Multipolicy Decision-Making for Autonomous Driving via Changepoint-based Behavior Prediction

Enric Galceran, Alexander G. Cunningham, Ryan M. Eustice, and Edwin Olson
University of Michigan
{egalcera, alexgc, eustice, ebolson}@umich.edu

***Abstract*—To operate reliably in real-world traffic, an autonomous car must evaluate the consequences of its potential actions by anticipating the uncertain intentions of other traffic participants. This paper presents an integrated behavioral inference and decision-making approach that models vehicle behavior for both our vehicle and nearby vehicles as a discrete set of closed-loop policies that react to the actions of other agents. Each policy captures a distinct high-level behavior and intention, such as driving along a lane or turning at an intersection. We first employ Bayesian changepoint detection on the observed history of states of nearby cars to estimate the distribution over potential policies that each nearby car might be executing. We then sample policies from these distributions to obtain high-likelihood actions for each participating vehicle. Through closed-loop forward simulation of these samples, we can evaluate the outcomes of the interaction of our vehicle with other participants (e.g., a merging vehicle accelerates and we slow down to make room for it, or the vehicle in front of ours suddenly slows down and we decide to pass it). Based on those samples, our vehicle then executes the policy with the maximum expected reward value. Thus, our system is able to make decisions based on coupled interactions between cars in a tractable manner. This work extends our previous multipolicy system [11] by incorporating behavioral anticipation into decision-making to evaluate sampled potential vehicle interactions. We evaluate our approach using real-world traffic-tracking data from our autonomous vehicle platform, and present decision-making results in simulation involving highway traffic scenarios.**

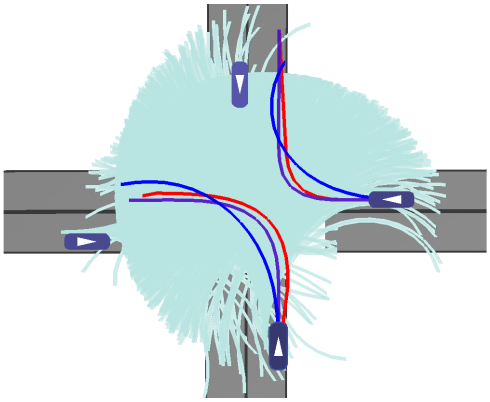


Fig. 1. Our multipolicy approach allows us to sample from the likely coupled interactions between traffic agents. In this simulation at a four-way stop-sign-regulated intersection (§VI-D), we evaluate the outcomes of the possible intentions of other cars to make a decision for our car. The bottom and right cars proceed through the intersection, while the other two cars yield. This experiment shows that our multipolicy sampling strategy generates high-likelihood samples over the coupled interactions of vehicles, and that is orders of magnitude faster than uninformed sampling strategies commonly used in the literature (§VI-D). Legend: human-driven trajectories (red); rollouts from our multipolicy sampling strategy (purple); high-likelihood trajectories obtained by an uninformed sampling strategy (dark blue); trajectories sampled by the uninformed strategy before finding a high-likelihood sample (light blue).

I. Introduction

Decision-making for autonomous driving is hard due to uncertainty on the continuous state of nearby vehicles and, in particular, due to uncertainty over their discrete potential intentions (such as turning at an intersection or changing lanes).

Previous approaches have employed hand-tuned heuristics [28, 29, 41] and numerical optimization [17, 21, 42], but these methods fail to capture the coupled dynamic effects of interacting traffic agents. Partially observable Markov decision process (POMDP) solvers [2, 26, 35] offer a theoretically-grounded framework to capture these interactions, but have difficulty scaling up to real-world scenarios. In addition, current approaches for anticipating future intentions of other traffic agents [1, 22, 24, 25] either consider only the current state of the target vehicle, ignoring the history of its past actions, or rather require expensive collection of training data.

In this paper, we present an integrated behavioral anticipation and decision-making system that models behavior for *both* our vehicle and nearby vehicles as the result of closed-loop policies. This approach is made tractable by considering only a finite set of *a priori* known policies. Each policy is designed to capture a different high-level behavior, such as following a lane, changing lanes, or turning at an intersection. Our system proceeds in a sequence of two interleaved stages of behavioral prediction and decision-making. In the first stage, we estimate the probability distribution over the potential policies other traffic agents may be executing. To this aim, we leverage Bayesian changepoint detection to estimate which policy a given vehicle was executing at each point in its history of actions, and then infer the likelihood of each potential intention of the vehicle. Furthermore, we propose a statistical test based on changepoint detection to identify anomalous behavior of other vehicles, such as driving in the wrong direction or swerving out of lanes. Individual policies can therefore adjust their behavior to react to anomalous cars.

In the second stage, we use this distribution to sample over permutations of other vehicle policies and the policies available for our car, with forward-simulation of these sampled intentions to evaluate their outcomes via a user-defined

reward function. Our vehicle finally executes the policy that maximizes the expected reward given the sampled outcomes. Thus, our system is able to make decisions based on closed-loop interactions between cars in a tractable manner.

We evaluate our behavioral prediction system using a real-world autonomous vehicle, and present decision-making results in simulation involving highway traffic scenarios.

The central contributions of this paper are:

- A changepoint-based behavioral prediction approach that leverages the history of actions of a target vehicle to infer the likelihood of its possible future actions and detect anomalous behavior online.
- A decision-making algorithm that evaluates the outcomes of modeled interactions between vehicles, being able to account for the effect of its actions on the future reactions of other participants.
- An evaluation of the proposed system using both traffic data obtained from a real-world autonomous vehicle and simulated traffic scenarios.

This work extends our earlier work [11], where we proposed the strategy of selecting between multiple policies for our car by evaluating them via forward simulation, and demonstrated passing maneuvers using a real-world autonomous vehicle. However, that work did not address anticipation of policies for other cars. In contrast, this paper presents a fully integrated behavioral anticipation and decision-making approach.

II. Related Work

A. Related Work on Behavioral Prediction

Despite the probabilistic nature of the anticipation problem, some approaches in the literature assume no uncertainty on the future states of other participants [10, 31, 33]. Such an approach could be justified in a scenario where vehicles broadcast their intentions over some communications channel, but it is an unrealistic assumption otherwise.

Some approaches assume a dynamic model of the obstacle and propagate its state using standard filtering techniques such as the extended Kalman filter [13, 18]. Despite providing rigorous probabilistic estimates over an obstacle's future states, these methods often perform poorly when dealing with nonlinearities in the assumed dynamics model and the multimodalities induced by discrete decisions (e.g. continuing straight, merging, or passing). Some researchers have explored using Gaussian mixture models (GMMs) [14, 22] and context-sensitive models [19, 20] to account for nonlinearities and multiple discrete decisions. However, this approach does not consider the history of previous states of the target object, assigning an equal likelihood to each discrete hypothesis and leading to a conservative estimate.

A common anticipation strategy in autonomous driving [7, 16, 21] consists in computing the possible goals of a target vehicle by planning from its standpoint, accounting for its current state. This strategy is similar to our factorization of potential driving behavior into a set of policies, but lacks our closed-loop simulation of vehicle interactions.

Recent work uses Gaussian process (GP) regression to learn typical motion patterns for classification and prediction of agent trajectories [24, 25, 40], particularly in autonomous driving [1, 38, 39]. Nonetheless, these methods require collecting training data to reflect all possible motion patterns the system may encounter, which can be time consuming. For instance, a lane change motion pattern learned in urban roads will not be representative of the same maneuver performed at higher speeds on the highway.

B. Related Work on Decision Making

The first instances of decision making systems for autonomous vehicles capable of handling urban traffic situations stem from the 2007 DARPA Urban Challenge [12]. In that event, participants tackled decision making using a variety of solutions ranging from finite state machines (FSMs) [29] and decision trees [28] to several heuristics [41]. However, these approaches were tailored for very specific and simplified situations and were, even according to their authors, "not robust to a varied world" [41].

More recent approaches have addressed the decision making problem for autonomous driving through the lens of trajectory optimization [17, 21, 42]. However, these methods do not model the closed-loop interactions between vehicles, failing to reason about their potential outcomes.

The POMDP model provides a mathematically rigorous formulation of the decision making problem in dynamic, uncertain scenarios such as autonomous driving. Unfortunately, finding an optimal solution to most POMDPs is intractable [27, 32]. A variety of general [2, 5, 26, 35, 37] and domain-specific [8] POMDP solvers exist in the literature that seek to approximate the solution. Nonetheless, online application of POMDP solvers [6] remains challenging because they often explore unlikely regions of the belief space.

The idea of assuming finite sets of policies to speed up planning has appeared before in the POMDP literature [3, 23, 36]. However, these approaches dedicate significant resources to compute their sets of policies, and as a result they are limited to short planning horizons and relatively small state, observation, and action spaces. In contrast, we propose to exploit domain knowledge to design a set of policies that are readily available at planning time.

III. Problem Formulation

We first formulate the problem of decision making in dynamic, uncertain environments with tightly coupled interactions between multiple agents as a multiagent POMDP. We then show how we exploit autonomous driving domain knowledge to make approximations to the POMDP formulation, thus enabling principled decisions in a tractable manner.

A. General Decision Process

Let V denote the set of vehicles interacting in a local neighborhood of our vehicle, including our controlled vehicle. At time t, a vehicle $v \in V$ can take an action $a_t^v \in \mathcal{A}^v$ to transition from state $x_t^v \in \mathcal{X}^v$ to x_{t+1}^v. In our system, a state

x_t^v is a tuple of the pose, velocity, and acceleration and an action a_t^v is a tuple of controls for steering, throttle, brake, shifter, and directionals. As a notational convenience, let x_t include all state variables x_t^v for all vehicles at time t, and similarly let $a_t \in \mathcal{A}$ be the actions of all vehicles.

We model the vehicle dynamics with a conditional probability function $T(x_t, a_t, x_{t+1}) = p(x_{t+1}|x_t, a_t)$. Similarly, we model observation uncertainty as $Z(x_t, z_t^v) = p(z_t^v|x_t)$, where $z_t^v \in \mathcal{Z}^v$ is the observation made by vehicle v at time t, and $z_t \in \mathcal{Z}$ is the vector of all sensor observations made by all vehicles. In our system, an observation z_t^v is a tuple including the estimated poses and velocities of nearby vehicles and an occupancy grid of static obstacles. Further, we model uncertainty on the behavior of other agents with the following driver model: $D(x_t, z_t^v, a_t^v) = p(a_t^v|x_t, z_t^v)$, where $a_t^v \in \mathcal{A}$ is a latent variable that must be inferred from sensor observations.

Our vehicle's goal is to find an optimal policy π^* that maximizes the expected reward over a given decision horizon H, where a policy is a mapping $\pi : \mathcal{X} \times \mathcal{Z}^v \to \mathcal{A}^v$ that yields an action from the current maximum *a posteriori* (MAP) estimate of the state and an observation:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}\left[\sum_{t=t_0}^{H} \int_{\mathcal{X}} R(x_t) p(x_t)\, dx_t\right], \tag{1}$$

where $R(x_t)$ is a real-valued reward function $R : \mathcal{X} \to \mathbb{R}$. The evolution of $p(x_t)$ over time is governed by

$$\begin{aligned} p(x_{t+1}) = \iiint_{\mathcal{X}\,\mathcal{Z}\,\mathcal{A}} & p(x_{t+1}|x_t, a_t) p(z_t|x_t) \\ & p(a_t|x_t, z_t) p(x_t)\, da_t\, dz_t\, dx_t. \end{aligned} \tag{2}$$

The driver model $D(x_t, z_t^v, a_t^v)$ implicitly assumes that the instantaneous actions of each vehicle are independent of each other, since a_t^v is conditioned only on x_t and z_t^v. However, modeled agents can still react to the observed states of nearby vehicles via z_t^v. That is to say that vehicles do not collaborate with each other, as would be implied by an action a_t^v dependent on a_t. Thus, the joint density for a single vehicle v can be written as

$$\begin{aligned} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v) = p(x_{t+1}^v|x_t^v, a_t^v) p(z_t^v|x_t^v) \\ p(a_t^v|x_t^v, z_t^v) p(x_t^v), \end{aligned} \tag{3}$$

and the independence assumption finally leads to

$$p(x_{t+1}) = \prod_{v \in V} \iiint_{\mathcal{X}^v\,\mathcal{Z}^v\,\mathcal{A}^v} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v)\, da_t^v\, dz_t^v\, dx_t^v. \tag{4}$$

Despite assuming independent vehicle actions, marginalizing over the large state, observation and action spaces in Eq. 4 is too expensive to find an optimal policy online in a timely manner. A possible approximation to speed up the process, commonly used by general POMDP solvers [2, 37] is to solve Eq. 1 by drawing samples from $p(x_t)$. However, sampling over the full probability space with random walks will yield a large number of low probability samples (see Fig. 1). This paper presents an approach designed to sample from high likelihood scenarios such that the decision-making process is tractable.

B. Multipolicy Approach

We make the following approximations to sample from the likely interactions of traffic agents:

1) At any given time, both our vehicle and other vehicles are executing a policy from a discrete set of policies.
2) We approximate the vehicle dynamics and observation models through deterministic, closed-loop forward simulation of all vehicles with assigned policies.

These approximations allow us to evaluate the consequences of our decisions over a limited set of high-level behaviors determined by the available policies (for both our vehicle and other agents), rather than performing the evaluation for every possible control input of every vehicle.

Let Π be a discrete set of policies, where each policy captures a specific high-level driving behavior. Let each policy $\pi \in \Pi$ be parameterized by a parameter vector θ capturing variations of the given policy. For example, for a lane-following policy, θ can capture the "driving style" of the policy by regulating its acceleration profile to be more or less aggressive. We thus reduce the search in Eq. 1 to a limited set of policies. By assuming each vehicle $v \in V$ is executing a policy $\pi_t^v \in \Pi$ at time t, the driver model for other agents can be now expressed as:

$$D(x_t, z_t^v, a_t^v, \pi_t^v) = p(a_t^v|x_t, z_t^v, \pi_t^v) p(\pi_t^v|x_t, \mathbf{z_{0:t}}), \tag{5}$$

where $p(\pi_t^v|x_t, \mathbf{z_{0:t}})$ is the probability that vehicle v is executing the policy π_t^v (we describe how we infer this probability in §IV). Thus, the per-vehicle joint density from Eq. 3 can now be approximated in terms of π_t^v:

$$\begin{aligned} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v, \pi_t^v) = p(x_{t+1}^v|x_t^v, a_t^v) p(z_t^v|x_t^v) \\ p(a_t^v|x_t^v, z_t^v, \pi_t^v) p(\pi_t^v|x_t, \mathbf{z_{0:t}}) p(x_t^v). \end{aligned} \tag{6}$$

Finally, since we have full authority over the policy executed by our controlled car $q \in V$, we can separate our vehicle from the other agents in $p(x_{t+1})$ as follows:

$$\begin{aligned} p(x_{t+1}) \approx & \iint_{\mathcal{X}^q\,\mathcal{Z}^q} p^q(x_t^q, x_{t+1}^q, z_t^q, a_t^q, \pi_t^q)\, dz_t^q\, dx_t^q \\ & \prod_{v \in V | v \neq q} \left[\sum_{\Pi} \iint_{\mathcal{X}^v\,\mathcal{Z}^v} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v, \pi_t^v)\, dz_t^v\, dx_t^v\right]. \end{aligned} \tag{7}$$

We have thus far factored out the action space from $p(x_{t+1})$ by assuming actions are given by the available policies. However, Eq. 7 still requires integration over the state and observation spaces. Our second approximation addresses this issue. Given samples from $p(\pi_t^v|x_t, \mathbf{z_{0:t}})$ that assign a policy to each vehicle, we simulate forward in time the interactions of our vehicle and other vehicles under their assigned policies, and obtain a corresponding sequence of future states and observations. We are thereby able to evaluate the reward function over the entire decision horizon.

IV. Behavioral Analysis and Prediction via Changepoint Detection

In this section, we describe how we infer the probability of the policies executed by other cars and their parameters. Our behavioral anticipation method is based on a segmentation of the history of observed states of each vehicle, where each segment is associated with the policy most likely to have generated the observations in the segment. We obtain this segmentation using Bayesian changepoint detection, which infers the points in the history of observations where the underlying policy generating the observations changes. Thereby, we can compute the likelihood of all available policies for the target car given the observations in the most recent segment, capturing the distribution $p(\pi_t^v | x_t, \mathbf{z_{0:t}})$ over the car's potential policies at the current timestep. Further, full history segmentation allows us to detect anomalous behavior that is not explained by the set of policies in our system. The changepoint-detection procedure is illustrated by the simulation in Fig. 2. We next describe the anticipation method for a single vehicle, which we then apply successively to all nearby vehicles.

A. Changepoint Detection

To segment a target car's history of observed states, we adopt the recently proposed CHAMP algorithm by Niekum et al. [30], which builds upon the work of Fearnhead and Liu [15]. Given the set of available policies Π and a time series of the observed states of a given vehicle $\mathbf{z_{1:n}} = (z_1, z_2, \ldots, z_n)$, CHAMP infers the MAP set of times $\tau_1, \tau_2, \ldots, \tau_m$, at which changepoints between policies have occurred, yielding $m+1$ segments. Thus, the i^{th} segment consists of observations $\mathbf{z}_{\tau_i+\mathbf{1}:\tau_{i+1}}$ and has an associated policy $\pi_i \in \Pi$ with parameters θ_i.

The changepoint positions are modeled as a Markov chain where the transition probabilites are a function of the time since the last changepoint:

$$p(\tau_{i+1} = t | \tau_i = s) = g(t-s), \tag{8}$$

where $g(\cdot)$ is a pdf over time, and $G(\cdot)$ denotes its cdf.

Given a segment from time s to t and a policy π, CHAMP approximates the logarithm of the policy evidence for that segment via the Bayesian information criterion (BIC) [4] as:

$$\log L(s,t,\pi) \approx \log p(\mathbf{z_{s+1:t}} | \pi, \hat{\theta}) - \frac{1}{2} k_\pi \log(t-s), \tag{9}$$

where k_π is the number of parameters of policy π and $\hat{\theta}$ are estimated parameters for policy π. The BIC is a well-known approximation that avoids marginalizing over the policy parameters and provides a principled penalty against complex policies by assuming a Gaussian posterior around the estimated parameters $\hat{\theta}$. Thus, only the ability to fit policies to the observed data is required, which can be achieved via a maximum likelihood estimation (MLE) method of choice (we elaborate on this in §IV-B).

As shown by Fearnhead and Liu [15], the distribution C_t over the position of the first changepoint before time t can be estimated efficiently using standard Bayesian filtering and an online Viterbi algorithm. Defining

$$P_t(j,q) = p(C_t = j, q, \mathcal{E}_j, \mathbf{z_{1:t}}) \tag{10}$$

$$P_t^{\text{MAP}} = p(\text{Changepoint at } t, \mathcal{E}_t, \mathbf{z_{1:t}}), \tag{11}$$

where $\mathcal{E}_j$ is the event that the MAP choice of changepoints has occurred prior to a given changepoint at time j, results in:

$$P_t(j,q) = (1 - G(t-j-1)) L(j,t,q) p(q) P_j^{\text{MAP}} \tag{12}$$

$$P_t^{\text{MAP}} = \max_{j,q} \left[\frac{g(t-j)}{1 - G(t-j-1)} P_t(j,q) \right]. \tag{13}$$

At any time, the most likely sequence of latent policies (called the Viterbi path) that results in the sequence of observations can be recovered by finding (j,q) that maximize P_t^{MAP}, and then repeating the maximization for P_j^{MAP}, successively until time zero is reached. Further details on this changepoint detection method are provided by Niekum et al. [30].

B. Behavioral Prediction

In contrast with other anticipation approaches in the literature which consider only the current state of the target vehicle and assign equal likelihood to all its potential intentions [16, 21, 22], here we compute the likelihood of each latent policy by leveraging changepoint detection on the history of observed vehicle states.

Consider the $(m+1)^{\text{th}}$ segment (the most recent), obtained via changepoint detection and consisting of observations $\mathbf{z}_{\tau_m+\mathbf{1:n}}$. The likelihood and parameters of each latent policy $\pi \in \Pi$ for the target vehicle given the present segment can be computed by solving the following MLE problem:

$$\forall \pi \in \Pi, \qquad \mathcal{L}(\pi) = \operatorname*{argmax}_{\theta} \log p(\mathbf{z}_{\tau_m+\mathbf{1:n}} | \pi, \theta). \tag{14}$$

Specifically, we assume $p(\mathbf{z}_{\tau_m+\mathbf{1:n}} | \pi, \theta)$ to be a multivariate Gaussian with mean at the trajectory $\psi^{\pi,\theta}$ obtained by simulating forward in time the execution of policy π under parameters θ from timestep $\tau_m + 1$:

$$p(\mathbf{z}_{\tau_m+\mathbf{1:n}} | \pi, \theta) = \mathcal{N}(\mathbf{z}_{\tau_m+\mathbf{1:n}}; \psi^{\pi,\theta}, \sigma I), \tag{15}$$

where σ is a nuisance parameter capturing modeling error and I is a suitable identity matrix (we discuss our forward simulation of policies further in §V-B). That is, Eq. 15 essentially measures the deviation of the observed states from those prescribed by the given policy. The policy likelihoods obtained via Eq. 14 capture the probability distribution over the possible policies that the observed vehicle might be executing at the current timestep, which can be represented, using delta functions, as a mixture distribution:

$$p(\pi_t^v | x_t, \mathbf{z_{0:t}}) = \eta \sum_{i=1}^{|\Pi|} \delta(\alpha_i) \cdot \mathcal{L}(\pi_i), \tag{16}$$

where α_i is the hypothesis over policy π_i and η is a normalizing constant. We can therefore compute the approximated posterior of Eq. 7 by sampling from this distribution for each vehicle, obtaining high-likelihood samples from the coupled interactions of traffic agents.

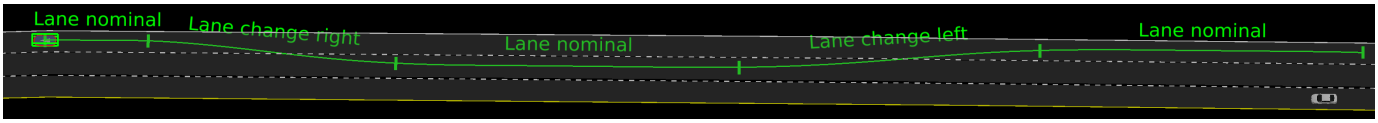


Fig. 2. Policy changepoint detection on a simulated passing maneuver on a highway. Our vehicle (far right) tracks the behavior of another traffic agent (far left) as it navigates through the highway segment from right to left. Using the tracked vehicle's history of past observations (green curve), we are able to infer which policies are most likely to have generated the maneuvers of the tracked vehicle.

C. Anomaly Detection

The time-series segmentation obtained via changepoint detection allows us to perform online detection of anomalous behavior not modeled by our policies. Inspired by prior work on anomaly detection [9, 25, 34], we first define the properties of anomalous behavior in terms of policy likelihoods, and then compare the observed data against labeled normal patterns in previously-recorded vehicle trajectories. Thus, we define the following two criteria for anomalous behavior:

1) Unlikelihood against available policies. Anomalous behavior is not likely to be explained by any of the available policies, since they are designed to abide by traffic rules and provide a smooth riding experience. Therefore, behaviors like driving in the wrong direction or crossing a solid line on the highway will not be captured by the available policies. We thus measure the average likelihood among all segments in the vehicle's history as the global similarity of the observed history to all available policies:

$$\mathcal{S} = \frac{1}{m+1} \sum_{i=1}^{m+1} \mathcal{L}(\pi_i), \tag{17}$$

where π_i is the policy associated with the i^{th} segment.

2) Ambiguity among policies. A history segmentation that fluctuates frequently among different policies might be a sign of ambiguity on the segmentation. To express this criterion formally, we first construct a histogram capturing the occurrences of each policy in the vehicle's segmented history. A histogram with a broad spread indicates frequent fluctuation, whereas one with a single mode is more likely to correspond to normal behavior. We measure this characteristic as the excess kurtosis of the histogram, $\kappa = \frac{\mu_4}{\sigma^4} - 3$, where μ_4 is the fourth moment of the mean and σ is the standard deviation. The excess kurtosis satisfies $-2 < \kappa < \infty$. If $\kappa = 0$, the histogram resembles a normal distribution, whereas if $\kappa < 0$, the histogram presents a broader spread. That is, we seek to identify changepoint sequences where there is no dominant policy.

Using these criteria, we define the following normality measure given a vehicle's MAP choice of changepoints:

$$N = \frac{1}{2}\left[(\kappa + 2)\mathcal{S}\right]. \tag{18}$$

This normality measure on the target car's history can then be compared to that of a set of previously recorded trajectories of other vehicles. We thus define the normality test for the current vehicle's history as $N < 0.5\gamma$, where γ is the minimum normality measure evaluated on the prior time-series.

V. Multipolicy Decision-Making

We now present the policy selection procedure for our car (Algorithm 1), which implements the formulation and approximations given in §III by leveraging the anticipation scheme from §IV. The algorithm begins by drawing a set of samples $s \in S$ from the distribution over policies of other cars via Eq. 16, where each sample assigns a policy $\pi^v \in \Pi$ to each nearby vehicle v, excluding our car. For each policy π available to our car and for each sample s, we roll out forward in time until the decision horizon H all vehicles under the policy assignments (π, s) with closed loop simulation to yield a set Ψ of simulated trajectories ψ. We then evaluate the reward $r_{\pi,s}$ for each rollout Ψ, and finally select the policy π^* maximizing the expected reward. The process continuously repeats in a receding horizon manner. Note that policies that are not applicable given the current state x_0, such as an intersection handling policy when driving on the highway, are not considered for selection (line 5). We next discuss three key points of our decision-making procedure: the design of the set of available policies, using forward simulation to roll out potential interactions, and the reward function.

Algorithm 1: Policy selection procedure.

Input:
- Current MAP estimate of the state, x_0.
- Set of available policies Π.
- Policy assignment probabilities (Eq. 16).
- Planning horizon H.

1 Draw a set of samples $s \in S$ via Eq. 16, where each sample assigns a policy to each nearby vehicle.
2 $\mathcal{R} \leftarrow \emptyset$ `// Rewards for each rollout`
3 **foreach** $\pi \in \Pi$ **do** `// Policies for our car`
4 **foreach** $s \in S$ **do** `// Policies for other cars`
5 **if** APPLICABLE(π, x_0) **then**
6 $\Psi^{\pi,s} \leftarrow$ SIMULATEFORWARD(x_0, π, s, H) `// $\Psi^{\pi,s}$ captures all vehicles`
7 $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\pi, s,$ COMPUTEREWARD$(\Psi^{\pi,s}))\}$
8 **return** $\pi^* \leftarrow$ SELECTBEST$(\mathcal{R})$

A. Policy Design

There are many possible design choices for engineering the set of available policies in our approach, which we wish to explore in future work. However, in this work we use a set

of policies that covers many in-lane and intersection driving situations, comprising the following policies: *lane-nominal*, drive in the current lane and maintain distance to the car directly in front; *lane-change-right/lane-change-left*, separate policies for a single lane change in each direction; and *turn-right*, *turn-left*, *go-straight*, or *yield* at an intersection.

B. Sample Rollout via Forward Simulation

While it is possible to perform high-fidelity simulation for rolling out sampled policy assignments, a lower-fidelity simulation can capture the necessary interactions between vehicles to make reasonable choices for our vehicle behavior, while providing faster performance. In practice, we use a simplified simulation model for each vehicle that assumes an idealized steering controller. Nonetheless, this simplification still faithfully describes the high-level behavior of the between-vehicle interactions our method reasons about. For vehicles classified as anomalous, we simulate them using a single policy accounting only for their current state and map of the environment, since they are not likely to be modeled by the set of behaviors in our system.

C. Reward Function

The reward function for evaluating the outcome of a rollout Ψ involving all vehicles is a weighted combination of metrics $m_q(\cdot) \in \mathcal{M}$, with weights w_q that express user importance. The construction of a reward function based on a flexible set of metrics derives from our previous work [11], which we extend here to handle multiple potential policies for other vehicles. In our system, typical metrics include the distance to the goal at the end of the evaluation horizon as a measure of accomplishment, minimum distance to obstacles to evaluate safety, a lane choice bias to add a preference for the right lane, and the maximum yaw rate and longitudinal jerk to measure passenger comfort. For a full policy assignment (π, s) with rollout $\Psi^{\pi,s}$, we compute the rollout reward $r_{\pi,s}$ as the weighted sum $r_{\pi,s} = \sum_{q=1}^{|\mathcal{M}|} w_q m_q(\Psi^{\pi,s})$. We normalize each $m_q(\Psi_{\pi,s})$ across all rollouts to ensure comparability between metrics. To avoid biasing decisions, we set the weight w_q to zero when the range of $m_q(\cdot)$ across all samples is too small to be informative.

We finally evaluate each policy reward r_π for our vehicle as the expected reward over all rollout rewards $r_{\pi,s}$, computed as $r_\pi = \sum_{k=1}^{|S|} r_{\pi,s_k} p(s_k)$, where $p(s_k)$ is the joint probability of the policy assignments in sample s_k, computed as a product of the per-vehicle assignment probabilities (Eq. 16). We use expected reward to target better average-case performance, as it is easy to become overly conservative when negotiating traffic if one only accounts for worst-case behavior. By weighting by the probability of each sample, we can avoid overcorrecting for low-probability events.

VI. Results

To evaluate our behavioral anticipation method and our multipolicy sampling strategy, we use traffic-tracking data collected using our autonomous vehicle platform. We first introduce the traffic-tracking dataset and the vehicle used to collect it. Next, we use this dataset to evaluate our prediction and anomaly detection method and the performance of our multipolicy sampling strategy. Finally, we evaluate our multipolicy approach performing integrated behavioral analysis and decision-making on highway traffic scenarios using our multivehicle simulation engine.

A. Autonomous Vehicle Platform, Dataset, and Setup

To collect the traffic-tracking dataset we use in this work, we have used our autonomous vehicle platform (shown in Fig. 3), a 2013 Ford Fusion equipped with a sensor suite including four Velodyne HDL-32E 3D LIDAR scanners, an Applanix POS-LV 420 inertial navigation system (INS), GPS, and several other sensors.

Fig. 3. Our autonomous car platform, used to record the traffic-tracking dataset we use in this work. The vehicle is equipped with a sensor suite including four LIDAR units and survey-grade INS.

The vehicle uses prior maps of the area it operates on that capture information about the environment such as LIDAR reflectivity and road height, and are used for localization and tracking of other agents. The road network is encoded as a metric-topological map that provides information about the location and connectivity of road segments, and lanes therein.

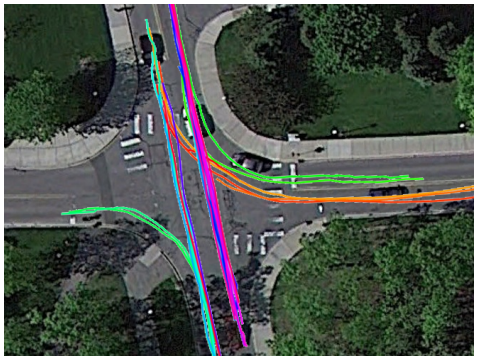
Estimates over the states of other traffic participants are provided by a dynamic object tracker running on the vehicle, which uses LIDAR range measurements. The geometry and location of static obstacles are also inferred onboard using LIDAR measurements.

The traffic-tracking dataset consists of 67 dynamic object trajectories recorded in an urban area. Of these 67 trajectories (shown in Fig. 4), 18 correspond to "follow the lane" maneuvers and 20 to lane change maneuvers, recorded on a divided highway. The remaining 29 trajectories correspond to maneuvers observed at a four-way intersection regulated by stop signs. All trajectories were recorded by the dynamic object tracker onboard the vehicle and extracted from approximately 3.5 h of total tracking data.

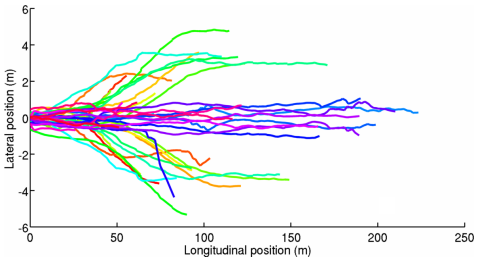
In all experiments we use a C implementation of our system running on a single 2.8GHz Intel i7 laptop computer.

B. Behavioral Prediction

For our system, we are interested in correctly identifying the behavior of target vehicles by associating it to the most likely policy according to the observations. Thus, we evaluate



(a)



(b)

Fig. 4. Trajectories in the traffic-tracking dataset used to evaluate our multipolicy framework. (a) 29 trajectories recorded at a four-way intersection. (b) 38 trajectories comprising lane change and "follow the lane" maneuvers on a divided highway, plotted on a common frame of reference.

our behavioral analysis method in the context of a classification problem, where we want to map each trajectory to the underlying policy (class) that is generating it at the current timestep. The available policies used in this evaluation are:

$$\Pi = \{\text{lane-nominal}, \text{lane-change-left}, \text{lane-change-right}\} \cup \{\text{turn-right}, \text{turn-left}, \text{go-straight}, \text{yield}\}, \tag{19}$$

where the first subset applies to in-lane maneuvers and the second subset applies to intersection maneuvers. For all policies we use a fixed set of parameters tuned empirically to control our autonomous vehicle platform, including maximum longitudinal and lateral accelerations, and allowed distances to nearby cars, among other parameters.

To assess each classification as correct or incorrect, we leverage the road network map and compare the final lane where the trajectory actually ends to that predicted by the declared policy. In addition, we assess behavioral prediction performance on subsequences of incremental duration of the input trajectory, measuring classification performance on increasingly longer observation sequences.

Fig. 5 shows the accuracy and precision curves for policy classification over the entire dataset. The ambiguity among hypotheses results in poor performance when only an early stage of the trajectories is used, especially under 30% completion. However, we are able to classify the trajectories with over 85% accuracy and precision after only 50% of the trajectory has been completed. Note, however, that the closed-loop nature of our policies allows us to maintain safety at all times regardless of anticipation performance.

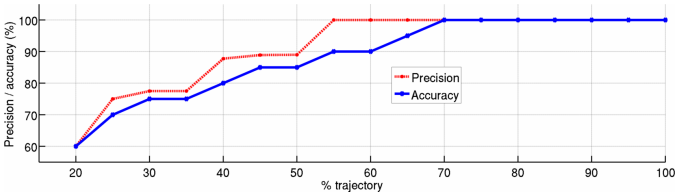
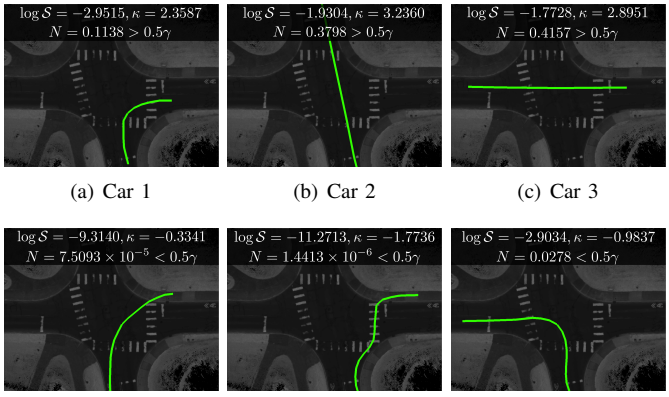


Fig. 5. Precision and accuracy curves of current policy identification via changepoint detection, evaluated at increasing subsequences of the trajectories. Our method provides over 85% accuracy and precision after only 50% of trajectory completion, while the closed loop nature of our policies guarantee safety at all times regardless of anticipation performance.

C. Anomaly Detection

We now qualitatively explore the performance of our anomaly detection test. We recorded three additional trajectories corresponding to two bikes and a bus. The bikes crossed the intersection from the sidewalk, while the bus made a significantly wide turn. We run the test on these trajectories and on three additional intersection trajectories using the minimum normality value on the intersection portion of the dataset, $\gamma = 0.1233$. As shown by the results in Fig. 6, our test is able to correctly detect the anomalous behaviors not modeled in our system.



(a) Car 1 (b) Car 2 (c) Car 3

(d) Bike 1 (e) Bike 2 (f) Bus

Fig. 6. Anomaly detection examples. Top row: normal trajectories driven by cars from the intersection dataset. Bottom row: anomalous trajectories driven by bikes (d), (e), and a bus (f). Our test is able to correctly detect the anomalous trajectories not modeled by our intersection policies ($\gamma = 0.1233$).

D. Multipolicy Sampling Performance

To show that our approach makes decision-making tractable, we assess the sampling performance in terms of the likelihood of the samples using the recorded intersection trajectories. We compare our multipolicy sampling strategy to an uninformed sampling strategy such as those used by general decision-making algorithms that do not account for domain knowledge to focus sampling (e.g., Silver and Veness [35], Thrun [37]).

We take groups of coupled trajectories from the dataset involving from one to four vehicles negotiating the intersection simultaneously. For each vehicle in each group, we compute, via Eq. 15, the likelihood of the most likely policy π^{ML} in $\{\text{turn-right}, \text{turn-left}, \text{go-straight}, \text{yield}\}$ according to the corresponding trajectory in the group. We then evaluate the computation time required by each of the two sampling strategies to find a sampled trajectory with a likelihood equal or greater than $\mathcal{L}(\pi^{\text{ML}})$.

The uninformed strategy generates, for each vehicle involved, a trajectory that either remains static for the duration of the trajectory to yield or crosses the intersection at constant speed. This decision is made at random. If the decision is to cross, the direction of the vehicle is determined via random steering wheel angle rates in a simple car kinematic model. Conversely, the multipolicy sampling strategy consists of randomly selecting policies for each vehicle and obtaining their rollouts. The computation times for each strategy are shown in Table I. Times are computed out of 100 simulations for each case (from one to four cars). Although the time required grows dramatically fast for both strategies due to the combinatorial explosion of vehicle intentions, these results show that our multipolicy sampling strategy is able to find high-likelihood samples orders of magnitude faster than an uninformed sampling strategy. A visualization of a sample simulation of this experiment is shown in Fig. 1.

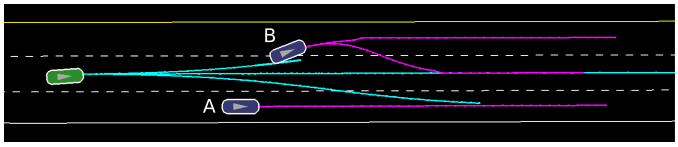
TABLE I
COMPARISON OF SAMPLING STRATEGIES.

STRATEGY	NUM. CARS	AVG. COMP. TIME	STD. DEVIATION
Uninformed	1	15.3990 s	9.1014 s
Multipolicy		0.0012 s	0.0004 s
Uninformed	2	39.6037 s	24.4575 s
Multipolicy		0.0036 s	0.0014 s
Uninformed	3	99.5785 s	76.3222 s
Multipolicy		0.0100 s	0.0050 s
Uninformed	4	296.9633 s	232.5125 s
Multipolicy		0.0247 s	0.0142 s

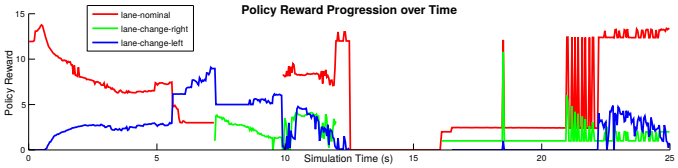
E. Decision-Making Results

We tested the full decision-making algorithm with behavioral prediction in a simulated environment with a multi-lane highway scenario involving two nearby cars. Fig. 7(a) shows the scenario used for testing at an illustrative point at half way through the scenario. This simulation uses the same policy models we have developed and tested on our real-world test car [11]. Fig. 7(b) shows the policy reward function, in which the chosen policy is the maximum of the available policies. Note that this decision process is instantaneous, which explains the oscillations when policies are near decision surfaces. We prevent the executed policy from oscillating with a simple pre-emption model that ensures we only switch policies when distinct maneuvers (such as lane-changes) are complete.

We collected timing information on different operations in the experiment to evaluate runtime performance. The main expense is forward simulation and metric evaluation for each



(a)



(b)

Fig. 7. (a) Results of a simulated multi-car interaction scenario, in which the car under our control (shown in green) approaches the slower vehicles A and B from behind. Vehicle B starts by executing a lane change from the center to left lane, which it is just completing at the time shown, while A remains in the right lane. Cyan lines show the simulated rollouts for our vehicle, while magenta lines show the simulated rollouts for each of the other vehicles. (b) Evaluation of the policy reward functions for each of the three policies over the course of the simulated scenario. Note that not all policies are applicable at all times, which we render as a discontinuity.

rollout, however, these tasks are easily parallelizable. In the test scenario in which we rollout all sample permutations, the theoretical maximum number of rollouts is 27 given 3 policy options per vehicle, but in practice the maximum number of rollouts was 12, with a mean of 8.6. This smaller number of rollouts is because not all policies are applicable at once. Parallel evaluation performance is bounded by the maximum time for a single rollout, for which the mean worst time was 84ms, and the worst time over the whole experiment was 106ms. Even in the worst case, our real-time decision-making target of 1 Hz is acheiveable.

VII. CONCLUSION

We introduced a principled framework for integrated behavioral anticipation and decision-making in environments with extensively coupled interactions between agents. By explicitly modeling reasonable behaviors of both our vehicle and other vehicles as policies, we make informed high-level behavioral decisions that account for the consequences of our actions.

We presented a behavior analysis and anticipation system based on Bayesian changepoint detection that infers the likelihood of policies of other vehicles. Furthermore, we provided a normality test to detect unexpected behavior of other traffic participants. We have shown that our behavioral anticipation approach can identify the most-likely underlying policies that explain the observed behavior of other cars, and to detect anomalous behavior not modeled by the policies in our system.

In future work we will explicitly model unexpected behavior, such as the appearance of a pedestrian or vehicles occluded by large objects. We can also extend the system to scale to larger environments by strategically sampling policies to focus on those outcomes that most affect our choices. Exploring principled methods for reacting to detected anomalous behavior is also an avenue for future work.

ACKNOWLEDGMENTS

This work was supported in part by a grant from Ford Motor Company via the Ford-UM Alliance under award N015392 and in part by DARPA under award D13AP00059.

The authors are sincerely grateful to Patrick Carmody for his help in collecting the traffic-tracking data used in this work and to Ryan Wolcott for his helpful comments.

REFERENCES

[1] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Auton. Robot.*, 35(1):51–76, 2013.

[2] H. Bai, D. Hsu, and W. S. Lee. Integrated perception and planning in the continuous space: A POMDP approach. *Int. J. Robot. Res.*, 33(9):1288–1302, 2014.

[3] T. Bandyopadhyay, K. Won, E. Frazzoli, D. Hsu, W. Lee, and D. Rus. Intention-aware motion planning. In E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, editors, *Proc. Int. Work. Alg. Foundation of Robotics*, volume 86 of *Springer Tracts in Advanced Robotics*, pages 475–491. Springer Berlin Heidelberg, 2013.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2007.

[5] S. Brechtel, T. Gindele, and R. Dillmann. Solving continuous pomdps: Value iteration with incremental learning of an efficient space representation. In S. Dasgupta and D. Mcallester, editors, *Proc. Int. Conf. Machine Learning*, pages 370–378, Atlanta, GA, USA, May 2013.

[6] S. Brechtel, T. Gindele, and R. Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. In *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, pages 392–399, Qingdao, China, Oct. 2014. doi: 10.1109/ITSC.2014.6957722.

[7] A. Broadhurst, S. Baker, and T. Kanade. Monte carlo road safety reasoning. In *Proc. IEEE Intell. Veh. Symp.*, pages 319–324, Las Vegas, NV, USA, June 2005.

[8] S. Candido, J. Davidson, and S. Hutchinson. Exploiting domain knowledge in planning for uncertain robot systems modeled as pomdps. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 3596–3603, Anchorage, AK, USA, May 2010.

[9] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3): 15, 2009.

[10] J. Choi, G. Eoh, J. Kim, Y. Yoon, J. Park, and B.-H. Lee. Analytic collision anticipation technology considering agents' future behavior. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 1656–1661, Taipei, Taiwan, Oct. 2010.

[11] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson. MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving. In *Proc. IEEE Int. Conf. Robot. and Automation*, Seattle, WA, USA, May 2015.

[12] DARPA. DARPA Urban Challenge. http://archive.darpa.mil/grandchallenge/, 2007.

[13] N. Du Toit and J. Burdick. Robotic motion planning in dynamic, cluttered, uncertain environments. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 966–973, Anchorage, AK, USA, May 2010.

[14] N. E. Du Toit and J. W. Burdick. Robot motion planning in dynamic, uncertain environments. *IEEE Trans. Robot.*, 28(1):101–115, 2012.

[15] P. Fearnhead and Z. Liu. On-line inference for multiple changepoint problems. *J. Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.

[16] D. Ferguson, M. Darms, C. Urmson, and S. Kolski. Detection, prediction, and avoidance of dynamic obstacles in urban environments. In *Proc. IEEE Intell. Veh. Symp.*, pages 1149–1154, Eindhoven, Netherlands, June 2008.

[17] D. Ferguson, T. M. Howard, and M. Likhachev. Motion planning in urban environments. *J. Field Robot.*, 25(11-12):939–960, 2008.

[18] 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. Intell. Robots and Syst.*, pages 1056–1062, Nice, France, Sept. 2008.

[19] T. Gindele, S. Brechtel, and R. Dillmann. A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments. In *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, pages 1625–1631, Madeira Island, Portugal, Sept. 2010. doi: 10.1109/ITSC.2010.5625262.

[20] T. Gindele, S. Brechtel, and R. Dillmann. Learning context sensitive behavior models from observations for predicting traffic situations. In *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, pages 1764–1771, The Hague, The Netherlands, Oct. 2013. doi: 10.1109/ITSC.2013.6728484.

[21] J. Hardy and M. Campbell. Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. *IEEE Trans. Robot.*, 29(4):913–929, 2013.

[22] F. Havlak and M. Campbell. Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments. *IEEE Trans. Robot.*, 30(2):461–474, 2014.

[23] R. He, E. Brunskill, and N. Roy. Efficient planning under uncertainty with macro-actions. *J. Artif. Intell. Res.*, 40: 523–570, 2011.

[24] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy. A Bayesian nonparametric approach to modeling motion patterns. *Auton. Robot.*, 31(4):383–400, 2011.

[25] K. Kim, D. Lee, and I. Essa. Gaussian process regression flow for analysis of motion trajectories. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1164–1171, Barcelona, Spain, Nov. 2011.

[26] H. Kurniawati, D. Hsu, and W. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. Robot.: Sci. & Syst. Conf.*, Zurich, Switzerland, June 2008.

[27] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and related stochastic

optimization problems. *Artificial Intelligence*, 147(1–2): 5–34, 2003.

[28] I. Miller et al. Team Cornell's Skynet: Robust perception and planning in an urban environment. *J. Field Robot.*, 25(8):493–527, 2008.

[29] M. Montemerlo et al. Junior: The Stanford entry in the Urban Challenge. *J. Field Robot.*, 25(9):569–597, 2008.

[30] S. Niekum, S. Osentoski, C. G. Atkeson, and A. G. Barto. CHAMP: Changepoint detection using approximate model parameters. Technical Report CMU-RI-TR-14-10, Robotics Institute, Carnegie Mellon University, 2014.

[31] T. Ohki, K. Nagatani, and K. Yoshida. Collision avoidance method for mobile robot considering motion and personal spaces of evacuees. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 1819–1824, Taipei, Taiwan, Oct. 2010.

[32] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[33] S. Petti and T. Fraichard. Safe motion planning in dynamic environments. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 2210–2215, Edmonton, AB, Canada, Aug. 2005.

[34] C. Piciarelli and G. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, 27(15):1835–1842, 2006.

[35] D. Silver and J. Veness. Monte-carlo planning in large POMDPs. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2164–2172. Curran Associates, Inc., 2010.

[36] A. Somani, N. Ye, D. Hsu, and W. S. Lee. DESPOT: Online POMDP planning with regularization. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1772–1780. Curran Associates, Inc., 2013.

[37] S. Thrun. Monte Carlo POMDPs. *Proc. Advances Neural Inform. Process. Syst. Conf.*, pages 1064–1070, 2000.

[38] Q. Tran and J. Firl. Modelling of traffic situations at urban intersections with probabilistic non-parametric regression. In *Proc. IEEE Intell. Veh. Symp.*, pages 334–339, Gold Coast City, Australia, June 2013.

[39] Q. Tran and J. Firl. Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. In *Proc. IEEE Intell. Veh. Symp.*, pages 918–923, Dearborn, MI, USA, June 2014.

[40] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 797–803, Taipei, Taiwan, Oct. 2010.

[41] C. Urmson et al. Autonomous driving in urban environments: Boss and the Urban Challenge. *J. Field Robot.*, 25(8):425–466, 2008.

[42] W. Xu, J. Wei, J. Dolan, H. Zhao, and H. Zha. A real-time motion planner with trajectory optimization for autonomous vehicles. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 2061–2067, Saint Paul, MN, USA, May 2012.