Submitted, 2013 International Conference on Robotics and Automation (ICRA)
http://www.cds.caltech.edu/~murray/papers/tmmk13-icra.html

Robot Navigation in Dense Human Crowds: the Case for Cooperation

Pete Trautman[1], Jeremy Ma[1,2], Richard M. Murray[1] and Andreas Krause[1,3]

***Abstract*— We consider mobile robot navigation in dense human crowds. In particular, we explore two questions. Can we design a navigation algorithm that encourages humans to cooperate with a robot? Would such cooperation improve navigation performance? We address the first question by developing a probabilistic predictive model of cooperative collision avoidance and goal-oriented behavior. Specifically, this model extends the recently introduced interacting Gaussian processes approach to the case of multiple goals and stochastic movement duration. We answer the second question by empirically validating our model in a natural environment (a university cafeteria), and in the process, carry out the first extensive quantitative study of robot navigation in dense human crowds (completing 488 runs). The "multiple goal" interacting Gaussian processes algorithm performs comparably with human teleoperators in crowd densities near 1 person/m^2, while a state of the art noncooperative planner exhibits unsafe behavior more than 3 times as often as our planner. Furthermore, a reactive planner based on the "dynamic window" approach—widely used for robotic tour guide experiments—fails for crowd densities above 0.55 people/m^2. We conclude that a *cooperation* model is critical for safe and efficient robot navigation in dense human crowds.**

I. Introduction

One of the first major deployments of an autonomous robot in an unscripted human environment occurred in the late 1990s at the Deutsches Museum in Bonn, Germany [1]. This RHINO experiment was quickly followed by another robotic tour guide experiment; the robot in the follow-on study, named MINERVA [2], was exhibited at the Smithsonian and at the National Museum of American History in Washington D.C. These studies inspired a wide variety of research in the broad area of robotic navigation in the presence of humans, ranging from additional work with robotic tour guides ([3], [4]), to field trials for interactive robots as social partners ([5], [6]).

Despite the many successes of the RHINO and MINERVA work and the studies they inspired, fundamental questions about navigation in dense crowds remain. In particular, prevailing algorithms ([7], [8]) and opinion ([9]) on navigation in dynamic environments emphasize deterministic and decoupled approaches. Critically, an experimental study of robotic navigation in dense human crowds is unavailable.

In this paper, we focus on these two major deficiencies: a dearth of human-robot cooperative navigation models and the complete absence of a systematic study of robot navigation in dense human crowds. We thus develop a novel cooperative navigation methodology and conduct the first extensive ($n_{\text{runs}} \approx 500$) field trial of robot navigation in the natural human crowds[1] of a university cafeteria (Figure 1). In these experiments, we quantify the degree to which our cooperation model improves navigation performance. We conclude that a cooperation model is required for safe and efficient navigation in crowds.

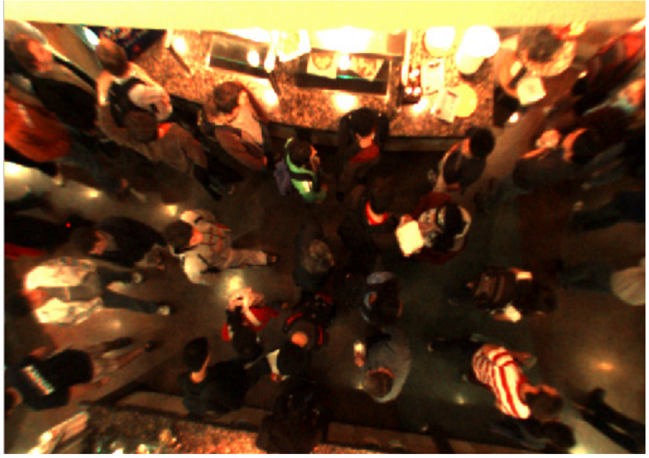


Fig. 1. Overhead still of the crowded university cafeteria testbed. The density of the crowd varies through the day, allowing for diverse experiments.

A. Related Work

Naively modeling the uncertainty in dynamic environments (e.g., with independent agent constant velocity Kalman filters) leads to an uncertainty explosion that makes safe and efficient navigation impossible ([10]). Some research has thus focused on *controlling* predictive uncertainty: in [11] and [12], high fidelity independent human motion models were developed, in the hope that reducing the uncertainty would lead to improved navigation performance. Similarly, [13] holds the individual agent predictive covariance *constant* at a low value as a surrogate for near perfect prediction (in the hope that as the robot gets close to the dynamic agents, prediction and replanning will be good enough for safe navigation to occur).

The work of [14] and [15] shares insight with the approach of [13], although more sophisticated individual dynamic models are developed: motion patterns are modeled as a mixture of Gaussian processes with a Dirichlet process prior over mixture weights. The Dirichlet process prior allows for representation of an unknown number of motion patterns, while the Gaussian process allows for variability within a particular motion pattern. Rapidly exploring random trees (RRTs) are used to find feasible paths. However, all the above

[1] These authors are with the California Institute of Technology.
[2] This author is with NASA/JPL.
[3] This author is with ETH Zurich.

[1] We point out the critical importance of an experiment with *unscripted* human subjects. Simulated humans do not capture the range of human behavior and response. Scripting humans to follow a routine introduces bias that is impossible to quantify.

approaches ignore human-robot interaction. As was argued in [10], unless the dependencies between agents is modeled, navigation will fail in dense crowds.

Interaction between agents has also been addressed. In [16] RRTs are combined with a potential field whose value is based on theories of human proximity relationships (called "proxemics"—see [17]). The authors of [18] take a similar proxemic potential function based approach. In [19], the interaction principles of [20] guide algorithm development. Although these navigation algorithms model a type of human-robot interaction—in particular, how robots should avoid engaging humans—none model human-robot *cooperation*. As we will validate in Section VI, models of cooperation (a special type of human-robot interaction) are required for safe and efficient navigation in dense crowds.

In [21], pedestrian decision making is first learned from a large trajectory example database using inverse reinforcement learning (IRL), and then the robot navigates such that the human's predicted path is minimally disrupted. In [22], the authors extend IRL to work in dynamic environments, and the planner is trained using *simulated* trajectories and tested in simulation. The method successfully recovers the behavior of the simulator.

In [23], a theory of *learning* interactions is developed using game theory and the principle of maximum entropy; only 2 agent simulations are tested. Similarly, the work of [24] leverages IRL to learn an interaction model from human trajectory data. This research pioneers IRL from human data (and explicitly models cooperation), but the experiments are limited in scale—one scripted human crosses paths with a single robot in a laboratory environment.

II. Background

We begin with a high level description of the *interacting Gaussian processes* (IGP) approach to cooperative navigation of [10] and provide details of the method in the subsections. Sections III and IV explain how to extend IGP to include multiple probabilistic goals with uncertain goal arrival times.

The IGP approach is motivated by the following: dynamic navigation algorithms typically assume agent motion to be independent of robot motion. As is shown in [10], this independence assumption leads to highly sub-optimal behavior in dense crowds; coupling agent action and robot action via a joint *trajectory* probability density $p(\mathbf{f}^{(R)}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}, \dots, \mathbf{f}^{(n_t)} \mid \mathbf{z}_{1:t})$ can dramatically improve navigation performance. In this density, t is the present time, $\mathbf{f}^{(i)}$ is a random function representing agent i's trajectory in $\mathbb{R}^2$ from time 1 to T where $T > t$, i ranges over $(R, 1, \dots, n_t)$, R is the robot superscript in $\mathbf{f}^{(R)}$, n_t is the number of agents at time t, and $\mathbf{z}_{1:t}$ are the measurements of all the agents from time 1 to T. The coupling in IGP is achieved with a multiplicative potential function that models cooperative collision avoidance. On the one hand, IGP is a forecast of the crowd's evolution in time. On the other, the density is interpretable as a navigation protocol: choose robot actions according to the *maximum a-posteriori* (MAP) value. This navigation interpretation is an instance of planning reducing to inference, a concept formalized in [25].

However, for each agent, IGP assumes a single, deterministic goal. Furthermore, the goal arrival time is assumed known in advance. Although these assumptions are suitable in simulation, real world agents (such as cafeteria patrons) have multiple probabilistic goals as well as stochastic goal arrival times. In Section III, we address the "single, deterministic goal with known arrival time" assumption with *multiple goal interacting Gaussian processes* (mgIGP). Section IV extends the planning and inference methods used for IGP in [10] to the case of mgIGP.

A. Gaussian Processes for Single Goal Trajectory Modeling

IGP models each agent's trajectory as a random function distributed as a Gaussian Process ([26]), $\mathbf{f}^{(i)} \sim GP(\mathbf{f}^{(i)}; 0, k)$. New measurements $\mathbf{z}_t^{(i)}$ update the GP to $p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}^{(i)}) = GP(\mathbf{f}^{(i)}; m_t^{(i)}, k_t^{(i)})$, where

$$m_t^{(i)}(t') = \Sigma_{1:t,t'}^{\mathrm{T}} (\Sigma_{1:t,1:t} + \sigma^2 I)^{-1} \mathbf{z}_{1:t}^{(i)},$$
$$k_t^{(i)}(t_1, t_2) = k(t_1, t_2) - \Sigma_{1:t,t_1}^{\mathrm{T}} (\Sigma_{1:t,1:t} + \sigma^2 I)^{-1} \Sigma_{1:t,t_2}.$$

Hereby, $\Sigma_{1:t,t'} = [k(1,t'), k(2,t'), \dots, k(t,t')]$, 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 to t. Lastly, σ^2 is the measurement noise (which is assumed to be Gaussian). Since GPs model entire trajectories, goal data is incorporated as a measurement on the final step of the trajectory.

B. Gaussian Process Kernels as Kinematic Models

The kernel function k is the crucial ingredient in GP trajectory models, since it captures "how" dynamic agents move (how smoothly, how linearly, etc.). A class of useful kernel functions are explained in [26]. These individual kernels can be combined to make new kernels via summation and product; we chose a sum of a Matern kernel, a linear kernel, and a noise kernel for the human and the robot. This kernel allowed us to capture nonlinear and linear motion, and sensor noise. We trained the four parameters of the *human* kernel (the "hyperparameters") with track data (Section V-A.3). We trained the *robot* kernel with robot odometric data.

Although the hyperparameters are fixed once they are learned, the actual mean and kernel functions change at each time step; with GPs then, the kinematic model for both the humans and the robot is *adaptive*. That is, online learning of the kinematic models of the agents occurs.

C. Modeling Robot Human Cooperation

IGP couples each individual agent model via an *interaction potential* $\psi(\mathbf{f}^{(R)}, \mathbf{f}) = \psi(\mathbf{f}^{(R)}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)} \dots, \mathbf{f}^{(n_t)})$ that results in a joint model over the $n_t + 1$ agent function space:

$$p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t}) = \frac{1}{Z} \psi(\mathbf{f}^{(R)}, \mathbf{f}) \prod_{i=R}^{n_t} p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}^{(i)}), \quad \text{(II.1)}$$

using the notation $\mathbf{f} = (\mathbf{f}^{(1)}, \dots, \mathbf{f}^{(n_t)})$. We choose ψ to be

$$\psi(\mathbf{f}^{(R)}, \mathbf{f}) = \prod_{i=R}^{n_t} \prod_{j=i+1}^{n_t} \prod_{\tau=t}^{T} \Big(1 - \frac{\alpha}{\exp(\frac{1}{2h^2} |\mathbf{f}^{(i)}(\tau) - \mathbf{f}^{(j)}(\tau)|)}\Big)$$

where $|\mathbf{f}^{(i)}(\tau) - \mathbf{f}^{(j)}(\tau)|$ is the Euclidean distance at time τ between agent i and agent j. The rationale behind our choice is that any set of paths $\mathbf{f}^{(R)}, \ldots, \mathbf{f}^{(n_t)}$ becomes very unlikely if, at time τ, two agents i and j are too close. The parameter h controls the "safety margin", and $\alpha \in [0,1]$ its strength.

D. *Reducing Planning to Inference*

Our model $p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t})$ immediately suggests a natural way to perform navigation: at time t, find the MAP value

$$(\mathbf{f}^{(R)*}, \mathbf{f}^*) = \underset{\mathbf{f}^{(R)}, \mathbf{f}}{\arg\max}\, p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t}),$$

and take $\mathbf{f}^{(R)*}(t{+}1)$ as the next action in the path (where $t{+}1$ means the next step of the estimation). At $t+1$, we receive observations, update the distribution to $p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t+1})$, find the MAP, and choose $\mathbf{f}^{(R)*}(t+2)$ as the next step. This process repeats until the robot arrives at its destination.

III. Multiple Goal Interacting Gaussian Processes

In practice there may be uncertainty between multiple, discrete goals that an agent could pursue; similarly, it is exceedingly rare to know in advance the time it takes to travel between these waypoints. For this reason, we develop a novel probabilistic model over waypoints and the transition time between these waypoints by generalizing the GPs of Section II-B to a mixture of GPs interpolating between waypoints. Figure 2 illustrates our motivation.

We begin with the assumption that the environment in which we will be doing trajectory prediction has a *fixed* number of goals G (corresponding roughly to the number of eating stations in the cafeteria):

$$\mathbf{g} = (\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_G).$$

For the purposes of this analysis, we restrict the distributions governing each goal random variable to be Gaussian. We also restrict our goals $\mathbf{g}_k$ $(k = 1, \ldots, G)$ to lie in the plane $\mathbb{R}^2$.

In order to learn the distribution of the goals $\mathbf{g}$, we gridded the cafeteria floor, collected frequency data on pedestrian linger time within each cell, and then used Gaussian Mixture Model clustering ([27]) to segment the pedestrian track data into "hot spots". In particular, we learned

$$p(\mathbf{g}) = \sum_{k=1}^{G} \beta_k \mathcal{N}\big(\mathbf{g}_k; \boldsymbol{\mu}_{\mathbf{g}_k}, \boldsymbol{\Sigma}_{\mathbf{g}_k}\big).$$

where β_k is the weight of each component, $\boldsymbol{\mu}_{\mathbf{g}_k}$ is the mean of the goal location, and $\boldsymbol{\Sigma}_{\mathbf{g}_k}$ is the uncertainty around the goal. The perimeter ovals in Figure 2 illustrate this idea.

Given $p(\mathbf{g})$, we derive, from experimental data, the transition probability $p(\mathbf{g}_a \to \mathbf{g}_b)$ for all $a, b \in \{1, 2, \ldots, G\}$. For transitions between two goals $\mathbf{g}_a \to \mathbf{g}_b$, we learn $p(T_{a\to b})$, the density governing the duration random variable $T_{a\to b}$.

Finally, we introduce a waypoint sequence $\bar{\mathbf{g}}_m = (\mathbf{g}_{m_1} \to \mathbf{g}_{m_2} \to \cdots \to \mathbf{g}_{m_F})$, composed of waypoints $\mathbf{g}_{m_k}$ with $m_k \in \{1, 2 \ldots, G\}$, for locations indexed by $m_1, m_2, \ldots, m_F$ where $F \in \mathbb{N}$, with associated way point durations $\bar{T}_m = \{T_{m_0 \to m_1}, T_{m_1 \to m_2}, \cdots, T_{m_{F-1} \to m_F}\}$ where $T_{m_0 \to m_1}$ is the time to the first goal.

A. *Generative process for a sequence of waypoints*

We now describe a generative process for a sequence of waypoints that we will use as a prior in our Bayesian model. Beginning with the set of goals $\mathbf{g}$, we draw indices from the set $\{1, 2, \ldots G\}$. The first index is drawn uniformly at random, with the following indices drawn according to $p(\mathbf{g}_a \to \mathbf{g}_b)$. Simultaneously, we draw transition times $T_{a\to b} \sim p(T_{a\to b})$. Thereby, a possibly infinite series of waypoints and transition times is generated.

We formulate agent i's prediction model by marginalizing over waypoint sequences $\bar{\mathbf{g}}_m$ and durations $\bar{T}_m$:

$$p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}^{(i)}) = \sum_{\bar{\mathbf{g}}_m} \left(\int_{\bar{T}_m} p(\mathbf{f}^{(i)}, \bar{\mathbf{g}}_m, \bar{T}_m \mid \mathbf{z}_{1:t}) \right).$$

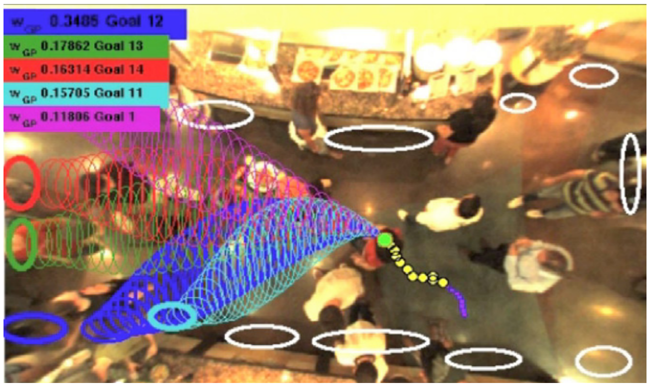


Fig. 2. A patron moves through the cafeteria (solid green circle). Trailing yellow dots are history, and tubes are GP mixture components. GP mixture weights are in the upper left corner. Colored ovals are hot spots.

Using the chain rule, we have

$$p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}^{(i)}) = \\ \sum_{\bar{\mathbf{g}}_m} \int_{\bar{T}_m} p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}, \bar{\mathbf{g}}_m, \bar{T}_m) p(\bar{\mathbf{g}}_m, \bar{T}_m \mid \mathbf{z}_{1:t}). \quad \text{(III.1)}$$

Notice that for each goal sequence $\bar{\mathbf{g}}_m$, we potentially have a different number of waypoints $\mathbf{g}_{m_k}$.

The mgIGP density is Equation II.1 with the mixture models (Equation III.1) substituted for $p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}^{(i)})$.

IV. Cooperative Planning and Inference

We introduce a sampling based inference algorithm for the mgIGP density. We interpret mgIGP as a navigation density, and derive action commands according to Section II-D. We employ two *different* sampling steps to approximate the mgIGP density: a sample based approximation of the mixture process Equation III.1, and a sample based approximation of the mgIGP posterior.

A. *Sample based approximation of mixture models*

Since Equation III.1 is intractable, we employ a sample based approximation:

$$p(\bar{\mathbf{g}}_m, \bar{T}_m \mid \mathbf{z}_{1:t}) \approx \sum_{k=1}^{N_p} w_k^{(i)} \delta\Big[(\bar{\mathbf{g}}_m, \bar{T}_m)_k - (\bar{\mathbf{g}}_m, \bar{T}_m)\Big],$$

where we utilize the empirically derived density $\big(\bar{\mathbf{g}}_m, \bar{T}_m\big)_k \sim p(\bar{\mathbf{g}}_m, \bar{T}_m)$ and N_p samples. Substituting

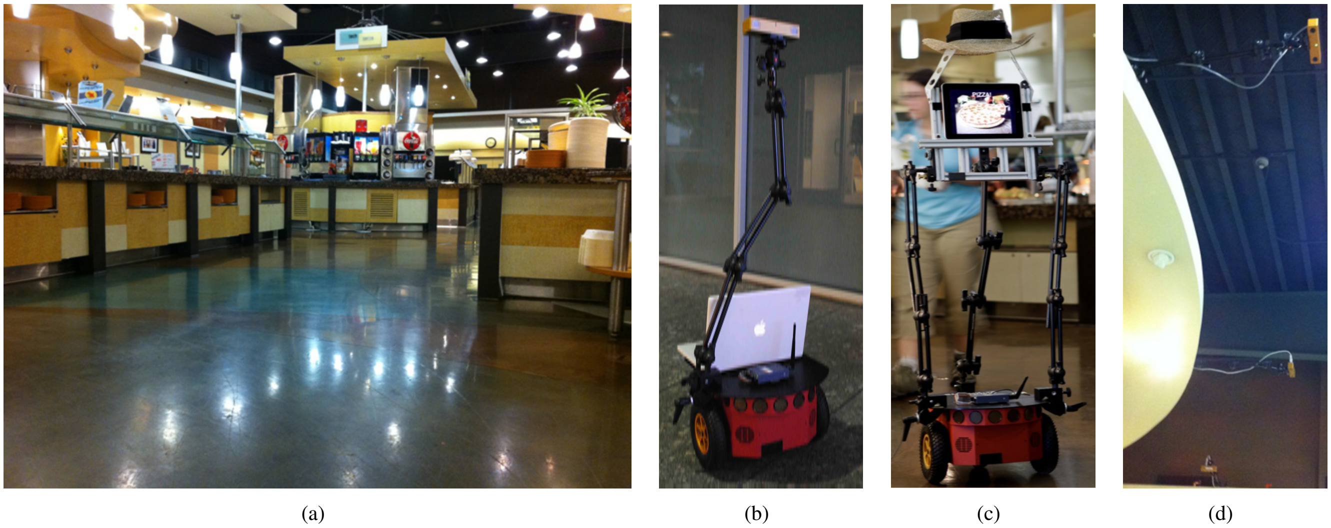


Fig. 3. **(a)** The robot workspace consists of a 20m^2 area surrounded by a buffet (left), a pizza station (right), and a soda fountain (background). Distance between start and goal was 6m. **(b)** Old form factor **(c)** New form factor **(d)** 3 overhead stereo cameras comprising the tracker.

$\sum_{k=1}^{N_p} w_k^{(i)} \delta\Big[(\bar{\mathbf{g}}_m, \bar{T}_m)_k - (\bar{\mathbf{g}}_m, \bar{T}_m)\Big]$ into Equation III.1, we generate

$$p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}^{(i)}) \approx \sum_{k=1}^{N_p} w_k^{(i)} p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}, \bar{\mathbf{g}}_k, \bar{T}_k). \tag{IV.1}$$

The samples collapse the infinite sum of integrals to one finite sum. This illustrated in Figure 2.

In order to generate samples $(\bar{\mathbf{g}}_k, \bar{T}_k)$, we draw a sequence of waypoints $\bar{\mathbf{g}}_k$ and then the corresponding sequence of waypoint durations $T_{k_a \rightarrow k_b}$. To draw the waypoints, we sample $\mathbf{g}_{k_1}$ uniformly from the G goals. We then draw $T_{k_0 \rightarrow k_1}$ according to a distribution with mean given by the average time to travel from the current point to $\mathbf{g}_{k_1}$. Then, $\mathbf{g}_{k_2}$ is drawn according to $p(\mathbf{g}_{k_1} \rightarrow \mathbf{g}_{k_2})$, and $T_{k_1 \rightarrow k_2}$ is consequently sampled. We continue until the sum of the duration waypoints reaches or exceeds T_{max}, and then drop the most recently sampled goal.

Additionally, we evaluate the individual mixture component weights according to

$$w_k^{(i)} = \frac{p\big((\bar{\mathbf{g}}_m, \bar{T}_m)_k \mid \mathbf{z}_{1:t}\big)}{p(\bar{\mathbf{g}}_m, \bar{T}_m)} \propto p\big(\mathbf{z}_{1:t} \mid (\bar{\mathbf{g}}_m, \bar{T}_m)_k\big).$$

That is, we evaluate the likelihood that $\mathbf{z}_{1:t}$ is true conditioned on the waypoint-duration pair $(\bar{\mathbf{g}}_m, \bar{T}_m)_k$. Specifically, $p\big(\mathbf{z}_{1:t} \mid (\bar{\mathbf{g}}_m, \bar{T}_m)_k\big)$ is a GP conditioned on $(\bar{\mathbf{g}}_m, \bar{T}_m)_k$ and the first measurement $\mathbf{z}_1$, and evaluated over $\mathbf{z}_{2:t}$.

B. *Sample based approximation of mgIGP*

We expand the mgIGP density to take goal and waypoint duration uncertainty into account by using Equation IV.1:

$$\begin{aligned} p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t}) &= \frac{1}{Z}\psi(\mathbf{f}) \prod_{i=1}^{N} p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}, \hat{\mathbf{g}}_i) \\ &= \frac{1}{Z}\psi(\mathbf{f}) \prod_{i=1}^{N} \left(\sum_{\bar{\mathbf{g}}_m} \int_{\bar{T}_m} p(\mathbf{f}^{(i)}, \bar{\mathbf{g}}_m, \bar{T}_m \mid \mathbf{z}_{1:t}) \right) \\ &\approx \frac{1}{Z}\psi(\mathbf{f}) \prod_{i=1}^{N} \left(\sum_{k=1}^{N_p} w_k^{(i)} p(\mathbf{f}^{(i)} \mid \mathbf{z}_{1:t}, \bar{\mathbf{g}}_k, \bar{T}_k) \right). \end{aligned}$$

We wish to approximate $p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t})$ using samples. To do this, we extend the method outlined in [10] by adding a step to account for the multiple GP components—that is, to draw the l'th joint sample $(\mathbf{f}^{(R)}, \mathbf{f})_l$ from the mgIGP density we first draw agent i's mixture index from the discrete distribution $\{w_1^{(i)}, w_2^{(i)}, \ldots, w_{N_p}^{(i)}\}$. Given the mixture index γ, we draw $(\mathbf{f}^{(i)})_l \sim p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}, \bar{\mathbf{g}}_\gamma, \bar{T}_\gamma)$. We iterate through all $n_t + 1$ agents (including the robot), and then arrive at the joint sample weight $\eta_l = \psi((\mathbf{f}^{(R)}, \mathbf{f})_l)$. With this collection of N_{mgIGP} weights, we arrive at the approximation

$$p(\mathbf{f}^{(R)}, \mathbf{f} \mid \mathbf{z}_{1:t}) \approx \sum_{l=1}^{N_{mgIGP}} \eta_l \delta[(\mathbf{f}^{(R)}, \mathbf{f})_l - (\mathbf{f}^{(R)}, \mathbf{f})].$$

V. Experiments

In this section, we perform the first comprehensive quantitative study of robot navigation in a crowded environment. In particular, we study the navigation of a Pioneer 3-DX®differential drive mobile robot through dense crowds in a public cafeteria. The purpose of these experiments is to understand how cooperative navigation models affect robot safety and efficiency in human environments. To that end, we tested the following five navigation protocols: a noncooperative planner detailed in Section V-B.1, the single goal IGP algorithm, the mgIGP algorithm, and a reactive planner, based on the Dynamic Window approach of [28], and detailed in Section V-B.4. As an "upper bound" on navigation safety and efficiency, we benchmarked line of sight teleoperation. Sections V-B and V-C, explains how these choices represent nearly *all* existing navigation algorithms.

A. *Experimental setup*

Our experiments were conducted in a university cafeteria (see Figure 3(a)). During typical lunch hours, the number of patrons ranged between five and thirty individuals. The robot's task was to travel through natural, lunchtime crowds from point $A = (0, 0)$ to point $B = (6, 0)$ (in meters). This brought the robot through the center of the scene in Figure

3(a). Cafeteria patrons were unscripted, although doorway signs warned of the presence of filming and a robot.

1) Salient human factors engineering: To build a salient, but not conspicuous, robot we began with a form factor that indicated to human observers that the robot was both sensing and comprehending its environment (see Figure 3(b))—a camera mounted at 3 feet, with a laptop set atop the robot. Unfortunately, this form factor was nearly invisible to cafeteria patrons, especially in crowds of density greater than 0.3 people/m^2. We thus filled out the volume, so that the robot had roughly the shape of a human torso; this was accomplished by mounting 3 camera arms, such that from any angle at least 2 arms were discernible. Additionally, we mounted an 80/20 "head" with a computer tablet "face" at around 4 feet, and adorned the robot's head with a sun hat. Patrons responded positively to this costume (Figure 3(c)).

2) Robotic workspace: Figure 3(a) provides an image of the actual robot workspace used in our experiments. Due to the available coverage of our pedestrian tracker (Figure 4), robot motions were limited to a 20m^2 area between the buffet station, the pizza counter, and the soda fountain.

3) Pedestrian Tracking System: Our pedestrian tracking system utilized three Point Grey Bumblebee2 stereo cameras mounted in an overhead configuration (Figure 3(d)) with overlapping workspace at a nominal height of 3.5m. The Point Grey Censys3D^{TM} software[2] was used to provide accurate tracks of observed pedestrians at an update rate of approximately 20Hz. All identified tracks (up to a maximum of 40) were tagged with an ID and broadcast wirelessly to the Pioneer robot, which used the tracks for navigation.

Figure 4 is a screenshot of the 3D tracker used in our experiments. The bottom pane of the screenshot shows three separate overhead images from each of the stereo camera pairs (only left camera image is displayed). The top pane is our OpenGL GUI displaying all the Censys3D^{TM} tracks in red with magenta circles used to indicate which tracks are currently being reasoned about by the robot. The green path indicates the robot's current planned path. Overlayed underneath all tracks is an image projection of the scene from the stereo cameras to provide scene context.

B. Testing Conditions and Baseline Navigation Algorithms

In our cafeteria experiments, a testing operator was required to stay within a few meters of the robot during every run for emergency stops and for pedestrian safety. The close proximity of the operator to the robot likely influenced the crowd, and probably biased the performance of the robot, *for any given run*. In order to buttress against any algorithm gaining an unfair advantage, every effort was made to reproduce identical testing conditions for each algorithm and for every run. Additionally, we collected as many runs per algorithm as was possible—approximately 3 months of testing, with 488 runs collected, and around 800 attempted.

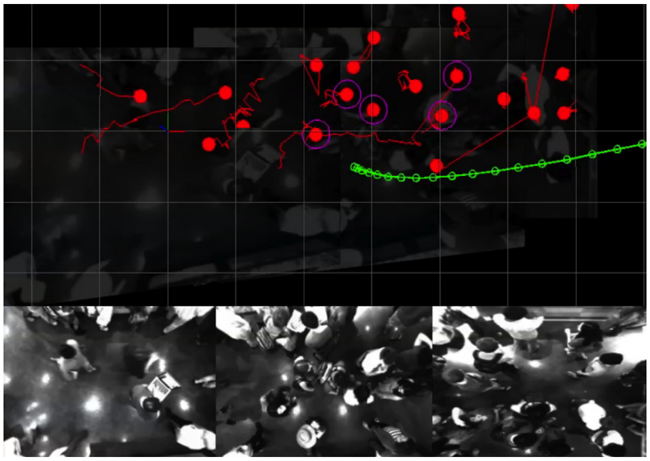


Fig. 4. Robot (wearing sun hat, bottom middle pane) navigating through densities nearing 1 person/m^2. Green dots are robot's present plan, red dots are cafeteria patrons, and magenta circles are "salient" patrons. See Section VII for movies of the robot in action.

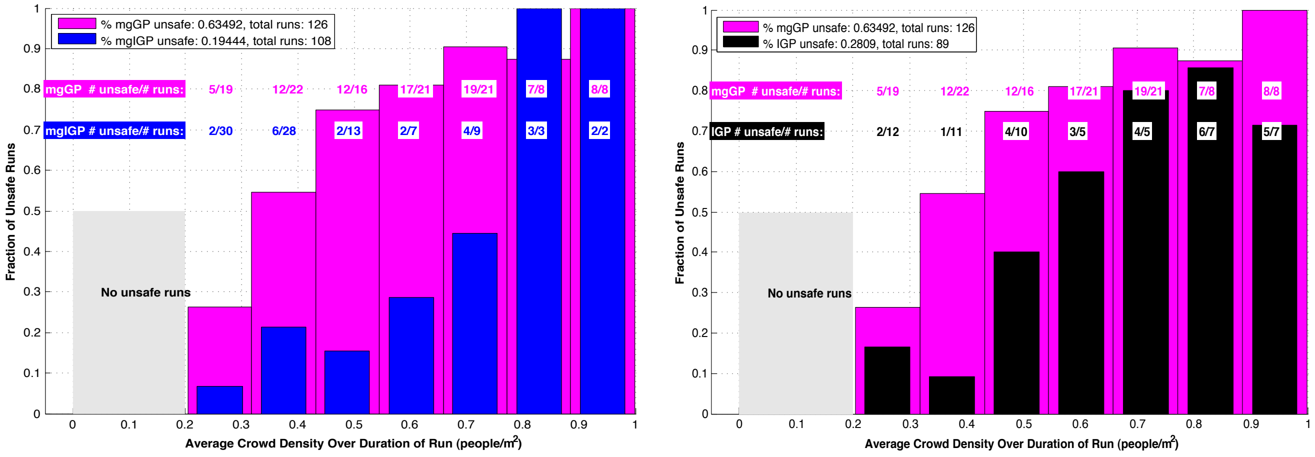
1) Noncooperative Gaussian Processes: Given crowd data from time $t' = 1, \dots, t$, this algorithm predicts individual trajectories using the Gaussian process mixture models. This prediction model is similar to the state of the art crowd prediction models of [29], [30] and [31]. Additionally, our mixture model is nearly identical to the state of the art models used for navigation in [14], [32] and [15]. We also point out that when pedestrian track data indicates linear movement, the Gaussian process mixture model predicts linear movement. Linear prediction models are common to many of the navigation algorithms that we did not test. Our noncooperative planner then uses importance sampling to produce a navigation command at time $t+1$ that approximately minimizes the time to goal while maximizing safety. These two steps are iterated in a receding horizon control manner. We remark that optimizing over the most probable trajectories is similar to the state of the art crowd navigation algorithm of [33].

2) Interacting Gaussian Processes: We often refer to this algorithm as the IGP planner. Implementation details of this algorithm are presented in [10].

3) Multiple Goal Interacting Gaussian Processes: We implement mgIGP as described in Sections IV-A and IV-B.

4) Reactive Navigation: This planner moves forward in a straight line towards the goal, replanning its velocity profile each time step $\Delta t \approx 0.1s$ (since the overhead tracking algorithm runs at about 10Hz, any planner in the cafeteria is limited by this constraint) so that it continues moving at the maximal speed while avoiding collision. This is accomplished in four steps. First, crowd agents are predicted forward in time $0.5s$ using a Gaussian process ($0.5s$ is how long it takes the robot to come to a complete stop from maximum velocity). Next, six potential robot trajectory velocity profiles are computed in the direction of the goal. The velocity profiles range from 0 m/s to 0.3 m/s, discretized in increments of 0.05 m/s. Then, each velocity profile is evaluated for potential collisions using the formula for "important patrons" detailed in the appendix to [34]; those velocity profiles with an unsafe probability of collision are discarded

[2] Censys3D^{TM} is a tracking system that utilizes background subtraction with a plane fit to extract a cumulative set of 3D points belonging to pedestrains from all available stereo cameras. A clustering algorithm segments point cloud data to generate pedestrian *blobs* which are then tracked using a simple motion model with nearest neighbor association.



(a)

(b)

Fig. 5. **(a)** Unsafe runs for the noncooperative planner (called mgGP, in magenta) and mgIGP (in blue). Overall, the noncooperative planner fails more than 3 times as often as the cooperative planner. At extremely high densities (above 0.8 $people/m^2$, when patrons are standing nearly shoulder to shoulder) all planners consistently fail. Anecdotally, it is extremely hard to teleoperate a robot at these densities. **(b)** Unsafe runs for the noncooperative planner and IGP (in black). Even without goal based prediction, the cooperative planner is more than twice as safe as the noncooperative planner.

(we tuned this threshold to be maximally aggressive yet always safe). The safest profile with the highest velocity is chosen. This approach is similar to [28], and is always safe.

5) Human Teleoperation: Human teleoperation was conducted at the discretion of the teleoperator: we allowed the operator to maintain as much line of sight as desired. In all, six teleoperators controlled the robot, for a total of 85 runs. The data produced served as an "upper bound" of dense crowd navigation performance: at all densities, the performance of the human teleoperator exceeded that of the autonomous navigation algorithm.

C. Description of Untested Navigation Algorithms

We survey existing navigation approaches and explain why our test algorithms are sufficiently representative. Inevitable collision states (ICS) are limited to deterministic settings, and so are inapplicable. Probabilistic ICS ([35]) is designed to handle predictive uncertainty. However, Probabilistic ICS is a special case of [14], and so V-B.1 (our noncooperative planner) is representative. Velocity obstacles (VOs) are limited to deterministic scenarios, and thus inappropriate. In [36], VOs are generalized for noise. However, Probabilistic VOs use linear extrapolation, and so V-B.1 is representative. We tested reciprocal velocity obstacles (RVOs, [37]). However, noisy pedestrian tracks caused RVO to behave erratically (unresponsive to a single person walking directly at the robot), and RVO assumes *all* agents choose velocities in a pre-specified manner, which is untrue for humans. This algorithm was thus deemed unsuitable for this application. Potential fields are combined with RRTs to find the minimal cost robot trajectory in [16]. The primary difference between this algorithm and V-B.1 is that our cost field is spherical (rather than ellipsoidal), so V-B.1 is representative.

We point out that the work of [24] and [22] are likely the most compelling alternatives to mgIGP. In particular, [24] uses a joint collision avoidance feature in their inverse reinforcement learning representation, and they learn the weight of that feature from captured human data. However, their experiments involve only a single person and a single robot, and, in their own words, "in more densely populated environments . . . it is not feasible to compute all topological variants". In other words, their current implementation is unsuitable for real time implementation in dense crowds.

VI. Experimental Results: Quantitative Studies

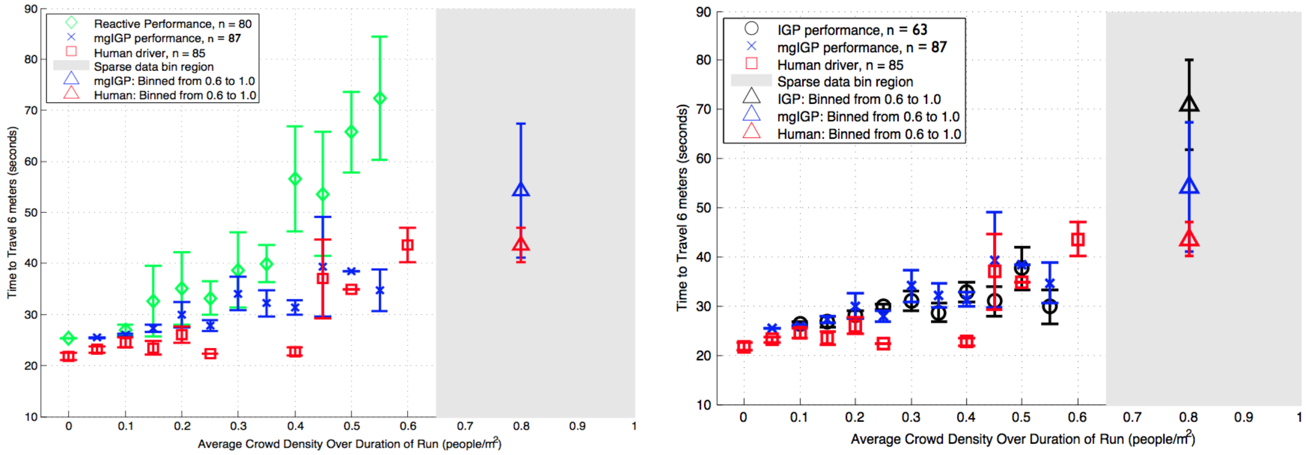
In [39] numerous metrics for evaluating human-robot interaction are presented. Importantly, *safety* is pinpointed as the most important. Accordingly, we evaluate the safety and efficiency of the algorithms of Section V-B.

A. Robot Safety in Dense Human Crowds

We discuss the human density metric. First, we have normalized to values between 0 and 1—thus, the highest density (1 person/m^2) is a *shoulder to shoulder crowd.* Further, patrons rarely stand still; this constant motion increases crowd complexity. Anecdotally, humans found crowd densities above 0.8 people/m^2 to be extremely difficult to teleoperate through, and densities above 0.4 people/m^2 challenging.

We define safety as a binary variable: either the robot was able to navigate through the crowd without collision or it was not. Obviously, we could not allow the robot to collide with either walls or people, and so a protocol for the test operator was put in place: if the robot came within 1 meter of a human, and neither the robot nor the human was making progress towards avoiding collision, then the robot was "emergency stopped" (the velocity command was set to zero). Note that both our reactive planner and human teleoperation were always safe, by design.

1) Noncooperative Planner and mgIGP Planner: In Figure 5(a), we compare the safety of our state of the art noncooperative planner to the mgIGP planner. This data suggests the following: cooperative collision avoidance models can improve overall safety by up to a factor of $0.63/0.19 \approx 3.31$. Additionally, the noncooperative planner is unsafe more than 50% of the time at densities as low as 0.3 people/m^2 and above. At densities of 0.55 people/m^2 and above, it is unsafe



(a)

(b)

Fig. 6. **a** Efficiency of reactive planner, mgIGP, and human teleoperation; mgIGP performs similarly to teleoperation, while reactive planning degrades super linearly. **(b)** Performance of mgIGP, IGP, and human teleoperation. Performance improvement due to goal inclusion is modest.

more than 80% of the time. In contrast, mgIGP is unsafe less than 30% of the time for densities up to 0.65 people/m^2. At densities near 0.8 people/m^2, mgIGP is still safe more than 50% of the time. The noncooperative planner is unsafe over 90% of the time at this high density. Finally, the safety of both planners degrades reliably as crowd density increases (both planners cease to be safe above 0.8 people/m^2).

We present the following explanation: the noncooperative planner believes itself invisible, and so has trouble finding safe paths *through* the crowd, and thus tries to creep along the walls of the testing area. This resulted in many failed runs: the robot's movement is not precise enough to avoid collisions when "wall hugging". More generally, this is a manifestation of the *freezing robot problem* of [10]. In contrast, failures for mgIGP were rare because the robot was more likely to engage the crowd. By engaging the crowd, the robot elicited cooperation, which made navigation safer.

2) Noncooperative Planner and IGP planner: In Figure 5(b), the noncooperative planner is compared to a "compromised" cooperative planner, IGP. The noncooperative planner retains the Gaussian process mixture model.

Although the results are not as stark as in Section VI-A.1, IGP is still $0.63/0.28 \approx 2.25$ times as safe as the noncooperative planner, overall. This result suggests that for navigation in dense crowds, modeling cooperation is more important than high fidelity individual trajectory predictive models.

B. Navigation Efficiency in Dense Human Crowds

Navigation efficiency is defined as the time elapsed from the start of the algorithm until arrival at the goal.

1) mgIGP Planner, Reactive Planner, and Human Tele-Operation: In Figure 6(a), we present the efficiency for the reactive planner, the mgIGP planner, and human tele-operation. This figure demonstrates that, for most crowd densities, mgIGP was nearly as efficient as human tele-operation. We point out that (by definition) the human tele-operators never had to be emergency stopped.

Whereas the efficiency of all the other planners (including human tele-operation) increased roughly linearly with crowd density, the reactive planner grows super linearly with crowd density. Additionally, no runs for the reactive planner were collected for densities above 0.55 people/m^2. This was a result of the following: when the reactive planner started at a density above 0.55 people/m^2, it moved extremely slowly. If the crowd density was any higher than 0.55 people/m^2, it stopped moving forward entirely. Essentially, the reactive algorithm was waiting until the density was low enough to ensure safety. By this time, however, the *average crowd density over the duration of the run* had dropped substantially from the *maximum* crowd density during the run. Thus, the reactive algorithm was unable to make progress through a crowd with an average density above 0.55 people/m^2.

2) mgIGP Planner, IGP Planner, and Human Tele-Operation: In Figure 6(b) we present the efficiency results for the mgIGP planner, the IGP planner, and human tele-operation. This figure provides insight into how efficiency is affected when the Gaussian process mixture model of independent trajectories is removed from the interactive formulation. We note that although the mixture model improves safety (Figure 5), it does not appear to improve efficiency. This data again suggests that modeling *cooperation* between agents is more important than modeling individual agent behavior. Human teleoperation serves as an upper bound on efficiency.

VII. Experimental Results: Qualitative Studies

A highly useful behavior of the robot was that it was always in motion. This was achieved safely by doing the following: if a collision was imminent, the forward velocity was set to zero. However, the rotational velocity was not set to zero. The navigation algorithm continued generating new plans (even though the forward velocity was held at zero until collision was not imminent), and each new plan potentially pointed the robot in a new direction. Indeed, the robot was searching for a way through a challenging crowd state—see the movie snippet at `http://resolver.caltech.edu/CaltechAUTHORS:20120911-130046401`).

Sometimes, this resulted in quite humorous situations: at the beginning of one run, while the navigation algorithm was still starting up, a patron came up and began inspecting the robot. The robot, sensing an imminent collision, set its velocity to zero, and began searching for a clear path (i.e., rotating in place). The patron realized what was happening, and moved along with the robot, constantly staying in front of the robot's forward velocity vector. This resulted in what we have since called the "robot dance"—see the movie snippet at `http://resolver.caltech.edu/CaltechAUTHORS:20120911-125945867`).

This behavior can be quite useful in dense crowds. For instance, the reactive robot did not display this behavior. When a collision was imminent, it stopped completely. Unfortunately, a completely stopped robot is very hard for a human to understand. Is this robot turned off? Is this robot waiting for me? Meanwhile, the mgIGP robot displayed intentionality—see the movie snippet at `http://resolver.caltech.edu/CaltechAUTHORS:20120911-125828298`). Animators call this behavior "readability", and it can be employed to create a more human like intelligence (see [40]).

VIII. Conclusion

We posed two questions: how should human-robot cooperation be modeled? And would such cooperation improve navigation in dense crowds? We answered the first question by introducing mgIGP, and treating that density as a prediction of how the robot *should* act in order to be cooperative. We answered the second question empirically: the mgIGP algorithm was shown to perform comparably with human teleoperators in crowd densities nearing 1 person/m^2, while a state of the art noncooperative planner exhibited unsafe behavior more than 3 times as often as our planner. Also, a state of the art reactive planner was insufficient for crowd densities above 0.55 people/m^2. These experimental results provide the first strong evidence that safe and efficient crowd navigation require a human-robot cooperation model.

References

[1] W. Burgard, A. B. Cremers, and et al., "The interactive museum tour-guide robot," in *AAAI*, 1998.

[2] S. Thrun, M. Beetz, and et al., "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *IJRR*, 2000.

[3] A. Bauer, K. Klasing, and et al., "The autonomous city explorer: Towards natural human-robot interaction in urban environments," *IJSR*, 2009.

[4] K. Hayashi, M. Shiomi, and et al., "Friendly patrolling: A model of natural encounters," in *RSS*, 2011.

[5] L. Saiki, S. Satake, and et al., "How do people walk side-by-side? – using a computational model of human behavior for a social robot," in *HRI*, 2012.

[6] T. Kruse, A. Kirsch, and et al., "Exploiting human cooperation in human-centered robot navigation," in *IEEE International Symposium on Robots and Human Interactive Communications*, 2010.

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

[8] H. Choset, K. Lynch, and et al., *Principles of Robot Motion*. MIT Press, 2005.

[9] T. Fraichard and J. Kuffner, "Guaranteeing motion safety for robots," in *Autonomous Robots*, April 2012.

[10] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense interacting crowds," in *IROS*, 2010.

[11] S. Thompson, T. Horiuchi, and S. Kagami, "A probabilistic model of human motion and navigation intent for mobile robot path planning," in *ICARA*, 2009.

[12] M. Bennewitz, W. Burgard, and et al., "Learning motion patterns of people for compliant robot motion," *IJRR*, 2005.

[13] N. duToit and J. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE-TRO*, 2012.

[14] G. Aoude, B. Luders, and et al., "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," in *Autonomous Robots*, 2011.

[15] J. Joseph, F. Doshi-Velez, and N. Roy, "A bayesian non- parametric approach to modeling mobility patterns," *Autonomous Robots*, 2011.

[16] M. Svenstrup, T. Bak, and J. Andersen, "Trajectory planning for robots in dynamic human environments," in *IROS*, 2010.

[17] E. Hall, "A system for notation of proxemic behavior," *American Anthropologist*, 1963.

[18] N. Pradhan, T. Burg, and S. Birchfield, "Robot crowd navigation using predictive position fields in the potential function framework," in *ACC*, 2011.

[19] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "Understanding human interaction for probabilistic autonomous navigation using risk-rrt approach," in *IROS*, 2011.

[20] C. Lam, C. Chou, and et al., "Human centered robot navigation: Towards a harmonious human-robot coexisting environment," *IEEE-TRO*, 2011.

[21] B. D. Ziebart, N. Ratliff, and et al., "Planning-based prediction for pedestrians," in *IROS*, 2009.

[22] P. Henry, C. Vollmer, and et al., "Learning to navigate through crowded environments," in *ICRA*, 2010.

[23] K. Waugh, B. D. Ziebart, and J. A. D. Bagnell, "Computational rationalization: The inverse equilibrium problem," in *ICML*, 2011.

[24] M. Kuderer, H. Kretzschmar, and et al., "Feature-based prediction of trajectories for socially compliant navigation," *RSS*, 2012.

[25] K. Rawlik, M. Toussaint, and S. Vijayakumar, "On stochastic optimal control and reinforcement learning by approximate inference," *RSS*, 2012.

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

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

[28] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, 1997.

[29] S. Pellegrini, A. Ess, and et al., "You'll never walk alone: modeling social behavior for multi-target tracking," in *ICCV*, 2009.

[30] S. Pellegrini, A. Ess, and M. T. L. van Gool, "Wrong turn - no dead end: a stochastic pedestrian motion model," in *International Workshop on Socially Intelligent Surveillance and Monitoring*, 2010.

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

[32] G. Aoude, J. Joseph, and et al., "Mobile agent trajectory prediction using bayesian nonparametric reachability trees," in *AIAA Infotech*, 2011.

[33] N. duToit, "Robotic motion planning in dynamic, cluttered, uncertain environments: the partially closed-loop receding horizon control approach," Ph.D. dissertation, Caltech, 2009.

[34] P. Trautman, "Robot navigation in dense crowds: Statistical models and experimental studies of human robot cooperation," Ph.D. dissertation, Caltech, 2012.

[35] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable collision states: a probabilistic perspective," *ICRA*, 2010.

[36] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid," in *ICRA*, 2007.

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

[38] B. Neuman and A. Stentz, "Anytime policy planning in large dynamic environments with interactive uncertainty," in *IROS*, 2012.

[39] D. F. Seifer, K. Skinner, and M. Mataric, "Benchmarks for evaluating socially assistive robotics," *Interaction Studies: Psychological Benchmarks of Human-Robot Interaction*, 2007.

[40] L. Takayama, D. Dooley, and W. Ju, "Expressing thought: Improving robot readability with animation principles," in *HRI*, 2011.