Article

The International Journal of
Robotics Research
2015, Vol. 34(3) 335–356

DOI: 10.1177/0278364914557874
ijr.sagepub.com

Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation

Pete Trautman[1], Jeremy Ma[2], Richard M. Murray[3] and Andreas Krause[4]

Abstract

We consider the problem of navigating a mobile robot through dense human crowds. We begin by exploring a fundamental impediment to classical motion planning algorithms called the "freezing robot problem": once the environment surpasses a certain level of dynamic complexity, the planner decides that all forward paths are unsafe, and the robot freezes in place (or performs unnecessary maneuvers) to avoid collisions. We argue that this problem can be avoided if the robot anticipates human cooperation, and accordingly we develop interacting Gaussian processes, a prediction density that captures cooperative collision avoidance, and a "multiple goal" extension that models the goal-driven nature of human decision making. We validate this model with an empirical study of robot navigation in dense human crowds (488 runs), specifically testing how cooperation models effect navigation performance. The multiple goal interacting Gaussian processes algorithm performs comparably with human teleoperators in crowd densities nearing 0.8 humans/m^2, while a state-of-the-art non-cooperative planner exhibits unsafe behavior more than three times as often as the multiple goal extension, and twice as often as the basic interacting Gaussian process approach. Furthermore, a reactive planner based on the widely used dynamic window approach proves insufficient for crowd densities above 0.55 people/m^2. We also show that our non-cooperative planner or our reactive planner capture the salient characteristics of nearly any dynamic navigation algorithm. Based on these experimental results and theoretical observations, we conclude that a cooperation model is critical for safe and efficient robot navigation in dense human crowds.

Keywords

Learning and adaptive systems, cognitive robotics, social human-robot interaction, human-centered and life-like robotics, adaptive control, mechanics, design and control

1. 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 (Burgard et al., 1998). This RHINO experiment was quickly followed by another robotic tour guide experiment; the robot in the follow-on study, named MINERVA (Thrun et al., 2000a), was exhibited at the Smithsonian and at the National Museum of American History in Washington, DC. Both the RHINO and MINERVA robots made extensive use of probabilistic methods for localization and mapping (Dellaert et al., 1999; Roy and Thrun, 1999). In addition, these experiments pioneered the nascent field of human–robot interaction in natural spaces: see Schulte et al. (1999) and Thrun et al. (2000b). Perhaps most importantly, the RHINO and MINERVA 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 (Shiomi et al., 2006, 2009; Hayashi et al., 2011), to work on nursing home robots (Pineau et al., 2003; Montemerlo et al., 2002), to robots that perform household chores (Kruse et al., 2010), to field trials for interacting robots as social partners (Kanda et al., 2004; Saiki et al., 2012), to decorum for robot hosts (Sidner and Lee, 2003), and even to protocols for social robot design (Glas et al., 2011).

Despite the many successes of the pioneering RHINO and MINERVA experiments, and the success of the work

[1]Matrix Research Inc., Dayton, OH, USA
[2]Jet Propulsion Laboratory, Pasadena, CA, USA
[3]California Institute of Technology, Pasadena, CA, USA
[4]ETH Zurich, Zurich, Switzerland

Corresponding author:
Peter F Trautman, 472 Irving Avenue, Dayton, OH 45409, USA.
Email: peter.trautman@gmail.com

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

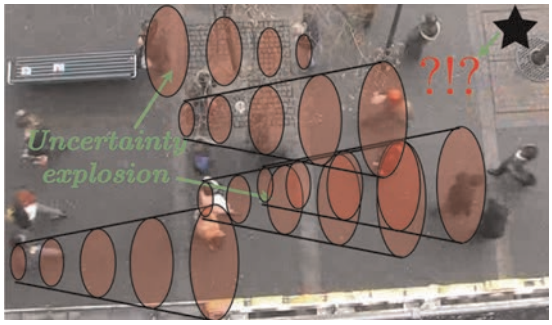

Fig. 2. Freezing robot problem as a result of unconstrained prediction. The robot is the black star, and the ellipses are the predictive covariance of the dynamic agents. The question marks indicate that the robot can find no clear path forward.

that followed it, fundamental questions about robotic navigation in dense human crowds remain unresolved. In particular, prevailing algorithms for navigation in dynamic environments emphasize deterministic and decoupled prediction algorithms (such as in LaValle, 2006, Latombe, 1991 and Choset et al., 2005), and are thus inappropriate for applications in highly uncertain environments or for situations in which the agent and the robot are dependent on one another. Critically, a large-scale experimental study of robotic navigation in dense human crowds is unavailable.

In this paper, we focus on these two deficiencies: a dearth of human–robot cooperative navigation models and the absence of a systematic study of robot navigation in dense human crowds. We develop a cooperative navigation methodology and conduct the first extensive ($n_{\text{runs}} \approx 500$) field trial of robot navigation in natural human crowds. (Figure 1). These experiments quantify the degree to which our cooperation model improves navigation performance. Coupled with the arguments of Section 2, we deduce the importance of a cooperation model for safe and efficient crowd navigation.

1.1. Motivation for cooperative navigation

In Figure 2, we illustrate an instance of the freezing robot problem (FRP). The black star (representing a mobile robot) predicts the individual trajectories (light red ellipses) of a crowd of people. In this case, the lack of any predictive covariance constraints results in a robot that cannot make an informed navigation decision: the deficiencies of the predictive models force the robot to come to a complete stop (or the robot chooses to follow an essentially arbitrary path through the crowd). As we discuss in Section 5.1.1, arbitrary and highly evasive paths can often be much worse than suboptimal: they can be dangerous.

Figure 2 suggests that the culprit behind the FRP could be the individual uncertainty growth. Indeed, if the amount of uncertainty was the primary reason for this suboptimal navigation, then using more precise individual dynamics models would prevent the FRP. As is illustrated in Figure 3, this approach works well for certain crowd configurations.

However, even under perfect individual prediction (i.e. each agent's trajectory is known to the planning algorithm) the FRP still occurs if the crowd adopts specific configurations. In Figures 5a and (b) we illustrate a very common crowd configuration that can cause any independent planner to fail; when people walk shoulder to shoulder, the robot is forced to walk around the crowd, even when the humans are willing to allow passage. In more demanding scenarios, like the cafeteria illustration in Figure 4, this behavior can lead to a failure mode; for instance, the robot in this run collided with the wall in an attempt to make way for the humans.

Given this observation, how is it possible that people can safely navigate through crowds? The key insight is that people typically engage in joint collision avoidance (as argued in the "social forces model" work of Helbing and Molnar (1995) and Helbing et al. (2001, 2000)): they adapt their trajectories to each other to make room for navigation.

Evidence of the usefulness of joint collision avoidance models occurs in other fields as well: work on multi-robot coordination in van den Berg et al. (2008), van den Berg et al. (2009), van den Berg et al. (2011) and Snape et al. (2011) shows that robots programmed to jointly avoid each other are guaranteed to be collision free and display vastly improved efficiency at navigation tasks. In addition, this joint collision avoidance criteria has been used to improve the data association and target tracking of individuals in human crowds (Pellegrini et al., 2009, 2010; Luber et al., 2010).

In the following section, we will show that existing robot navigation approaches commonly ignore mathematical models of cooperation between humans and robots. Unfortunately, under this modeling assumption, the FRP will always occur, given dense enough crowds.

1.2. Related work

Independent agent constant velocity Kalman filters are a starting point for modeling the uncertainty in dynamic environments. Unfortunately, this prediction engine can lead to unconstrained uncertainty growth that makes safe and efficient navigation impossible (Figure 2). Some

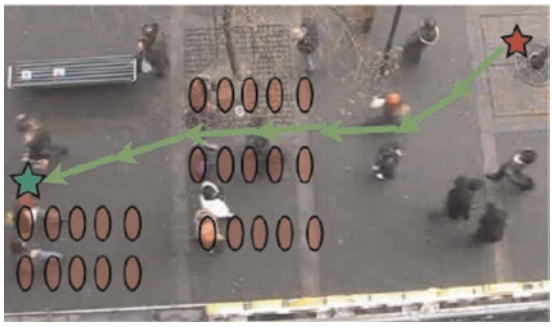

Fig. 3. If the predictive covariance of individual agents is held to a small value, navigation can proceed in an optimal manner, if the crowd is sparse enough.

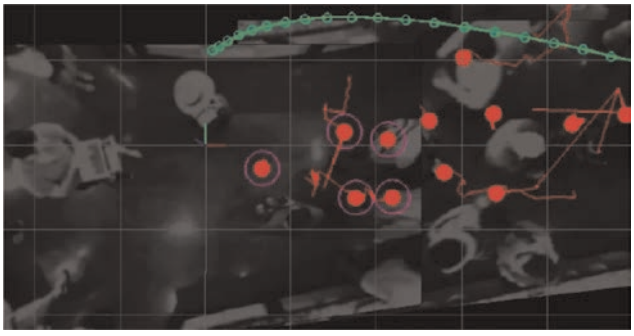

Fig. 4. Example of FRP in cafeteria. The robot was not anticipating interaction, and so chose a highly evasive maneuver (green line). Inspection of human tracks (red lines), in contrast, show people passing between each other. Imagine a crowd of agents unaware of joint interaction; that is, imagine a room full of agents all trying to move along the wall.

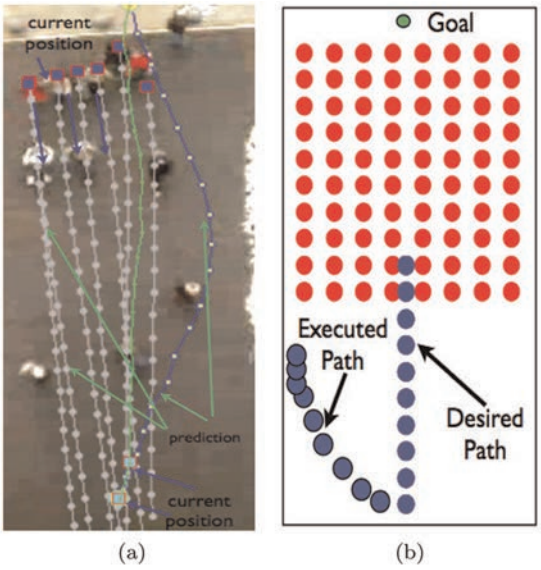

Fig. 5. (a) Even if we hold pedestrian predictive covariance to be extremely small (grey circles), common crowd configurations (shoulder to shoulder walking, sparse crowd) can lead to evasive maneuvering by the robot (b) An illustration of what is occurring in (a). Red dots represent crowd prediction, blue dots represent robot decision making

research has thus focused on limiting this predictive uncertainty. For instance, in Thompson et al. (2009), Bennewitz et al. (2005), Helble and Cameron (2007) and Large et al. (2004), high-fidelity independent human motion models were developed, in the hope that controlling the predictive uncertainty would lead to improved navigation performance. The work of Du Toit and Burdick (2012) and Du Toit (2009) improves navigation performance by directly limiting individual agent predictive uncertainty. Specifically, they formalize robot motion planning in dynamic, uncertain environments as a stochastic dynamic program; intractability is avoided with receding horizon control (RHC) techniques. Furthermore, the collision-checking algorithms developed in earlier work (Du Toit and Burdick, 2011) keeps the navigation protocol safe. The insight is that since replanning is used, the predictive covariance can be held constant at measurement noise. Although robot–agent interaction models are developed for a few cases, the primary contribution from this line of research comes in the form of independent agent dynamics models. Section 2 argues that only limiting the uncertainty growth is insufficient for robot navigation in dense crowds.

The work of Aoude et al. (2011b), Aoude et al. (2011a) and Joseph et al. (2011) shares insight with the approach of Du Toit (2009), although more sophisticated individual models are developed: motion patterns are modeled as a Gaussian process mixture (Rasmussen and Williams, 2006) with a Dirichlet process prior over mixture weights (Teh, 2010). 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 are used to find feasible paths. No work is done on modeling agent interaction.

The field of proxemics (Hall, 1966) has much to say about the interaction between a navigating robot and a human crowd. Specifically, proxemics tries to understand human proximity relationships, and in so doing, can provide insight about the design of social robots. Mead et al. (2011) and Takayama and Pantofaru (2009) developed various robots in accordance with proxemic rules, while in Mead and Matarić (2012) a probabilistic framework for identifying specific proxemic indicators is developed. Similarly, Castro-Gonzalez et al. (2010) study pedestrian crossing behaviors using proxemics. However, this work only studies sparse crowd interactions in scripted settings.

Svenstrup et al. (2010) combined rapidly exploring random trees (RRT) with a potential field based on proxemics. Pradhan et al. (2011) took a similar proxemic potential function-based approach. Although these navigation algorithms model human–robot interaction, they do not model human–robot cooperation. Instead, the emphasis is placed

on respecting a proper distance between the robot and the humans. Further, the algorithm is implemented in simulation only, and the density of humans in the simulated robotic workspace is kept quite low (approximately 0.1 person/m^2).

Rios-Martinez et al. (2011) adopted a "human-centric" approach as well, but instead of using the proxemic rules of Hall (1966), they use the criteria of Lam et al. (2011) instead. They incorporate these rules of personal space into the robot's behavior by extending the Risk-RRT algorithm developed by Fulgenzi et al. (2009). The Risk-RRT algorithm extends the traditional RRT algorithm to include risk, or the probability of collision along any candidate trajectory.

The mobile robot navigation research of Althoff et al. (2012) is more agnostic about the specific cultural considerations of the dynamic agents. A "probabilistic collision cost" is introduced (to assess the fitness of candidate robot trajectories in human crowds) that is based on the idea of inevitable collision states, described by Fraichard and Asama (2003) and expanded by Bautin et al. (2010) (inevitable collision states are robot configurations that are guaranteed to result in a collision with another agent). In particular, Fraichard (2007) advocated three quantities as essential to the proper evaluation of motion safety: the dynamics of the robot, the dynamics of the environment, and a long enough time horizon. Furthermore, Fraichard (2007) argued that full knowledge of these quantities would enable perfect prediction, which in turn would guarantee perfect collision avoidance. The cost function of Althoff et al. (2012) encodes an approximation of these rules. Importantly, collision avoidance capabilities of neighboring dynamic agents are modeled. However, experiments are carried out entirely in simulation.

Importantly, work has been done on learning navigation strategies by observing many example trajectories. Ziebart et al. (2009) used a combination of inverse reinforcement learning (IRL) and the principle of maximum entropy to learn taxi cab driver decision-making protocols from large volumes of data. These methods are extended to the case of a robot navigating through an office environment of Ziebart et al. (2008): pedestrian decision making is first learned from a large trajectory example database, and then the robot navigates in a way that causes the least disruption to the human's anticipated paths. Henry et al. (2010) extend IRL to work in dynamic environments. Their planner is trained using simulated trajectories, and the method recovers a planner which duplicates the behavior of the simulator. In the work of Waugh et al. (2010), agents learn how to act in multi-agent settings using game theory and the principle of maximum entropy. Kuderer et al. (2012) leveraged IRL to learn an interaction model from human trajectory data. Critically, the IRL feature vector is an extension of the cooperation model that was developed by Trautman and Krause (2010); thus, not only does this work model cooperation, it pioneers IRL navigation strategies from real human interaction data as well. However, the experiments are limited in scope: one scripted human crosses paths with a single robot in a laboratory environment.

We mention briefly that (although not developed in the field of robotic navigation) models capturing crowd interaction are explored by Pellegrini et al. (2009), Pellegrini et al. (2010) and Luber et al. (2010) for the purposes of crowd prediction. These papers rely on the social forces model, developed by Helbing and Molnar (1995). The ideas introduced by Helbing and Molnar (1995) underpin the interaction model of Section 3.

We thus suggest that there is a dearth of human–robot cooperative navigation models, and no extensive study of robot navigation in dense human crowds has taken place. In this paper, we address these two deficiencies.

2. The freezing robot problem

In Section 1.1, the FRP was presented as a conceptual motivation for the development of human–robot cooperative navigation protocols. In this section, we provide a more detailed discussion of the FRP, and discuss approaches that can solve the FRP (Section 2.2).

2.1. Mathematical details of the FRP

Consider agent i, where the index i can take values in the set $\{R, 1, 2,\ldots, n\}$, such that $\{1, 2,\ldots, n\}$ are human agents and $i = R$ is a robot. Suppose we have a distribution $p\left(\mathbf{f}^{(i)}_{1:T}\right)$ over each agent's trajectory where

$$\mathbf{f}^{(i)}_{1:T} = \left(\mathbf{f}^{(i)}(1),\ \ldots, \mathbf{f}^{(i)}(T)\right) \tag{2.1}$$

over T timesteps, where each $\mathbf{f}^{(i)}(t) = (x(t), y(t)) \in \mathbb{R}^2$ is the planar location of agent i at time t. We also have a likelihood function $p(\mathbf{z}^{(i)}_t|\mathbf{f}^{(i)}(t))$ for our observations. Since we are dealing with the case of multiple agents, we let

$$\mathbf{z}_{1:t} = (\mathbf{z}^{(1)}_{1:t}, \mathbf{z}^{(2)}_{1:t} \ldots, \mathbf{z}^{(n)}_{1:t}) \tag{2.2}$$

acknowledging that for some times t', we may not observe agent i, in which case $\mathbf{z}^{(i)}_{t'} = \emptyset$.

In the following, we will assume that data association is solved. Note that an observation of agent i is not necessarily independent of the robot's actions. For instance, if the robot's movement influences another agent's movement, then that observation explicitly depends on the robot's actions.

Our goal in dynamic navigation is to pick a policy $\boldsymbol{\pi}$ that adaptively chooses a path $\mathbf{f}^{(R)}_{1:T}$ for the robot based on the observations $\mathbf{z}_{1:t}$ and any ancillary information (such as agent goal location, boundary locations, etc.). The policy $\boldsymbol{\pi}$ is typically specified by stating the next location $\mathbf{f}^{(R)}(t\ +\ 1)$ the robot should choose given all observational and ancillary information.

Thus, for any complete sequence of observations $\mathbf{z}_{1:T}$, the robot can potentially end up choosing a different path

$\mathbf{f}^{(R)}_{1:T} = \pi(\mathbf{z}_{1:T})$. The cost $J(\boldsymbol{\pi}, n)$ of a policy $\boldsymbol{\pi}$ is the expected cost

$$J(\pi, n) = \int p(\mathbf{f}_{1:T}, \mathbf{z}_{1:T}) c(\pi(\mathbf{z}_{1:T}), \mathbf{f}^{(1)}_{1:T}, \ldots, \mathbf{f}^{(n)}_{1:T}) d\mathbf{f}_{1:T} d\mathbf{z}_{1:T} \tag{2.3}$$

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

$$\mathbf{f}_{1:T} = (\mathbf{f}^{(1)}_{1:T}, \ldots, \mathbf{f}^{(n)}_{1:T}) \tag{2.4}$$

Unfortunately, solving for the optimal policy $\boldsymbol{\pi}$ requires solving a continuous-state Markov decision process (MDP), where the dimensionality grows linearly with the number of agents, which is intractable. Intuitively, the intractability is a consequence of attempting exhaustive enumeration; in the above expected cost $J(\boldsymbol{\pi}, n)$, we are attempting to search over the policy space for all possible measurement sequences.

This insolubility is fairly common. In the path planning community, a state-of-the-art, tractable approximation to this MDP is 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. If we let $J(\mathbf{f}^{(R)}_{1:T}, n|\mathbf{z}_{1:t})$ be the objective function that calculates the "cost" of each path $\mathbf{f}^{(R)}_{1:T}$ based on the observations $\mathbf{z}_{1:t}$, that is

$$J(\mathbf{f}^{(R)}_{1:T}, n|\mathbf{z}_{1:t}) = \int c(\mathbf{f}^{(R)}_{1:T}, \mathbf{f}_{1:T}) p(\mathbf{f}_{1:T}|\mathbf{z}_{1:t})\, d\mathbf{f}_{1:T}$$

where $\mathbf{f}^{(R)}_{1:T}$ is the trajectory of the robot, then RHC finds $\mathbf{f}^{(R*)}(t)$, where

$$\mathbf{f}^{(R*)}(t) = \underset{\mathbf{f}^{(R)}_{1:T}}{\operatorname{argmin}} J(\mathbf{f}^{(R)}_{1:T}, n|\mathbf{z}_{1:t}) \tag{2.5}$$

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

Unfortunately, certain assumptions about the distribution $p(\mathbf{f}_{1:T}|\mathbf{z}_{1:t})$ cause the minimum value of the objective function $J(\mathbf{f}^{(R*)}_{1:T}, n|\mathbf{z}_{1:t})$ to increase without bound as the number of agents n increases. This behavior of the objective function is what we call the FRP.

2.2. Approaches for solving the FRP

In order to fix the FRP, nearly all state-of-the-art approaches (see Section 1) focus on individual agent prediction. In particular, Du Toit (2009) anticipates the observations (effectively assuming that a certain measurement sequence of the entire trajectory sequence has already taken place at time $t < T$); the approach is motivated by the assumption that the culprit of the FRP is the uncertainty growth, as illustrated in Figure 2. The claim is that if you can control the covariance, then you can keep the minimum value of $J(\mathbf{f}^{(R)}_{1:T}, n|\mathbf{z}_{1:t})$ low for moderately dense crowds, and thus solve the FRP (other approaches, which incorporate more accurate agent modeling, are similar in motivation, since better dynamic models would reduce predictive covariance as well). However, as discussed above, approaches that work at improving the independent agent prediction or reducing the covariance only solve the FRP for crowd densities below a certain threshold; importantly, they cannot be expected to solve the FRP in general, no matter how favorable the circumstances (even for the case of perfect knowledge of the future).

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

$$p(\mathbf{f}_{1:T} \mid \mathbf{z}_{1:t}) \tag{2.6}$$

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 cost. We additionally remark that in the crowd experiments catalogued in the research of Helbing and Molnar (1995) and Helbing et al. (2001, 2000), the multi-robot coordination theorems of van den Berg et al. (2008, 2009), and the tracking experiments of Pellegrini et al. (2009), Pellegrini et al. (2010) and Luber et al. (2010), 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.

We discuss two ways that human–robot interaction (or human–robot cooperation) may be modeled. One approach to modeling this interaction would be to use a conditional density $p(\mathbf{f}_{1:T}|\mathbf{z}_{1:t}, \mathbf{f}^{(R)}_{1:T})$, 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 assumes that the robot has the ability to fully 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 probably unsuitable for crowded situations.

The other alternative, which we advocate in this paper, is to model the robot as one of the agents, and subsequently model a joint distribution describing their interaction:

$$p(\mathbf{f}^{(R)}_{1:T}, \mathbf{f}_{1:T}|\mathbf{z}_{1:t}) \tag{2.7}$$

This distribution encodes the idea of cooperative planning (e.g. cooperative collision avoidance) by treating robot and agent behaviors as equivalent (unlike the conditional density, where the robot was given priority, or the non-cooperative density $p(\mathbf{f}_{1:T}|\mathbf{z}_{1:t})$, where the agents were given priority).

We point out an important characteristic of this formulation. Although the robot anticipates agent cooperation, the data ultimately takes precedence. Consider the situation where an agent does not cooperate with the robot (perhaps the agent does not see the robot, or perhaps the agent just does not want to cooperate). As the robot approaches this agent, it will predict that the agent will eventually act to cooperatively create space. However, as the robot moves closer to the agent, the evidence (and, thus, the prediction) that the agent is not going to cooperate will outweigh the prior belief that the agent will cooperate. Thus, the robot will compensate, maneuvering around the unyielding agent (we observe this behavior in Sections 5 and 6).

In addition, since the density $p(\mathbf{f}_{1:T}^{(R)}, \mathbf{f}_{1:T}|\mathbf{z}_{1:t})$ models collision avoidance and optimal path lengths (see Section 3), planning corresponds to computing

$$\underset{\mathbf{f}_{1:T}^{(R)}, \mathbf{f}_{1:T}}{\operatorname{argmax}}\, p(\mathbf{f}_{1:T}^{(R)}, \mathbf{f}_{1:T}|\mathbf{z}_{1:t}) \tag{2.8}$$

i.e. inferring with what the robot should do given observations of the other agents (this approach is an example of "reducing planning to inference", that we discuss in Section 3.4). This is an important alternative to the traditional strategy of optimizing an objective function, since we can now interpret navigation as a density estimation problem. We thereby inherit a powerful suite of approximation methodologies that are unavailable for traditional cost-based formulations. In particular, we make use of importance sampling in Section 3.3, and the corresponding convergence results of that method. In Section 7, we briefly discuss how to improve convergence rates using a variant of Gibbs' sampling. The work of Kuderer et al. (2012) explores other novel methods to approximate the navigation density (that would be unavailable if traditional cost-based optimization methods were used).

Finally, we remark that the formulation $p(\mathbf{f}_{1:T}^{(R)}, \mathbf{f}_{1:T}|\mathbf{z}_{1:t})$ does not require complete datasets; that is, $\mathbf{z}_{1:t}$ can range from being globally complete (deterministic access to each agent's state at each time t') to complete data outage. As data reliability decreases, navigation performance will accordingly degrade, but the method does not require perfect information.

3. Interacting Gaussian processes

We begin this section with a description of Gaussian processes, and then we derive *interacting Gaussian processes* (IGP) for crowd prediction. We then describe how approximate inference is performed on the IGP density, followed by a discussion of "the navigation density", or how robotic navigation in dense human crowds can be interpreted as a statistic of the IGP density. We conclude with the results of a simulation experiment.

3.1. Gaussian-process-based prediction

A Gaussian process (Rasmussen and Williams, 2006) is a distribution over (typically smooth) functions, and thus well-suited to model wheeled mobile robot trajectories. Formally, a Gaussian process is a collection of Gaussian random variables indexed by a set (in our case, the continuum of time steps $[1, T]$) that is parameterized uniquely by a mean function

$$m : [1, T] \to \mathbb{R} \tag{3.1}$$

(typically taken as zero without loss of generality) and a covariance (or kernel) function

$$k : [1, T] \times [1, T] \to \mathbb{R} \tag{3.2}$$

We will write

$$\mathbf{f}^{(i)} \sim GP(m^{(i)}, k^{(i)}) \tag{3.3}$$

to mean that the random function $\mathbf{f}^{(i)} : [1, T] \to \mathbb{R}$ is distributed as a Gaussian process with mean $m^{(i)}$ and covariance $k^{(i)}$; since we will be generalizing to the case of multiple dynamic agents $i = 1,\ldots n$, we introduce the superscript notation i to indicate a particular agent i. For clarification, we draw a comparison: with a Gaussian vector $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$, the matrix element $\Sigma_{l,j}$ encodes the covariance between the elements of the state vector $\mathbf{x}_l$ and $\mathbf{x}_j$. Likewise, with Gaussian processes, the kernel function parameterizes the smoothness of the function: recalling that points $t \in [1, T]$ act as our index set, we see that $\mathbf{f}^{(i)}(t)$ and $\mathbf{f}^{(i)}(t')$ are related according to the value of $k^{(i)}(t, t')$.

Notionally, we believe the true trajectory $\mathbf{f}_*^{(i)}$ exists (or will exist, since we have only gathered prior data about this trajectory; see Section 3.1.2). The Gaussian process $GP(m^{(i)}, k^{(i)})$ encodes all our prior knowledge about the function $\mathbf{f}_*^{(i)}$. In contrast, for sequential Bayesian estimation, the prior model is typically derived from first principles (such as the physics of the moving object), and encoded as the distribution $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$. With Gaussian processes, the prior model $GP(m^{(i)}, k^{(i)})$ is learned from training data. The dearth of high-fidelity first principles models of human behavior, combined with the abundance of example human trajectory data, make Gaussian processes especially appealing for our application.

3.1.1. Posterior Gaussian process. For simplicity of notation, we formalize our Gaussian process trajectory model for one-dimensional locations only. Multiple dimensions are easily incorporated by modeling each dimension as a separate Gaussian process.

Suppose that we collect the set of noisy measurements $\mathbf{z}_{1:t}^{(i)} = (\mathbf{z}_1^{(i)}, \ldots, \mathbf{z}_t^{(i)})$ of the trajectory, where

$$\mathbf{z}_{t'}^{(i)} = \mathbf{f}^{(i)}(t') + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_{noise}^2) \tag{3.4}$$

Then we can calculate the posterior Gaussian process $p(\mathbf{f}^{(i)}|\mathbf{z}_{1:t}^{(i)}) = GP(m_t^{(i)}, k_t^{(i)})$, where

$$m_t^{(i)}(t') = \Sigma_{1:t,\,t'}^{\mathrm{T}}(\Sigma_{1:t,\,1:t} + \sigma_{noise}^2\mathbf{I})^{-1}\mathbf{z}_{1:t}^{(i)} \tag{3.5}$$

$$k_t^{(i)}(t_1, t_2) = k^{(i)}(t_1, t_2) - \Sigma_{1:t,\,t_1}^{\mathrm{T}}(\Sigma_{1:t,\,1:t} + \sigma_{noise}^2\mathbf{I})^{-1}\Sigma_{1:t,\,t_2} \tag{3.6}$$

Hereby, $\Sigma_{1:t,t'} = [k^{(i)}(1,t'), \quad k^{(i)}(2,t'),\ldots,k^{(i)}(t,t')]$, and $\Sigma_{1:t,1:t}$ is the matrix such that the $(l,\ j)$ entry is $\Sigma_{l,j} = k^{(i)}(l,\ j)$ and the indices $(l,\ j)$ take values from 1: t. The quantity σ_{noise}^2 is the measurement noise (which is assumed to be Gaussian, and as shown in Section 3.1.3, can be learned from training data). Since the entire trajectory $\mathbf{f}^{(i)} : [1, T] \to \mathbb{R}$ is being modeled, information about the goal of the agent (such as an eating station in a cafeteria) can be treated as a measurement $\mathbf{z}_T^{(i)}$. The information $\mathbf{z}_T^{(i)}$ constrains the predictive uncertainty along the entirety of the trajectory $\mathbf{f}^{(i)}$ (not only at time T).

3.1.2. Training the Gaussian process. The kernel function $k^{(i)}$ is the crucial ingredient in a Gaussian process model, since it encodes "how" the underlying function behaves: in our case, how a dynamic agent moves (e.g. how smoothly, how linearly, length scales of behavior modes, etc.). For a kernel function to be valid it must first be positive semidefinite. That is, for all sets A that take values in the indexing set (for our case, the indexing set is the closed continuum $[1,\ T]$), $\Sigma_{A,A}$ must be positive definite. A class of useful kernel functions are known and are discussed in detail by Rasmussen and Williams (2006). These individual kernels can be combined to make new kernels via summation and multiplication.

However, even with this set of predefined kernel functions and rules for combining them, choices still have to be made. What combination of discrete kernel functions should be used for a particular application? And once we decide on the kernel functions, how should the kernel hyperparameters be chosen?

To answer these questions, we begin by assuming that we are presented with a training set of input–output pairs. For our pedestrian dynamics models, the inputs are the times $t' = 1, 2,\ldots, t$ and the outputs are the trajectory measurements $\mathbf{z}_{1:t}^{(i)}$. Using this training data we can optimize over both specific kernel functions as well as the hyperparameters of those particular kernel functions. In addition, we describe how *a priori* information can be leveraged to inform our choice of kernel function.

3.1.3. Gaussian process kernels as pedestrian dynamics models. We describe our particular choice of kernel function in this section (up to the hyperparameters, which are trained using the methods outlined above). Because of the nature of our application (humans walking through a cafeteria), and the way that we modeled portions of agent trajectories (see Section 3.2.2), we had *a priori* insight about which kernel functions were appropriate. In particular, we chose to model pedestrian dynamics as the summation of a linear kernel (the nominal movement mode of humans between waypoints is linear)

Fig. 6. An example trajectory of a cafeteria patron. The trajectory was hand labeled and segmented; blue dots are part of the nominal trajectory (modeled with the kernel function $k = k_{linear} + k_{matern} + k_{noise}$, as in Section 3.1.3), green dots are goals, and red represents interaction between agents (see Section 3).

$$k_{linear}(t, t') = t \cdot t' + \frac{1}{\gamma_{linear}^2} \tag{3.7}$$

a Matern kernel (it captures mild curving in the trajectory, common to pedestrian dynamics)

$$k_{Matern}(t, t') = s_{Matern} \cdot \left(1 + \frac{\sqrt{5}(t-t')}{\ell_{Matern}} + \frac{5(t-t')^2}{3\ell_{Matern}^2}\right) \exp\left(-\frac{\sqrt{5}(t-t')}{\ell_{Matern}}\right) \tag{3.8}$$

and a noise kernel (to account for sensor measurement noise)

$$k_{noise}(t, t') = \sigma_{noise}^2\delta(t, t') \tag{3.9}$$

where $\delta(t,\ t') = 1$ if $t = t'$ and is zero otherwise. Thus, our final kernel was

$$k^{(i)}(t, t') = s_{Matern} \cdot \left(1 + \frac{\sqrt{5}(t-t')}{\ell_{Matern}} + \frac{5(t-t')^2}{3\ell_{Matern}^2}\right) \cdot \exp\left(-\frac{\sqrt{5}(t-t')}{\ell_{Matern}}\right) + t \cdot t' + \frac{1}{\gamma_{linear}^2} + \sigma_{noise}^2\delta(t, t') \tag{3.10}$$

(Figure 6 presents an actual human trajectory exhibiting each of these behavior modes: we observe linear and curvy motion, and noise in the measurements). Thus, four hyperparameters had to be learned: s_{Matern}, ℓ_{Matern}, γ_{linear} and σ_{noise}. We used the methods detailed in Section 3.1.2 to train these parameters from sample trajectories.

We point out that, in the absence of *a priori* information about what kernel function should be used, the methods of

Section 3.1.2 can be used to compare different candidate kernel functions (e.g. squared exponential versus linear), since the values of the log marginal likelihoods can be compared across different kernel functions. In addition, comparing marginal likelihood values can be used to guard against local minima when optimizing a fixed kernel function: for instance, one might randomly restart the hyperparameter optimization multiple times, and compare the marginal likelihood for the specific hyperparameter values found for each run, and then choose the most likely hyperparameter set.

3.2. Crowd prediction modeling with IGP

In this section we introduce IGP. Although we ultimately interpret this density for robot navigation, IGP is also a crowd prediction model. We begin by deriving individual models of goal-driven human motion using mixtures of Gaussian processes. We then couple these individual models with the interaction potential.

We begin by assuming that the trajectory prediction region 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,…, 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 (Bishop, 2006) to segment the pedestrian track data into "hot spots". In particular, we learned $p(\mathbf{g}) = \sum_{k=1}^{G} \beta_k \mathcal{N}(\mu_{\mathbf{g}_k}, \Sigma_{\mathbf{g}_k})$, where β_k is component weight, $\boldsymbol{\mu}_{\mathbf{g}k}$ is the goal location mean, $\Sigma_{\mathbf{g}k}$ is goal uncertainty, and $\mathcal{N}(\cdot)$ is a normal density. The perimeter ovals in Figure 8 illustrate this idea. We note that the availability of such data before runtime is a restrictive assumption; nevertheless, such situations are ubiquitous enough to merit consideration.

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

Finally, we introduce a waypoint sequence $\overline{\mathbf{g}}_m = (\mathbf{g}_{m_1} \rightarrow \mathbf{g}_{m_2} \rightarrow \cdots \rightarrow \mathbf{g}_{m_F})$, composed of waypoints $\mathbf{g}_{mk}$ with $m_k \in \{1, 2\ldots, G\}$, for locations indexed by m_1, m_2,…, m_F where $F \in \mathbb{N}$, with associated way point durations $\bar{T}_m = \{T_{m_0\rightarrow m_1}, T_{m_1\rightarrow m_2}, \cdots, T_{m_{F-1}\rightarrow m_F}\}$ where $T_{m0\rightarrow m1}$ is the time to the first goal.

3.2.1. Generative process for a sequence of waypoints. We describe a generative process for a waypoint sequence that we will use as a prior. Beginning with $\mathbf{g}$, we draw indices from the set $\{1, 2,\ldots G\}$. The first index is drawn uniformly at random; $p(\mathbf{g}_a \rightarrow \mathbf{g}_b)$ determines the following indices. Simultaneously, we draw transition times from $p(T_{a\rightarrow b})$. An infinite series of waypoints and transition times is generated.

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

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

Using the chain rule, we have

$$p(\mathbf{f}^{(i)}|\mathbf{z}^{(i)}_{1:t}) = \sum_{\overline{\mathbf{g}}_m} \int_{\bar{T}_m} p(\mathbf{f}^{(i)}|\mathbf{z}_{1:t}, \overline{\mathbf{g}}_m, \bar{T}_m) p(\overline{\mathbf{g}}_m, \bar{T}_m|\mathbf{z}_{1:t}) \tag{3.11}$$

Note that for each goal sequence $\overline{\mathbf{g}}_m$, we potentially have a different number of waypoints $\mathbf{g}_{mk}$.

3.2.2. Gaussian process mixtures for modeling multiple goal trajectories. In practice there may be uncertainty between multiple, discrete goals that an agent could pursue (Figure 6); similarly, it is exceedingly rare to know in advance the time it takes to travel between these waypoints. For these reasons, we introduce a novel probabilistic model over waypoints and the transition time between these waypoints. The motion model is then a mixture of Gaussian processes interpolating between these waypoints.

3.2.3. Interacting Gaussian processes. Our key modeling idea is to capture the dynamic interactions by introducing dependencies between the Gaussian processes. We begin with the independent Gaussian process models $p(\mathbf{f}^{(R)}|\mathbf{z}^{(R)}_{1:t})$, $p(\mathbf{f}^{(1)}|\mathbf{z}^{(1)}_{1:t}), \ldots, p(\mathbf{f}^{(n)}|\mathbf{z}^{(n)}_{1:t})$ and couple them by multiplying in an interaction potential $\psi(\mathbf{f}^{(R)}, \mathbf{f})$. Thus,

$$p(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t}) = \frac{1}{Z} \psi(\mathbf{f}^{(R)}, \mathbf{f}) \prod_{i=R}^{n} p(\mathbf{f}^{(i)}|\mathbf{z}^{(i)}_{1:t}) \tag{3.12}$$

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

$$\psi(\mathbf{f}^{(R)}, \mathbf{f}) = \prod_{i=R}^{n} \prod_{j=i+1}^{n} \prod_{\tau=t}^{T} \left(1 - \alpha \exp\left(-\frac{1}{2h^2} |\mathbf{f}^{(i)}(\tau) - \mathbf{f}^{(j)}(\tau)|\right)\right) \tag{3.13}$$

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 specific instantiation of paths $\mathbf{f}^{(R)}_l, \mathbf{f}^{(1)}_l, \mathbf{f}^{(2)}_l, \ldots, \mathbf{f}^{(n)}_l$ 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 (in both simulation (Section 3.5) and in the dining hall experiments (Sections 5 and 6)), while α was chosen to be in the range [0.90.99].

3.2.4. Approximations introduced by IGP. We recall the argument of Section 2.2, in which the model

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

was advocated as the solution to the FRP. We note that

$$\begin{aligned} p(\mathbf{f}^{(R)}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)} | \mathbf{z}_{1:t}) = \\ p(\mathbf{f}^{(R)} | \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)}, \mathbf{z}_{1:t}) p(\mathbf{f}^{(1)} | \mathbf{f}^{(2)}, \ldots, \mathbf{f}^{(n)}, \mathbf{z}_{1:t}) \\ \times p(\mathbf{f}^{(2)} | \mathbf{f}^{(3)}, \ldots, \mathbf{f}^{(n)}, \mathbf{z}_{1:t}) \cdots p(\mathbf{f}^{(n-1)} | \mathbf{f}^{(n)}, \mathbf{z}_{1:t}) \\ \times p(\mathbf{f}^{(n)} | \mathbf{z}_{1:t}) \end{aligned} \tag{3.14}$$

while the IGP distribution models the full joint as

$$\begin{aligned} p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t}) \propto \psi(\mathbf{f}^{(R)}, \mathbf{f}) p(\mathbf{f}^{(R)} | \mathbf{z}_{1:t}^{(R)}) p(\mathbf{f}^{(1)} | \mathbf{z}_{1:t}^{(1)}) \\ \times p(\mathbf{f}^{(n-1)} | \mathbf{z}_{1:t}^{(n-1)}) p(\mathbf{f}^{(n)} | \mathbf{z}_{1:t}^{(n)}) \end{aligned}$$

We point out that while each distribution over $\mathbf{f}^{(i)}$ in Equation (3.14) is a conditional distribution, IGP assumes that we can model each distribution over $\mathbf{f}^{(i)}$ independently of the other agents $\mathbf{f}^{j \neq i}$, and then capture the interaction via $\psi(\mathbf{f}^{(R)}, \mathbf{f})$. This is a strong modeling assumption; however, we emphasize that our intention with IGP is to capture the notion of "cooperative collision avoidance" (described in Section 2.2 and illustrated in Figure 7). Given the extensive experimental results of Sections 5 and 6, and the simulation results of Section 3.5, we believe that this modeling assumption is justified.

3.2.5. Multi-goal interacting Gaussian processes. Importantly, we point out that if we expand the IGP density to take goal and waypoint duration uncertainty into account by using the motion mixture model approximation, then we have *multi-goal interacting Gaussian processes* (mgIGP):

$$\begin{aligned} p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t}) &= \frac{1}{Z} \psi(\mathbf{f}^{(R)}, \mathbf{f}) \prod_{i=1}^{n} p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)}) \\ &= \frac{1}{Z} \psi(\mathbf{f}^{(R)}, \mathbf{f}) \prod_{i=1}^{n} \left(\sum_{\bar{\mathbf{g}}_m} \int_{\bar{T}_m} p(\mathbf{f}^{(i)}, \bar{\mathbf{g}}_m, \bar{T}_m | \mathbf{z}_{1:t}^{(i)}) \right) \end{aligned} \tag{3.15}$$

3.3. Approximate inference for IGP. For Gaussian processes, exact and efficient inference is possible. However, the introduction of the interaction potential makes the posterior $p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t})$ non-Gaussian and thus approximate inference is required. Standard approaches to approximate inference in models derived from Gaussian processes include the Laplace approximation (Bishop, 2006) and expectation propagation (Minka, 2001). 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). With 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.

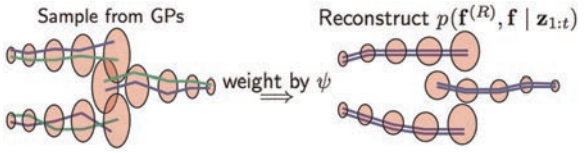

Fig. 7. First samples $\mathbf{f}_l^{(i)}$ are drawn for each agent i. In this illustration, three agents are under consideration; we represent one joint sample of the three agents with three green lines, and another joint sample with three blue lines. The samples are then weighted and combined to produce an estimate of the IGP density.

To cope with the multimodality, we use an approximate inference technique based on importance sampling, a well-understood approximate inference method for Bayesian statistics (Arulampalam et al., 2002).

In this section, we first describe importance sampling for the special case of IGP that have a single known goal for each agent. We then generalize the importance sampling procedure for individual agent models that follow Equation (3.11), with multiple goals and unknown times to goal. That is, we employ two different sampling steps: first we compute (online) a sample-based approximation of each agent's mixture process (Section 3.3.2)

$$p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)}) = \sum_{\bar{\mathbf{g}}_m} \left(\int_{\bar{T}_m} p(\mathbf{f}^{(i)}, \bar{\mathbf{g}}_m, \bar{T}_m | \mathbf{z}_{1:t}^{(i)}) \right) \tag{3.16}$$

and then we compute a sample based approximation of the full mgIGP posterior $p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t})$ in Section 3.3.3.

3.3.1. Sample-based approximation of IGP. We implement importance sampling for approximate inference of the single known goal IGP density as follows.

- For all agents i, sample independent trajectories of agent i from the prior $(\mathbf{f}^{(i)})_l \sim p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)})$. A joint sample is the collection of $n + 1$ such agent samples: $(\mathbf{f}^{(R)}, \mathbf{f})_l$ (see the left-hand side of Figure 7 for an illustration of this idea). Since we are approximating the density $p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t})$, the joint sample is our quantity of interest.
- Evaluate the weight of each sample $(\mathbf{f}^{(R)}, \mathbf{f})_l$ using the rules of importance sampling:

$$\eta_l = \frac{p((\mathbf{f}^{(R)}, \mathbf{f})_l | \mathbf{z}_{1:t})}{\prod_{i=R}^{n} p((\mathbf{f}^{(i)})_l | \mathbf{z}_{1:t}^{(i)})} \tag{3.17}$$

$$= \frac{\psi((\mathbf{f}^{(R)}, \mathbf{f})_l) \prod_{j=R}^{n} p((\mathbf{f}^{(j)})_l | \mathbf{z}_{1:t}^{(j)})}{\prod_{i=R}^{n} p((\mathbf{f}^{(i)})_l | \mathbf{z}_{1:t}^{(i)})} \tag{3.18}$$

$$= \psi((\mathbf{f}^{(R)}, \mathbf{f})_l) \tag{3.19}$$

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

$$p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t}) \approx \frac{1}{\sum_{s=1}^{N} \eta_s} \sum_{l=1}^{N} \eta_l \delta([\mathbf{f}^{(R)}, \mathbf{f}]_l - [\mathbf{f}^{(R)}, \mathbf{f}]) \tag{3.20}$$

where $\delta([\mathbf{f}^{(R)}, \mathbf{f}]_l - [\mathbf{f}^{(R)}, \mathbf{f}])$ is the delta function centered at sample $[\mathbf{f}^{(R)}, \mathbf{f}]_l$. The right-hand side of Figure 7 illustrates this reconstruction.

As we let the number N of samples grow, we approximate $p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t})$ to arbitrary accuracy. Note that all samples are independent of one another. Thus, the technique can be parallelized.

3.3.2. Sample-based approximation of Gaussian process mixtures. Unfortunately, the expansion in Equation (3.11) is intractable, so we employ a sample-based approximation for the distribution over goal waypoints and durations for each agent $p(\overline{\mathbf{g}}_m, \bar{T}_m | \mathbf{z}_{1:t}^{(i)}) \approx \sum_{k=1}^{N_p} w_k^{(i)} \delta([\overline{\mathbf{g}}_m, \bar{T}_m]_k - [\overline{\mathbf{g}}_m, \bar{T}_m])$, where we utilize the empirically derived density $(\overline{\mathbf{g}}_m, \bar{T}_m)_k (\overline{\mathbf{g}}_m, \bar{T}_m)$. Substituting $\sum_{k=1}^{N_p} w_k^{(i)} \delta[(\overline{\mathbf{g}}_m, \bar{T}_m)_k - (\overline{\mathbf{g}}_m, \bar{T}_m)]$ into Equation (3.11), we generate the approximation

$$\begin{aligned} p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)}) &= \sum_{\overline{\mathbf{g}}_m} \int_{\bar{T}_m} p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)}, \overline{\mathbf{g}}_m, \bar{T}_m) \\ &\quad \cdot (\overline{\mathbf{g}}_m, \bar{T}_m | \mathbf{z}_{1:t}^{(i)}) \\ &= \sum_{\overline{\mathbf{g}}_m} \int_{\bar{T}_m} p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)}, \overline{\mathbf{g}}_m, \bar{T}_m) \\ &\quad \cdot \sum_{k=1}^{N_p} w_k^{(i)} \delta([\overline{\mathbf{g}}_m, \bar{T}_m]_k - [\overline{\mathbf{g}}_m, \bar{T}_m]) \\ &\approx \sum_{k=1}^{N_p} w_k^{(i)} p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)}, \overline{\mathbf{g}}_k, \bar{T}_k) \end{aligned}$$

The samples collapse the infinite sum of integrals to one finite sum. This finite component mixture process is illustrated in Figure 8.

In order to generate particles $(\overline{\mathbf{g}}_k, \bar{T}_k)$, we first draw a sequence of waypoints $\overline{\mathbf{g}}_k$ and then the corresponding sequence of waypoint durations $T_{ka \to kb}$. To draw the waypoints, we begin by first sampling $\mathbf{g}_{k1}$ uniformly from the G goals. We then draw $T_{k0 \to k1}$ according to a distribution with mean given by the average time to travel from the current point to $\mathbf{g}_{k1}$. Then, $\mathbf{g}_{k2}$ is drawn according to the transition probabilities $p(\mathbf{g}_{k1} \to \mathbf{g}_b)$, and $T_{k1 \to k2}$ 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. In addition, we evaluate the individual mixture component weights according to

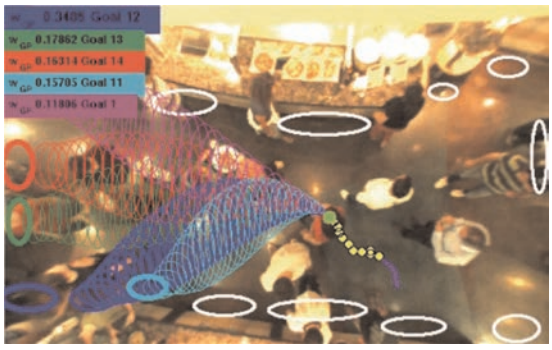

Fig. 8. A patron moves through the cafeteria (solid green circle). Trailing yellow dots are history, and tubes are Gaussian process mixture components. Gaussian process mixture weights are in the upper left corner.

$$w_k^{(i)} = \frac{p((\overline{\mathbf{g}}_m, \bar{T}_m)_k | \mathbf{z}_{1:t}^{(i)})}{p(\overline{\mathbf{g}}_m, \bar{T}_m)} \propto p(\mathbf{z}_{1:t}^{(i)} | (\overline{\mathbf{g}}_m, \bar{T}_m)_k) \tag{3.23}$$

that is, we evaluate the likelihood of the observed data $\mathbf{z}_{1:t}^{(i)}$ given a specific $(\overline{\mathbf{g}}_m, \bar{T}_m)_k$.

3.3.3. Sample-based approximation of mgIGP. We expand the IGP density to take goal and waypoint duration uncertainty into account by using the motion mixture model approximation:

$$p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t}) \propto \psi(\mathbf{f}^{(R)}, \mathbf{f}) \prod_{i=1}^{n} p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)}) \tag{3.24}$$

$$\approx \psi(\mathbf{f}^{(R)}, \mathbf{f}) \prod_{i=1}^{n} \left(\sum_{k=1}^{N_p} w_k^{(i)} p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)}, \overline{\mathbf{g}}_k, \bar{T}_k) \right) \tag{3.25}$$

We wish to approximate $p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t})$ using samples. To do this, we extend the method outlined in Section 3.3.1 by adding a step to account for the Gaussian process mixture components; that is, to draw a joint sample $(\mathbf{f}^{(R)}, \mathbf{f})_l$ from the mgIGP density we first draw agent i's mixture index ζ from the discrete distribution $\{w_1^{(i)}, w_2^{(i)}, \ldots, w_N^{(i)}\}$. Given the mixture index ζ, we draw $(\mathbf{f}^{(i)})_l \sim p(\mathbf{f}^{(i)} | \mathbf{z}_{1:t}^{(i)}, \overline{\mathbf{g}}_\zeta, \bar{T}_\zeta)$. We iterate through all $N + 1$ 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 weights, we arrive at the approximation

$$p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t}) \approx \frac{1}{\sum_{s=1}^{N} \eta_s} \sum_{l=1}^{N} \eta_l \delta([\mathbf{f}^{(R)}, \mathbf{f}]_l - [\mathbf{f}^{(R)}, \mathbf{f}]) \tag{3.26}$$

3.4. Reducing planning to inference

In this section, we explain how the IGP (and mgIGP) density $p(\mathbf{f}^{(R)},\mathbf{f}|\mathbf{z}_{1:t})$ can be interpreted as a "navigation density"; that is, in our model, navigation can be understood as a statistic of prediction. We also explain how a noncooperative planner can be implemented in this manner.

3.4.1. Interacting Gaussian processes for navigation. Our model $p(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t})$ immediately suggests a natural way to perform navigation: at time t, find the *maximum a-posteriori* (MAP) assignment for the posterior

$$(\mathbf{f}^{(R)}, \mathbf{f})^* = \arg\max_{\mathbf{f}^{(R)}, \mathbf{f}} p(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t}) \tag{3.27}$$

and then take $\mathbf{f}^{(R)*}(t+1)$ 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 $p(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t+1})$, find the MAP assignment again and choose $\mathbf{f}^{(R)*}(t+2)$ as the next step in the path. We repeat this process until the robot has arrived at its destination.

We point out that this approach is a special case of the duality between stochastic optimal control and approximate inference discovered by Toussaint (2009) and fully formalized by Rawlik et al. (2012).

3.4.2. Non-cooperative planner. We can also leverage the Gaussian process prediction models to do noncooperative planning. That is, we can plan through a crowd that we do not expect to respond to the robot: the robot merely maximizes the distance between itself and the expected independent trajectories of each pedestrian while minimizing the length of the path to the goal. In Section 5, we test this noncooperative planner in dense human crowds.

A slight modification of the importance sampling technique detailed in Section 3.3.1 allows us to do this: for agent i, instead of drawing samples such that $(\mathbf{f}^{(i)})_l \sim p(\mathbf{f}^{(i)}|\mathbf{z}_{1:t}^{(i)}) = GP(m_t^{(i)}, k_t^{(i)})$, we only draw one sample, the most probable sample, according to $(\mathbf{f}^{(i)})_l = m_t^{(i)}$. For the robot, we continue to draw samples according to $(\mathbf{f}^{(R)})_l \sim p(\mathbf{f}^{(R)}|\mathbf{z}_{1:t}^{(R)}) = GP(m_t^{(R)}, k_t^{(R)})$. Each joint sample $(\mathbf{f}^{(R)}, \mathbf{f})_l$ is then weighted according to the potential function $\psi((\mathbf{f}^{(R)}, \mathbf{f})_l)$, and the sample with the highest weight is chosen as the navigation command for time t. Once we receive new data $\mathbf{z}_{t+1}$, the process is repeated, and the navigation command for time $t+1$ is found. More generally, this procedure is just the sampling-based approximation of

$$(\mathbf{f}^{(R)})^* = \operatorname*{argmax}_{\mathbf{f}^{(R)}} \psi(\mathbf{f}^{(R)}, m_t^{(1)}, \ldots, m_t^{(n)}) p(\mathbf{f}^{(R)}|\mathbf{z}_{1:t}^{(R)}) \tag{3.28}$$

In addition, if we wish to use the Gaussian process mixture models

$$p(\mathbf{f}^{(i)}|\mathbf{z}_{1:t}^{(i)}) \approx \sum_{k=1}^{N_p} w_k^{(i)} p(\mathbf{f}^{(i)}|\mathbf{z}_{1:t}^{(i)}, \bar{\mathbf{g}}_k, \bar{T}_k) \tag{3.29}$$

then we follow the same procedure: for each agent i, we choose the most probable agent trajectory and then sample the robot path with highest potential function value. The most probable trajectory for agent i is the mean of the most likely mixture component. Thus, we find the largest mixture weight $w_k^{(i)}$, and then choose $(\mathbf{f}^{(i)})_l = w_t^{(i,k)}$, where we add the additional superscript k in $m_t^{(i,k)}$ to indicate mixture component k. By optimizing the robot's trajectory against the most probable agent predictions, we produce an algorithm that is highly similar to the planners described by Du Toit (2009), Aoude et al. (2011b,a) and Joseph et al. (2011).

We point out that other non-cooperative planners were available. For instance, we could have minimized the expected probability of collision using Kalman filter prediction or Gaussian process mixture model prediction. However, both of these approaches are the RHC implementation of $\mathbf{f}_t^* = \arg\min_{\mathbf{f}(R)}(\mathbf{f}(R)|\mathbf{z}_{1:t})$. As shown in Section 2, both these approaches have larger objective function costs than the planner of Equation (3.28), meaning that average performance in dense crowds is guaranteed to be inferior. Accordingly, we chose to experiment with the planner of Equation (3.28).

3.5. Simulation experiments

3.5.1. Experimental setup: Crowded pedestrian data. Before we instrumented Caltech's Chandler dining hall, we first evaluated the IGP approach on a dataset of over 8 minutes of video recorded from above a doorway of a university building at ETH Zurich (see Pellegrini et al. (2009) for more details of the video collection process and how to access the data). This dataset exhibits high crowd density, i.e. people frequently pass by one another fairly closely. As an example, see Figure 9 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 number of other people (red arrows) head *into* and *through* the crowd.

We tested the IGP algorithm on variations of just these types of scenarios (one crowd or person intersecting another crowd); our task was to utilize the navigation density in combination with the particle filtering inference method to do navigation through these crowds.

Given the type of data that we experimented with, we now explain our performance metric. For navigation, we are interested in two quantities: *path length* (the Euclidean path distance in $\mathbb{R}^2$ 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 hope to minimize the path length while maximizing the safety margin.

We measure both of 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

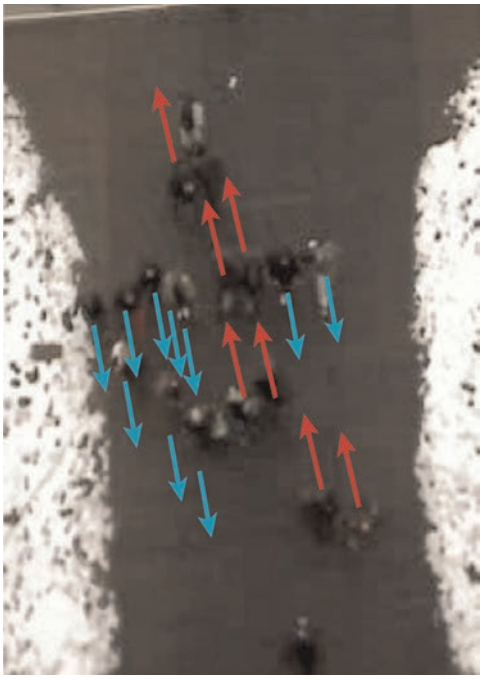

Fig. 9. Crowded still from the ETH data sequence. Near the center of the group is a subgroup of about 6 people moving upwards (red arrows) through a crowd of about 10 people moving down (cyan arrows).

observed pedestrians within 11–12 pixels of one another, although never any closer. Based on this empirical observation of human behavior, we chose any separation distance above 13 pixels to be "safe". Furthermore, we can roughly estimate 13 pixels to be about the width of a person from shoulder to shoulder. Based on this, we chose the value of h in our potential function ψ to be 13 pixels.

As a validation of the methods developed above, we tested against a dataset of human crowds, rather than simulated dynamic agents. In order to test joint collision avoidance, we gave the IGP planner and the non-cooperative planner 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 the IGP or non-cooperative planner would *anticipate* that space. The fact that the IGP took nearly identical paths to the humans and the non-cooperative planner chose highly conservative paths justified, to some extent, our approach. Furthermore, examinations of *planned* paths at early stages in the experiment showed the IGP expecting the opening in the crowd, while the non-cooperative planner expected no such event.

3.5.2 Navigation performance. In Figure 10, we present the results of the various algorithms over 10 experiments. Each box surrounding the colored dots represents the standard error bars over the 10 experiments. The IGP (green dot) had a mean safety of around 22 pixels, with standard

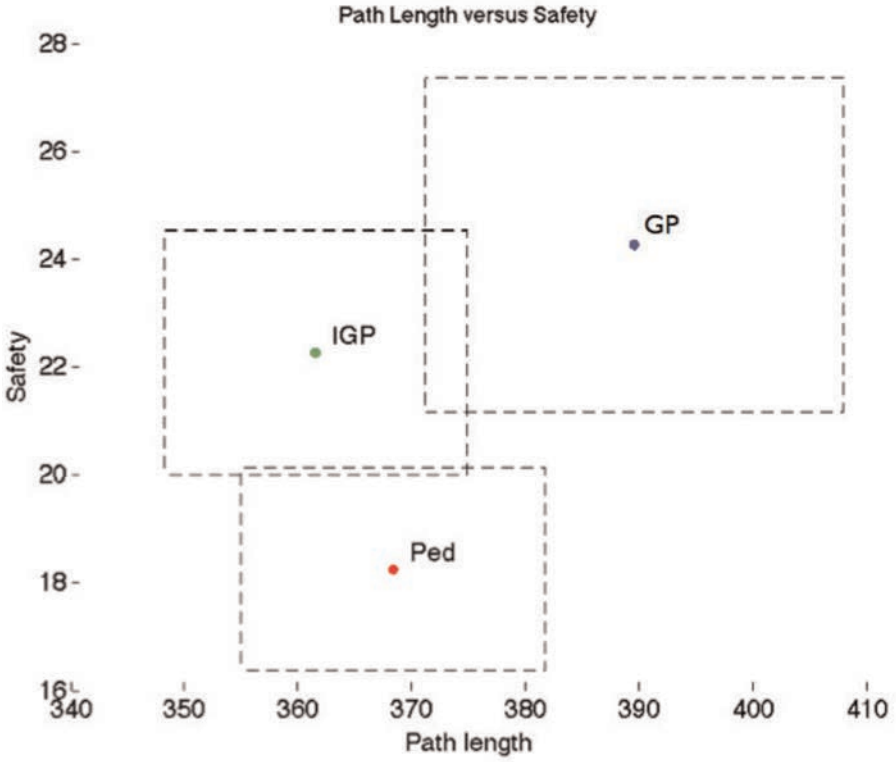

Fig. 10. Path length versus safety over 10 runs. IGP outperforms pedestrians in both safety and path length, while the non-cooperative planner (GP) is inappropriate for this application.

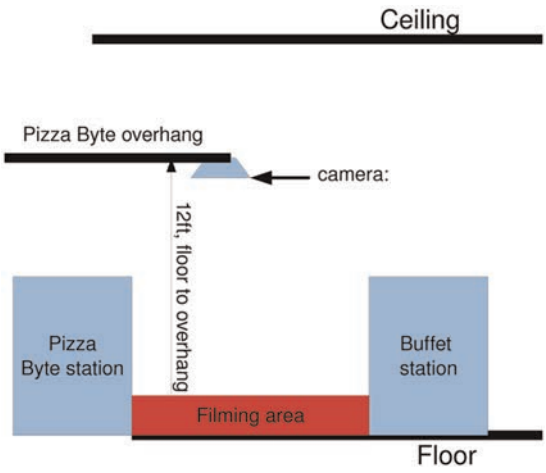

Fig. 11. Side diagram of the observation space.

Fig. 12. Same perspective as Figure 11, but for an actual cafeteria.

error ranging over 2 pixels, and mean path length of around 362, with standard error around 12.

Figure 10 shows the IGP outperforming pedestrians in both safety and path length by a fairly large margin. Furthermore, the non-cooperative planner is, as theoretically demonstrated earlier, inappropriate for very dense crowds: the non-cooperative planner almost always takes evasive maneuvers (long path length) in an effort to avoid the crowds (large safety margin).

True validation of the IGP algorithm demanded "live" interaction, however. That is, in order to test the concept of joint collision avoidance, a robot must actually interact with human beings. This is the motivation for the content of Sections 5 and 6.

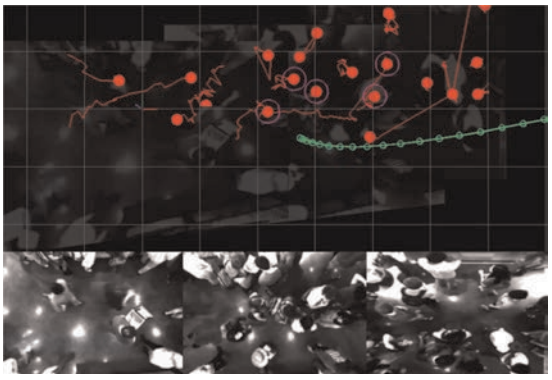

Fig. 13. Robot (wearing sun hat, bottom middle pane) navigating through densities nearing 1 person/m^2. Green dots are robot's present plan, red dots are cafeteria patrons, and magenta circles are "important" patrons (Section 4.5). See Section 6 for movies of the robot in action.

4. Experimental setup

Our experiments were conducted in a university cafeteria. During typical lunch hours, the number of patrons ranged between 5 and 30 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 12. Cafeteria patrons were unscripted, although doorway signs warned of the presence of filming and a robot.

4.1. Robotic workspace

Figure 12 provides an image of the actual robot workspace used in our experiments. Due to the available coverage of our pedestrian tracking system (Figures 13 and 14), robot motions were limited to a 20 m^2 area between the buffet station, the pizza counter, and the soda fountain.

4.2. Pedestrian tracking system

Our pedestrian tracking system used three Point Grey Bumblebee2 stereo cameras mounted 3.5 m overhead (Figure 14b). The Point Grey Censys3D^{TM} software[1] provided pedestrian tracks at an update rate of approximately 20 Hz. However, only the five most "salient" tracks (those most likely to collide with the robot; see Trautman (2013) and Section 4.5) were provided to the navigation algorithm.

Figure 13 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 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. Underneath the tracks is an image projection from the stereo cameras to provide scene context.

(a) (b)

Fig. 14. (a) Stereo camera used by our pedestrian tracker and (b) three stereo cameras, configured to maximize coverage and track quality.

(a) (b)

Fig. 15. (a) Old form factor and (b) new form factor.

4.3. 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 15a): 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 three camera arms, such that from any angle at least two arms were discernible. In addition, we mounted an aluminum fixture "head" with a computer tablet "face" at around 4 feet, and adorned the robot's head with a sun hat. Patrons responded positively.

4.4. 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. In addition, we collected as many runs per algorithm as was possible: approximately 3 months of testing, with 488 runs collected, and around 800 attempted.

We emphasize that although an overhead tracker was used, the data provided to the navigation algorithms was *local* (five most salient tracks; see Section 4.5).

4.5. "Important" cafeteria patrons

In our cafeteria experiments, we computed the five most "important" patrons to perform prediction over. This was done so that the mgIGP planner could operate fast enough (if, in a crowd of 30 people, the planner were to do prediction over each individual, it would replan far too slowly). Performing inference over five people allowed the planner to operate at around 10 Hz, the slowest possible replanning time for safe operation. The five most important patrons were taken to be the five patrons with the highest probability of collision with the robot; following the derivation of Du Toit and Burdick (2011) we first define the *collision condition* between agent i and the robot R to be $\kappa(\mathbf{f}^{(R)}, \mathbf{f}^{(i)}) \neq \emptyset$ where κ measures the overlap (or collision) in $\mathbb{R}^2$ between two agents. The probability of collision is thus

$$P(\kappa) = \int_{\mathbf{f}^{(R)}} \int_{\mathbf{f}^{(i)}} I_\kappa(\mathbf{f}^{(R)}, \mathbf{f}^{(i)}) p(\mathbf{f}^{(R)}, \mathbf{f}^{(i)})\, d\mathbf{f}^{(R)}\, d\mathbf{f}^{(i)}$$

where I_κ is the indicator function for whether or not a collision has occurred between $\mathbf{f}^{(R)}$ and $\mathbf{f}^{(i)}$. However, as derived by Du Toit and Burdick (2011), we can approximate the probability of collision between the robot and an agent as

$$P(\kappa) \approx A_\epsilon \times (2\pi)^{-D/2} |\Sigma_R + \Sigma_i|^{-1/2} \cdot \exp\left(-\frac{1}{2}(\mu_R - \mu_i)^\top (\Sigma_R + \Sigma_i)^{-1} (\mu_R - \mu_i)\right) \quad (4.1)$$

We use Equation (4.1) to determine the patrons most likely to collide with the robot.

4.6. Tested and untested navigation algorithms

We present the implementation details of the navigation algorithms developed in this article: mgIGP and IGP. We also present the implementation details of the baseline algorithms that mgIGP and IGP were tested against.

4.6.1. Interacting Gaussian processes. We often refer to this algorithm as the IGP planner. Implementation details of this algorithm are presented in Section 3.3.1 and in Trautman and Krause (2010) and Trautman et al. (2013). Simulation studies for this algorithm were presented in Section 3.5. As argued in Section 1, IGP is the first algorithm that explicitly models human cooperative collision avoidance for navigation in dense human crowds.

4.6.2. Multi-goal interacting Gaussian processes. Using the goal model $p(\mathbf{g})$, we implement the mgIGP as described in Sections 3.3.2, 3.3.3 and 3.4. This approach augments IGP with a Gaussian process mixture model for individual trajectory prediction.

In dense crowds, new navigation plans must be generated at around 10 Hz. To accomplish this, the navigation algorithm only performed prediction over the five most "important" people: to the robot, the people with whom it was most likely to collide were deemed the most important.

In addition, the mgIGP algorithm only computed the top three Gaussian process mixture components. As shown in Section 3.3.2, the mixture weight can be computed without full knowledge of the mixture component, saving substantial computational resources. The work of Trautman et al. (2013) explains this in detail.

4.6.3. Non-cooperative Gaussian processes. This planner proceeds in the following manner. First, given crowd data from time $t' = 1,\ldots, t$, the 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 Pellegrini et al. (2009), Pellegrini et al. (2010) and Luber et al. (2010). In addition, our mixture model is nearly identical to the state-of-the-art prediction models used for navigation by Aoude et al. (2011b), Aoude et al. (2011a) and Joseph et al. (2011). 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.

Second, our non-cooperative planner uses importance sampling to produce a navigation command at time $t + 1$ that (approximately, importance sampling is still vulnerable to finding local minima) minimizes the time to goal while maximizing safety. These two steps are iterated in a RHC manner. This sampling based approximation procedure is very similar to the rapidly exploring random trees navigation method implemented by Aoude et al. (2011b) and Aoude et al. (2011a). The presence of Gaussian process mixture models in both approaches, and the absence of cooperation modeling in both approaches, suggests a high degree of similarity between the two planning methods. Furthermore, optimizing over the most probable trajectories (rather than over distributions) is similar to the state of the art crowd navigation algorithm of Du Toit (2009).

4.6.4. Reactive navigation. This planner moves forward in a straight line along the x -axis, replanning its velocity profile each time step $\Delta t \approx 0.1s$ (since the overhead tracking algorithm runs at about 10 Hz, 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, the agents in the crowd are predicted forward in time approximately 0.5 s using the Gaussian process that is not conditioned on any goals (0.5 s is about how long it takes the robot to come to a complete stop from maximum velocity). Second, six potential robot trajectory velocity profiles are computed (using Gaussian processes that have been conditioned on the robot's goal) along the x-axis. The velocity profiles range from 0 to 0.3 m/s (0.3 m/s was deemed the maximum safe velocity of the robot in dense crowds), discretized in increments of 0.05 m/s. Third, each velocity profile is evaluated for potential collisions using Equation (4.1); those velocity profiles with a probability of collision above 0.3 are deemed unsafe, while those velocity profiles with a collision probability below 0.3 are considered safe (if no velocity profiles are safe, then the 0 m/s profile is chosen). Fourth, of the safe profiles, the one with the highest velocity is chosen (to maximize efficiency and safety simultaneously). This approach is motivated by the "dynamic window approach" of Fox et al. (1997).

4.6.5. Human teleoperation. Human teleoperation was conducted at the discretion of the teleoperator, so much as was possible: we allowed the operator to maintain as much line of sight as the teleoperator considered necessary (i.e. safety was the priority). Occasionally, this meant that some operators followed the robot (some operators were more confident than others, and some operators were more confident under certain conditions).

In all, 6 operators teleoperated the robot, for a total of 85 runs. The data produced was low variance (as would be expected), and served as an equitable "upper bound" of dense crowd navigation performance: at all densities, the performance of the human teleoperator exceeded that of the autonomous navigation algorithm.

4.6.6. Untested navigation algorithms. Unfortunately, not all dynamic navigation algorithms could be tested.

However, we made every effort to capture the essential characteristics of existing navigation algorithms with the algorithms we did test. In Section 5.4 of Trautman (2013), an overview of untested navigation approaches is provided, along with an explanation for why our test algorithms capture the essential characteristics of those algorithms.

We do call attention to the methods of Kuderer et al. (2012), Kretzschmar et al. (2013) and Kuderer et al. (2013), which leverage the method of maximum entropy (maxEnt) IRL. The advantage to this approach is that one can learn the feature vectors associated with human crowd navigation; for instance, the authors postulate collision avoidance, time to goal, velocity, and acceleration feature vectors, and then train the features using laboratory data on human navigation interactions. While this method is a compelling alternative to mgIGP, it is unclear how the approach can be scaled to the crowd densities and size encountered in the cafeteria experiments detailed in this paper. In particular, it remains unclear how one would capture enough data to adequately satisfy the training needs of maxEnt IRL.

Our approach, although arguably less expressive than approaches based on maxEnt IRL, leveraged prior knowledge to reduce the needed training data (in particular, only two parameters were needed for the interaction function, the Gaussian processes only required individual trajectories for training, and the goals were trained using histograms of stopping points). This allowed us to deploy our method in more dynamically complex environments (we point out that the maxEnt IRL approach has only been applied to crowds of four individuals, in spaces nearly as large as the cafeteria).

5. Experimental results: Quantitative studies

Seifer et al. (2007) presented a lengthy catalogue of metrics for determining the efficacy of a robot interacting with a human. However, the authors point out that the most important metric to consider in human–robot interaction experiments is safety. Accordingly, we first evaluate the safety of the test algorithms of Section 4.6. We follow this safety study with an efficiency study. Although efficiency does not always reflect the nuanced behavior of a probabilistic algorithm interacting with humans, we felt that this study, when considered in combination with the safety study, accurately reflected the salient behaviors of our test algorithms.

We point out how to interpret the crowd density values. The scale of the crowd density can be somewhat misleading since we have normalized to values between 0 and 1, bear in mind that the highest density (1 person/m^2) is a shoulder to shoulder crowd; see Figure 1. Also remember that patrons rarely stand still; this constant motion increases the complexity, confusion, and chaos of the situation. Anecdotally, the human drivers found crowd densities above 0.8 people/m^2 to be extremely difficult to teleoperate the robot through. Densities between 0.4 and 0.8 people/m^2 were challenging, while navigation at densities below 0.4 people/m^2 was reasonable.

5.1. Robot navigational safety in dense human crowds

We define safety to be a binary variable: either the robot was able to navigate through the crowd without collision or it was not. For obvious reasons, however, we could not allow the robot to actually collide with objects (either walls or people), and so a protocol for the human monitor (Pete Trautman) was put in place: if the robot came within 1 meter of an object, and the robot did not appear to be making progress towards avoiding the collision (or, likewise, the human did not appear to be making progress towards avoiding the collision), then the robot was "emergency stopped".[2] In other words, if the human monitor believed that a collision was imminent, then an emergency stop was required.

5.1.1. Non-cooperative planner versus mgIGP planner. We first compare the safety performance of our state-of-the-art non-cooperative planner (recall Section 4.6.3) to that of the mgIGP planner in Figure 16. The data presented in this figure suggests the following: modeling cooperative collision avoidance between the crowd and the robot can improve overall safety by up to a factor of $0.63/0.19 \approx 3.31$. Further inspection of Figure 16 reveals additional interesting structure: the safety performance of both planners degrades reliably as crowd density increases (while at densities above 0.8 people/m^2, both planners essentially cease to be safe).

We point out that the non-cooperative 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 more than 80% of the time. In contrast, the interacting planner is unsafe less than 30% of the time for densities up to 0.65 people/m^2. The interacting planner is still safe more than 50% of the time at densities nearing 0.8 people/m^2, while the noncooperative planner is unsafe over 90% of the time at this high density.

We present the following explanation for the unsafe behavior of the non-cooperative planner. Because the non-cooperative robot believes itself invisible, it has trouble finding safe paths through the crowd, and thus oftentimes tries to creep along the perimeter of the testing area (the testing area is bounded by walls). In our specific testing environment, this resulted in many unsafe runs: the robot's movement is simply not precise enough to avoid collisions when "wall hugging". More generally, this is a manifestation of the FRP, explained in Sections 2 and 3.5, and illustrated in Figure 5. In contrast, the number of unsafe runs for the interacting planner were comparatively small because the robot was more likely to engage the crowd (we point out that, to illustrate fairness of the operator, the interacting planner occasionally came to close to the wall, and had to be emergency stopped). By engaging the crowd,

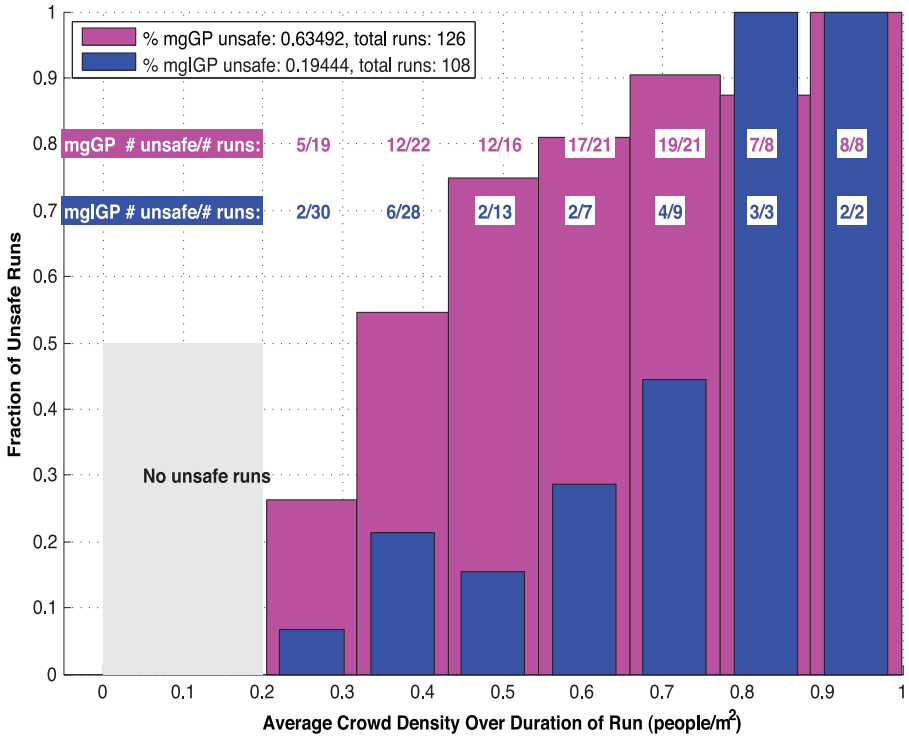

Fig. 16. Unsafe runs for the non-cooperative planner (called mgGP, in magenta) and mgIGP (in blue). Overall, the non-cooperative planner fails more than three times as often. We also point out that at extremely high densities (above 0.8 people/m^2, when patrons are standing nearly shoulder to shoulder) all of the planners consistently fail. Anecdotally, it is extremely hard to teleoperate a robot at these densities.

the robot elicited patron cooperation, which made navigation through the crowd safer. In addition, by navigating in the center of the workspace, the robot was able to stay clear of hard to navigate zones, such as next to walls.

5.1.2. Non-cooperative planner versus IGP planner. In Figure 17, we compare the non-cooperative planner to a "compromised" interacting planner; that is, we remove the Gaussian process mixture model individual trajectory prediction from the interacting planner (leaving it with single goal Gaussian process prediction). The non-cooperative planner retains the Gaussian process mixture model prediction.

Although the results are not as stark as in Section 5.1.1, the IGP is still around $0.63/0.28 \approx 2.25$ times as safe as the non-cooperative planner. This result suggests that for robot navigation in dense crowds, modeling cooperation is more important than high-fidelity individual trajectory predictive models.

5.2. Robot navigational efficiency in dense human crowds

The robot's task for every algorithm and for every run 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 "filming area". Cafeteria patrons were almost entirely unscripted: they were not trained in any way, although they were warned (with signs at every entrance to the Chandler dining hall) that a robot would be present during their lunchtime routine.

5.2.1. mgIGP planner, non-cooperative planner, and human teleoperation. In Figure 18, we present the results of a set of nearly 200 runs in Chandler dining hall during lunch hours. We point out a few things. First, the number of example runs for the noncooperative planner is relatively low ($n = 40$). This is due to the typically unsafe behavior of this planner, as discussed in Section 5.1.1.

Indeed, more runs were attempted for the non-cooperative planner than for any of the other planners, precisely because the completion rate was so low. To wit, 126 runs were attempted for the non-cooperative planner, 89 for the IGP planner, and 108 for the mgIGP planner.

In addition, we point out that when the non-cooperative planner did complete runs, it did so with respectable efficiency. This is easy to understand in light of the discussion of "crowd configurations" of Section 1.1. That is, the non-cooperative planner was able to complete runs primarily when the crowd adopted configurations amenable to efficiency. For instance, if the patrons were standing along the perimeter of the testing space, leaving an opening through the middle, then the correct navigation strategy did not require interaction, and so the non-cooperative planner would produce an efficient run.

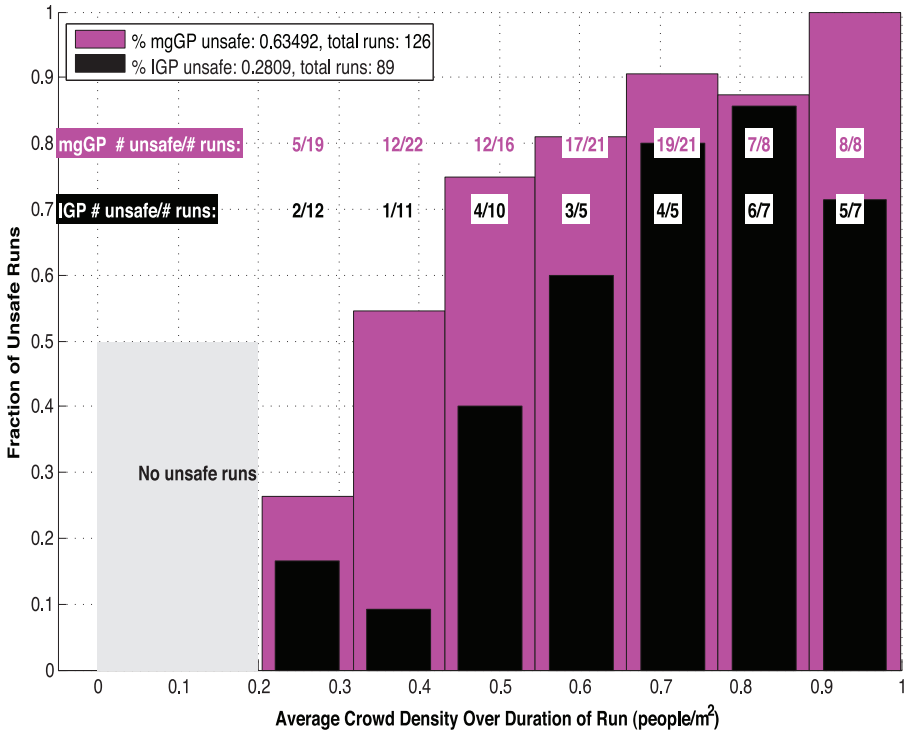

Fig. 17. Unsafe runs for the non-cooperative planner (called mgGP, in magenta) and IGP (in black). Even without goal-based prediction, the interacting planner is more than twice as safe as the non-cooperative planner.

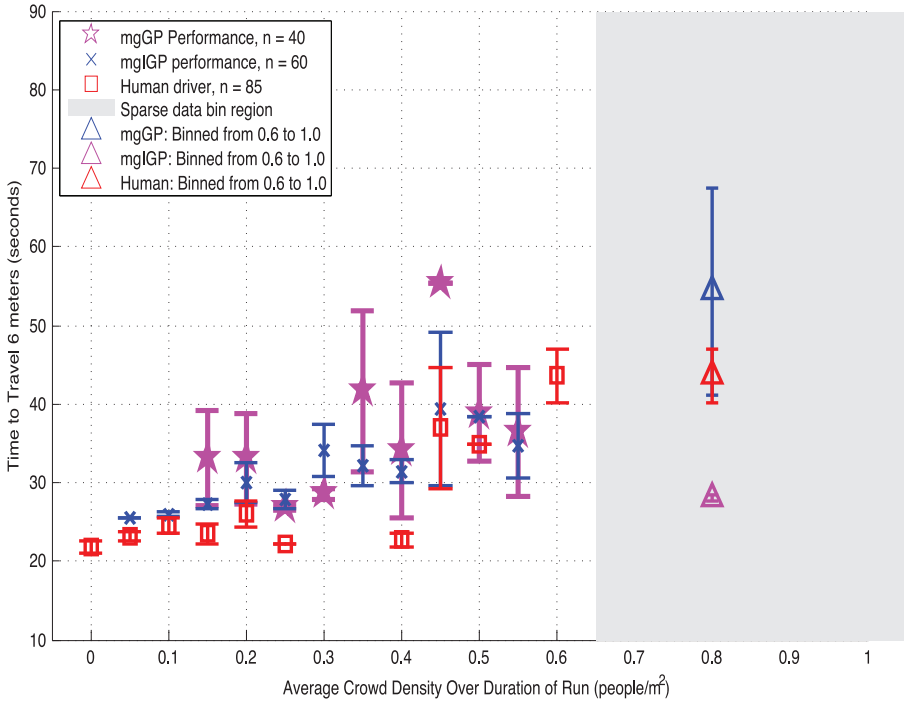

Fig. 18. Efficiency of the non-cooperative planner, mgIGP, and human teleoperation.

5.2.2. mgIGP planner, reactive planner, and human teleoperation. In Figure 19, we present the efficiency for the reactive planner, the mgIGP planner, and human teleoperation. This figure demonstrates that, for most crowd densities, mgIGP was nearly as efficient as human teleoperation. We point out that, by definition, the human teleoperators never had to be emergency stopped: obviously, the safety of the human teleoperators was superior to any of the autonomous algorithms.

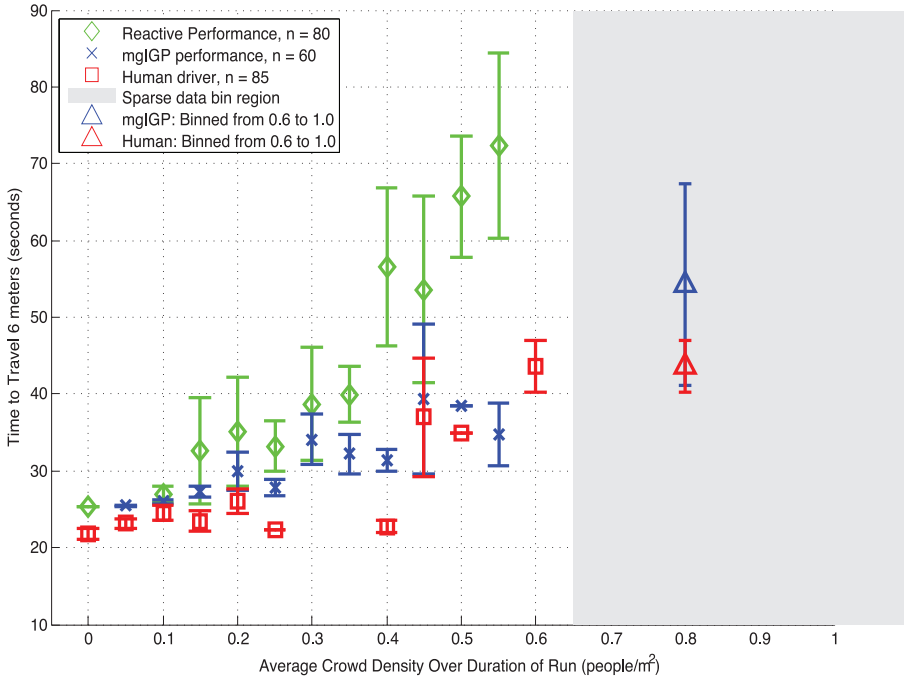

Fig. 19. Efficiency of the reactive planner, mgIGP, and human teleoperation.

The results for the reactive planner are particularly intriguing: whereas for all of the other planners (including human teleoperation) efficiency roughly increased linearly with crowd density, the reactive planner appears to grow nonlinearly with crowd density. In addition, it is important to note that 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 a run at a high density, it moved extremely slowly. Indeed, while the crowd density was above a certain amount, it almost never moved forward: the algorithm was just too cautious. So, essentially, the reactive algorithm waited until the density was low enough, and then it proceeded forward. By this time, however, the average crowd density over the duration of the run had dropped substantially from the maximum crowd density. Effectively, the reactive algorithm was unable to make progress through a crowd with an average density above 0.55 people/m^2.

5.2.3. IGP planner, non-cooperative planner, and human teleoperation. In Figure 20 we present the efficiency results for the IGP planner, the non-cooperative planner, and human teleoperation. This figure provides insight into how "bare" interaction compares with a more sophisticated prediction model. Human teleoperation serves as an upper bound on efficiency.

6. Experimental results: Qualitative studies

In this section, we present qualitative details of the robot's performance. We begin the section by recalling an image frame from a successful run of the mgIGP planner in dense crowds (Figure 13), and follow with a discussion of three movies, each of which illustrate various aspects of the robot's behavior in human crowds.

6.1. Motion for saliency

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 at http://resolver.caltech.edu/CaltechAUTHORS:20120911-130046401).

6.2. Dancing with a robot

Sometimes, this "motion for saliency" resulted in quite humorous situations: at the beginning of one run, while the navigation algorithm was still starting up, a patron approached 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 at http://resolver.caltech.edu/CaltechAUTHORS:20120911-125945867).

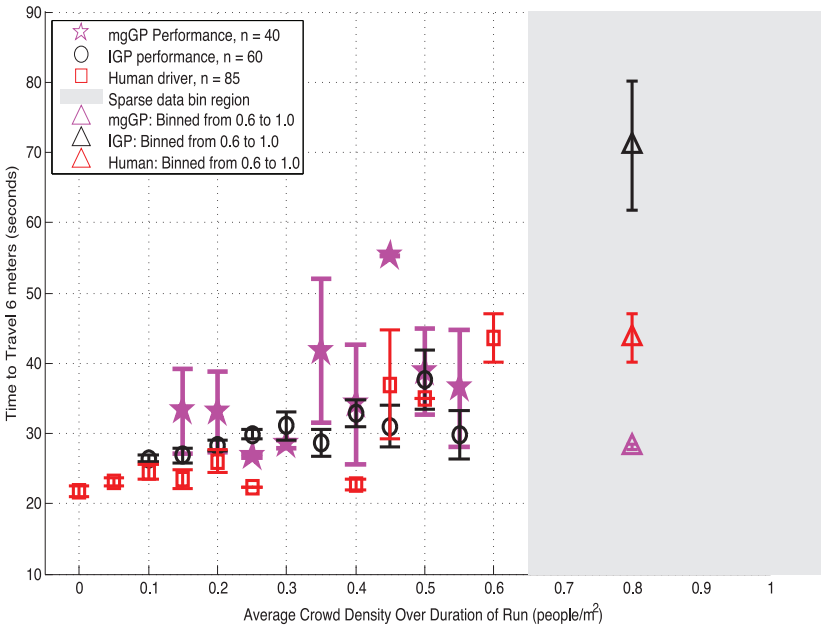

Fig. 20. Efficiency of the non-cooperative planner, IGP, and human teleoperation.

6.3. Motion "readability"

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 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 Takayama et al., 2011).

7. Future work

A practical challenge to robotic navigation in dense crowds is real-time operation. Indeed, the robot needs to replan at around 10 Hz; otherwise, *trajectory following* errors begin to accumulate. We thus suggest that a profitable direction for future research could be on the topic of efficient inference methods for the nonlinearly coupled Gaussian process model IGP (Kuderer et al. (2013), Kuderer et al. (2012) and Kretzschmar et al. (2013) explore alternative inference methods for a similar distribution). In particular, preliminary experiments that use Gibbs sampling with a modified Metropolis–Hastings step have shown promise. This approximate inference method biases the samples towards high probability regions of the distribution, possibly achieving a more efficient sampling procedure.

8. Conclusion

In this paper, we considered mobile robot navigation in dense human crowds. In particular, we explored two questions. Can we design a navigation algorithm that encourages humans to cooperate with a robot? And would such cooperation improve navigation performance? We addressed the first question by developing a probabilistic predictive model of cooperative collision avoidance that we called IGP; we then extended IGP to include multiple goals and stochastic movement duration, which we called mgIGP. We answered the second question by conducting an extensive quantitative study of robot navigation in dense human crowds (488 runs completed), specifically testing how cooperation models effect navigation performance. We found that the mgIGP algorithm performed comparably with human teleoperators in crowd densities near 0.8 humans/m^2, while a state-of-the-art non-cooperative planner exhibited unsafe behavior more than three times as often as this multiple goal extension, and more than twice as often as the basic IGP. Furthermore, a reactive planner based on the widely used "dynamic window" approach failed for crowd densities above 0.55 people/m^2.

Funding

This work was supported in part by the Boeing company through a grant to the California Institute of Technology.

Notes

1. Censys3D^{TM} uses background subtraction to extract a 3D point cloud of pedestrians. A clustering algorithm generates pedestrian *blobs* that are then tracked using a simple motion model with nearest neighbor association.
2. By emergency stop, we mean that the navigation algorithm was terminated. By default, the action command immediately following termination of the navigation algorithm is the zero velocity command. Since the robot's maximum velocity is 0.3 m/s, the robot is thus halted almost instantaneously.

References

Althoff D, Kuffner JJ, Wollherr D and Buss M (2012) Safety assessment of trajectories for navigation in uncertain and dynamic environments. *Autonomous Robots* 32(3): 285–302.

Aoude G, Joseph J, Roy N and How J (2011a) Mobile agent trajectory prediction using Bayesian nonparametric reachability trees. In: *American institute of aeronautics and astronautics infotech at aerospace conference*.

Aoude G, Luders B, Joseph J, Roy N and How J (2011b) Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Autonomous Robots* 35(1): 51–76.

Arulampalam S, Maskell S, Gordon N and Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 50: 174–188.

Bautin A, Martinez-Gomez L and Fraichard T (2010) Inevitable collision states: A probabilistic perspective. In: *IEEE international conference on robotics and automation*.

Bennewitz M, Burgard W, Cielniak G and Thrun S (2005) Learning motion patterns of people for compliant robot motion. *The International Journal of Robotics Research* 24: 31–48.

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

Burgard W, Cremers AB, Fox D, et al. (1998) The interactive museum tour-guide robot. In: *Association for the Advancement of Artificial Intelligence*.

Castro-Gonzalez A, Shiomi M, Kanda T, Salichs M, Ishiguro H and Hagita N (2010) Position prediction in crossing behaviors. In: *IEEE international conference on intelligent robots and systems*.

Choset H, Lynch K, Hutchinson S, et al. (2005) *Principles of Robot Motion*. Cambridge, MA: MIT Press.

Dellaert F, Fox D, Burgard W and Thrun S (1999) Monte carlo localization for mobile robots. In: *IEEE international conference on robotics and automation*.

Du Toit N (2009) *Robotic Motion Planning in Dynamic, Cluttered, Uncertain Environments: the Partially Closed-loop Receding Horizon Control Approach*. PhD Thesis, California Institute of Technology.

Du Toit N and Burdick J (2011) Probabilistic collision checking with chance constraints. *IEEE Transactions on Robotics* 27(4): 809–815.

Du Toit N and Burdick J (2012) Robot motion planning in dynamic, uncertain environments. *IEEE Transactions on Robotics* 28(1): 101–115.

Fox D, Burgard W and Thrun S (1997) The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine* 4(1): 23–33.

Fraichard T (2007) A short paper about motion safety. In: *IEEE International Conference on Robotics and Automation*.

Fraichard T and Asama H (2003) Inevitable collision states: a step towards safer robots? In: *IEEE International Conference on Intelligent Robots and Systems*.

Fulgenzi C, Spalanzani A and Laugier C (2009) Probabilistic motion planning among moving obstacles following typical motion patterns. In: *IEEE International Conference on Intelligent Robots and Systems*.

Glas D, Satake S, Kanda T and Hagita N (2011) An interaction design framework for social robots. In: *Robotics: Science and Systems*.

Hall E (1966) *The Hidden Dimension*. New York: Doubleday.

Hayashi K, Shiomi M, Kanda T and Hagita N (2011) Friendly patrolling: A model of natural encounters. In: *Robotics: Science and Systems*.

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

Helbing D and Molnar P (1995) Social force model for pedestrian dynamics. *Physical Review E* 51: 4282.

Helbing D, Molnar P, Farkas I and Bolay K (2001) Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design* 28(3): 361–383.

Helble H and Cameron S (2007) 3-D path planning and target trajectory prediction for the oxford aerial tracking system. In: *IEEE international conference on robotics and automation*.

Henry P, Vollmer C, Ferris B and Fox D (2010) Learning to navigate through crowded environments. In: *IEEE international conference on robotics and automation*.

Joseph J, Doshi-Velez F and Roy N (2011) A Bayesian nonparametric approach to modeling mobility patterns. *Autonomous Robots* 31(4): 383–400.

Kanda T, Hirano T, Eaton D and Ishiguro H (2004) Interactive robots as social partners and peer tutors. In: *International conference on human–computer interaction*.

Kretzschmar H, Kuderer M and Burgard W (2013) Inferring navigation policies for mobile robots from demonstrations. In: *Proceedings of the autonomous learning workshop at the IEEE international conference on robotics and automation*.

Kruse T, Kirsch A, Sisbot E and Alami R (2010) Dynamic generation and execution of human aware navigation plans. In: *International conference on autonomous agents and multiagent systems*.

Kuderer M, Kretzschmar H and Burgard W (2013) Teaching mobile robots to cooperatively navigate in populated environments. In: *IEEE international conference on intelligent robots and systems*.

Kuderer M, Kretzschmar H, Sprunk C and Burgard W (2012) Feature-based prediction of trajectories for socially compliant navigation. In: *Robotics: science and systems*.

Lam C, Chou C, Chiang K and Fu L (2011) Human centered robot navigation: Towards a harmonious human-robot coexisting environment. *IEEE Transactions on Robotics* 27(1): 99–112.

Large F, Vasquez D, Fraichard T and Laugier C (2004) Avoiding cars and pedestrians using velocity obstacles and motion predictionotion prediction. In: *Intelligent vehicles symposium*.

Latombe J (1991) *Robot Motion Planning*. Dordrecht: Kluwer Academic Publishers.

LaValle S (2006) *Planning Algorithms*. Cambridge: Cambridge University Press.

Luber M, Tipaldi G and Arras K (2010) People tracking with human motion predictions from social forces. In: *IEEE international conference on robotics and automation*.

Mead R, Atrash A and Matarić M (2011) Proxemic feature recognition for interactive robots: automating metrics from the social sciences. In: *International conference on social robotics*.

Mead R and Matarić M (2012) A probabilistic framework for autonomous proxemic control in situated and mobile human–robot interaction. In: *ACM/IEEE international conference on human–robot interaction*.

Minka TP (2001) Expectation propagation for approximate Bayesian inference. In: *Uncertainty in artificial intelligence*.

Montemerlo M, Pineau J, Roy N, Thrun S and Verma V (2002) Experiences with a mobile robotic guide for the elderly. In: *Association for the advancement of artificial intelligence*.

Pellegrini S, Ess A, Schindler K and Van Gool L (2009) You'll never walk alone: modeling social behavior for multi-target tracking. In: *International conference on computer vision*.

Pellegrini S, Ess A, Tanaskovic M and Van Gool L (2010) Wrong turn - no dead end: a stochastic pedestrian motion model. In: *International workshop on socially intelligent surveillance and monitoring*.

Pineau J, Montemerlo M, Pollack M, Roy N and Thrun S (2003) Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*.

Pradhan N, Burg T and Birchfield S (2011) Robot crowd navigation using predictive position fields in the potential function framework. In: *American control conference*.

Rasmussen CE and Williams C (2006) *Gaussian Processes for Machine Learning*. Cambridge: MIT Press. http://www.gaussianprocess.org/gpml/.

Rawlik K, Toussaint M and Vijayakumar S (2012) On stochastic optimal control and reinforcement learning by approximate inference. In: *Robotics: science and systems*.

Rios-Martinez J, Spalanzani A and Laugier C (2011) Understanding human interaction for probabilistic autonomous navigation using risk-RRT approach. In: *IEEE international conference on intelligent robots and systems*.

Roy N and Thrun S (1999) Online self-calibration for mobile robots. In: *IEEE international conference on robotics and automation*.

Saiki L, Satake S, Huq R, Glas D, Kanda T and Hagita N (2012) How do people walk side-by-side? Using a computational model of human behavior for a social robot. In: *ACM/IEEE international conference on human–robot interaction*.

Schulte J, Rosenberg C and Thrun S (1999) Spontaneous, short-term interaction with mobile robots. In: *IEEE international conference on robotics and automation*.

Seifer DF, Skinner K and Matarić M (2007) Benchmarks for evaluating socially assistive robotics. *Interaction Studies: Psychological Benchmarks of Human–Robot Interaction* 17: 423–439.

Shiomi M, Kanda T, Glas D, Satake S, Ishiguro H and Hagita N (2009) Field trial of networked social robots in a shopping mall. In: *IEEE international conference on intelligent robots and systems*.

Shiomi M, Kanda T, Ishiguro H and Hagita N (2006) Interactive humanoid robots for a science museum. In: *ACM/IEEE international conference on human–robot interaction*.

Sidner C and Lee C (2003) Engagement rules for human–robot collaborative interactions. *IEEE international conference on systems, man and cybernetics*.

Snape J, van den Berg J, Guy S and Manocha D (2011) The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics* 27(4): 696–706.

Svenstrup M, Bak T and Andersen J (2010) Trajectory planning for robots in dynamic human environments. In: *IEEE international conference on intelligent robots and systems*.

Takayama L, Dooley D and Ju W (2011) Expressing thought: Improving robot readability with animation principles. In: *ACM/IEEE international conference on human–robot interaction*.

Takayama L and Pantofaru C (2009) Influences on proxemic behaviors in human–robot interaction. In: *IEEE international conference on intelligent robots and systems*.

Teh Y (2010) Dirichlet processes. In: *Enclyclopedia of Machine Learning*. Berlin: Springer.

Thompson S, Horiuchi T and Kagami S (2009) A probabilistic model of human motion and navigation intent for mobile robot path planning. In: *International conference on autonomous robots and agents*.

Thrun S, Beetz M, et al (2000a) Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *The International Journal of Robotics Research* 19: 972–999.

Thrun S, Schulte J and Rosenberg C (2000b) Interaction with mobile robots in public places. *IEEE Intelligent Systems* (July/August): 7–11.

Toussaint M (2009) Robot trajectory optimization using approximate inference. In: *International conference on machine learning*.

Trautman P (2013) *Robot Navigation in Dense Crowds: Statistical Models and Experimental Studies of Human Robot Cooperation*. PhD Thesis, California Institute of Technology.

Trautman P and Krause A (2010) Unfreezing the robot: Navigation in dense interacting crowds. In: *IEEE international conference on intelligent robots and systems*.

Trautman P, Ma J, Murray R and Krause A (2013) Robot navigation in dense crowds: the case for cooperation. In: *IEEE international conference on robotics and automation*.

van den Berg J, Abbeel P and Goldberg K (2011) LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research* 7: 895–913.

van den Berg J, Guy S, Lin M and Manocha D (2009) Reciprocal *n* -body collision avoidance. In: *International symposium on robotics research*.

van den Berg J, Lin M and Manocha D (2008) Reciprocal velocity obstacles for real-time multi-agent navigation. In: *IEEE international conference on robotics and automation*.

Waugh K, Ziebart BD and Bagnell JAD (2010) Computational rationalization: the inverse equilibrium problem. In: *International conference on machine learning*.

Ziebart BD, Maas A, Dey A and Bagnell JA (2008) Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior. In: *International conference on ubiquitous computing*.

Ziebart BD, Ratliff N and Gallagher G (2009) Planning-based prediction for pedestrians. In: *IEEE international conference on intelligent robots and systems*.