Intention-Aware Motion Planning

Citation: Bandyopadhyay, Tirthankar, Kok Sung Won, Emilio Frazzoli, David Hsu, Wee Sun Lee, and Daniela Rus. "Intention-Aware Motion Planning." Algorithmic Foundations of Robotics X (2013): 475–491.

As Published: http://dx.doi.org/10.1007/978-3-642-36279-8_29

Publisher: Springer-Velag

Persistent URL: http://hdl.handle.net/1721.1/112770

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Intention-Aware Motion Planning

Tirthankar Bandyopadhyay, Kok Sung Won, Emilio Frazzoli, David Hsu,
Wee Sun Lee and Daniela Rus

Abstract As robots venture into new application domains as autonomous vehicles on the road or as domestic helpers at home, they must recognize human intentions and behaviors in order to operate effectively. This paper investigates a new class of motion planning problems with uncertainty in human intention. We propose a method for constructing a practical model by assuming a finite set of unknown intentions. We first construct a motion model for each intention in the set and then combine these models together into a single Mixed Observability Markov Decision Process (MOMDP), which is a structured variant of the more common Partially Observable Markov Decision Process (POMDP). By leveraging the latest advances in POMDP/MOMDP approximation algorithms, we can construct and solve moderately complex models for interesting robotic tasks. Experiments in simulation and with an autonomous vehicle show that the proposed method outperforms common alternatives because of its ability in recognizing intentions and using the information effectively for decision making.

1 Introduction

Motion planning is a critical capability for autonomous robots. The key issues of motion planning—geometry, kinematics, dynamics, uncertainties in robot control and sensing, *etc.* [13]—were identified many years ago. At the time, autonomous robots were mostly confined to tightly controlled environments, such as manufacturing factory floors, where they seldom actively interact with humans. As robots venture into new application domains as autonomous vehicles on the road or as domestic helpers at home, they must recognize human intentions and behaviors in order to operate effectively.

Recognizing human intention is difficult. In principle, estimating intention is similar to estimating other more common quantities such as the robot position and velocity. The true state is inferred from sensor data with some uncertainty. However,

T. Bandyopadhyay
SingaporeMIT Alliance for Research and Technology
K.S. Won, D. Hsu, W.S. Lee
National University of Singapore
Emilio Frazzoli, Daniela Rus
Massachusetts Institute of Technology

Fig. 1 Our experimental platform, a robot golf cart, during an encounter with pedestrians in a parking lot. Each pedestrian is walking towards one of several goals. Unaware of the pedestrians' intentions in advance, the vehicle must pass over them as quickly as possible and avoid accidents.

recognizing intention is often more difficult, because of the diversity and subtlety of human behaviors and the lack of a powerful "intention sensor". Further, a robot may actively gather information for intention recognition by taking sensing actions, but must balance such actions against those contributing directly to the goal of motion planning. The robot's ultimate goal is to complete the specified tasks and not to recognize intention. It thus should not gather more information than necessary.

We propose to treat intention-aware motion planning as planning under uncertainty and model it as a partially observable Markov decision process (POMDP) [9, 19], in fact, a recently introduced variant called the *mixed observability Markov decision process* (MOMDP) [15]. Our MOMDP model assumes that the robot interacts with an intentional agent, *e.g.*, a human. The agent has a finite set of intentions, each embodied in an observable behavior. Given the intention, the agent's behavioral dynamics is modeled in advance and known to the robot. The agent's intention is then the primary source of uncertainty for motion planning and the main partially observable state variable in the MOMDP. The remaining state variables, which specify the agent's and the robot's dynamics, may be either fully or partially observable. Consider, for example, an autonomous robot vehicle in an encounter with pedestrians (Fig. 1). Each pedestrian is walking towards one of several goals, some of which may lead him to cross the road. The robot does not know the pedestrian intention, in this case, the goal in advance and must infer it based on observed pedestrian behavior. The robot then acts accordingly. By modeling the encounter as a MOMDP and solving it, we obtain a conditional plan that enables the robot to act optimally (with respect to the model) despite uncertainty on the pedestrians' intentions.

This work introduces the intention of an interacting agent into motion planning and proposes the MOMDP as a model for it. The MOMDP formulation of intention-aware motion planning provides several key advantages:

- The MOMDP is a rich probabilistic model that captures uncertainties in intention as well as robot control and sensing.
- It divides the complex task of recognizing agent intentions into two simpler parts. We first construct a model of the agent's behavioral dynamics for each intention

and solve the resulting MOMDP model for a plan. The execution of the plan performs inference over a finite number of intentions already modeled, based on observed agent behavior. This approach simplifies both modeling and inference.
- Our MOMDP model treats intention as a single partially observable state variable and limits uncertainty over intention to a small portion of the state space. The latest MOMDP algorithm exploits this modeling feature to achieve dramatic computational efficiency gain over standard POMDP algorithms [12, 15]. The scalability of MOMDPs makes them useful for modeling moderately complex robotic tasks.
- Similar to its POMDP counterpart, the MOMDP model provides a principled general approach to planning under uncertainty and optimally balances information-gathering and task completion actions.

We evaluated our approach on two navigation tasks both in simulation and on a robot golf cart: pedestrian interaction and interaction navigation. In the first task, intentions represent pedestrian goals. In the second task, the autonomous vehicle navigates through an uncontrolled intersection and encounters another vehicle. Here, intention reflects the preference of the approaching vehicle's driver, *e.g.*, the level of caution. Experimental results show that the MOMDP approach outperforms the more common approaches such as reactive planning and Bayesian intention inference in terms of safety and navigation efficiency.

2 Background

2.1 Related Work

Our work touches on several distinct lines of research in the literature. Motion planning is a vast field [13]. Over the years, many important issues have been identified, including the geometry of robots and environments, the kinematics and dynamics of robots, uncertainties in robot control and sensing, *etc.*. Our work introduces a new aspect, the intention of an interacting agent, into motion planning.

Both plan recognition and activity recognition deal with agent intentions (see, *e.g.*, [10, 21]). While they solve the state estimation problem of identifying an agent's plan or activity, we solve a control problem, in which a robot's goal is to complete specified tasks and resolve the intention only when necessary. One distinguishing feature is the need to balance actions that gather information for intention recognition and actions that contribute directly to task completion.

Our work is more closely related to the Hidden Goal Markov Decision Process (HGMDP) and the Helper Action Markov Decision Processes (HAMDP) [6]. Both model a helper agent trying to recognize another agent's intention and perform assistive actions. Solving HGMDPs exactly is PSPACE-hard [6]. The earlier work does not provide a practical way of solving HGMDPs, but it introduces HAMDPs, which place restrictions on how the two agents interact, for computational efficiency.

We model intention-aware motion planning as MOMDPs, which are structured variants of POMDPs. POMDPs provide a general framework for planning under uncertainty. Although solving POMDPs exactly is computationally intractable in

the worst case [16], point-based approximation algorithms have greatly improved the speed of POMDP planning in recent years [12, 17, 20]. Today the fastest algorithms, such as HSVI [20] and SARSOP [12], can solve POMDPs with hundreds of thousands states in reasonable time. MOMDPs specify additional structural information on the corresponding POMDPs and dramatically improve efficiency of point-based approximation algorithms under suitable conditions [15].

2.2 Preliminaries on MDPs and POMDPs

A Markov decision process (MDP) models a system taking a sequence of actions under uncertainty to maximize its total rewards. Formally, an infinite-horizon discrete MDP is a tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the system's state space and action space, respectively. At each time step, the system takes an action $a \in \mathcal{A}$ and moves from the current state $s \in \mathcal{S}$ to a new state $s' \in \mathcal{S}$. To model action uncertainty, we specify the system dynamics with a conditional probability function $T(s, a, s') = p(s'|s, a)$, which gives the probability that the system lies in s', after taking action a in state s. At each time step, the system receives a real-valued reward $R(s, a)$ that depends on its state s and action a. The goal of the system is to choose a sequence of actions that maximizes the expected total reward $\mathrm{E}\big(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\big)$, where s_t and a_t denote the system's state and action at time t, and $\gamma \in (0, 1)$ is a discount factor that reflects the preference of immediate rewards over future ones.

After solving an MDP, we obtain a *policy* $\pi \colon \mathcal{S} \to \mathcal{A}$, which prescribes an action $a \in \mathcal{A}$ for each system state $s \in \mathcal{S}$. An optimal policy π^* maximizes the agent's expected total reward.

The MDP allows us to model action uncertainty only and assumes that the system state is fully observable. To allow for observation uncertainty, due to, *e.g.*, imperfect sensors, we need the POMDP, which adds two additional elements: the observation space $\mathcal{O}$ and the observation function Z. At each time step, the system receives an observation o after taking action a and arriving in state s'. The observation function is again specified as a conditional probability function $Z(s', a, o) = p(o|s', a)$, which models observation uncertainty.

In a POMDP, the system state is not known exactly and is represented as a probability distribution $b(s)$ over $\mathcal{S}$, commonly called a *belief*. Suppose that the beliefs of a discrete POMDP are represented as vectors and $|\mathcal{S}|$ is the number of states in the POMDP. The space $\mathcal{B}$ of all possible beliefs then forms an $(|\mathcal{S}| - 1)$-dimensional simplex, as the probabilities over $\mathcal{S}$ must sum up to 1.

In contrast to an MDP policy, a POMDP policy is a mapping $\pi \colon \mathcal{B} \to \mathcal{A}$, which prescribes a action $a \in \mathcal{A}$ for each belief $b \in \mathcal{B}$. The policy π induces a *value function* $V_\pi \colon \mathcal{B} \to \mathbb{R}$. The *value* of b with respect to π is the system's expected total reward of executing π with initial belief b: $V_\pi(b) = \mathrm{E}\big(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \;\big|\; \pi, b\big)$. The value function V^* for an optimal policy π^* can be approximated arbitrarily closely by a piecewise-linear convex function

$$V(b) = \max_{\alpha \in \Gamma} \sum_{s \in S} \alpha(s) b(s), \tag{1}$$

where each $\alpha \in \Gamma$ is represented as a vector and called an α-vector. Each α-vector defines a hyperplane $h(b) = \sum_{s \in S} \alpha(s)b(s)$ over $\mathcal{B}$. The value function V can be then represented as a finite set of hyperplanes. Most of the fastest discrete-state POMDP algorithms [12, 17, 20] represent a policy by its value function and exploit the α-vector representation for efficient computation. As the value function is defined over $\mathcal{B}$, a high-dimensional belief simplex is a major obstacle to computational efficiency.

Each α-vector is associated with an action. Once a value function is computed, the corresponding policy can be executed by selecting the action associated with the best α-vector at the current belief b, using (1).

3 Intention-Aware Motion Planning as a MOMDP

Consider a robot interacting with an intentional agent. We divide the complex task of recognizing agent intentions and acting optimally into two stages. In the off-line stage, we construct a motion model for each agent intention (Section 3.1) and solve the resulting MOMDP model for a policy (Section 3.2). In the on-line stage, the policy enables the robot to make an inference over a finite set of agent intentions and act accordingly, based on observed agent behavior Section 3.3.

3.1 Modeling

Let $\mathcal{X}$ and $\mathcal{A}$ denote the robot's state space and action space, respectively. Let $\mathcal{Y}$ denote the agent's state space. The robot's motion is governed by the probabilistic transition function $T_{\mathcal{X}}(x, a, x') = p(x'|x, a)$, where $x, x' \in \mathcal{X}$ are the robot's current and next state and $a \in \mathcal{A}$ is an admissible robot action (Fig. 2). The robot may observe both its own and the agent's state, and the probabilistic observation function is given by $Z(x, y, o) = p(o|x, y)$, for a robot state $x \in \mathcal{X}$, an agent state $y \in \mathcal{Y}$, and an observation $o \in \mathcal{O}$.

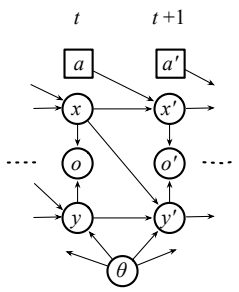


Fig. 2 A MOMDP model for intention-aware motion planning.

The agent's motion is governed by another probabilistic transition function $T_{\mathcal{Y}}(y, a', y') = p(y'|y, a')$ for $y, y' \in \mathcal{Y}$ and some agent action $a' \in \mathcal{A}'$. To relate the agent's action a' to its intention $g \in \Theta$, we further assume that a' is the result of the agent executing a policy $\rho\colon \mathcal{X} \times \mathcal{Y} \times \Theta \to \mathcal{A}'$, which chooses a' based on the current robot state x, the current agent state y, and the agent's intention θ. Recall our earlier example in which a robot vehicle encounters a pedestrian. It is reasonable to expect that the pedestrian chooses an action based on the robot's and his own position and velocity as well as his intention, in this case, the goal location. The form of ρ indicates that the agent has perfect information on the robot's and its own state. Although these assumptions may not hold exactly, it provides a reasonable trade-off between model fidelity and computational complexity, as we show in Section 4.

There are various ways to construct the policy ρ. One possibility is to specify ρ manually. This potentially provides high infidelity in reproducing the agent's behavior, but becomes tedious for an agent with complex behavior. Another possibility is to compute ρ by solving a simplified MDP model. We illustrate this approach with our vehicle-pedestrian encounter example. Assume that the pedestrian approaches the goal location by optimizing an objective function, *e.g.*, following the shortest path and avoiding collision with the vehicle. Assume also that the pedestrian's actions may be imperfect. All these can be captured in an MDP, which is then solved to generate a policy ρ for the pedestrian. Now substituting $a' = \rho(x, y, \theta)$ into $T_{\mathcal{Y}}$, we obtain $T_{\mathcal{Y}}(y, \rho(x, y, \theta), y')$, which clearly shows the dependence of the agent's motion on its intention. For simplicity, we assume that y' does not depend on x', but the dependence can be added if necessary.

To summarize, our MOMDP model is a tuple $(\mathcal{X}, \mathcal{Y}, \Theta, \mathcal{A}, \mathcal{O}, T_{\mathcal{X}}, T_{\mathcal{Y}}, Z, R, \gamma)$, which consists of the joint state space $\mathcal{X} \times \mathcal{Y} \times \Theta$, the robot action space $\mathcal{A}$, the robot observation space $\mathcal{O}$, the transition functions $T_{\mathcal{X}}$ and $T_{\mathcal{Y}}$, the observation function Z, a reward function $R\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, and a discount factor γ. To complete the model, we must also specify which variables are fully or partially observable. In general, the intention is always partially observable. Other variables, which describe the robot's or the agent's state, may be partially observable as well.

3.2 Policy Computation

To compute a policy, we apply SARSOP [12, 15], a leading point-based approximation algorithm, to our MOMDP model. We give a brief description of the algorithm here for completeness. SARSOP samples incrementally a set of points from the belief space $\mathcal{B}$ and maintains a set of α-vectors, which represents a piecewise-linear lower-bound approximation $\underline{V}$ to the optimal value function V^*. We can view the sampling process as building a *belief search tree* rooted at a given initial belief $b_0 \in \mathcal{B}$. The nodes of the tree correspond to beliefs in $\mathcal{B}$, and the edges correspond to action-observation pairs. A child node b' is connected to its parent b by an edge (a, o), if performing the action a and receiving the observation o update the belief b to a new belief b'. Thus every belief in the tree is reachable from b_0 through a sequence of action and observation updates, and the approximation $\underline{V}$ is constructed over a subset $\mathcal{R}(b_0)$ of beliefs reachable from b_0. To compute $\underline{V}$, SARSOP uses value iteration [18], which is based on the idea of dynamic programming. Exploiting the fact that V^* must satisfy the Bellman equation, value iteration starts with an initial set of α-vectors and performs backup operations on the α-vectors at the sampled points by iterating on the Bellman equation, until the iteration converges.

Compared with the standard POMDP model, a major advantage of our MOMDP model is computational efficiency. A POMDP models all components of a system's state in a single variable, thus forcing the state variable to be partially observable even if only one component is partially observable. To simplify the presentation, let us assume that in the rest of this section, intention θ is the only partially observable component in our model. In this case, the dimensionality of $\mathcal{B}$ is nevertheless $|\mathcal{X}||\mathcal{Y}||\Theta| - 1$ for the POMDP model (see Section 2.2). The belief search tree must

be constructed in this high-dimensional space. Furthermore, the primitive objects in SARSOP—the beliefs and α-vectors—must be represented and computed over this space as well. All these increase computational cost, unnecessarily.

The MOMDP model separates the fully and partially observable state components through a factored model. If θ is the only partially observable variable, a belief is represented as $b = (x, y, b_\Theta)$, where x and y are the fully observable robot and agent states and b_Θ is a belief over the partially observable agent intentions. The belief space $\mathcal{B}$ then becomes a union of subspaces: $\mathcal{B} = \{\mathcal{B}_\Theta(x, y) \mid x \in \mathcal{X}, y \in \mathcal{Y}\}$. For each x and y, $\mathcal{B}_\Theta(x, y)$ is a space of beliefs over the intentions, and its dimensionality is only $|\Theta| - 1$, a drastic reduction compared with the POMDP model. Correspondingly, a MOMDP value function $V(b) = V(x, y, b_\Theta)$ is represented as a collection of α-vector sets: $\{\Gamma_\Theta(x, y) \mid x \in \mathcal{X}, y \in \mathcal{Y}\}$, where for each x and y, $\Gamma_\Theta(x, y)$ is a set of α-vectors defined over $\mathcal{B}_\Theta(x, y)$. Geometrically, each α-vector set $\Gamma_\Theta(x, y)$ represents a *restriction* of V to the subspace $\mathcal{B}_\Theta(x, y)$, obtained by restricting the domain of V from $\mathcal{B}$ to $\mathcal{B}_\Theta(x, y)$. In the MOMDP policy computation, SARSOP computes only these restrictions in the lower-dimensional subspaces $\mathcal{B}_\Theta(x, y)$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, because there is no uncertainty over x or y.

3.3 Policy Execution

After computing a MOMDP policy, represented as a value function $V(x, y, b_\Theta)$, we execute the policy by repeating two steps. The first step selects an action for the current belief (x, y, b_Θ). To evaluate $V(x, y, b_\Theta)$, we use x and y as an index to find the right α-vector set and then find the maximum α-vector from the set:

$$V(x, y, b_\Theta) = \max_{\alpha \in \Gamma_\theta(x,y)} \{\alpha \cdot b_\Theta\}. \tag{2}$$

We then execute the action associated with the chosen α-vector. The second step updates the belief. Suppose that after the robot takes action a, the new robot state is x' and the new agent state is y'. The robot receives an observation o, which is exactly (x', y'), as the robot state and the agent state are fully observable. The new belief is then (x', y', b'_Θ), where

$$b'_\Theta(\theta) = \eta\, T_\mathcal{Y}(x, y, \theta, y') b_\Theta(\theta) \tag{3}$$

and η is a normalizing constant.

In general, some of the constituent variables in x and y may be partially observable as well. Let s_f denote all the fully observable variables, and let s_p denote all the partially observable variables, including, in particular, the intention θ. Let $\mathcal{S}_\mathrm{f}$ and $\mathcal{S}_\mathrm{p}$ be the subspaces in $\mathcal{S}$ for s_f and s_p, respectively. We then have

$$V(s_\mathrm{f}, b_{\mathcal{S}_\mathrm{p}}) = \max_{\alpha \in \Gamma_{\mathcal{S}_\mathrm{p}}(s_\mathrm{f})} \{\alpha \cdot b_{\mathcal{S}_\mathrm{p}}\}. \tag{4}$$

and

$$b_{\mathcal{S}_\mathrm{p}}(s'_\mathrm{p}) = \eta Z(x', y', o) \sum_{\theta \in \Theta} T_\mathcal{X}(x, a, x') T_\mathcal{Y}(x, y, \theta, y') b_{\mathcal{S}_\mathrm{p}}(s_\mathrm{p}). \tag{5}$$

Equations (2) and (3) are merely the special case when $s_{\mathrm{f}} = (x, y)$ and $s_{\mathrm{p}} = \theta$. The MOMDP model's computational advantage decreases with an increasing number of partially observable variables. However, it is always at least as efficient as the corresponding POMDP and significantly reduce the computational cost when a system has only a few partially observable variables.

4 Experiments

We now present experimental results on our approach evaluated on two autonomous vehicle navigation tasks: pedestrian interaction (Sections 4.1–4.2) and intersection navigation (Section 4.3). There has significant interest in safe navigation of autonomous vehicles in recent years (see, *e.g.*, [1, 4, 7]). However, the main objective of our work here is is to propose a general approach to intention-aware motion planning. A comparison with these alternative approaches in the specific context of autonomous vehicle navigation will be explored in future work.

4.1 Pedestrian Interaction

Recall the pedestrian avoidance task from Section 1. We model it as a MOMDP. The robot vehicle and the pedestrian traverse in an environment discretized into a uniform grid. Each grid cell has size $1\,\mathrm{m} \times 1\,\mathrm{m}$. The robot has three velocity levels: $0\,\mathrm{m/s}$, $1\,\mathrm{m/s}$, and $2\,\mathrm{m/s}$. The robot's state consists of its position and velocity. It has three actions that control the acceleration: ACCELERATE ($1\,\mathrm{m/s^2}$), MAINTAIN ($0\,\mathrm{m/s^2}$), and DECELERATE ($-1\,\mathrm{m/s^2}$). The robot motion may be noisy. We assume that the robot can sense its own position and the pedestrian's position perfectly, as the laser range finder on-board our robot golf cart (Fig. 1) is sufficiently accurate, with respect to the discretized environment. The robot gets a reward for safely passing over the pedestrian and gets penalties for collision with the pedestrian, speeding, and time delay. Each time step has duration 1 second.

To model the pedestrian behavior, we use a simplified version of Helbing *et al.*'s pedestrian motion model [8], which has been carefully validated on empirical data. This model resembles the potential field method [11] for motion planning: the pedestrian is attracted to a goal and is repelled by the robot vehicle, which is treated as a quasi-static obstacle. The pedestrian movement may be noisy as well. The hidden intention is the pedestrian's goal location, which the robot does not know in advance and cannot sense directly.

For comparison, we consider two common pedestrian avoidance methods. The first one is simple reactive planning. At each time step, if the pedestrian is within a predefined distance from the robot according to the current observation, the robot slows down to a full stop and waits until the pedestrian passes by. The second method, Bayes-ML, pre-computes a set of MDP policies, each assuming a known pedestrian goal. It then performs Bayesian inference on the pedestrian's goal based on the received observations. Just as our MOMDP policy, Bayes-ML maintains a belief over pedestrian goals by incorporating the observation at each time step (implementing the same update Eqn. 3). However, it chooses the action by determining

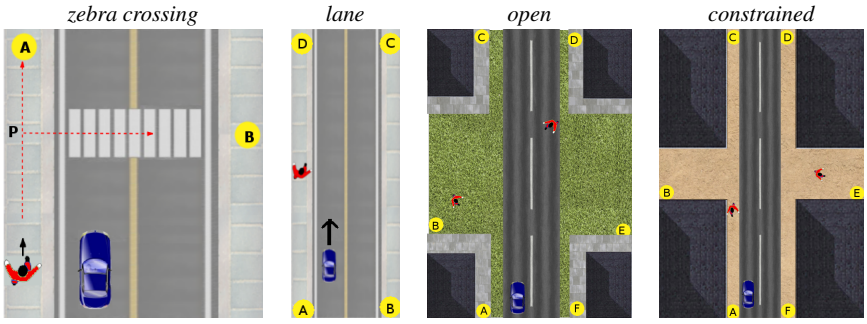
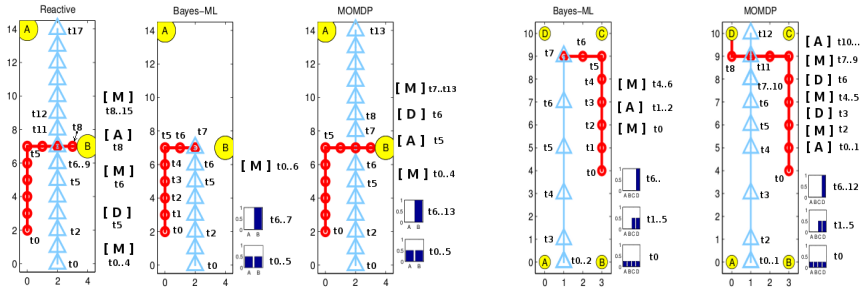


Fig. 3 Four test environments. The letters in yellow circles mark pedestrian goals. The dashed curves in the leftmost environment mark the pedestrian paths towards the goals.



(*a*) (*b*)

Fig. 4 Simulation runs comparing MOMDP policies with two other methods in the (*a*) *zebra crossing* and the (*b*) *lane* environments. Red curves mark the pedestrian's trajectories. Light blue curves mark the robot's trajectories. The histograms show the robot's belief on the pedestrian's goal over time. "A", "M", and "D" indicate the robot's three actions.

a most likely goal from the current belief and looking up the action from the precomputed MDP policy for this goal.

We compared these methods in four environments in simulation (Fig. 3). In the *zebra crossing* environment, the robot vehicle encounters a pedestrian, who may follow two possible trajectories, one proceeding straight ahead and one crossing the road over the zebra stripes. The *lane* environment is similar, but has four pedestrian goals. Furthermore, as there is no zebra marking, the pedestrian can "jaywalk" towards the goals directly. The *open* environment emulates a road going through open space, *e.g.*, in a park. The *constrained* environment emulates a narrow lane between city blocks with pedestrian walkways across. The *open* and the *constrained* environments look similar. However, the latter is more difficult, because many pedestrian trajectories overlap due to the constrained space, making it challenging to recognize the pedestrian's true goal. The performance statistics are shown in Table 1, and some sample simulation runs are shown in Fig. 4.

Fig. 4*a* shows that that for reactive planning, the robot slows down to a full stop. It waits until the pedestrian crosses the road and is sufficiently far. It then proceeds. If the pedestrian stands at the location P (see Fig. 3, leftmost) for a prolonged time, the

Table 1 Performance comparison of Bayes-ML and MOMDP in four test environments.

Environment	Noise	Bayes-ML		MOMDP	
		Time	Accident	Time	Accident
zebra	zero	14.8 (0.4)	4.8%	14.7 (0.5)	0%
crossing	high	17.0 (3.6)	2.8%	16.8 (3.5)	2.1%
lane	zero	7.6 (1.0)	2.5%	7.7 (1.2)	0.3%
	low	9.1 (1.8)	3.1%	9.1 (2.2)	1.9%
	med	8.6 (1.8)	6.1%	8.6 (1.8)	5.9%
	high	10.1 (4.0)	5.8%	10.7 (4.4)	5.0%
open	zero	11.3 (0.9)	0.8%	11.4 (1.1)	0.02%
	low	13.5 (2.6)	1.7%	13.6 (2.6)	1.3%
	med	14.3 (3.8)	2.5%	14.5 (4.0)	2.2%
	high	14.6 (4.5)	3.0%	14.5 (4.4)	2.7%
constrained	zero	11.3 (0.8)	1.6%	11.5 (1.2)	0.07%
	low	13.5 (2.5)	1.7%	13.7 (2.5)	0.9%
	med	14.6 (4.0)	3.6%	15.1 (4.3)	3.3%
	high	18.4 (10.4)	4.0%	21.9 (13.5)	3.2%

robot must wait, as the pedestrian is within the predefined distance. This is clearly unacceptable and is an inherent limitation of simple reactive planning, which does not look ahead and plan for the future.

We now focus on the comparison between Bayes-ML and our MOMDP policy, which both perform probabilistic inference on the pedestrian's goal. For each test environment, we varied the system noise level, and constructed and solved a MOMDP model. For the *zebra crossing* environment, we varied the noise level in robot motion. In the other three environments, we varied pedestrian movement noise, which makes intention inference more difficult. Each data entry in Table 1 is the result over $4,500$ simulation runs. Columns 3 and 5 of the table report the average time and the standard deviation for the robot to pass over the pedestrian and reach the goal. Columns 4 and 6 report the rate of accident, which occurs when the robot and pedestrian are within a predefined distance. For easy comparison, we tuned the MOMDP reward functions so that the time to clear is roughly the same for Bayes-ML and the MOMDP policies. The statistics show that the MOMDP policies consistently have lower accident rate, sometimes substantially.

To understand the reasons behind MOMDP policies' better performance, let us look at some simulation runs (Fig. 4). In the *zebra crossing* environment (Fig. 4*a*), Bayes-ML maintains a belief over the pedestrian's goal based on the observations. However, the initial stretches of the two pedestrian trajectories towards A and B overlap. The robot has no information to distinguish the pedestrian's goal until time step $t = 5\,\mathrm{s}$, when the pedestrian reaches the location P. The histogram shows that the belief at $t = 5\,\mathrm{s}$ is roughly uniform, but not exactly because of noise in pedestrian movement. Bayes-ML forces the robot to act according to the most likely goal and ignores the alternative completely. If goal A happens to have slightly higher probability, the robot will maintain its current velocity. At $t = 6\,\mathrm{s}$, the new observation changes the belief and shows clearly that the pedestrian is crossing the road towards B. However, it is too late by then. Because of the robot's dynamics, no action can prevent an accident. It cannot stop in time with DECELERATE. Neither can it

overtake the pedestrian fast enough with ACCELERATE. Bayes-ML applies MAINTAIN simply because it incurs the lowest cost. One main weakness of Bayes-ML is its failure to exploit the full information in the belief and instead to choose the action based solely on the mostly likely estimate. It is overly confident and does not hedge against all possible situations.

When faced with the same roughly uniform belief at $t = 5$ s, the MOMDP policy reasons about the future effects of both goals A and B and realizes the danger if the pedestrian crosses the road. It chooses ACCELERATE so that the robot passes over the pedestrian before any dangerous situation occurs. At $t = 6$ s, the robot is assured of safely overtaking the pedestrian at its current velocity. The policy then chooses DECELERATE in order to avoid the penalty incurred in high-velocity states. It is interesting to note that the key action, ACCELERATE, is decided at $t = 5$ s, when the pedestrian's true goal is still largely unresolved (see the belief histogram in Fig. 4*a*).

The simulation run in the *lane* environment (Fig. 4*b*) provides more information on the MOMDP policies' behavior. Initially, the robot's belief on the pedestrian's goal is uniform. At $t = 1$ s, the robot observes the pedestrian's forward movement. The belief now peaks on goals C and D. The robot then chooses ACCELERATE. As the robot catches up with the pedestrian, it chooses DECELERATES at $t = 3$ s to stay a safe distance behind, in case the pedestrian suddenly crosses the road. At $t = 6$ s, the pedestrian steps off the curb. The updated belief concentrates almost entirely on the goal D, and the pedestrian's intention is now resolved. The robot decelerates until it fully stops, waits for the pedestrian to cross the road, and then proceeds.

Comparison of the MOMDP policies' behavior in Fig. 4*a* and Fig. 4*b* clearly demonstrates the MOMDP's unique ability in optimally balancing exploration and exploitation. It gathers information and resolves the uncertainty in the pedestrian's intention only when necessary.

4.2 A Robot Golf Cart Interacting with Multiple Pedestrians

We implemented and tested a MOMDP controller on a robot golf cart (Fig. 1). The vehicle uses a SICK LMS200 laser range finder and a Logitech webcam for pedestrian detection and localization. The MOMDP controller provides high-level advisory actions at 2 Hz to a low-level PID controller, which handles motor and brake commands. For safety, the vehicle has a maximum speed limit when a pedestrian is within a predetermined distance. More details of the hardware platform are described in [5].

Our method is extended to handle multiple pedestrians. We instantiate a MOMDP controller for each pedestrian detected. These controllers operate independently. We then combine their output actions by choosing the most conservative one.

Fig. 5 shows a test run of the vehicle in a parking lot. There are three pedestrians. Initially the beliefs on all the pedestrians' goals are uniform. The time step $t = 2$ s is quite interesting. As a result of inherent stochastic noise in human movement, the beliefs seem to indicate that none of the pedestrians will cross the road. In this case, Bayes-ML would choose to accelerate. The red color bars in the picture show that for every pedestrian, the MOMDP controller chooses to either maintain low speed or

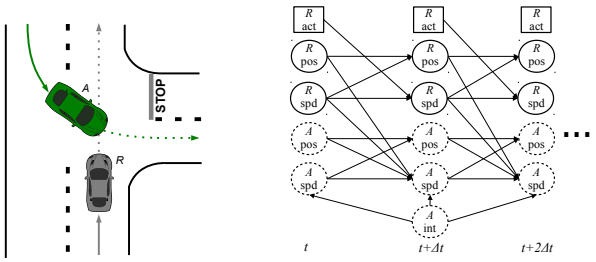
Fig. 5 A test run of the robot golf cart under the control of a MOMDP policy. The pictures are taken from a camera on-board the vehicle. A full view of the test environment is shown in Fig. 1. There are three pedestrians. For each pedestrian, the blue histogram shows the summary belief of whether the pedestrian will stay on the left or right side of the road. The color bar indicates the action chosen with respect to the pedestrian. Red means to decelerate from high speed or maintain low speed. Green means to accelerate from low speed or maintain high speed.

decelerate. The reason is that the probability that a pedestrian suddenly crosses the road is still substantial and the penalty for an accident is high. With the additional observation received at $t = 3\,\mathrm{s}$, the belief indicates that the rightmost pedestrian will not cross; the MOMDP controller decides that it is safe to move on with respect to him (see the green bar in the picture). Then, the belief at $t = 5\,\mathrm{s}$ indicates that the leftmost pedestrian will also not cross, and it is safe to move on. However, the middle pedestrian is crossing the road. The action chosen with respect to him is to decelerate or maintain low speed (see the red bar). This is the most conservative action and is actually executed. At $t = 8\,\mathrm{s}$, the rightmost pedestrian is far away and is no longer detected as a potential threat. At $t = 11\,\mathrm{s}$, the middle pedestrian has crossed. The green bars show that the actions chosen now for all pedestrians are to accelerate. The vehicle moves on and safely pass over the pedestrians.

This test uses the same model as that for the *lane* environment with pedestrian movement noise. The result shows that our approach is robust under uncertainty in robot control and sensing, and is scalable with respect to multiple pedestrians.

4.3 Intersection Navigation

The intersection navigation task is motivated by a near-miss accident in the 2007 DARPA Urban Challenge [14]. Two autonomous vehicles, R and A, approach an



(*a*) (*b*)

Fig. 6 Intersection navigation. (*a*) A near-miss accident during the 2007 DARPA Urban Challenge. (*b*) A MOMDP model for vehicle R to recognize the intention of vehicle A and navigate safely through the intersection. Observation variables are omitted to avoid clutter.

uncontrolled traffic intersection (Fig. 6*a*). R comes to a stop and then resumes its forward motion after checking for precedence. A wants to make a left turn, but fails to yield. Without understanding A's intention, R continues its forward motion. The two vehicles got so close that an emergency mechanism was activated to stop both. We want to model this scenario as intention-aware motion planning and evaluate whether a MOMDP policy can potentially improve safety and efficiency when an autonomous vehicle navigates through an intersection. In the special case of two autonomous vehicles, intentions can in principle be communicated directly. However, in the foreseeable future, autonomous vehicles must interact with other vehicles with human drivers. Such direct communication will not be possible then.

Following the approach in Section 3, we built a MOMDP model for vehicle R (Fig. 6*b*). The state of R consists of its position and velocity. The environment is discretized into a uniform grid. Each grid cell has size $2.5\,\mathrm{m} \times 2.5\,\mathrm{m}$. There are five velocity levels from $0\,\mathrm{m/s}$ to $4\,\mathrm{m/s}$. Each time step has duration $\Delta t = 0.25\,\mathrm{s}$. Vehicle R has four actions: ACCELERATE ($1\,\mathrm{m/s^2}$), MAINTAIN ($0\,\mathrm{m/s^2}$), DECELERATE ($-1\,\mathrm{m/s^2}$), EMERGENCYSTOP ($-3\,\mathrm{m/s^2}$). The actions are noisy with probability 0.05 of failing to achieve the intended outcomes. We use probabilistic transition to compensate for the coarse discretization by matching the *expected* distance of travel per time step. For example, if the vehicle's speed is $1\,\mathrm{m/s}$, it then travels a distance of 0.25 m per time step. Let x and x' be two adjacent grid cells along the vehicle's path. We set $p(x'|x) = 0.1$ and $p(x|x) = 0.9$ so that the expected distance that vehicle travels is also $2.5 \times 0.1 + 0 \times 0.9 = 0.25\,\mathrm{m}$ in our model. Vehicle R receives observations on its own state and vehicle A's state, but the observations may be noisy. It gets a reward for crossing the intersection safely. It gets a penalty for time delay and a high penalty for collision with A. By adjusting reward and penalty values, we can trade-off accident rate and time to clear the intersection. This is an additional practical advantage of the MOMDP formulation.

Vehicle A has a motion model similar to R's. Its intention is unknown to R. We assume that A exhibits one of four driving behaviors, each reflecting an intention:

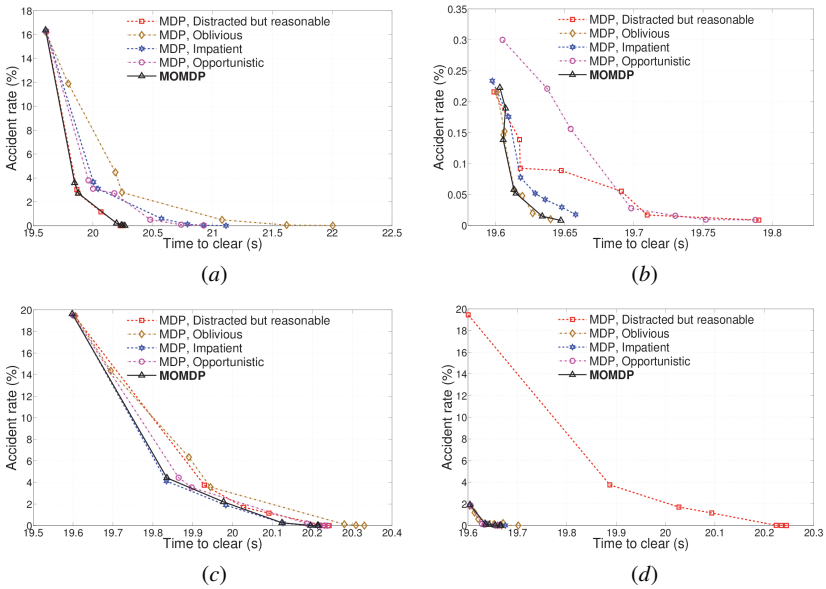


Fig. 7 Performance of the robot vehicle when faced with another vehicle that is (*a*) reasonable but distracted, (*b*) oblivious, (*c*) impatient, and (*d*) opportunistic but rational.

- *Reasonable but distracted*. This driver usually slows and stops before the intersection, but with probability 0.1, he may not stop.
- *Oblivious*. This driver increases his speed to $2\,\mathrm{m/s}$ and maintains it, totally ignoring the presence of other vehicles.
- *Impatient*. This driver seeks to cross the intersection as fast as possible. He reacts to the speed of vehicle R, increasing his speed if R slows down and vice versa. He never comes to a complete stop at the intersection.
- *Opportunistic*. Similar to the impatient driver, this driver increases his speed if R slows down and vice versa. However, he will come to a complete stop at the intersection to avoid a collision.

Although not exhaustive, the list includes some common behaviors encountered in intersection navigation, and more can be added if desired.

We built several MOMDP models by varying the reward function, in order to trade off accident rate and time to clear the intersection. We evaluated the resulting policies in SUMO, an established open-source package for microscopic road traffic simulation [3]. For comparison, we also computed and evaluated four MDP policies, which assume that the intended behavior of vehicle A is known. The MDP policies must perform better than our MOMDP policies, as they know A's intention in advance. The results are shown in Figs. 7 and 8, which plot the performance of vehicle R measured in accident rate versus time to clear the intersection. Each data point was obtained from $10,000$ simulation runs.

The results in Fig. 7 assume that vehicle R has perfect observations. Each plot shows the performance of the MOMDP and MDP policies, when vehicle A has a

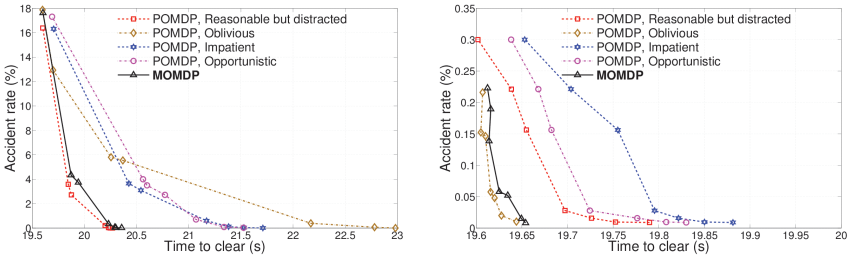


Fig. 8 Performance of the robot vehicle with noisy observations, when faced with another vehicle that is (*a*) reasonable but distracted and (*b*) oblivious.

particular driving behavior. The plots show that the MOMDP policy consistently performs almost as well as the MDP policy that knows the true underlying driving behavior. The performance gap between the MOMDP and MDP policies is very small. In contrast, the MDP policies with the wrong assumption of vehicle A's behavior usually perform poorly. This suggests that identifying A's underlying behavior is crucial and the MOMDP policies do so successfully.

We performed further testing by adding noise to the observations. The noise causes the position of vehicle A to be observed at the adjoining grid cells with probability 0.3 rather than its actual cell. Due to space limitation, Fig. 8 shows the results for two of the four driving behaviors only. The remaining results are similar. Observation noise causes moderate performance degradation of the MOMDP policies. This is expected, because with observation noise, it is more difficult to infer vehicle A's true behavior. However, the performance gaps between the MOMDP policy and the best POMDP policy is still relatively small.

5 Conclusion

This paper introduced the intention of an interacting agent into motion planning. We treat intention-aware motion planning as planning under uncertainty and model it as a MOMDP. The MOMDP is a structured variant of the more common POMDP. It is a rich probabilistic model that can accommodate not only uncertainty in the intention of an interacting agent but also uncertainty in robot control and sensing. It provides a principled general approach to planning under uncertainty and optimally balances information-gathering and task completion actions. By separating the fully and partially observable variables through a factored model, the MOMDP significantly improves computational efficiency for policy computation under suitable conditions, making it a practical tool for modeling interesting robotic tasks.

Our current model assumes that the interacting agent's intention is fixed and does not change over time. To allow changing intentions, we can generalize our MOMDP model by adding a new component for the intention dynamics.

SARSOP, the MOMDP solver used in our implementation, assumes a discrete state state. We are investigating the possibility of using a continuous POMDP solver [2] to remove this modeling restriction.

Acknowledgments. We thank Leslie Kaelbling and Tomás Lozano-Pérez from MIT for many insightful discussions on POMDPs. This work is supported in part by SMART IRG grant R-252-000-447-592, MoE AcRF grant 2010-T2-2-071, and MDA GAMBIT grant R-252-000-398-490.

References

1. G.S. Aoude, B.D. Luders, D.S. Levine, and J.P. How. Threat-aware path planning in uncertain urban environments. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2010.
2. H.Y. Bai, D. Hsu, W.S. Lee, and V.A. Ngo. Monte Carlo value iteration for continuous-state POMDPs. In D. Hsu et al., editors, *Algorithmic Foundations of Robotics IX—Proc. Int. Workshop on the Algorithmic Foundations of Robotics (WAFR)*. Springer, 2010.
3. M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. Sumo – simulation of urban mobility: An overview. In *Proc. Int. Conf. on Advances in System Simulation*, pages 63–68, 2011.
4. M. Bennewitz and W. Burgard. Adapting navigation strategies using motion patterns of people. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2003.
5. Z.J. Chong et al. Autonomous personal vehicle in crowded campus environments. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems, Workshop on Perception and Navigation for Autonomous Vehicles in Human Environment*, 2011.
6. A. Fern and P. Tadepalli. A computational decision theory for interactive assistants. In *Advances in Neural Information Processing Systems (NIPS)*. 2010.
7. C. Fulgenzi, C. Tay, A. Spalanzani, and C. Laugier. Probabilistic navigation in dynamic environment using rapidly-exploring random trees and Gaussian processes. In *Proc. IEEE/RSJ Int. Conf. on Intelligent RObots & Systems*, 2008.
8. D. Helbing, L. Buzna, A. Johansson, and T. Werner. Self-organized pedestrian crowd dynamics and design solutions: Experiments, simulations and design solutions. *Transportation Science*, 39(1):1–24, 2005.
9. L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.
10. H. Kautz and J.F. Allen. Generalized plan recognition. In *Proc. AAAI Conf. on Artificial Intelligence*, volume 19, page 86, 1986.
11. O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robotics Research*, 5(1):90–98, 1986.
12. H. Kurniawati, D. Hsu, and W.S. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. Robotics: Science and Systems*, 2008.
13. J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.
14. J. Leonard et al. A perception driven autonomous urban vehicle. *J. Field Robotics*, 25(10):727–774, 2008.
15. S.C.W. Ong, S.W. Png, D. Hsu, and W.S. Lee. Planning under uncertainty for robotic tasks with mixed observability. *Int. J. Robotics Research*, 29(8):1053–1068, 2010.
16. C. Papadimitriou and J.N. Tsisiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.
17. J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc. Int. Jnt. Conf. on Artificial Intelligence*, pages 477–484, 2003.
18. S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.
19. R.D. Smallwood and E.J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21:1071–1088, 1973.
20. T. Smith and R. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. In *Proc. Uncertainty in Artificial Intelligence*, 2005.
21. P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Trans. on Circuits & Systems for Video Technology,*, 18(11):1473–1488, 2008.