

High-Speed Highway Scene Prediction Based on Driver Models Learned From Demonstrations

David Sierra González, Jilles Steeve Dibangoye, Christian Laugier

▶ **To cite this version:**

David Sierra González, Jilles Steeve Dibangoye, Christian Laugier. High-Speed Highway Scene Prediction Based on Driver Models Learned From Demonstrations. Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC 2016), Nov 2016, Rio de Janeiro, Brazil. 10.1109/ITSC.2016.7795546 . hal-01396047

HAL Id: hal-01396047

https://inria.hal.science/hal-01396047

Submitted on 13 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

High-Speed Highway Scene Prediction Based on Driver Models Learned From Demonstrations

David Sierra González, Jilles Steeve Dibangoye, Christian Laugier

***Abstract*— One of the key factors to ensure the safe operation of autonomous and semi-autonomous vehicles in dynamic environments is the ability to accurately predict the motion of the dynamic obstacles in the scene. In this work, we show how to use a realistic driver model learned from demonstrations via Inverse Reinforcement Learning to predict the long-term evolution of highway traffic scenes. We model each traffic participant as a Markov Decision Process in which the cost function is a linear combination of static and dynamic features. In particular, the static features capture the preferences of the driver while the dynamic features, which change over time depending on the actions of the other traffic participants, capture the driver's risk-aversive behavior. Using such a model for prediction enables us to explicitly consider the interactions between traffic participants while keeping the computational complexity quadratic in the number of vehicles in the scene. Preliminary experiments in simulated and real scenarios show the capability of our approach to produce reliable, human-like scene predictions.**

I. INTRODUCTION

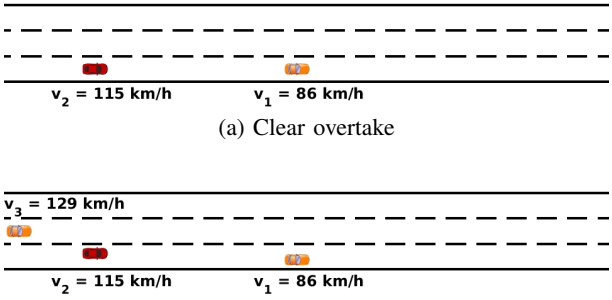
Recent years have seen a significant increase in research on Advanced Driver Assistance Systems (ADAS). Especially, autonomous driving has become a critical concern around the world. Whether it is to increase safety and security of drivers, passengers or pedestrians; enhance mobility for people and freights; or protect the environment. However, techniques to predict accurately the trajectories and intents of the involved traffic participants are time and memory demanding, preventing us from safe and secure autonomous driving systems [1].

Instead of exhaustively estimating the intent of all road users [2], one can rely on planning based prediction approaches to overcome the computational complexity [3], [4], [5]. Planning-based approaches assume that people, when they move, they do so by minimizing a cost function which depends on their preferences and the context. Thus, given the cost function and the context, their future actions can be predicted. Unfortunately, most of the work in this area is limited to motion prediction for pedestrians in which the cost function is time-invariant and models the behavior of an agent in a static environment (i.e. an environment in which the obstacles do not move).

In contrast, the highway is a dynamic environment in which the actions of a driver are strongly conditioned by the states of the surrounding vehicles. Predicting the future becomes then a difficult task, as it involves predicting the actions of all agents in the scene while taking into account the interactions between them.

This work has been supported by Toyota Motor Europe.

The authors are with INRIA Grenoble Rhone-Alpes, 38330 Montbonnot-Saint-Martin, France - {david.sierra-gonzalez, jilles.dibangoye, christian.laugier}@inria.fr



(a) Clear overtake

(b) Delayed overtake

Fig. 1: Two traffic situations commonly found in highway driving.

Despite the complexity of the task, humans are extremely good at predicting the future intents of other traffic participants. Moreover, we do this instinctively, without thinking too much about it. We know that their actions are constrained by the road network, by the traffic rules, and by a risk-aversive common sense. As an example of this, we can consider the situations shown in Fig. 1. When a driver approaches the vehicle in front at a high relative speed, and there is no obstacle preventing him from switching lanes, we can predict with high probability that he will be indeed switching lanes (Fig. 1a). However, if this maneuver carries a high risk of collision due to the presence of an obstacle (Fig. 1b), the most likely response of the driver will be a braking maneuver. In this work, we model this "not so complex" behavior of drivers with a cost function that is a linear combination of (a) static features, which model the preferences of the driver (e.g. how fast to drive); and (b) dynamic features, which model the risk-aversive behavior of the driver and take into account the dynamic environment (e.g. the time to reach the closest vehicle in front at the current speed).

The problem of learning a cost function that encodes the behavior of an agent is not new in robotics. Often, this cost shaping is done manually, balancing the multiple trade-offs by trial and error. However, such a technique gets in trouble as the complexity of the model increases. Inverse Reinforcement Learning (IRL) algorithms enable us to obtain the cost function that encodes the behavior of an agent from a set of demonstrations in a systematic and mathematically consistent way [6].

In this paper, we use IRL to learn the cost function of a regular highway driver from a set of demonstrated highway

driving scenes. We use such cost function to predict the behavior of each traffic participant and thus, the long-term evolution of highway traffic scenes. The main contributions of the paper are: (1) A novel highway scene prediction framework that takes into account the interactions between traffic participants through the use of a cost function with static and dynamic features; (2) A procedure that exploits the constraints of the highway environment to sequentially predict the cost induced by the dynamic features for each traffic participant in future timesteps; and (3) Preliminary results in simulated and real-world scenarios that validate our approach and show the potential of driver models learned from demonstrated driving tasks to capture and predict the interactions between traffic participants.

The rest of the document is organized as follows: Section II presents the current state of research in the fields of motion and scene prediction for intelligent vehicles; Section III explains the theoretical background of the work; Section IV describes the proposed approach; Section V presents the different test scenarios and shows the experimental results. Lastly, Section VI concludes.

II. RELATED WORK

In this work, we focus on long-term highway scene prediction. Such a task requires necessarily to take into consideration the interactions between vehicles. An interaction-aware approach to predict the development of traffic scenes is presented by Lawitzky et al. [2]. They assume a set of discrete maneuvers for each vehicle and calculate the collision probabilities associated to each maneuver by explicitly considering all possible interactions between vehicles. This leads to exponential complexity in the number of vehicles in the scene.

A more refined approach is presented by Schwarting and Pascheka [7]; they assume that not all dependencies between the vehicles in the scene are equally strong, and consider that a vehicle's decision is dominated by its predecessors'. Initially, egoistic decisions are predicted for all vehicles in the scene. If this results in a conflict, they examine all possible maneuver combinations for each pair of vehicles involved in order to find the best cooperative behavior according to a hand-tuned cost function. In contrast to this work, we do not assume any cooperative behavior but rather a risk-averse behavior for all traffic participants.

Dynamic Bayesian Networks (DBNs) have also been used to address the problem of prediction in traffic situations. They are a convenient approach to avoid the exhaustive enumeration of other approaches by exploiting dependencies among vehicles. In the work of Gindele et al. [8], each vehicle is represented within a DBN by a set of nodes representing its internal states and the relationships with the other vehicles and the infrastructure. The policies are learned from data using random forests. It is not obvious how the policy model can guarantee reasonable predictions when presented with new traffic scenes not included in the training set.

Planning-based motion prediction is an intuitive, sound framework that predicts human behavior by incorporating all the different variables relevant to a task. It is based on two assumptions: (1) The relationships between human agents performing a task and the environment can be modeled in a cost function; and (2) When people perform the task (e.g. walking in the city, driving) they minimize such cost function. Regarding the first assumption, IRL has already been shown as a viable approach to learn the cost function that describes the highway driving task [9], [10]. Regarding the second assumption, planning-based approaches have been successfully applied to pedestrian motion prediction [3], [4], [5]. However, these approaches assume that the pedestrians move in static environments and that the cost function is time-invariant. Recently, Lee and Kitani applied a planning-based approach to predict the trajectories of an attacking player in American football [11]. They used a dynamic feature dependent on the motion of the defenders, which was predicted using supervised learning.

In this work, we bring the planning-based motion prediction framework to the intelligent vehicles domain. We propose a planning-based approach to predict the future actions of every traffic participant in high-speed highway traffic scenes. The interactions between drivers are explicitly considered through the dynamic features of a driver model learned using IRL. Such a model captures the common-sense risk-aversive behavior of a driver and it can thus induce reliable predictions even when presented with situations never seen before. The predictive potential of the approach is demonstrated on simulated and real-world scenarios.

III. BACKGROUND

This section presents the theoretical foundations upon which our approach is built. We use the Markov Decision Process framework to model the dynamics of each traffic participant in the scene. With IRL, we can obtain a model that captures the behavior of a driver from driving demonstrations.

A. Markov Decision Processes

Markov Decision Processes (MDPs) provide a framework for modeling sequential decision making problems. A finite MDP $(\mathcal{S}, \mathcal{A}, \{P_{sa}\}, \mathcal{C}, \gamma)$ is defined by:

- A finite set $\mathcal{S}$ of n states
- A finite set $\mathcal{A}$ of k actions
- A state transition probability $P_{sa}(\cdot)$ upon taking action a in state s.
- A cost function $\mathcal{C} : \mathcal{S} \mapsto \mathbb{R}$
- A discount factor $\gamma \in [0, 1)$

A policy is defined as any map $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The value function for a policy π, can be evaluated at any arbitrary state $s \in \mathcal{S}$ as follows:

$$V^{\pi}(s) = E[\sum_{t=0}^{\infty} \gamma^t \mathcal{C}(S_t) \mid \pi, S_0 = s]$$

Solving an MDP implies finding the policy π that minimizes V^π, that is, for any arbitrary state $s \in \mathcal{S}$ we have that:

$$\begin{aligned} V^*(s) &= \min_\pi \; V^\pi(s) \\ &= \mathcal{C}(s) + \min_{a \in \mathcal{A}} \; E_{s' \sim P_{sa}(\cdot)}[V^*(s')]. \end{aligned}$$

where notation $s' \sim P_{sa}(\cdot)$ means the expectation is with respect to s' distributed according to $P_{sa}(\cdot)$.

B. Inverse Reinforcement Learning

Inverse reinforcement learning algorithms [6] aim to find the cost function $\mathcal{C} : \mathcal{S} \mapsto \mathbb{R}$ that can explain observed behavior. In this work, the observed behavior will be a set of M demonstrated, possibly suboptimal, trajectories $\{\xi_1, \xi_2, \ldots, \xi_M\}$. A trajectory is defined as a sequence of states of a vehicle $\xi = \{s_1, \ldots, s_T\}$. The majority of IRL algorithms assume that the cost function can be fully specified as a linear combination of features:

$$\mathcal{C}(s) = \boldsymbol{\theta} \cdot \mathbf{f}(s) \tag{1}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ is the unknown weight vector and $\mathbf{f}(s) = (f_1(s), \ldots, f_K(s))$ is the feature vector that parameterizes state s, both of dimension K. The goal is to find the weight parameters for which, if we found the optimal policy by solving the underlying planning problem, we would obtain similar trajectories to those demonstrated according to a given statistic. Unfortunately, this is an ill-posed problem, as there are many different weights that make the demonstrated behavior optimal.

C. Maximum Entropy Inverse Reinforcement Learning

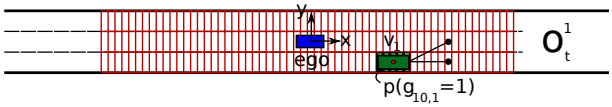
In this work, we use the Maximum Entropy IRL variant presented by Ziebart [12]. The similarity statistic of this approach is the sum of features. It resolves the ambiguity described above by applying the principle of maximum entropy by which the distribution over trajectories $P(\xi|\boldsymbol{\theta})$ that does not exhibit any additional preferences beyond matching the empirical sum of features is chosen. The solution has the form:

$$P(\xi_i|\boldsymbol{\theta}) \propto e^{\sum_{s_j \in \xi_i} -\boldsymbol{\theta}^T \mathbf{f}(s_j)} \tag{2}$$

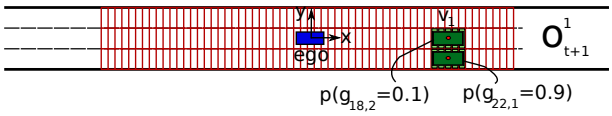
The probability of a certain trajectory ξ_i is defined to be proportional to the exponentiated weighted sum of features encountered over the trajectory. It can be shown that the gradient of the constrained optimization problem with respect to $\boldsymbol{\theta}$ is the difference between the empirical and expected sum of features. The cost weights can thus be calculated using gradient-based optimization methods. Further details can be found in [12].

D. IRL in dynamic environments

Classic IRL was not originally designed to work on dynamic environments, where the features can change over time. In order to adapt the IRL framework to our problem, we follow the procedure previously presented in [13]. In a nutshell, the solution involves training on a small window of H timesteps with the environment fixed. Then, successive re-training steps are performed by updating the state of the environment and shifting the window $t \leq H$ timesteps into the future. The solution is no longer guaranteed to be optimal but it is accepted as a good approximation [14].



(a) Occupancy distribution for vehicle $i = 1$ at timestep t. The red dot marks the center of mass of the vehicle's state.



(b) Occupancy distribution for vehicle $i = 1$ at timestep $t + 1$

Fig. 2: Evolution of the occupancy distribution of a vehicle over the cells of grid $\mathcal{G}$ in one timestep. The occupancy probabilities at timestep t + 1 will depend on the available actions in the previous timestep.

IV. APPROACH

We address the problem of predicting the development of highway traffic scenes. The proposed approach uses a driver model learned offline from demonstrations to predict the actions of all the vehicles in a highway traffic scene. The model is shaped as a linear feature-based cost function, which takes explicitly into account the interactions between vehicles. In this section, we formalize the problem and show how such a driver model can be used to determine a human-like policy model.

A. Discretization and notation

Our framework for motion prediction in high-speed highway scenes $(\mathcal{V}, \mathcal{G}, \{\mathcal{S}^i\}, \{\mathcal{A}^i\}, \{\mathcal{T}^i\}, \{S_0^i\})$ is given by:

- $\mathcal{V}$ a finite set of m vehicles involved in the traffic scene.
- A grid $\mathcal{G}$ centered in the position of the ego-vehicle.
- $\mathcal{S}^i$ a finite set of states for vehicle $i \in \{1, 2, \ldots, m\}$. Each vehicle i's state s^i is a tuple $((x^i, y^i), z^i, z^i_\bullet)$, where (x^i, y^i) denotes the position of the vehicle i in grid $\mathcal{G}$; z^i is its absolute speed, and $z^i_\bullet$ is the driver's desired speed.
- $\mathcal{A}^i$ a finite set of actions (or maneuvers) available to vehicle $i \in \{1, 2, \ldots, m\}$.
- $\mathcal{T}^i$: $\mathcal{S}^i \times \mathcal{A}^i \mapsto \mathcal{S}^i$ denotes the transition rule, which describes the next-state $s'^i = \mathcal{T}^i(s^i, a^i)$ of vehicle i after taking action a^i in state s^i.
- $\mathcal{S}_0^i$ the initial state for vehicle $i \in \{1, 2, \ldots, m\}$.

The cell (x^i, y^i) occupied by vehicle i in grid $\mathcal{G}$ can be uniquely identified given: a) the lane occupied by vehicle i; and b) the relative distance between the center of mass of vehicle i and that of the ego-vehicle (Fig. 2). In the tuple (x^i, y^i), x^i indicates the cell index along the longitudinal dimension of the road, and y^i indicates the lane index. We denote by r_x the longitudinal resolution of the grid.

B. Feature-based cost function in dynamic environments

For any state $s^i \in \mathcal{S}^i$, we can calculate a feature vector $\mathbf{f}(s^i)$ that describes the vehicle's state and its context. The highway is a dynamic environment and therefore we need to consider static and dynamic features. Static features are time-invariant and can be directly calculated given a state. Dynamic features are those that change over time in response to the actions of the dynamic agents in the environment. Consequently, the cost associated to a given state, which is a weighted linear combination of the features, will also change over time. We can rewrite equation 1 to explicitly reflect this:

$$\mathcal{C}_t^i(s^i) = \underbrace{\boldsymbol{\theta}_s^i \cdot \mathbf{f}_s(s^i)}_{\mathcal{C}_s^i(s^i)} + \underbrace{\boldsymbol{\theta}_d^i \cdot \mathbf{f}_{d,t}(s^i)}_{\mathcal{C}_{d,t}^i(s^i)} \tag{3}$$

where $\boldsymbol{\theta}_s^i$ and $\boldsymbol{\theta}_d^i$ are the components of the weight vector associated to the static and dynamic features respectively, $\mathbf{f}_s : \mathcal{S}^i \mapsto \mathbb{R}^{K_s}$ is the time-invariant vector of static features, and $\mathbf{f}_{d,t} : \mathcal{S}^i \times \mathbb{T} \mapsto \{0,1\}^{K_d}$ is the vector of dynamic features at the different timesteps $t \in \{0, 1, \ldots, T-1\}$.

C. Feature selection

In order for the model to capture the preferences of a highway driver we have selected the following static features:

- Lane: These features aim to capture the preference of an agent to drive on a particular lane. For a highway with n_l lanes, the vector of lane features consists of n_l mutually-exclusive binary features: $\mathbf{f}_{s1} \colon \mathcal{S}^i \mapsto \{0,1\}^{nl}$ for any vehicle $i \in \{1, \ldots, m\}$
- Speed deviation: This feature encodes the penalty of deviating from the agent's desired speed: $\mathbf{f}_{s2} \colon \mathcal{S}^i \mapsto \mathbb{R}$ for any vehicle $i \in \{1, \ldots, m\}$, where

  $$\mathbf{f}_{s2}(s^i) = z^i - z_\bullet^i$$

  for each state $s^i \in \mathcal{S}^i$. The preferred speed of a driver $z_\bullet^i$ is estimated as the maximum speed observed since the last change in the speed limit.

The dynamic features capture the risk-aversive behavior of drivers. We propose the following features:

- Time-headway: We define the time-headway as the time elapsed between the back of the lead vehicle passing a point and the front of the following vehicle passing the same point. It indicates potentially dangerous situations [15]. To calculate the time-headway we define function $\phi : \mathcal{S}^i \times \mathcal{S}^i \mapsto \mathbb{R}$ that calculates the time-headway value between any two vehicles $i, j \in \{1, \ldots, m\}$:

  $$\phi(s^i, s^j) = \begin{cases} \frac{(x^i - x^j)\ r_x}{z^j} & \text{if } x^i - x^j \geq 0 \\ \frac{(x^j - x^i)\ r_x}{z^i} & \text{otherwise} \end{cases}$$

  We consider two separate dynamic features: the time-headway to the closest car in front in the current lane $\mathbf{f}_{d_1,t} : \mathcal{S}^i \times \mathbb{T} \mapsto \{0,1\}^{n_{th}}$, and the time-headway to the closest trailing car in the current lane $\mathbf{f}_{d_2,t} : \mathcal{S}^i \times \mathbb{T} \mapsto \{0,1\}^{n_{th}}$. They are defined as:

  $$\mathbf{f}_{d_1,t}(s_t^i) = d\Big(\min_{\substack{j: \\ x_t^i - x_t^j \leq 0 \\ y^i = y^j}} \phi(s_t^i, s_t^j) \Big) \tag{4}$$

  $$\mathbf{f}_{d_2,t}(s_t^i) = d\Big(\min_{\substack{j: \\ x_t^i - x_t^j \geq 0 \\ y^i = y^j}} \phi(s_t^i, s_t^j) \Big) \tag{5}$$

  where s_t^i and s_t^j are the states of vehicles i and j at timestep t, and d is a discretization operator $d : \mathbb{R} \mapsto \{0,1\}^{n_{th}}$ that returns mutually exclusive binary vectors of dimension n_{th}. If there is no other vehicle at the front or at the back the time-headway is set to infinity.

D. Occupancy distribution

The generic driver model obtained via IRL enables us to predict what action is a driver likely to perform given his preferences and the current state of the environment. In order to keep track of the state of the environment in the current and future timesteps, we maintain a distribution over the cells of grid $\mathcal{G}$ for each vehicle. We shall call such distribution an *occupancy distribution of vehicle* i denoted $o_{t+1}^i \triangleq \chi(b_{t+1}^i)$ and given by:

$$o_{t+1}^i(x^i, y^i) = \sum_{z^i, z_\bullet^i} \sum_{s'^i = (x^i, y^i, z^i, z_\bullet^i)} b_{t+1}^i(s'^i) \tag{6}$$

where $b_{t+1}^i \triangleq \tau(b_t^i, \pi_t^i)$ is the probability distribution over the state space of vehicle i, namely the belief state of vehicle i and given by:

$$b_{t+1}^i(s'^i) = \sum_{s^i} \sum_{a^i} P(s'^i | s^i, a^i)\, \pi_t^i(a^i | s^i)\, b_t^i(s^i) \tag{7}$$

We assume deterministic transitions and thus the transition probability term $P(s'^i | s^i, a^i) = 1\{s'^i = \mathcal{T}^i(s^i, a^i)\}$ only takes values 0 or 1, where $1\{\cdot\}$ is an indicator function. The term $\pi_t^i(a^i | s^i)$ is the policy model.

E. Policy model

Using IRL, we have obtained a generic driver model that encodes the behavior of highway drivers. This model tells us how comfortable a driver feels in a given state. For example, if the learned model associates high costs to states in which the distance to the car in front is too small, which is usually an indicator of dangerous situations, the driver will tend to avoid taking actions that would lead him to such states. Thus, we propose the following heuristic as a policy model that captures the risk-aversive behavior of drivers:

$$\pi_t^i(a^i | s^i) \approx 1 - \frac{\mathcal{C}_t^i(\mathcal{T}^i(s^i, a^i))}{\sum_{a'^i \in \mathcal{A}^i} \mathcal{C}_t^i(\mathcal{T}^i(s^i, a'^i))} \tag{8}$$

The probability of taking one action will be low if it transitions the agent to a state s'^i in which the cost $\mathcal{C}_t^i(s'^i)$ is high and there are other actions available that would lead to states with a lower cost. In this work, we consider only a one timestep look-ahead.

F. Dynamic cost estimation

In order to forecast the development of a traffic scene, we need to predict the state transitions of each agent following (8). This implies calculating the cost associated to each state and by extension, the value of the dynamic features of the state, in the future. In subsection IV-C, we defined the dynamic features $\mathbf{f}_{d_1,t}(s_t^i)$ and $\mathbf{f}_{d_2,t}(s_t^i)$ for the deterministic case in which each vehicle occupies a single state. The calculation of the cost associated to the dynamic features (hereafter, dynamic cost) is then trivial:

$$\mathcal{C}^i_{d,t}(s^i) = \boldsymbol{\theta}^i_d \cdot \mathbf{f}_{d,t}(s^i) = \underbrace{\boldsymbol{\theta}^i_{d_1} \cdot \mathbf{f}_{d_1,t}(s^i)}_{\mathcal{C}^i_{d_1,t}(s^i)} + \underbrace{\boldsymbol{\theta}^i_{d_2} \cdot \mathbf{f}_{d_2,t}(s^i)}_{\mathcal{C}^i_{d_2,t}(s^i)}$$

where we have decomposed the dynamic cost into its components $\mathcal{C}^i_{d_1,t}(s^i)$ and $\mathcal{C}^i_{d_2,t}(s^i)$.

However, the concept of time-headway to the closest car is no longer defined in future timesteps where we have a state distribution for each vehicle in the scene. Instead, we traverse the occupancy distributions of all opposite vehicles calculating the dynamic costs associated to each possible location, and weighting them by the occupancy probability:

$$\varphi_{d1,t}(s_t^i) \triangleq \left\{ o_t^j(x^j, y^j) \;\; \boldsymbol{\theta}^i_{d_1} \cdot d\left(\phi(s_t^i, \hat{s}_t^j)\right) \right\} \quad (9)$$
$$\forall \hat{s}_t^j = ((x^j, y^j), z_t'^j, -) : j \neq i,\, y^j = y^i,\, x^j - x^i \geq 0$$

$$\varphi_{d2,t}(s_t^i) \triangleq \left\{ o_t^j(x^j, y^j) \;\; \boldsymbol{\theta}^i_{d_2} \cdot d\left(\phi(s_t^i, \hat{s}_t^j)\right) \right\} \quad (10)$$
$$\forall \hat{s}_t^j = ((x^j, y^j), z_t'^j, -) : j \neq i,\, y^j = y^i,\, x^j - x^i \leq 0$$

where in the state $\hat{s}_t^j$ induced by cell (x^j, y^j) we calculate the speed $z_t'^j$ as the weighted average of the states that contributed to the occupancy probability $o_t^j(x^j, y^j)$. The dynamic costs $\mathcal{C}^i_{d_1,t}(s^i)$ and $\mathcal{C}^i_{d_2,t}(s^i)$ are then set to be the maximum of the weighted costs obtained above, that is:

$$\mathcal{C}^i_{d_1,t}(s^i) = \|\varphi_{d1,t}(s_t^i)\|_\infty \quad (11)$$
$$\mathcal{C}^i_{d_2,t}(s^i) = \|\varphi_{d2,t}(s_t^i)\|_\infty \quad (12)$$

Note that this is a maximum both across vehicles and across possible locations in grid $\mathcal{G}$.

This approach mimics the way human drivers deal with risk: for a given vehicle i, if the probability of another driver cutting in front (and causing thus a low front time-headway feature value) is high, a risk-aversive preemptive maneuver (e.g. braking, changing lane) is expected. Equivalently, if a cell close in front of the vehicle has a high probability of being occupied, the policy model should encourage actions that transition the vehicle to low risk states (either by reducing the speed or changing lanes). In contrast, if a vehicle can cut in front but this is not likely to happen (i.e. there is no reason or evidence that this will happen), vehicle i will not alter its trajectory. In other words, cells with very low occupancy probability should generate a low dynamic cost. The occupancy probability acts here as a gate, letting through the cost associated to dangerous situations when they are likely to occur.

G. Highway scene prediction algorithm

Algorithm 1 sums up the proposed approach. For each prediction timestep within the desired time horizon, and for each vehicle $i \in \mathcal{V}$ from the front to the back of the scene we calculate the cost function $\mathcal{C}_t^i(s^i)$ using (3) for all states $s^i \in \mathcal{S}^i$. To calculate the dynamic component of the cost, we use (11) and (12). Once the costs are computed, we can calculate each vehicle's risk-aversive policy using (8). Finally, we update the probability distribution over the state space b_{t+1}^i and the occupancy distribution $o_{t+1}^i = \chi(b_{t+1}^i)$.

Algorithm 1 Highway scene prediction with feature-based driver models

Require: T

```
for t = 0 ... T − 1 do
    for all i ∈ V from front to back do
        Update cost function C_t^i using (3,11,12)
        Compute vehicle i's policy π_t^i using (8)
        Maintain belief and occupancy dist. using (7,6)
    end for
end for
```

The algorithm outputs at each timestep the expected occupancy of the road. This can be used by a planner to estimate the risk of different candidate trajectories.

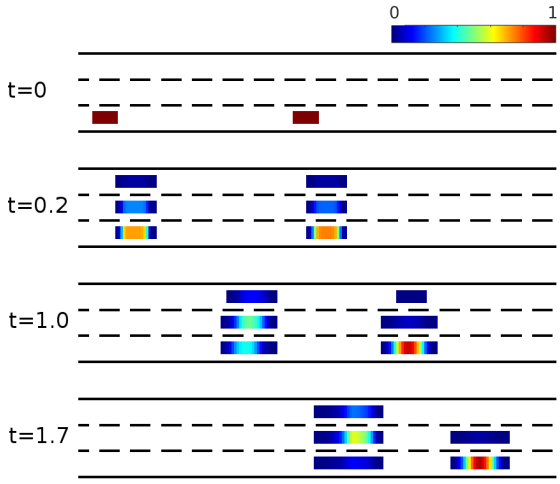
H. Complexity of algorithm 1

Algorithm 1 consists of three main operations (3), (8) and (6). Let $|\mathcal{A}^*|$ and $|\mathcal{S}^*|$ be the maximum number of actions and states for all vehicles; and G_x and G_y the transversal and longitudinal sizes of grid $\mathcal{G}$, respectively.

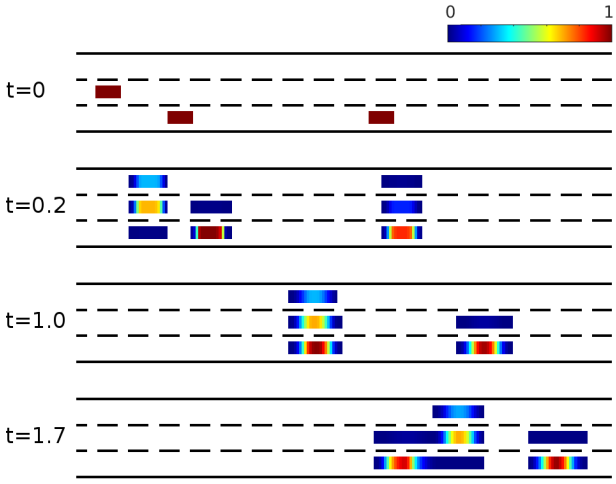
a) Complexity of cost-function updates: The first operation (3) is the update of the cost function for all states. This involves the update of the static and dynamic parts of the cost function for all states. However, the static part can be computed offline so it is not considered. The complexity of computing the dynamic part of the cost function is linear in the number of states for each vehicle $i \in \mathcal{V}$ and each timestep $t \in \{0, 1, \ldots, T-1\}$. As seen in subsection IV-F, to calculate these costs it is necessary to traverse longitudinally the occupancy grids of all opponent vehicles. Hence, it results in complexity $O(T \cdot |\mathcal{V}||S^*||\mathcal{V}-1|G_y)$.

b) Complexity of decision-rule updates: The second operation is the computation of the decision rule for all state and action of a vehicle $i \in \mathcal{V}$ and each timestep $t \in \{0, 1, \ldots, T-1\}$. That is about time complexity $O(|S^*||\mathcal{A}^*|)$. Notice that the denominator of (8) is computed only once for each state and vehicle, i.e., $O(|S^*||\mathcal{A}^*|)$. Overall the complexity of maintaining the decision rules for all vehicles and all timesteps is about $O(T \cdot |\mathcal{V}||\mathcal{S}^*||\mathcal{A}^*|)$

c) Complexity of information-measure updates: The belief- and occupancy-update rules require time complexities $O(|\mathcal{A}^*||\mathcal{S}^*|^2)$ and $O(|\mathcal{S}^*|)$, respectively for each vehicle at every timestep. That results in an overall complexity of about $O(T \cdot |\mathcal{V}||\mathcal{A}^*||\mathcal{S}^*|^2)$.



(a) Simulation of the traffic scene presented in figure 1a. The model predicts an overtaking maneuver from the trailing vehicle.



(b) Simulation of the traffic scene presented in figure 1b. The model predicts that the trailing vehicle on lane 1 will slow down (notice the low occupancy probabilities on lane 1 at $t = 1.7s$ that correspond to states in which the vehicle would maintain/increase its speed).

Fig. 3: Prediction of the traffic scenes from figures 1a and 1b using the driver model learned with the simulator. We show the result of summing up the occupancy probabilities across the occupancy distributions of all vehicles in the scene. One timestep corresponds to $0.1s$. Each action (lane change, speed change) takes one timestep to be completed.

By aggregating together each operation, we end up with a complexity for Algorithm 1 that is linear with respect to the planning horizon T and the number of actions per vehicle $|\mathcal{A}^*|$, and quadratic in the number of vehicles $|\mathcal{V}|$, and the number of states $|\mathcal{S}^*|$ — i.e., $O(T \cdot |\mathcal{A}^*||\mathcal{S}^*|^2|\mathcal{V}|^2 G_y)$.

V. EVALUATION

To evaluate the presented approach we developed a 3-lane highway simulator inspired by the one presented in [9]. We used this simulator to generate demonstrated trajectories from which a driver model was learned. Using this model, we tested our scene prediction algorithm on simulated highway traffic scenarios and on real-world traffic scenes from data gathered on a French highway.

A. Settings

In the simulator, the discrete set of possible actions $\mathcal{A}^i$ for each vehicle i includes lane and speed changes, and any combination of the two. The vehicle speeds are discretized into a set $\mathcal{Z}$ of 11 evenly spaced bins between 0 m/s and 40 m/s (144 km/h). We assume that a vehicle can switch to an adjacent lane or speed per timestep.

The simulator was used to generate training data by driving a vehicle through different highway traffic scenes. A total of 2 minutes of demonstrations were recorded and sampled at 10Hz. Using this dataset, we trained a driver model following the procedure described in subsection IV-B.

B. Simulation Results

The driver model learned was used to predict the evolution of different simulated highway traffic scenes. The length of the grid was set to twice the range of a Velodyne LiDAR, i.e. 240m., with a resolution in the longitudinal dimension of 0.5m. Fig. 3 shows the results obtained for the two highway traffic situations presented in the introduction in figures 1a and 1b. The figure shows the result of summing up across the occupancy distributions of all the vehicles in the scene (note that the result is no longer a probability distribution, but it is convenient to visualize the results). We assume that a vehicle changes lanes each time a new lane contains the majority of occupancy probability mass for that vehicle. The model captured correctly the demonstrated behavior. In Fig. 3a, the system predicted that the trailing car would switch lanes in order to overtake the slower vehicle and maintain its speed. In Fig. 3b, the system anticipated that the trailing vehicle on lane 1, would not switch lanes to overtake the slower vehicle in front, but instead slow down to maintain a security distance while waiting for lane 2 to be free of obstacles.

C. Real-World Highway Traffic Scenario

In order to verify if the intuitive results obtained in simulation apply as well to real-world situations, we used an instrumented vehicle to gather data on a French 3-lane highway. Fig. 4 shows a given traffic scene encountered

TABLE I: Data from the traffic scene in Fig. 4

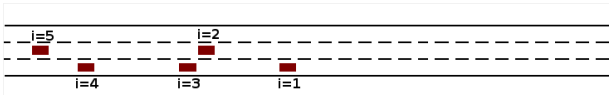
i	z^i[km/h]	rel. dist. [m]	lane
1	86	24.2	1
2	101	2.1	2
3	86	0	1
4	84	-26.6	1
5	107	-36.7	2



(a) Front view



(b) Back view



(c) Sum of the initial probability occupancy distributions



(d) Front view



(e) Back view



(f) Sum of the probability occupancy distributions at $t = 2.8s$

Fig. 4: Typical 3-lane highway scene ($t = 2.8s$). Our approach predicted a lane change for vehicle $i = 4$, that was advancing fast in the middle lane

during our data gathering. Aside from the ego-vehicle ($i = 3$), 4 other vehicles are in the scene. Their velocities and positions were tracked using the Conditional Monte Carlo Dense Occupancy Tracker [16] and are shown in table I (the vehicles are indexed from front to back). Vehicle $i = 5$ was approaching fast in the middle lane. Our approach successfully predicted the evolution of this traffic scene. In particular, it was predicted that vehicle $i = 5$ would change lanes to the left around $t = 2.8s$ in order to keep its high speed and overtake the vehicle in front (Fig. 4f).

VI. CONCLUSIONS

This paper proposes a novel highway scene prediction framework that takes into account the interactions between traffic participants through the use of a driver model learned from demonstrated driving tasks. The model is composed of static and dynamic features that capture the preferences and the risk-aversive behavior of drivers. The potential of this framework to predict the long-term development of highway traffic scenes has been demonstrated in simulated and real-world scenarios

In contrast with other approaches that take into account the mutual influence between drivers, the complexity of our approach does not grow exponentially in the number of vehicles in the scene. Instead, we showed that the complexity scales quadratically in the number of vehicles. As a direct application of the framework presented here, we can think of an advanced collision avoidance system that automatically learns the preferences of the driver and uses such knowledge to predict the driver's future maneuvers and to monitor the collision risk.

ACKNOWLEDGMENT

The authors would like to thank Nicolas Vignard, Gabriel Othmezouri, Dizan Vasquez, and Victor Romero-Cano for the helpful discussions and the feedback on earlier versions of the paper.

REFERENCES

[1] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, and F.-R. Kline, "The MIT-Cornell collision and why it happened," *Journal of Field Robotics*, vol. 25, no. 10, pp. 775–807, 2008.

[2] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss, "Interactive scene prediction for automotive applications," in *2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, June 23-26, 2013*, 2013, pp. 1028–1033.

[3] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, ser. IROS'09, Piscataway, NJ, USA, 2009, pp. 3931–3936.

[4] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, *Activity Forecasting*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 201–214.

[5] D. Vasquez, "Novel Planning-based Algorithms for Human Motion Prediction," in *IEEE Conference on Robotics and Automation*, ser. To appear in Proceedings of the IEEE Conference on Robotics and Automation, Stockholm, Sweden, May 2016.

[6] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, 2000, pp. 663–670.

[7] W. Schwarting and P. Pascheka, "Recursive conflict resolution for cooperative motion planning in dynamic highway traffic," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, Oct 2014, pp. 1039–1044.

[8] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, 2015.

[9] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, 2004.

[10] S. Levine and V. Koltun, "Continuous Inverse Optimal Control with Locally Optimal Examples," in *ICML '12: Proceedings of the 29th International Conference on Machine Learning*, 2012.

[11] N. Lee and K. M. Kitani, "Predicting wide receiver trajectories in american football," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2016, pp. 1–9.

[12] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, 2008, pp. 1433–1438.

[13] P. Henry, C. Vollmer, B. Ferris, and D. Fox, "Learning to navigate through crowded environments," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 981–986, 2010.

[14] D. Vasquez, B. Okal, and K. O. Arras, "Inverse reinforcement learning algorithms and features for robot navigation in crowds: An experimental comparison," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 1341–1346.

[15] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident Analysis & Prevention*, vol. 35, no. 3, pp. 427 – 433, 2003.

[16] L. Rummelhard, A. Nègre, and C. Laugier, "Conditional monte carlo dense occupancy tracker," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 2485–2490.