorded from above a doorway of a university building at ETH (see [18] for more details of the video collect and how to access the data). This data set exhibits high



(a) *Example pedestrian interaction* (b) *PCLRHC* (c) *IGP* (d) *IGP*

Fig. 5. (a) Crowded still from the ETH data sequence. Near the center of the group is a subgroup of about 6 people moving upwards (red arrows) through a crowd of about 10 people moving down (cyan arrows). Experiments were run on this particular scenario, with IGP performing (in terms of safety and path length) about the same or slightly better than the actual pedestrians, and greatly outperforming state of the art methods, such as seen in [6]. (b-d) The blue squares over the gray lines are the agents traveling downward (lowest dot is current position), the cyan diamond over the green line is the pedestrian walking upwards through the crowd; IGP is red circles on top of blue prediction line, and PCLRHC is blue prediction line. In (b) PCLRHC chooses an overcautious path because the crowd is too dense. In (c) and (d), IGP follows nearly the same path as the pedestrian in green, validating the model. This set of figures illustrates the free space created by the pedestrian walking through the crowd—this is the interaction we capture with the IGP model.

TABLE I

NAVIGATION RESULTS: IGP VERSUS PEDESTRIAN

Run	ℓ_{ped}	s_{ped}	ℓ_{IGP}	s_{IGP}	ℓ_{PCLRHC}	s_{PCLRHC}
1	343	13	341	12	353	22
2	343	14	344	18	349	8
3	316	71	305	26	317	73
4	383	12	358	21	420	18
5	361	12	363	42	409	65
6	337	21	321	22	330	23
7	439	16	439	23	489	26
8	428	19	423	20	466	11
9	416	20	402	18	448	24
10	415	11	407	24	445	13

crowd density, i.e., people frequently pass by one another fairly closely). As an example, see Figure 5 for one frame of the data sequence in which the crowds are dense. In this frame, a number of pedestrians are heading down towards the doorway (cyan arrows) while a few other people (red arrows) head *into* and *through* the crowd.

We test the IGP algorithm on variations of just these types of scenarios (one crowd or person intersecting another crowd); our task is to utilize the navigation density in combination with the particle filtering inference method to do navigation through these crowds. Videos demonstrating the navigation performance are available at http://www.cds.caltech.edu/∼trautman

Given the type of data that we are going to be experimenting with, we now explain our performance metric: For navigation, we are interested in two quantities: *path length* (the euclidean path distance in $x-y$ space taken by the robot from start to finish), and *safety margin* (the nearest distance that the robot ever came to another pedestrian during a run).

We measure both these quantities in pixel values, because transforming back to "real" distances (meters, for instance) would be too inaccurate. Importantly, we have baselines for the two metrics in pixels. For path length, we tended to see pedestrians take paths which ranged from about 350-390 pixels. For the safety margin, we often observed pedestrians within 11-12 pixels of one another, although never any closer. So we take as "safe" any separation distance above 13 pixels. Furthermore, we can roughly estimate 13 pixels to be about the width of a person from shoulder to shoulder.

An experimental explanation is in order. True validation of the IGP algorithm demands "live" interaction—that is, in order to test the concept of joint collision avoidance, a robot must actually interact with human beings. Unfortunately, conducting such an experiment was infeasible at this time (see Section VI for details on future experiments).

Instead, we conducted what we felt was the next best experiment. First, we used a dataset of human crowds, rather than simulated dynamic agents. Second, in order to test joint collision avoidance, we gave IGP and PCLRHC the same start and goal states as a human navigating through a crowd, and ran the algorithms simultaneously with the human. In other words, the person created space, and we tested the algorithms to see if they would *anticipate* that space. The fact that IGP took nearly identical paths to the humans and PCLRHC chose highly conservative paths justified, to some

extent, our approach. Furthermore, examinations of *planned* paths at early stages in the experiment showed IGP expecting the opening in the crowd, while PCLRHC expected no such event.

B. Navigation performance

We begin this section with anecdotal evidence of how our algorithm performs in comparison to both pedestrians and PCLRHC, in Figure 5. Note that for all 10 experiments we ran, this behavior was typical: IGP performed similarly or better than the pedestrian, and PCLRHC took evasive action, usually going to the far outside to avoid the crowds.

Figure 4 is the main experimental result of this paper. In Figure 4, we present the results of our algorithm over 10 experiments. Each box surrounding the colored dots represents the standard error bars over the 10 experiments. IGP (green dot) had a mean safety of around 22 pixels, with standard error ranging over 2 pixels, and mean path length of around 362, with standard error around 12. Table I presents details for the 10 individual experiments. Columns labeled s refer to safety (in pixels), ℓ refers to path length (pixels).

Figure 4 shows IGP outperforming pedestrians in both safety and path length by a fairly large margin. Furthermore, PCLRHC is, as theoretically demonstrated earlier, inappropriate for very dense crowds—PCLRHC almost always takes evasive maneuvers (long path length) in an effort to avoid the crowds (large safety margin).

VI. CONCLUSIONS

In this paper, we studied the Freezing Robot Problem (FRP), a phenomenon where planning algorithms exhibit overcautious or evasive behavior due to anticipated collisions with stochastically moving agents. While most existing techniques for dealing with the FRP focus on more informed (or less uncertain) models, we show that the FRP can occur even with perfect prediction, and that the key to safely navigating through dense crowds is to capture the cooperative collision avoidance inherent in real world behavior. We develop IGP, a nonparametric statistical model based on dependent output Gaussian processes, coupled through a nonlinear interaction potential. We show how navigation in this model is naturally cast as an inference task, which can be approximately solved using importance sampling. Lastly, we demonstrated the efficacy of this algorithm on real world pedestrian data. Our results show that IGP leads to paths which are both safer and shorter than those taken by actual pedestrians and existing state of the art path planning algorithms.

Finally, we mention that work is currently underway on a live experiment in the student cafeteria at Caltech (see http://www.cds.caltech.edu/∼trautman for details). An overhead camera has been mounted in the style of the experiments of this paper, and the plan is to have a small nonholonomic robot robot attempt to navigate through human crowds at lunch time.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of Richard Murray and reviewers' comments. This research was partially supported by Microsoft Corporation through a gift, and by ONR Grant N000140911044.

REFERENCES

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50:174–188, 2002.

[2] M. Bennewitz, W. Burgard, G. Cielniak, and S Thrun. Learning Motion Patterns of People for Compliant Robot Motion. *IJRR*, 24(1):31–48, 2005.

[3] C. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, New York, NY, 2006.

[4] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[5] A. Doucet and Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. Technical report, 2008.

[6] N. Dutoit. *Robotic Motion Planning in Dynamic, Cluttered, Uncertain Environments: the Partially Closed-loop Receding Horizon Control Approach*. PhD thesis, California Institute of Technology, 2009.

[7] D. Helbing, I. Farkas, and T. Viscek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

[8] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, May 1995.

[9] D. Helbing, P. Molnar, I. Farkas, and K. Bolay. Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design*, 2001.

[10] H. Helble and S. Cameron. 3-d path planning and target trajectory prediction for the oxford aerial tracking system. In *ICRA*, pages 1042–1048, 2007.

[11] P. Henry, C. Vollmer, B. Ferris, and D. Fox. Learning to navigate through crowded environments. In *ICRA*, 2010.

[12] S.M. Herman. *A Particle Filtering Approach to Joint Passive Radar Tracking and Target Classification*. PhD thesis, University of Illinois, 2002.

[13] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[14] F. Large, D. Vasquez, T. Fraichard, and C. Laugier. Avoiding cars and pedestrians using velocity obstacles and motion prediction. pages 375–379, 2004.

[15] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[16] M. Luber, G. Tipaldi, and K. Arras. People tracking with human motion predictions from social forces. In *ICRA*, 2010.

[17] T. P. Minka. Expectation propagation for approximate bayesian inference. In *UAI '01*, pages 362–369, San Francisco, CA, USA, 2001.

[18] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: modeling social behavior for multi-target tracking. In *ICCV*, 2009.

[19] S. Pellegrini, A. Ess, and M. Tanaskovic L. van Gool. Wrong turn - no dead end: a stochastic pedestrian motion model. In *International Workshop on Socially Intelligent Surveillance and Monitoring (SISM)*, 2010.

[20] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[21] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston, MA, 2004.

[22] N. Roy, G. Gordon, and S. Thrun. Planning under uncertainty for reliable health care robotics. In *FSR*, 2003.

[23] S. Thompson, T. Horiuchi, and S. Kagami. A probabilistic model of human motion and navigation intent for mobile robot path planning. In *Proceedings of the 4th ICARA*, Wellington, New Zealand, February 2009.

[24] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

[25] J. van den Berg, S. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *ISRR*, 2009.

[26] J. van den Berg, M. Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *ICRA*, 2008.

[27] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, Martial H., A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Proc. IROS*, 2009.